Technical Report

UCAM-CL-TR-782
ISSN 1476-2986

Number 782

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Petri net semantics

Jonathan M. Hayman

June 2010

15 JJ Thomson Avenue
Cambridge CB3 0FD
United Kingdom
phone +44 1223 763500

http://www.cl.cam.ac.uk/

This technical report is based on a dissertation submitted January 2009 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Darwin College.

Technical reports published by the University of Cambridge Computer Laboratory are freely available via the Internet:

http://www.cl.cam.ac.uk/techreports/

ISSN 1476-2986

Summary

Petri nets are a widely-used model for concurrency. By modelling the effect of events on local components of state, they reveal how the events of a process interact each other with whether they can occur independently of each other by operating on disjoint regions of state.

Despite their popularity, we are lacking systematic syntax-driven techniques for defining the semantics of programming languages inside Petri nets in an analogous way that Plotkin's Structural Operational Semantics defines a transition system semantics. The first part of this thesis studies a generally-applicable framework for the definition of the net semantics of a programming language.

The net semantics is used to study concurrent separation logic, a Hoare-style logic used to prove partial correctness of pointer-manipulating concurrent programs. At the core of the logic is the notion of separation of ownership of state, allowing us to infer that proven parallel processes operate on the disjoint regions of the state that they are seen to own. In this thesis, a notion of validity of the judgements capturing the subtle notion of ownership is given and soundness of the logic with respect to this model is shown. The model is then used to study the independence of processes arising from the separation of ownership. Following from this, a form of refinement is given capable of changing the granularity assumed of the program's atomic actions.

Amongst the many different models for concurrency, there are several forms of Petri net. Category theory has been used in the past to establish connections between them via adjunctions, often coreflections, yielding common constructions across the models and relating concepts such as bisimulation. The most general forms of Petri net have, however, fallen outside this framework. Essentially, this is due to the most general forms of net having an implicit symmetry in their behaviour that other forms of net cannot directly represent.

The final part of this thesis shows how an abstract framework for defining symmetry in models can be applied to obtain categories of Petri net with symmetry. This is shown to recover, up to symmetry, the universal characterization of unfolding operations on general Petri nets, allowing coreflections up to symmetry between the category of general Petri nets and other categories of net.

Acknowledgements

During the course of my time as a PhD student, I have been fortunate to benefit from the generosity and kindness of many people.

First and foremost, I would like to thank my supervisor, Glynn Winskel. It is impossible to overstate the importance of his support over the past few years. I am very grateful for his patient tuition, his guidance through many of the technical challenges that I encountered and his generosity in sharing his insight into models for concurrency, and more generally giving me the confidence needed to pursue new research ideas.

Though it would be a never-ending task to provide a comprehensive list of all those who have contributed their time and ideas, I would like to acknowledge especially the contributions of Matthew Parkinson to the work on concurrent separation logic and Sam Staton to the work on symmetry in nets. I would also like to thank Peter O'Hearn for giving a talk in Cambridge early on in my time as a PhD student giving the initial motivation for the first part of this thesis, and supporting the work through its development, including being my external examiner. Thanks, too, to Alan Mycroft, my internal examiner, for his careful reading of the thesis and useful suggestions on how to improve its presentation.

Less formally, Matt, Sam and all the other members of the theory group provided a wonderfully stimulating, supportive and sociable place in which to work, for which I am very grateful. I should not forget to explicitly mention my office-mates, Chung-Kil Hur, David Turner and Alex Gurney, for their pleasant company over the past four years.

Looking back, I would also like to thank the two people who encouraged me most to enter into research in theoretical computer science, Philippa Gardner and Chris Tofts. Philippa provided a great deal of guidance whilst I was an undergraduate at Imperial, and Chris allowed me a first proper taste of research during an industrial placement at Hewlett-Packard Laboratories in Bristol, when my interest in concurrency theory truly arose.

Finally, I would like to thank my family for all their love and support.

Contents

Chapter 1

Introduction

The development of computer processors over the past four decades has been described and, in a sense, led by Moore's Law [Moo65, Moo05] which predicts that the average number of transistors on CPUs being manufactured doubles approximately every two years. The law has broadly held over the the past forty-five years and is predicted to hold for some time into the future. Roughly speaking, in the past this increase in the number of transistors has allowed chip designers to increase the sophistication of the processor so that it can perform primitive instructions more rapidly. Broadly, this means that programs written years ago run faster when compiled to run on modern processors, and has allowed more complex programs to be developed that run in acceptable time. However, though Moore's Law is expected to continue to hold, this will be through new processors having multiple cores; essentially, through CPUs having several separate sub-processors each running their own program on the system's shared memory. Successive generations of CPU will have more cores. At present, commonly-available CPUs are either dual- or quad-core, and in the future it is expected that CPUs will have several hundreds of cores. The computational power of each core will not, however, be substantially greater than that of the previous generation. It follows that the programs of the future will have to run concurrently if they are to harness the power of future CPUs, and programming language systems will have to change to accommodate this.

The difficulty of writing correct concurrent programs has been appreciated for several decades. Critically, the inadequacy of the standard development cycle of coding, testing and debugging becomes severe when writing all but the most trivial concurrent programs. This is due to there possibly being many ways in which concurrent processes might interact with each other depending on the order in which they are scheduled to run, so each such interaction might not be seen during the testing phase.

The failure of the traditional development cycle of coding, testing and debugging means that the development of complex concurrent programs requires new techniques. One approach is to use compilers that extract a degree of concurrency from sequential code. In many situations, though, programmers will wish to exercise more control of how their code is run concurrently. New programming languages might well be of use in the future [Pey08, Myc07]. Yet, at the moment, the most commonly-encountered way of programming for parallelism is through programmers designing their own imperative programs in familiar languages such as C and Java, with concurrency obtained through the execution of concurrent 'threads' of programs. The threads may interact either through operation on a shared memory or through sending messages to each other, an idea taken to the limit in languages such as Occam and Esterel where the only form of interaction is by sending and receiving messages.

If programmers are to have direct control over the concurrency of their programs, they will need techniques to prove that their programs are correct and, if not, to help them determine where they are wrong. Generally, to prove that a program is correct we must have a formal model to represent the behaviour of the program — called its *semantics* — and then prove that the model has some particular property. The correctness property established might not be a full specification of the whole program since it might well be the case that a full specification is as complex as the program itself. It might instead be some form of *safety* property, that no 'bad' state is ever encountered (such as a state where a race occurs or that whenever the program terminates the correct result is given), or some form of *liveness* property, asserting that the process eventually makes some form of progress.

There are many ways in which the proof may be obtained: by some form of 'push-button' analysis such as a program analysis or model-checking a formula of temporal logic; establishing the correctness property directly by a proof on the model, perhaps in a formal system; or by applying a set of rules to the syntax of the program that establish that the model satisfies the property.

All these approaches rely on *models* of programs, *i.e.* formal structures that represent (in some level of abstraction) the behaviour of the program. The centre of this thesis is the development of the models themselves. One approach to modelling concurrent processes is through *interleaving*. This regards concurrency solely as a nondeterministic choice of which concurrent process to execute, for example regarding the program

$$x := 1 \parallel x := 2$$

which has two concurrent assignments to the variable x, one giving it value 1 and the other 2, as being indistinguishable from the process

$$(x := 1; x := 2) + (x := 2; x := 1).$$

This process represents the nondeterministic choice of the parallel composition, that either the assignment of 1 to x occurs and then the assignment of 2 or the assignments occur in the other order.

By adopting an interleaved semantics for programs, we lose information on what is perhaps the most important feature of the programs: the concurrency of their actions. For example, the interleaved semantics given to the program above does not represent that the program has a *race*, a concurrent assignment to the same variable. Problems with giving an interleaved semantics have been long-understood. The foundational work of Hoare on parallel programming [Hoa72] identified the fact that attributing an interleaved semantics to parallel languages is problematic. Three areas of difficulty were isolated, quoted directly:

- *That of defining a 'unit of action'.*
- *That of implementing the interleaving on genuinely parallel hardware.*
- *That of designing programs to control the fantastic number of combinations involved in arbitrary interleaving.*

The first two of these problems relate to the fact that an interleaved semantics can fail to correspond to the way in which concurrent programs run. The final problem relates to the fact that arbitrary interleaving naturally gives rise to intractably-large state spaces, and so to reason about concurrent programs it is necessary to constrain their interaction.

As Hoare went on to explain, a feature of concurrent systems in the physical world is that they are often *spatially separated*, operating on completely different resources and not interacting. When this is so, the systems are *independent* of each other and therefore it is unnecessary to consider how they interact. Spatial separation of programs can be made explicit through the use of structures such as semaphores [Dij68] or monitors [Bri72] to provide an explicit representation of the points where processes may interact. Programs should then be designed so that, for example, they only access a shared memory location when they are inside a critical region protecting that location [Hoa72]. As such, interaction is limited *only* to the points of synchronization. This embodies Dijkstra's principle of loose coupling [Dij68]:

> *...we have stipulated that the processes should be connected loosely; by this we mean that apart from the (rare) moments of explicit intercommunication, the individual processes themselves are to be regarded as completely independent of each other.*

The independence of programs motivates the use of the other broad class of models for concurrency, called *independence models*. These are sometimes called non-interleaving models or, more provocatively, 'true concurrency' models. Their common core is that they record the independence (which we might think of as 'allowed concurrency') of events. There are many forms of independence model such as Petri nets [Pet62, Rei85], event structures [NPW81, Win86], Mazurkiewicz trace languages [Maz89, Maz77] and asynchronous transition systems [Bed88, Shi85].

Petri nets are a widely-used independence model. They play a fundamental role analogous to that of transition systems, but, by capturing the effect of events on *local* components of state called *conditions*, it becomes possible to describe how events might occur concurrently, how they might conflict with each other and how they might causally depend on each other. In the form of Petri net that we shall consider first, conditions either hold a *token*, in which case they are said to hold, or they do not. The ability of an event to occur depends on and affects the holding of conditions. Conditions are an entirely abstract representation of components of state: they might correspond to a particular value being held in a location in a memory or they might correspond to a program being at a particular control point. As such, Petri nets make an excellent candidate for defining the semantics of programming languages.

Remarkably, though Petri nets and other independence models have been used successfully to give the semantics of several particular systems, they are lacking a *comprehensive* account of *systematic* ways of defining the semantics of programming languages in them. The first component of this thesis develops a net semantics for a programming language. This involves identifying the structure that the nets formed have, such as splitting the conditions into those that represent the control point of the program and those that represent the state, and establishing that the nets have particular control properties that ensure that they have the desired behaviour. The techniques developed indicate what appears to be a general method (an alternative to Plotkin's structural operational semantics [Plo81]) for giving a *structural Petri net semantics* to a variety of languages.

The development of the net semantics was motivated by the emergence of *concurrent separation logic* [O'H07, O'H04]. This extends Owicki and Gries' rules for shared-variable concurrency [OG76], themselves an extension of Hoare logic for sequential programs [Hoa69], to concurrent pointer-manipulating programs. Reasoning about such programs has traditionally proved difficult due to the problem of *variable aliasing*. For instance, Owicki and Gries' system essentially requires that the programs operate on disjoint

collections of variables, thereby allowing judgements to be composed. In the presence of pointers, the same syntactic condition cannot be imposed to yield a sound logic since distinct variables may point to the same memory location, allowing arbitrary interaction between the processes. To give a specific example, Owicki and Gries' system would allow a judgement of the form

$$\{x \mapsto 0 \wedge y \mapsto 0\} x := 1 \parallel y := 2 \{x \mapsto 1 \wedge y \mapsto 2\},$$

indicating that the result of assigning 1 to the program variable x concurrently with assigning 2 to y from a state where x and y both initially hold value 0 is a state where x holds value 1 and y holds value 2. The judgement is sound because the variables x and y are distinct. If pointers are introduced to the language, however, it is not sound to conclude that

$$\{[x] \mapsto 0 \wedge [y] \mapsto 0\} [x] := 1 \parallel [y] := 2 \{[x] \mapsto 1 \wedge [y] \mapsto 2\},$$

which would indicate that assigning 1 to the location pointed to by x and 2 to the location pointed to by y yields a state in which x points to a location holding 1 and y points to a location holding 2, since x and y may both point to the same location.

At the core of separation logic [Rey00, IO01], initially presented for non-concurrent programs, is the *separating conjunction*, $\varphi \star \psi$, which asserts that the state in which processes execute may be split into two parts, one part satisfying φ and the other ψ. The separating conjunction was used by O'Hearn to adapt Owicki and Gries' system to provide a rule for parallel composition suitable for pointer-manipulating programs [O'H07].

As we shall see, the rule for parallel composition is informally understood by splitting the initial state into two parts, one *owned* by the first process and the other by the second. Ownership can be seen as a dynamic constraint on the interference to be assumed: parallel processes always own disjoint sets of locations and only ever act on locations that they own. As processes evolve, ownership of locations may be transferred using a system of *invariants*. A consequence of this notion of ownership is that the rules discriminate between the parallel composition of processes and their interleaved expansion. For example, the logic does *not* allow the judgement

$$\{x \mapsto 0\} x := 1 \parallel x := 1 \{x \mapsto 1\},$$

which informally means that the effect of two processes acting in parallel which both assign the value 1 to x from a state in which x holds 0 is to yield a state in which x holds 1. However, if we adopt the usual rule for the nondeterministic sum of processes, the corresponding judgement *is* derivable for their interleaved expansion,

$$\{x \mapsto 0\}(x := 1; x := 1) + (x := 1; x := 1)\{x \mapsto 1\}.$$

One would hope that the distinction that the logic makes between concurrent processes and their interleaved expansion is captured by the semantics; the Petri net model that we give does so directly.

The rules of concurrent separation logic contain a good deal of subtlety, and so lacked a completely formal account until the pioneering proof of their soundness due to Brookes [Bro07]. The proof that Brookes gives is based on a form of interleaved trace semantics. The presence of pointers within the model alongside the possibility that ownership of locations is transferred means, however, that the way in which processes are separated is absolutely non-trivial, which motivates strongly the study of the language within an independence model. We therefore give a proof of soundness using our net model and then

characterize entirely semantically the independence of concurrent processes in Theorem 5.5.

It should be emphasized that the model that we present is different from Brookes' since it provides an *explicit* account of the intuitions behind ownership presented by O'Hearn. This is no criticism of Brookes' more abstract model. The model here involves taking the original semantics of the process being reasoned about and embellishing it to capture the meaning of the judgement. The proof technique that we employ defines validity of assertions in a way that captures the rely-guarantee reasoning [Jon83] emanating from ownership in separation logic directly, and in a way that might be applied in other situations.

In [Rey04], Reynolds argues that the separation of parallel processes that concurrent separation logic enforces allows store actions that were assumed to be atomic, in fact, to be implemented as composite actions (seen as a change in their *granularity*) with no effect on the validity of the judgement. Interleaving models do not permit simple techniques for studying the atomicity of actions [Lam86] since they assume atomicity of actions. Independence models, however, can be read without making this assumption [Pra86]. We introduce a novel form of refinement, inspired by that of [vGG89], and show how this may be applied to address the issue of granularity using our characterization of the independence of processes arising from the logic.

In the final part of this thesis, we move in a related but different direction to consider the semantics of Petri nets. In particular, we show how a general framework for defining *symmetry* on models can be applied to enrich nets with symmetry.

Without doubt symmetry is important and plays a role, at least informally, in many models, and often in the analysis of processes. It is, for instance, present in security protocols due to the repetition of essentially similar sessions [DGT07, FHG98, CW01a], can be exploited to increase efficiency in model checking [Sis04], and is present whenever abstract names are involved [GP01].

There is a wide array of models for concurrency. In [WN95], it is shown how category theory can be applied to describe the relationships between them by establishing adjunctions between their categories; the adjunctions often take the form of coreflections. This leads to uniform ways of defining constructions on models and provides links between concepts such as bisimulation in the models [JNW95].

Only partial results have been achieved in relating Petri nets to other models for concurrency since, in general, there is no coreflection between occurrence nets, a class of net suited to connecting nets to other models for concurrency, and more general forms of net that allow transitions to deposit more than one token in any condition or in which a condition can initially hold more than one token. The reason for this, as we shall see, is that the operation of unfolding such a net to form its associated occurrence net does not account for the *symmetry* in the behaviour of the original net due to conditions being marked more than once. We shall define the symmetry in the unfolding and use this to obtain a coreflection between general nets and occurrence nets *up to symmetry*.

Of course, there are undoubtedly several ways of adjoining symmetry to nets. The method we use was motivated by the need to extend the expressive power of event structures and the maps between them [Win07a, Win07b]. Roughly, a symmetry on a Petri net is described as a relation between its runs as causal nets, the relation specifying when one run is similar to another up to symmetry; of course, if runs are to be similar then they should have similar futures as well as pasts. Technically and generally, a relation of symmetry is expressed as a span of open maps which form a pseudo-equivalence.

This general algebraic method of adjoining symmetry is adopted to define symmetry

in (the paths of) nets, which we use to relate the categories of general nets with symmetry and occurrence nets with symmetry. Another motivation for this work is that Petri nets provide a useful testing ground for the general method of adjoining symmetries. For example, the present work has led us to drop the constraint in [Win07a, Win07b] that the morphisms of the span should be jointly monic, in which case the span would be an equivalence rather than a pseudo-equivalence. (A similar issue is encountered in the slightly simpler setting of nets without multiplicities — see Section 8.5.) Motivated by the categories of nets encountered, the method for adjoining symmetry is also extended to deal with more general forms of model such as those without all pullbacks.

Overview

In Chapter 2, we give some background material on Petri nets. The net semantics of a programming language is developed in Chapter 3. We then give an overview of concurrent separation logic in Chapter 4 followed by the formal semantics of the logic and proof of its soundness in Chapter 5. In the same chapter, we capture the independence of events obtained from the logic and use this in Chapter 6 to define the refinement operation. These chapters expand the journal paper [HW08a]. Unfoldings of general Petri nets are described in Chapter 7 followed by application of symmetry in Chapter 8 to obtain an abstract characterization of the unfolding. These chapters expand [HW08c, HW08b]. We then conclude, discussing related work and directions for further research.

Chapter 2

Categories of Petri nets

In this chapter, we review some basic definitions describing Petri nets. We shall begin by defining *basic Petri nets*. These will form the basis of the net semantics to be presented in the following chapter. There are many good introductions to Petri nets such as [Rei85, BRR87], to which the reader is referred for a fuller account. Since we wish to deal with categories of Petri nets, the account presented here shall follow that in [WN95].

2.1 Petri nets

Petri nets can be thought of as being a generalization of transition systems, where the ability of events to occur depends on local components of the distributed state called conditions. This allows a derived notion of independence of events; two events are independent if their neighbourhoods of conditions do not intersect.

Formally, nets are built only from these *conditions* to represent the local components of state and *events*. Petri nets have an appealing graphical interpretation where conditions are drawn as circles and events as rectangles. In the basic form of net to be introduced in this chapter, local state is represented by conditions either *holding* or not. If a condition holds, it is drawn containing a *token* and is said to be *marked*. The local components of state that must hold for an event to occur are called its *preconditions*; the conditions that an event causes to hold are called its *postconditions*. The precondition relation is indicated by drawing an arrow from a condition to an event if it is a precondition, and the postcondition relation is indicated by drawing arrows from events to conditions. The complete definition of a Petri net is:

Definition 2.1.1 (Basic Petri net)**.** *A Petri net is a five-tuple,*

$$(B, E, {}^\bullet(-), (-)^\bullet, M_0)$$

where

- B *is the set of* conditions *of the net,*
- E *is the set of* events *of the net, disjoint from* B*,*
- ${}^\bullet(-)$ *is the* precondition *map,* ${}^\bullet(-){:}E \to \mathcal{P}ow(B)$*,*
- ${}^\bullet(-)$ *is the* postcondition *map,* $(-)^\bullet{:}E \to \mathcal{P}ow(B)$*, and*
- M_0 *is the* initial marking *of the net,* $M_0 \subseteq B$*.*

A *marking* is a set of conditions, representing the global state of a net by indicating which conditions hold. As seen, Petri nets are defined with an *initial marking*. This is the set of conditions that hold in the initial state.

The set of preconditions of an event e is written ${}^\bullet e$ and its set of postconditions is $e^\bullet$. Its *neighbourhood* is the union of these two sets,

$$ {}^\bullet e^\bullet = {}^\bullet e \cup e^\bullet. $$

It is convenient to further restrict nets so that they contain no *isolated* conditions, which are conditions that are neither in the initial marking of the net nor are they in the neighbourhood of any event.

Action within nets is defined according to a *token game* which defines how the marking of the net changes according to the occurrence of the events. An event e can occur if all its preconditions are marked and, following their un-marking, all the postconditions are not marked. That is, in marking M,

$$ \begin{array}{ll} (1) & {}^\bullet e \subseteq M \\ (2) & (M \setminus {}^\bullet e) \cap e^\bullet = \emptyset. \end{array} $$

Such an event is said to have *concession* or to be *enabled*. The marking following the occurrence of e is obtained by removing the tokens from the preconditions of e and placing a token in every postcondition of e. We write $M \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ where

$$ M' = (M \setminus {}^\bullet e) \cup e^\bullet. $$

Let π be a sequence of events, so $\pi = (e_1, \ldots, e_n)$. The transition relation $M \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ can be extended to sequences:

$$ \begin{array}{rcl} M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M' & \iff & \text{there exist } M_0, \ldots, M_n \text{ such that } M = M_0 \ \&\ M' = M_n \\ & & \&\ \forall i \text{ s.t. } 0 < i \leq n : M_{i-1} \stackrel{e_i}{\longrightarrow\!\!\!\!\!\rightarrow} M_i. \end{array} $$

When we wish to make clear that the sequence occurs in net N, we shall write $N{:}M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$. We sometimes write $M \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ to mean that there exists π such that $M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$.

A marking M is said to be *reachable* if there is a sequence of events from the initial marking that yields M. That is, M is reachable iff

$$ N{:}M_0 \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} M, $$

where M_0 is the initial marking of N.

Lemma 2.1.1. *Any path gives rise to a unique marking: For any basic net N, markings M, M'_1 and M'_2 and sequence π:*

$$ M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'_1 \ \&\ M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'_2 \implies M'_1 = M'_2. $$

Proof. A straightforward induction on the length of π. □

Contact and safety

If constraint (2) in the definition of the token game above does not hold but constraint (1) does, so the preconditions are all marked (have a token inside) but following removal of

the tokens from the preconditions there is a token in some postcondition, there is said to be *contact* in the marking and the event cannot fire. The idea that the occurrence of an event should be inhibited by one of its postconditions already holding can be reconciled with the account above by thinking that an event can occur when its preconditions hold and must *begin* the holding of its postconditions. We shall give an example of a net with contact shortly. After that, we shall move on to consider morphisms of nets. In [WN95], two kinds of morphism are described: one restricted kind, not so useful for us, that is appropriate for all basic nets, and the one that we shall describe that is only suitable for basic nets without contact.

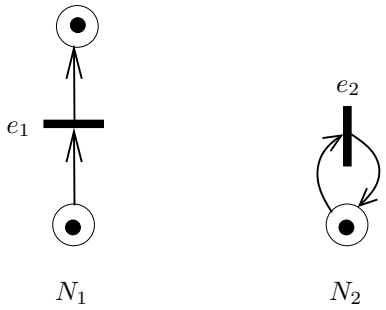
Definition 2.1.2. *A basic net is* safe *if there is no contact in any reachable marking.*

2.2 Behaviour of nets and examples

Through specifying events by their action on local components of state, the behaviour of nets gives rise to several phenomena.

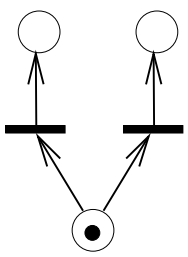
Contact

In the following net, the event e_1 in the net N_1 cannot occur in the marking drawn due to contact: one of its postconditions still holds a token once its preconditions are un-marked. The event e_2 in the net N_2 does not have contact since the event can occur according to the definition.

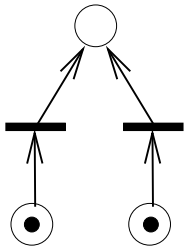


Conflict

Conflict describes how the occurrence of one event can inhibit the occurrence of another in a marking. There are two forms of conflict: *forwards* and *backwards*. Backwards conflict occurs where events compete to consume a condition. In the following net, only one of the events can occur from the marking drawn since the occurrence of one will consume the only token, causing the other not to have concession.

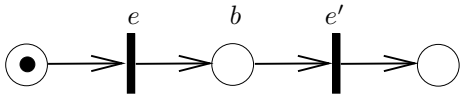


Forwards conflict is where events compete to place a token in a condition — they compete on their postconditions. In the following net, only one of the events can occur. The other will not be able to occur in the resulting marking due to contact.



Causality

The causal relationships between the occurrences of events can also be extracted from nets. For example, in the net below the event e_2 can only occur once the event e_1 has occurred. This is due to the requirement that the condition b be marked for the occurrence of e_2, and b can only be marked through the occurrence of e_1.

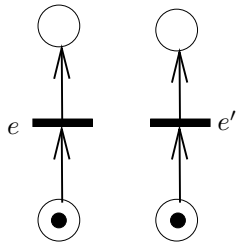


Concurrency

The final important concept in the behaviour of processes represented by nets is that independent events are allowed to occur concurrently. Independence of events is derived from their operation on non-overlapping sets of conditions. For events e and e', we write eIe' defined as:

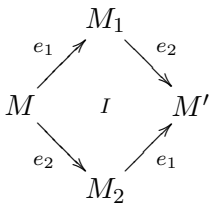
$$eIe' \iff {}^\bullet e^\bullet \cap {}^\bullet e'^\bullet = \emptyset.$$

For example, the events e and e' drawn below are independent:



This gives rise to the 'diamond property' for independent events. The first part describes that if two independent events occur consecutively then the occurrence of the second event does not depend on the occurrence of the first event, and the order in which they occur has no effect on the resultant marking. The second part describes that if two independent events can both occur in a marking then the occurrence of one does not inhibit the occurrence of the other. They both describe how the following diamond of

independent events may be derived:



Lemma 2.2.1. *Let M be a marking of a net N and e_1 and e_2 be independent events.*

- *If $M \xrightarrow{e_1} M_1 \xrightarrow{e_2} M'$ then there exists M_2 such that $M \xrightarrow{e_2} M_2 \xrightarrow{e_1} M'$.*
- *If $M \xrightarrow{e_1} M_1$ and $M \xrightarrow{e_2} M_2$ then there exists M' such that $M_1 \xrightarrow{e_2} M'$ and $M_2 \xrightarrow{e_1} M'$.*

Since the events e_1 and e_2 are independent, we can see the net as allowing them to occur concurrently in any marking.

2.3 Morphisms

A *morphism* from net N to net N' expresses how the structure of N embeds into N' in a way that preserves the behaviour of N. We shall use them later when we show that the net semantics corresponds to an operational semantics and when we consider unfoldings of nets in Chapters 7 and 8 (where a more general morphism accompanies the more general nets used there). If just interested in the definition of the net semantics and separation logic, the reader may safely proceed directly to Chapter 3.

The form of morphism that we now give, presented in [WN95], is suited to safe nets, and relies on a little notation. Morphisms of nets as presented here were first introduced in [Win84].

Let $N = (B, E, {}^\bullet(-), (-)^\bullet, M_0)$ and $N' = (B', E', {}^\bullet(-), (-)^\bullet, M_0')$ be safe Petri nets. The morphism shall consist of a partial function η from E to E'. If the partial function is undefined on the event e, we write $\eta(e) = *$. The notation for preconditions and postconditions is extended to $*$ so that

$${}^\bullet* = \emptyset \qquad *^\bullet = \emptyset.$$

To emphasize the role of $*$ in describing partiality, we write $\eta{:}E \to_* E'$.

The morphism shall also consist of a relation $\beta \subseteq B \times B'$. For any subset $X \subseteq B$ of conditions of N, we write βX for the result of applying β to X:

$$\beta X = \{b' \mid b' \in B' \ \&\ \exists b \in X : \beta(b, b')\}$$

We also write $\exists!$ to mean "there exists unique".

Definition 2.3.1. *A morphism $(\eta, \beta){:}N \to N'$ consists of a partial function $\eta{:}N \to_* N'$ and a relation $\beta \subseteq B \times B'$ that jointly satisfy:*

- *$\beta M_0 \subseteq M_0'$ and $\forall b' \in M_0' : \exists! b \in M_0 : \beta(b, b')$,*
- *for any $e \in E$: ${}^\bullet\beta e \subseteq {}^\bullet\eta(e)$ and $\forall b' \in {}^\bullet\eta(e) : \exists! b \in {}^\bullet e : \beta(b, b')$, and*
- *for any $e \in E$: $\beta e^\bullet \subseteq \eta(e)^\bullet$ and $\forall b' \in \eta(e)^\bullet : \exists! b \in e^\bullet : \beta(b, b')$.*

The morphism is synchronous *if η is a total function.*

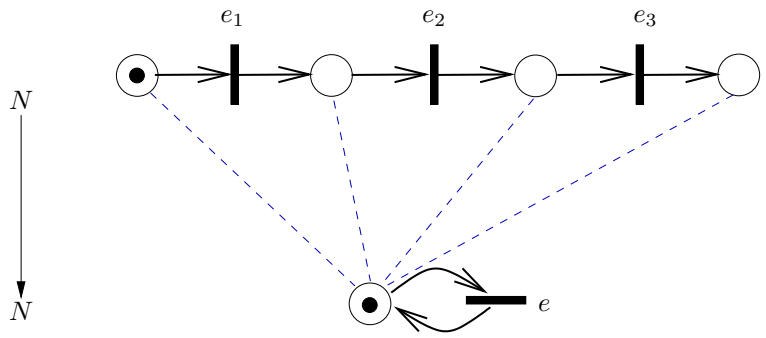
The constraint $\forall b' \in {}^\bullet\eta(e) : \exists! b \in {}^\bullet e : \beta(b, b')$ asserts that the morphism is *locally injective* on the preconditions of e. Generally, we say that a relation $R \subseteq A \times B$ is locally injective on a set $X \subseteq B$ if

$$\forall y \in RX : \exists! x \in X : R(x, y).$$

Taking composition of morphisms to be the composition of partial functions on events and the composition of relations on conditions, we obtain a category **Safe** of safe Petri nets with no isolated conditions.

Any morphism $(\eta, \beta){:}N \to N'$ preserves the token game described earlier. As shown in Section 10.3.2 of [WN95], it need not be the case that morphisms preserve the independence relation on events. An example of this is the representation of the following *folding* as a morphism:

Example 2.3.1. *The following dashed blue lines represent the β component of a morphism $(\eta, \beta){:}N \to N'$. On events, the morphism sends the event e_i to e for all i.*



This morphism represents that the net representing the sequence of events e_1, e_2, e_3 can be folded into the net with a single event e that can occur repeatedly. Tthe events e_1 and e_3 are independent (they operate on disjoint neighbourhoods) but their image under the morphism, e, is trivially not independent of itself.

However, importantly morphisms do preserve the *concurrency* of events. For any two events e_1 and e_2 of N, define e_1 co e_2 iff there exists a reachable marking M such that $M \overset{\{e_1,e_2\}}{\longrightarrow\!\!\!\!\!\rightarrow} M'$. This means that the events are independent and both have concession in some reachable marking.

The preservation of the token game and of independence is captured in the following lemma.

Lemma 2.3.1. *If $M \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ then $\beta M \overset{\eta(e)}{\longrightarrow\!\!\!\!\!\rightarrow} \beta M'$. If e_1 co e_2 then $\eta(e_1)$ co $\eta(e_2)$.*

Proof. Straightforward calculation. □

As seen, the occurrence of an event in a marking entails the possible occurrence of the image of the event in the image of the marking. The morphism is still a morphism between the resulting nets.

Lemma 2.3.2. *Let N and N' be safe Petri nets with no isolated conditions with $N = (B, E, {}^\bullet(-), (-)^\bullet, M_0)$ and $N' = (B', E', {}^\bullet(-), (-)^\bullet, M_0')$, and let $(\eta, \beta){:}N \to N'$ be a morphism. For any event e such that $M_0 \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} M$ and hence $M_0' \overset{\eta(e)}{\longrightarrow\!\!\!\!\!\rightarrow} \beta M$, the following holds:*

$$\forall b' \in \beta M : \exists! b \in M : \beta(b, b')$$

Hence the pair (η, β) *is a morphism*

$$(\eta, \beta):(B, E, {}^\bullet(-), (-)^\bullet, M) \to (B', E', {}^\bullet(-), (-)^\bullet, \beta M).$$

Proof. The first part is a straightforward calculation. The second part follows immediately. □

2.4 Open maps

One of the key reasons for studying categories of models for concurrency is to provide an abstract setting in which concepts such as process equivalence may be studied. For example, an account of bisimulation in models for concurrency is described in [JNW95] based on *open maps*. In any model for concurrency, given an interpretation of the computational paths of the model, the abstract framework can be applied to obtain a form of equivalence called *open map bisimulation*.

A proper account of the rich theory of open maps is provided in [JNW95]. It shall not be repeated fully here, but we shall for completeness give the essential definitions.

Abstractly, the definition can be applied to obtain a form bisimulation in any category $\mathcal{C}$ that has a subcategory of path objects $\mathcal{P}$. For instance, by taking paths of labelled transition systems to be sequences of labels, we obtain Milner and Park's standard form of bisimulation [JNW95].

The first part of the definition of open map bisimulation is the definition of *open maps*. These are morphisms that represent 'functional' bisimulations. The full relational definition of bisimulation is then obtained regarding these as the legs of a *span* of morphisms.

According to the definition there, a morphism $f:X \to Y$ is said to be $\mathcal{P}$*-open* if it satisfies the following path lifting condition:

> For any objects P and P' and morphism $s:P \to P'$ in $\mathcal{P}$ and morphisms $p:P \to X$ and $p':P' \to X'$ in $\mathcal{C}$ such that the following diagram commutes
>
> $$\begin{array}{ccc} P & \xrightarrow{p} & X \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle f} \\ P' & \xrightarrow[p']{} & X', \end{array}$$
>
> *i.e.* $f \circ p = p' \circ s$, there is a morphism $h:P' \to X$ such that the two triangles in the following diagram commute
>
> $$\begin{array}{ccc} P & \xrightarrow{p} & X \\ {\scriptstyle s}\downarrow & \overset{h}{\nearrow} & \downarrow{\scriptstyle f} \\ P' & \xrightarrow[p']{} & X', \end{array}$$
>
> *i.e.* $p = h \circ s$ and $f \circ h = p'$.

In general, objects Y and Y' are $\mathcal{P}$-open map bisimilar if there exists a span of $\mathcal{P}$-open maps relating them. That is, if there exists an object X of $\mathcal{C}$ and $\mathcal{P}$-open morphisms $f:X \to Y$ and $f':X \to Y'$:

$$\begin{array}{ccc} & X & \\ {\scriptstyle f}\swarrow & & \searrow{\scriptstyle f'} \\ Y & & Y' \end{array}$$

Spans of open maps compose by taking pullbacks as demonstrated in [JNW95]. This yields a bisimulation since the pullback of open maps is open [JNW95, Proposition 3].

Open maps and labelled nets

The particular equivalence that we shall be interested in later will be on *labelled* Petri nets. A labelled Petri net is just a standard Petri net with a function attaching a label to each event. Morphisms between labelled nets are required to be label-preserving, so it follows that the ensuing form of open map bisimilarity shall be label-preserving.

We must choose a path category with respect to which we draw open maps. There are two well-known choices of path for nets. The first form of path is *causal nets*, which shall be covered in Section 7.4, and the second form is *pomsets* [Pra86] (an abbreviation of 'partially ordered multisets'). In the following two chapters, we shall consider pomsets as paths.

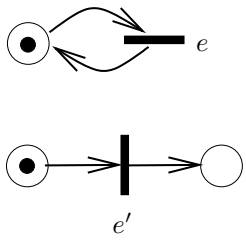
Fundamentally, a pomset path is just a multiset of the events to occur alongside an order to indicate the order in which the occurrences happened.

Definition 2.4.1. *A pomset over the labelling set Lab is a tuple* $(X, \leq, \lambda)$ *comprising:*

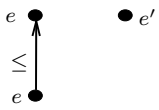
- *a finite set of occurrences* X*,*
- *a partial order* $\leq$ *on* X*, and*
- *a labelling function* $\lambda{:}X \to Lab$*.*

The elements of a pomset X can be thought of via λ as labelled occurrences of events. Where two occurrences are unrelated through the order $\leq$, they can be thought of as occurring concurrently. (Event structures have not yet been defined in this thesis, but the reader familiar with them might find it useful to note that a pomset is just a labelled elementary event structure [Win86] — *i.e.* a labelled event structure that has empty conflict relation.)

For example, consider the following Petri net:



One run of this net is for the event e to occur twice, one time after the other, and the event e' also to occur. The pomset is drawn below, with dots representing occurrences drawn with their label. Note that there are two occurrences of the event e with the order indicating that one occurs after the other. There is also an occurrence of the event e'. This is independent of the occurrences of the event e and so is unrelated to them through the order.



As described in [NW96], pomsets can be regarded as Petri nets. In the category **Safe**, a pomset $P = (X, \leq, \lambda)$ embeds into a net N through a morphism $p{:}P \to N$ if P represents a path of the net N in which $x \in X$ represents the occurrence of the event $p(x)$.

A morphism in the category **Pom** of pomsets from $(X, \leq, \lambda)$ to $(X', \leq', \lambda')$ is a label-preserving function f from X to X' such that if $f(x) \leq' f(y)$ then $x \leq y$. As such, morphisms show how paths can be extended by adding more events or allowing more concurrency.

It is shown in [NW96] that taking pomsets as paths yields a form of open map bisimulation that corresponds to a strengthening of a known equivalence on nets, namely history preserving bisimulation [vGG89, RT88]. Giving a span of **Pom**-open morphisms is, therefore, just a way of exhibiting a bisimulation and a **Pom**-open map simply exhibits a functional bisimulation.

With the categories **Pom** and **Safe** now defined, the abstract definition of open map bisimulation can now be applied. **Pom**-openness of a morphism in the category **Safe** can be characterized as follows:

Lemma 2.4.1 (Proposition 16 in [NW96])**.** *Let $N = (B, E, {}^\bullet(-), (-)^\bullet, M_0)$ and $N' = (B', E', {}^\bullet(-), (-)^\bullet, M_0')$ be safe Petri nets with labelling function $|-|$. A morphism $(\eta, \beta){:}N \to N'$ is* **Pom***-open if for any marking M reachable from M_0 in N:*

- *η is total and label preserving,*
- *if $\beta M \xrightarrow{e'} M''$ in N' then there exist e and M' such that $M \xrightarrow{e} M'$ in N and $\eta(e) = e'$ and $\beta M' = M''$, and*
- *(η, β) reflects the independence of consecutive events in the sense that whenever $M \xrightarrow{e_1} M_1 \xrightarrow{e_2} M_2$ for some M_1 and M_2 and $\eta(e_1) I \eta(e_2)$ in N' then $e_1 I e_2$ in N.* □

Chapter 3

Structural net semantics

In this chapter, we introduce a Petri net semantics for a simple concurrent programming language. The semantics shall be used in the following chapter to give a semantics to (an interpretation of the judgements derivable in) concurrent separation logic. We therefore introduce the net semantics for the programming language with the specific features which will be important to the logic, such as that the programs operate on a heap and may enforce mutual exclusion through the use of critical regions, though the techniques for defining the net semantics have wider applicability.

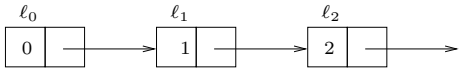
3.1 Terms and states

Concurrent separation logic is a logic for programs that operate on a *heap*. A heap is a structure recording the values held by memory locations that allows the existence of pointers as well as providing primitives for the allocation and deallocation of memory locations. A heap can be seen as a finite partial function from a set of locations Loc to a set of values Val:

$$\text{Heap} \quad \triangleq \quad \text{Loc} \rightharpoonup_{\mathsf{fin}} \text{Val}$$

We will use ℓ to range over elements of Loc and v to range over elements of Val. As stated, a heap location can point to another location, so we require that $\text{Loc} \subseteq \text{Val}$. We shall say that a location is *current* (or *allocated*) in a heap if the heap is defined at that location. The procedure of making a non-current location current is *allocation*, and the reverse procedure is called *deallocation*. If h is a heap and $h(\ell) = \ell'$, there is no implicit assumption that $h(\ell')$ is defined. Consequently, heaps may contain *dangling* pointers.

Example 3.1.1. *The partial function* $\{\ell_0 \mapsto (0, \ell_1), \ell_1 \mapsto (1, \ell_2), \ell_2 \mapsto (2, \ell_3)\}$ *represents the heap*



which is a list with a dangling pointer to ℓ_3 *at location* ℓ_2*.*

In addition to operating on a heap, the programs that we shall consider shall make use of *critical regions* [Dij68] protected by *resources*. The syntax for declaring a critical region is

$$\texttt{with}\ r\ \texttt{do}\ t\ \texttt{od}.$$

The mutual exclusion property that critical regions provide is that no two parallel processes may be inside critical regions protected by the same resource. For example, there is no

Terms:

t	::=	ε	nil process
	|	α	heap action
	|	$\texttt{alloc}(\ell)$	heap allocation
	|	$\texttt{dealloc}(\ell)$	heap disposal
	|	$t_1; t_2$	sequential composition
	|	$t_1 \parallel t_2$	parallel composition
	|	$\alpha_1.t_1 + \alpha_2.t_2$	guarded sum
	|	$\texttt{while}\ b\ \texttt{do}\ t\ \texttt{od}$	iteration
	|	$\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}$	resource declaration
	|	$\texttt{with}\ re\ \texttt{do}\ t\ \texttt{od}$	critical region.
re	::=	r	resource value
	|	w	local resource variable

Figure 3.1: Syntax of terms

run of

$$\texttt{with}\ r\ \texttt{do}\ t_1\ \texttt{od} \parallel \texttt{with}\ r\ \texttt{do}\ t_2\ \texttt{od},$$

the parallel composition of two critical regions, to a state where both t_1 and t_2 are active.

We will write Res for the set of resource values and use r to range over its elements. Critical regions are straightforwardly implemented by recording, for each resource, whether the resource is available or unavailable (assuming the existence of atomic operations to make the resource available or unavailable). A process may enter a critical region protected by r only if r is available; otherwise it is blocked and may not resume execution until the resource becomes available. The process makes r unavailable upon entering the critical region and makes r available again when it leaves the critical region.

To accompany the syntax for critical regions, there is also a syntax for defining new resources *local* to a particular subprocess. The term $\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}$ has the resource variable w bound within t, asserting that a resource is to be chosen that is local to t and used for w. Consequently, in the process

$$(\texttt{resource}\ w\ \texttt{do}\ \texttt{with}\ w\ \texttt{do}\ t_1\ \texttt{od}\ \texttt{od}) \parallel (\texttt{resource}\ w\ \texttt{do}\ \texttt{with}\ w\ \texttt{do}\ t_2\ \texttt{od}\ \texttt{od})$$

the sub-processes t_1 and t_2 may run concurrently since they must be protected by different resources, one local to the process on the left and the other local to the process on the right.

The semantics of this is obtained in the standard way by substitution, in the normal way for any syntax with abstraction. We shall say that the construct $\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}$ *binds* the variable w within t, and the variable w is *free* in $\texttt{with}\ w\ \texttt{do}\ t\ \texttt{od}$. We write $\mathrm{fv}(t)$ for the free variables in t and say that a term *closed* if it contains no free resource variables; we shall restrict attention to such terms. We write $[r/w]t$ for the term obtained by substituting r for free occurrences of the variable w within t. Free variables and substitution are formally defined in Figure 3.2. As standard, we will identify terms 'up to' the standard alpha-equivalence induced by renaming bound occurrences of variables. The notation $\mathrm{res}(t)$ is adopted to represent the resources occurring in t. As standard, the semantics will be defined in such a way to ensure that free variables never occur in actions that occur.

$$
\begin{array}{rcl}
\mathrm{fv}(\varepsilon) & = & \emptyset \\
\mathrm{fv}(\alpha) & = & \emptyset \\
\mathrm{fv}(\texttt{alloc}(\ell)) & = & \emptyset \\
\mathrm{fv}(\texttt{dealloc}(\ell)) & = & \emptyset \\
\mathrm{fv}(t_1; t_2) & = & \mathrm{fv}(t_1) \cup \mathrm{fv}(t_2) \\
\mathrm{fv}(t_1 \parallel t_2) & = & \mathrm{fv}(t_1) \cup \mathrm{fv}(t_2) \\
\mathrm{fv}(\alpha_1.t_1 + \alpha_2.t_2) & = & \mathrm{fv}(t_1) \cup \mathrm{fv}(t_2) \\
\mathrm{fv}(\texttt{while}\ b\ \texttt{do}\ t\ \texttt{od}) & = & \mathrm{fv}(t) \\
\mathrm{fv}(\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}) & = & \mathrm{fv}(t) \setminus \{w\} \\
\mathrm{fv}(\texttt{with}\ r\ \texttt{do}\ t\ \texttt{od}) & = & \mathrm{fv}(t) \\
\mathrm{fv}(\texttt{with}\ w\ \texttt{do}\ t\ \texttt{od}) & = & \mathrm{fv}(t) \cup \{w\}
\end{array}
$$

$$
\begin{array}{rcl}
\mathrm{res}(\varepsilon) & = & \emptyset \\
\mathrm{res}(\alpha) & = & \emptyset \\
\mathrm{res}(\texttt{alloc}(\ell)) & = & \emptyset \\
\mathrm{res}(\texttt{dealloc}(\ell)) & = & \emptyset \\
\mathrm{res}(t_1; t_2) & = & \mathrm{res}(t_1) \cup \mathrm{res}(t_2) \\
\mathrm{res}(t_1 \parallel t_2) & = & \mathrm{res}(t_1) \cup \mathrm{res}(t_2) \\
\mathrm{res}(\alpha_1.t_1; \alpha_2.t_2) & = & \mathrm{res}(t_1) \cup \mathrm{res}(t_2) \\
\mathrm{res}(\texttt{while}\ b\ \texttt{do}\ t\ \texttt{od}) & = & \mathrm{res}(t) \\
\mathrm{res}(\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}) & = & \mathrm{res}(t) \\
\mathrm{res}(\texttt{with}\ r\ \texttt{do}\ t\ \texttt{od}) & = & \mathrm{res}(t) \cup \{r\} \\
\mathrm{res}(\texttt{with}\ w\ \texttt{do}\ t\ \texttt{od}) & = & \mathrm{res}(t)
\end{array}
$$

$$
\begin{array}{llcll}
[r/w] & \varepsilon & = & \varepsilon & \\
[r/w] & \alpha & = & \alpha & \\
[r/w] & \texttt{alloc}(\ell) & = & \texttt{alloc}(\ell) & \\
[r/w] & \texttt{dealloc}(\ell) & = & \texttt{dealloc}(\ell) & \\
[r/w] & t_1; t_2 & = & ([r/w]\, t_1); ([r/w]\, t_2) & \\
[r/w] & t_1 \parallel t_2 & = & ([r/w]\, t_1) \parallel ([r/w]\, t_2) & \\
[r/w] & \alpha_1.t_1 + \alpha_2.t_2 & = & \alpha_1.([r/w]\, t_1) + \alpha_2.([r/w]\, t_2) & \\
[r/w] & \texttt{while}\ b\ \texttt{do}\ t\ \texttt{od} & = & \texttt{while}\ b\ \texttt{do}\ [r/w]\, t\ \texttt{od} & \\
[r/w] & \texttt{resource}\ w'\ \texttt{do}\ t\ \texttt{od} & = & \texttt{resource}\ w'\ \texttt{do}\ [r/w]\, t\ \texttt{od} & \text{if } w \neq w' \\
[r/w] & \texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od} & = & \texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od} & \\
[r/w] & \texttt{with}\ r'\ \texttt{do}\ t\ \texttt{od} & = & \texttt{with}\ r'\ \texttt{do}\ [r/w]\, t\ \texttt{od} & \\
[r/w] & \texttt{with}\ w'\ \texttt{do}\ t\ \texttt{od} & = & \begin{cases} \texttt{with}\ r\ \texttt{do}\ [r/w]\, t\ \texttt{od} & \text{if } w = w' \\ \texttt{with}\ w'\ \texttt{do}\ [r/w]\, t\ \texttt{od} & \text{otherwise} \end{cases} &
\end{array}
$$

Figure 3.2: Free variables, free resources and substitution

The syntax of the language that we will consider is presented in Figure 3.1. We write ε for the nil process, which can perform no action and is considered to have terminated whenever it is in its initial state. The symbol α is used to range over *heap actions*, which are actions on the heap that might change the values held at locations but do not affect the domain of definition of the heap. That is, they neither allocate nor deallocate locations. We reserve the symbol b for *boolean guards*, which are heap actions that may only proceed, without changing the heap, if the boolean b holds.

Provision for allocation within our language is made via the $\texttt{alloc}(\ell)$ primitive, which makes a location current (sometimes said to be 'allocated', *i.e.* part of the active heap), giving it an initial value, and sets ℓ to point at this location. It is worth repeating that the location ℓ is not itself being allocated; the location ℓ must initially be current and will end up pointing to the newly-allocated location. Consequently, some locations must be assumed to be allocated (current) before the process starts running. The justification for

this rather subtle definition is to avoid having to introduce program variables. By restricting attention just to heap locations, we avoid some untidiness in the original presentation of concurrent separation logic; in a sense, the semantics presented here abstracts away the difference between stack and heap locations (see the introduction to Chapter 4 on page 61 for a slightly fuller account).

For symmetry, the command $\texttt{dealloc}(\ell)$ makes the location pointed to by ℓ non-current. Writing a heap as the set of values that it holds for each allocated location, the effect of the command $\texttt{alloc}(\ell)$ on the heap $\{\ell \mapsto 0\}$ might be to form a heap $\{\ell \mapsto \ell', \ell' \mapsto 1\}$ if the location ℓ', not ℓ itself, is chosen to be allocated and is assigned initial value 1. The effect of the command $\texttt{dealloc}(\ell)$ on the heap $\{\ell \mapsto \ell', \ell' \mapsto 1\}$ would be to form the heap $\{\ell \mapsto \ell'\}$.

The *guarded sum* $\alpha.t + \alpha'.t'$ is a process that executes as t if α takes place or as t' if α' takes place. We refer the reader to Section 3.3 for a brief justification for disallowing non-guarded sums.

The semantics of the term $\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}$ will involve first picking a 'fresh' resource r and then using r for w in t, so running $[r/w]t$. It will therefore be necessary to record during the execution of processes which resources are *current* (*i.e.* not fresh) as well as which current resources are *available* (*i.e.* not held by any process).

The way in which we shall formally model the state in which processes execute is motivated by the way in which we shall give the net semantics to closed terms. We begin by defining the 'constructor' curr which takes either a location ℓ or resource r to give an element $\mathsf{curr}(\ell)$ or $\mathsf{curr}(r)$ to represent that the location or resource is current. Now define the following sets:

$$\begin{array}{rcl} \mathbf{D} & \triangleq & \mathrm{Loc} \times \mathrm{Val} \\ \mathbf{L} & \triangleq & \{\mathsf{curr}(\ell) \mid \ell \in \mathrm{Loc}\} \\ \mathbf{R} & \triangleq & \mathrm{Res} \\ \mathbf{N} & \triangleq & \{\mathsf{curr}(r) \mid r \in \mathrm{Res}\}. \end{array}$$

A state, ranged over by σ, is defined to be a tuple

$$(D, L, R, N)$$

where $D \subseteq \mathbf{D}$ represents the values held by locations in the heap; $L \subseteq \mathbf{L}$ represents the set of current, or allocated, locations of the heap; $R \subseteq \mathbf{R}$ represents the set of available resources; and $N \subseteq \mathbf{N}$ represents the set of current resources. The sets $\mathbf{D}$, $\mathbf{L}$, $\mathbf{R}$ and $\mathbf{N}$ are disjoint, so no ambiguity arises from writing, for example, $(\ell, v) \in \sigma$ to mean $(\ell, v) \in D$.

The interpretation of a state for the heap is that $(\ell, v) \in D$ if ℓ holds value v and that $\mathsf{curr}(\ell) \in L$ if ℓ is current. For resources, $r \in R$ if the resource r is available and $\mathsf{curr}(r) \in N$ if r is current. It is clear that only certain such tuples of subsets are sensible. In particular, the heap must be defined precisely on the set of current locations, and only current resources may be available.

Definition 3.1.1 (Consistent state)**.** *The state* (D, L, R, N) *is* consistent *if we have:*

- *the sets* D*,* L*,* R *and* N *are all finite,*
- D *is a partial function: for all* ℓ, v *and* v'*, if* $(\ell, v) \in D$ *and* $(\ell, v') \in D$ *then* $v = v'$*,*
- L *represents the locations current consistent with the domain of* D*:* $L = \{\mathsf{curr}(\ell) \mid \exists v : (\ell, v) \in D\}$*, and*
- *all available resources are current:* $R \subseteq \{r \mid \mathsf{curr}(r) \in N\}$*.*

It is clear to see that the L component of any given consistent state may be inferred from the D component. It will, however, be useful to retain this information separately for when the net semantics is given. We shall call $D \subseteq \mathbf{D}$ a *heap* when it is a finite partial function from locations to values, and write Heap for the set of all heaps. (Partial functions are identified here with their graphs, *i.e.* pairs of locations and values.) The notation $\ell \mapsto v$ is used for elements of heaps rather than (ℓ, v). We shall frequently make use of the standard definition of the domain of a heap D regarded as a partial function:

$$\mathrm{dom}(D) \triangleq \{\ell \mid \exists v.(\ell \mapsto v) \in D\}.$$

3.2 Process models

The definition of state that we have adopted permits a net semantics to be defined. Before doing so, we shall define how heap actions are to be interpreted and then give a transition semantics to closed terms.

Actions

The earlier definition of state allows a very general form of heap action to be defined that forms a basis for both the transition and net semantics. We assume that we are given the semantics of primitive actions α as $\mathcal{A}[\![\alpha]\!]$ comprising a set of heap pairs:

$$\mathcal{A}[\![\alpha]\!] \subseteq \mathrm{Heap} \times \mathrm{Heap}$$

The interpretation is that α can proceed in heap D if there are $(D_1, D_2) \in \mathcal{A}[\![\alpha]\!]$ such that D has the same value as D_1 wherever D_1 is defined. The resulting heap is formed by updating D to have the same value as D_2 wherever it is defined. It is significant that this definition allows us to infer precisely the set of locations upon which an action depends.

We require that whenever $(D_1, D_2) \in \mathcal{A}[\![\alpha]\!]$, the heaps D_1 and D_2 have the same domain. This means that the primitive actions ranged over by α will not allow allocation or deallocation, though the generality of the construction does allow them to represent abstractions over procedures. This requirement on the domains of D_1 and D_2 ensures that actions preserve consistent markings (Proposition 3.3).

As an example, we introduce some syntax to show how assignment of some very simple expressions might be defined.

Example 3.2.1 (Assignment)**.** *For any location ℓ and value v* that is not a location*, the action $\ell := v$ assigns the value v to location ℓ. Its semantics is:*

$$\mathcal{A}[\![\ell := v]\!] \triangleq \{(\{\ell \mapsto v'\}, \{\ell \mapsto v\}) \mid v' \in \mathrm{Val}\} \qquad \text{if } v \notin \mathrm{Loc}$$

If we wish to assign a particular location, we use the syntax $\ell := \&\ell'$. Its semantics is:

$$\mathcal{A}[\![\ell := \&\ell']\!] \triangleq \{(\{\ell \mapsto v'\}, \{\ell \mapsto \ell'\}) \mid v' \in \mathrm{Val}\}$$

The syntax $\ell := \ell'$ represents the action that copies the value held at location ℓ' to location ℓ. Its semantics is as follows:

$$\mathcal{A}[\![\ell := \ell']\!] \triangleq \{(\{\ell \mapsto v, \ell' \mapsto v'\}, \{\ell \mapsto v', \ell' \mapsto v'\}) \mid v, v' \in \mathrm{Val}\}$$

The syntax $\ell := [\ell']$ represents the action that copies the value at the location pointed to by ℓ' to ℓ.

$$\mathcal{A}\,[\![\ell := [\ell']]\!] \triangleq \begin{array}{l}\{(\{\ell \mapsto v, \ell' \mapsto \ell'', \ell'' \mapsto v''\},\\ \quad \{\ell \mapsto v'', \ell' \mapsto \ell', \ell'' \mapsto v''\}) \mid \ell \in \ell' \ \&\ v, v'' \in \mathrm{Val}\}\end{array}$$

Finally, the syntax $[\ell] := v$ indicates that the value v is assigned to the location pointed to by ℓ.

$$\mathcal{A}\,[\![[\ell] := v]\!] \triangleq \begin{array}{l}\{(\{\ell \mapsto \ell', \ell' \mapsto v'\},\\ \quad \{\ell \mapsto \ell', \ell' \mapsto v\}) \mid v' \in \mathrm{Val}\}\end{array}$$

Other forms of assignment such as $[\ell] := \&\ell'$ and $[\ell] := [\ell']$ can also be defined straightforwardly.

We can also define booleans as special kinds of guard action within this framework.

Example 3.2.2 (Booleans)**.** *Boolean guards b are actions that wait until the boolean expression holds and may then take place; they do not update the state. A selection of literals may be defined. For example:*

$$\begin{array}{rcll}\mathcal{A}\,[\![\ell = v]\!] & \triangleq & \{(\{\ell \mapsto v\}, \{\ell \mapsto v\})\} & \text{if } v \notin \mathrm{Loc}\\ \mathcal{A}\,[\![\ell = \&\ell']\!] & \triangleq & \{(\{\ell \mapsto \ell'\}, \{\ell \mapsto \ell'\})\} & \\ \mathcal{A}\,[\![\ell = \ell']\!] & \triangleq & \{(\{\ell \mapsto v, \ell' \mapsto v\}, \{\ell \mapsto v, \ell' \mapsto v\}) \mid v \in \mathrm{Val}\} & \end{array}$$

The first gives the semantics of an action that proceeds only if ℓ holds value v, the second gives the semantics of an action that proceeds only if ℓ points to ℓ', and the third gives the semantics of an action that proceeds only if the locations ℓ and ℓ' hold the same value.

Since boolean actions shall not modify the heap, they shall possess the property that:

$$\text{if } (D_1, D_2) \in \mathcal{A}\,[\![b]\!] \text{ then } D_1 = D_2.$$

This is preserved by the operations defined below. For heaps D and D', we use $D \uparrow D'$ to mean that D and D' are compatible as partial functions. This means that their union $D \cup D'$ must also be a partial function. Otherwise, i.e. if they disagree on the values assigned to a common location, we write $D \not\uparrow D'$.

$$\begin{array}{rcl}\mathcal{A}\,[\![\mathtt{true}]\!] & \triangleq & \{(\emptyset, \emptyset)\}\\ \mathcal{A}\,[\![\mathtt{false}]\!] & \triangleq & \emptyset\\ \mathcal{A}\,[\![b \wedge b']\!] & \triangleq & \{(D \cup D', D \cup D') \mid D \uparrow D' \text{ and } (D, D) \in \mathcal{A}\,[\![b]\!]\\ & & \qquad \text{and } (D', D') \in \mathcal{A}\,[\![b']\!]\}\\ \mathcal{A}\,[\![b \vee b']\!] & \triangleq & \mathcal{A}\,[\![b]\!] \cup \mathcal{A}\,[\![b']\!]\\ \mathcal{A}\,[\![\neg b]\!] & \triangleq & \{(D, D) \mid D \text{ is a } \subseteq\text{-minimal heap s.t.}\\ & & \qquad \forall D' \text{ if } (D', D') \in \mathcal{A}\,[\![b]\!] \text{ then } D \not\uparrow D'\}\end{array}$$

By insisting on minimality in the clause for $\neg b$, we form an action that is defined at as few locations as possible to refute all grounds for b (i.e. show that b cannot hold in the given heap).

Transition semantics

As an aid to understanding the net model, and in particular to give a model with respect to which we can prove its correspondence, a transition semantics for *closed* terms (terms such that $\mathrm{fv}(t) = \emptyset$) is given in Figure 3.3. A formal relationship between the two semantics

is presented in Theorem 3.4. The transition semantics is given by means of a labelled transition relation of the form $\langle t, \sigma\rangle \xrightarrow{\lambda} \langle t', \sigma'\rangle$ indicating that t performs an action labelled λ in the state σ to yield a resumption t' and a state σ'.

Labels in the transition semantics form a set Lab ranged over by λ following the grammar

λ	::=	$\mathsf{act}(D_1, D_2)$	heap action
	|	$\mathsf{bool}(D_0)$	boolean
	|	$\mathsf{alloc}(\ell, v, \ell', v')$	heap allocation
	|	$\mathsf{dealloc}(\ell, \ell', v)$	heap disposal
	|	$\mathsf{decl}(r)$	resource declaration
	|	$\mathsf{end}(r)$	end of resource scope
	|	$\mathsf{acq}(r)$	resource acquisition (critical region entry)
	|	$\mathsf{rel}(r)$	resource release (critical region exit).

The label $\mathsf{act}(D_1, D_2)$ indicates that an action, α say, with $(D_1, D_2) \in \mathcal{A}[\![\alpha]\!]$ occurs. Recalling how actions are specified, this means that D_1 and D_2 have the same domain and the resulting state is obtained by removing the subheap D_1 from the initial heap and replacing it with D_2. The label $\mathsf{bool}(D_0)$, representing a boolean action, is considered to be synonymous with $\mathsf{act}(D_0, D_0)$ (it is only included in the grammar above for clarity in later reference). The label $\mathsf{alloc}(\ell, v, \ell', v')$ represents the location ℓ' being chosen to become current, initially receiving value v', with a pointer to the new location being placed in ℓ which previously held value v. The label $\mathsf{dealloc}(\ell, \ell', v')$ indicates that the location ℓ', which initially holds value v', pointed to by ℓ is deallocated. The other labels are hopefully easily understood.

We write $\sigma \oplus \sigma'$ for the union of the components of two states where they are disjoint and impose the implicit side-condition in the transition semantics that this is defined wherever it is used. For example, this implicit side-condition means, in the rule (Alloc), that for a transition from $\langle t, \sigma\rangle$ labelled $\mathsf{alloc}(\ell, v, \ell', v')$ to take place we must have $\mathsf{curr}(\ell') \notin \sigma$, and hence ℓ' was initially non-current. Similarly, the rule (Res) can only be applied to derive a transition labelled $\mathsf{decl}(r)$ if the resource r was not initially current.

In the transition semantics, recalling that the nil process is written ε, we have $\langle t, \sigma\rangle \xrightarrow{\lambda} \langle \varepsilon, \sigma'\rangle$ if the process t can perform an action labelled λ to terminate, leaving the state σ'. For actions α_1 and α_2, using the rule (Seq) and (Act), we might derive $\langle \alpha_1; \alpha_2, \sigma\rangle \xrightarrow{\lambda_1} \langle \varepsilon; \alpha_2, \sigma_1\rangle$ for appropriate λ, σ and σ_1. To remove the leading ε from the resumption in order to derive a subsequent transition, we introduce *equivalence* on terms.

Definition 3.2.1 (Structural equivalence)**.** *Equivalence on closed terms $\equiv$ is the least equivalence relation satisfying the usual rules for the associativity of ;, + and $\|$ and for the commutativity of $\|$ and +, along with the following axioms and congruence rules:*

$$\varepsilon; t \equiv t \qquad \varepsilon \parallel t \equiv t$$

$$\textit{if } t_1 \equiv t_2 \textit{ then } \begin{cases} t_1; t \equiv t_2; t \\ t; t_1 \equiv t; t_2 \\ t_1 \parallel t \equiv t_2 \parallel t \\ \alpha.t_1 + \alpha_0.t_0 \equiv \alpha.t_2 + \alpha_0.t_0 \\ \texttt{while}\ b\ \texttt{do}\ t_1\ \texttt{od} \equiv \texttt{while}\ b\ \texttt{do}\ t_2\ \texttt{od} \\ \texttt{with}\ r\ \texttt{do}\ t_1\ \texttt{od} \equiv \texttt{with}\ r\ \texttt{do}\ t_2\ \texttt{od} \end{cases}$$

If there exists r such that $t_1[r/w] \equiv t_2[r/w]$ and $r \notin \mathrm{res}(t_1) \cup \mathrm{res}(t_2)$ then

$$\texttt{resource}\ w\ \texttt{do}\ t_1\ \texttt{od} \equiv \texttt{resource}\ w\ \texttt{do}\ t_2\ \texttt{od}$$

The syntax of terms is extended temporarily to include `rel` r and `end` r which are special terms used in the rules (Rel) and (End) in Figure 3.3 to follow. These, respectively, are attached to the ends of terms protected by critical regions and the ends of terms in which a resource was declared.

For conciseness, we do not give an error semantics to situations in which non-current locations or resources are used; instead, the process will become stuck. We show in Section 5.4 that such situations are excluded by concurrent separation logic.

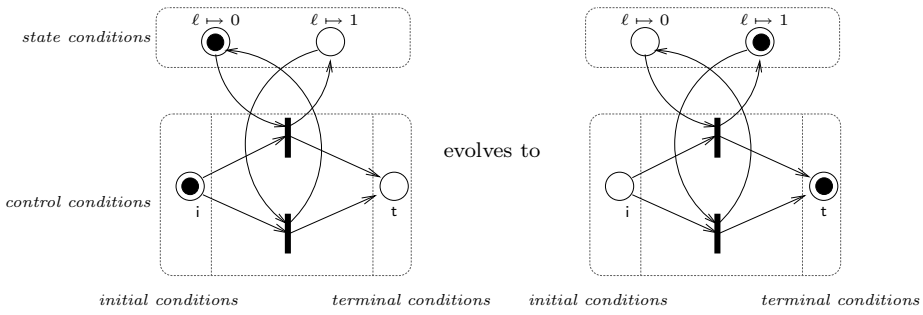
It is worth repeating a couple of points on the operational semantics:

- In the rule (Act), if α is a boolean guard b then $D = D'$.

- The notation $\{\ell \mapsto v\}$ is used to represent the state (D, L, R, N) where $R = N = \emptyset$ and $D = \{\ell \mapsto v\}$ and $L = \{\mathsf{curr}(\ell)\}$. Similarly, $\{r\}$ represents the state (D, L, R, N) where $D = L = N = \emptyset$ and $R = \{r\}$, and so on.

3.3 Net semantics

Before giving the formal definition of the net semantics of closed terms, by means of an example we shall illustrate how our semantics shall be defined. First, we shall draw the semantics of an action `toggle`$(\ell, 0, 1)$ that toggles the value held at a location ℓ between 0 and 1.



Notice that in the above net, as in general, there are conditions to represent the *shared state* in which processes execute, including for example the values held at locations (here ℓ; in diagrams, we shall tend only to draw conditions that are actually used by events in the net). There are also conditions to represent the *control* point of the process. The net pictured on the left is in its *initial marking* of control conditions and the net on the right is in its *terminal marking* of control conditions, indicating successful completion of the process following the toggle of the value; the marking of the net initially had the state condition $\ell \mapsto 0$ marked and finished with the condition $\ell \mapsto 1$ marked. There is an event present in the net for each way that the action could take place: one event for toggling the value from 0 to 1 and another event for toggling the value from 1 to 0. Only the first event could occur in the initial marking of the net on the left, and no event can occur in the marking on the right since the control conditions are not appropriately marked.

The parallel composition `toggle`$(\ell, 0, 1) \parallel$ `toggle`$(\ell, 0, 1)$ can be formed by taking two copies of the net `toggle`$(\ell, 0, 1)$ and forcing them to operate on disjoint sets of control conditions.

(Act) : $$\frac{(D_1, D_2) \in \mathcal{A}[\![\alpha]\!] \qquad D_1 \subseteq D \qquad D' = (D \setminus D_1) \cup D_2}{\langle \alpha, (D, L, R, N)\rangle \xrightarrow{\mathsf{act}(D_1, D_2)} \langle \varepsilon, (D', L, R, N)\rangle}$$

(Alloc) : $$\langle \mathtt{alloc}(\ell), \sigma \oplus \{\ell \mapsto v\}\rangle \xrightarrow{\mathsf{alloc}(\ell, v, \ell', v')} \langle \varepsilon, \sigma \oplus \{\ell \mapsto \ell', \ell' \mapsto v', \mathsf{curr}(\ell')\}\rangle$$

(Dealloc) : $$\langle \mathtt{dealloc}(\ell), \sigma \oplus \{\ell \mapsto \ell', \ell' \mapsto v', \mathsf{curr}(\ell')\}\rangle \xrightarrow{\mathsf{dealloc}(\ell, \ell', v')} \langle \varepsilon, \sigma \oplus \{\ell \mapsto \ell'\}\rangle$$

(Seq) : $$\frac{\langle t_1, \sigma\rangle \xrightarrow{\lambda} \langle t_1', \sigma'\rangle}{\langle t_1; t_2, \sigma\rangle \xrightarrow{\lambda} \langle t_1'; t_2, \sigma'\rangle}$$

(Par-1) : $$\frac{\langle t_1, \sigma\rangle \xrightarrow{\lambda} \langle t_1', \sigma'\rangle}{\langle t_1 \parallel t_2, \sigma\rangle \xrightarrow{\lambda} \langle t_1' \parallel t_2, \sigma'\rangle}$$

(Sum-1) : $$\frac{\langle \alpha_1, \sigma\rangle \xrightarrow{\lambda} \langle \varepsilon, \sigma'\rangle}{\langle \alpha_1.t_1 + \alpha_2.t_2, \sigma\rangle \xrightarrow{\lambda} \langle t_1, \sigma'\rangle}$$

(While) : $$\frac{\langle b, \sigma\rangle \xrightarrow{\lambda} \langle \varepsilon, \sigma\rangle}{\langle \mathtt{while}\ b\ \mathtt{do}\ t\ \mathtt{od}, \sigma\rangle \xrightarrow{\lambda} \langle t; \mathtt{while}\ b\ \mathtt{do}\ t\ \mathtt{od}, \sigma\rangle}$$

(While′) : $$\frac{\langle \neg b, \sigma\rangle \xrightarrow{\lambda} \langle \varepsilon, \sigma\rangle}{\langle \mathtt{while}\ b\ \mathtt{do}\ t\ \mathtt{od}, \sigma\rangle \xrightarrow{\lambda} \langle \varepsilon, \sigma\rangle}$$

(With) : $$\langle \mathtt{with}\ r\ \mathtt{do}\ t\ \mathtt{od}, \sigma \oplus \{r\}\rangle \xrightarrow{\mathsf{acq}(r)} \langle t; \mathtt{rel}\ r, \sigma\rangle$$

(Rel) : $$\langle \mathtt{rel}\ r, \sigma\rangle \xrightarrow{\mathsf{rel}(r)} \langle \varepsilon, \sigma \oplus \{r\}\rangle$$

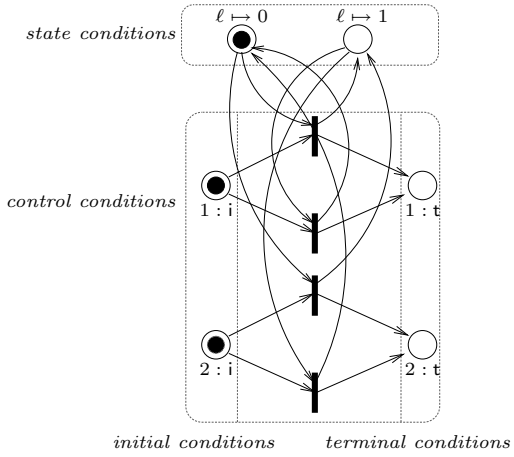
(Res) : $$\langle \mathtt{resource}\ w\ \mathtt{do}\ t\ \mathtt{od}, \sigma\rangle \xrightarrow{\mathsf{decl}(r)} \langle [r/w]t; \mathtt{end}\ r, \sigma \oplus \{r, \mathsf{curr}(r)\}\rangle \quad (r \notin \mathrm{res}(t))$$

(End) : $$\langle \mathtt{end}\ r, \sigma \oplus \{r, \mathsf{curr}(r)\}\rangle \xrightarrow{\mathsf{end}(r)} \langle \varepsilon, \sigma\rangle$$

(Equiv) : $$\frac{t \equiv t_0 \qquad \langle t_0, \sigma\rangle \xrightarrow{\lambda} \langle t_0', \sigma'\rangle \qquad t_0' \equiv t'}{\langle t, \sigma\rangle \xrightarrow{\lambda} \langle t', \sigma'\rangle}$$

Symmetric rules (Par-2) *and* (Sum-2) *omitted.*

Figure 3.3: Transition semantics

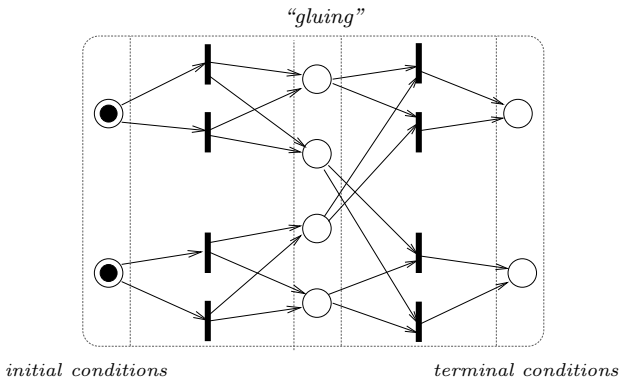


An example run of this net would involve first the top event changing the value of ℓ from 0 to 1 and then the bottom event changing ℓ back from 1 to 0. The resulting marking of control conditions would be equal to the terminal conditions of the net, so no event would have concession in this marking.

The net representing the sequential composition

$$(\texttt{toggle}(\ell, 0, 1) \parallel \texttt{toggle}(\ell, 0, 1)); (\texttt{toggle}(\ell, 0, 1) \parallel \texttt{toggle}(\ell, 0, 1))$$

is formed by a 'gluing' operation that joins the terminal conditions of one copy of the net for $\texttt{toggle}(\ell, 0, 1) \parallel \texttt{toggle}(\ell, 0, 1)$ to the initial conditions of another copy of the net for $\texttt{toggle}(\ell, 0, 1) \parallel \texttt{toggle}(\ell, 0, 1)$. (In this example net, for clarity we shall not show the state conditions.)



The set of 'glued' conditions can only all be marked once the left hand side of the sequential composition has terminated. It is only at this point that the net on the right hand side can begin.

Net structure

As outlined above, within the nets that we give for processes we distinguish two forms of condition, namely *state conditions* and *control conditions*. The markings of these sets of conditions determine the state in which the process is executing and the its control point, respectively.

The marking of state conditions shall correspond to a state introduced earlier as a tuple (D, L, R, N) where D represents the heap, L represents the set of current locations, R represents the set of available resources and N represents the set of current resources.

When we give the net semantics of a term, we will make use of the closure of the set of control conditions under various operations. The use of the following definition will hopefully become clear as the semantics is given. Roughly, we use the ability to pair control conditions to define the gluing operation above. We use the ability to tag control conditions to force the sub-nets representing components of the term to be disjoint (for example, to separate the control conditions used by t_1 from those used by t_2 in the net for $t_1 \parallel t_2$).

Definition 3.3.1 (Conditions)**.** *Define the set of* state conditions $\mathbf{S}$ *to be* $\mathbf{D} \cup \mathbf{L} \cup \mathbf{R} \cup \mathbf{N}$. *Define the set of control conditions* $\mathbf{C}$*, ranged over by* c*, to be the least set such that:*

- $\mathbf{C}$ *contains distinguished elements* i *and* t*, standing for 'initial' and 'terminal', respectively.*
- *If* $c \in \mathbf{C}$ *and* $c' \in \mathbf{C}$ *then* $(c, c') \in \mathbf{C}$ *to allow the 'gluing' operation.*
- *If* $c \in \mathbf{C}$ *then* $a : c \in \mathbf{C}$ *for a 'tag'*

$$\begin{array}{rcllll} a & ::= & \mathsf{seq}\,1 & | & \mathsf{seq}\,2 & \\ & | & \mathsf{par}\,1 & | & \mathsf{par}\,2 & \\ & | & \mathsf{sum}\,1 & | & \mathsf{sum}\,2 & \\ & | & \mathsf{body} & & & \\ & | & \mathsf{res}\,r & & \textit{for any } r \in \mathrm{Res} & \end{array}$$

A state $\sigma = (D, L, R, N)$ corresponds to the marking $D \cup L \cup R \cup N$ of state conditions in the obvious way. Similarly, if C is a marking of control conditions and σ is a state, the pair (C, σ) corresponds to the marking $C \cup \sigma$. We therefore use the notations interchangeably.

The nets that we form shall be *extensional* in the sense that two events are equal if they have the same preconditions and the same postconditions. An event can therefore be regarded as a tuple

$$e = (C, \sigma, C', \sigma')$$

with preconditions ${}^\bullet e \triangleq C \cup \sigma$ and postconditions $e^\bullet \triangleq C' \cup \sigma'$. To obtain a concise notation for working with events, we write ${}^{\mathbf{C}}e$ for the pre-control conditions of e:

$${}^{\mathbf{C}}e \triangleq {}^\bullet e \cap \mathbf{C}.$$

We likewise define notations $e^{\mathbf{C}}$, ${}^{\mathbf{D}}e$, ${}^{\mathbf{L}}e$ *etc.*, and call these the *components* of e by virtue of the fact that it is sufficient to define an event through the definition of its components. The pre-state conditions of e are ${}^{\mathbf{S}}e = {}^{\mathbf{D}}e \cup {}^{\mathbf{L}}e \cup {}^{\mathbf{R}}e \cup {}^{\mathbf{N}}e$, and we define the post-state conditions $e^{\mathbf{S}}$ similarly.

Two markings of control conditions are of particular importance: those marked when the process starts executing and those marked when the process has terminated. We call these the *initial* control conditions I and *terminal* control conditions T, respectively. We shall call a net with a partition of its conditions into control and state with the subsets of control conditions I and T an *embedded net*. For an embedded net N, we write $\mathrm{Ic}(N)$ for I and $\mathrm{Tc}(N)$ for T, and we write $\mathrm{Ev}(N)$ for its set of events. Observe that no initial marking of state conditions is yet specified.

The semantics of a closed term t shall be an embedded net, written $\mathcal{N}[\![t]\!]$. No confusion arises, so we shall write $\mathrm{Ic}(t)$ for $\mathrm{Ic}(\mathcal{N}[\![t]\!])$, and $\mathrm{Tc}(t)$ and $\mathrm{Ev}(t)$ for $\mathrm{Tc}(\mathcal{N}[\![t]\!])$ and $\mathrm{Ev}(\mathcal{N}[\![t]\!])$, respectively. The nets formed shall always have the same sets of control and state conditions; the difference shall arise in the events present in the nets. It is a trivial

matter to restrict to the conditions that are actually used, and we specify these after we give the semantics.

As we give the semantics of closed terms, we will make use of several constructions on nets. For example, we wish the events of parallel processes to operate on disjoint sets of control conditions. This is conducted using a *tagging* operation on events. For a tag a as defined in Definition 3.3.1, we define $a : e$ to be the event e changed so that

$$^{\mathbf{c}}(a : e) \triangleq \{a : c \mid c \in {}^{\mathbf{c}}e\} \qquad (a : e)^{\mathbf{c}} \triangleq \{a : c \mid c \in e^{\mathbf{c}}\}$$

but otherwise unchanged in its action on state conditions. The notations are extended pointwise to sets of events:

$$a : E \triangleq \{a : e \mid e \in E\}$$

Another useful operation is what we call *gluing* two embedded nets together. For example, when forming the sequential composition of processes $t_1; t_2$, we want to enable the events of t_2 when t_1 has terminated. This is done by 'gluing' the two nets together at the terminal conditions of t_1 and the initial conditions of t_2, having made them disjoint on control conditions using tagging. Wherever a terminal condition c of $\mathrm{Tc}(t_1)$ occurs as a pre- or a postcondition of an event of t_1, every element of the set $\{\mathsf{seq}\ 1{:}c\} \times (\mathsf{seq}\ 2{:}\mathrm{Ic}(t_2))$ would occur in its place. Similarly, the events of t_2 use the set of conditions $(\mathsf{seq}\ 1{:}\mathrm{Tc}(t_1)) \times \{\mathsf{seq}\ 2{:}c'\}$ instead of an initial condition c' of $\mathrm{Ic}(t_2)$. A variety of control properties that the nets we form possess (Lemma 3.5.1), such as that all events have at least one pre-control condition, allows us to infer that it is impossible for an event of t_2 to occur before t_1 has terminated, and thereon it is impossible for t_1 to resume. An example follows shortly.

Assume a set $P \subseteq \mathbf{C} \times \mathbf{C}$. Useful definitions to represent gluing are:

$$\begin{aligned} P \triangleleft C \quad \triangleq \quad & \{(c_1, c_2) \mid c_1 \in C \text{ and } (c_1, c_2) \in P\} \\ & \cup \{c_1 \mid c_1 \in C \text{ and } \nexists c_2.(c_1, c_2) \in P\} \end{aligned}$$

$$\begin{aligned} P \triangleright C \quad \triangleq \quad & \{(c_1, c_2) \mid c_2 \in C \text{ and } (c_1, c_2) \in P\} \\ & \cup \{c_2 \mid c_2 \in C \text{ and } \nexists c_1.(c_1, c_2) \in P\} \end{aligned}$$

The first definition, $P \triangleleft C$, indicates that an occurrence of c_1 in C is to be replaced by occurrences of (c_1, c_2) for every c_2 such that (c_1, c_2) occurs in P. The second definition, $P \triangleright C$, indicates that an occurrence of c_2 in C is to be replaced by occurrences of (c_1, c_2) for every c_1 such that (c_1, c_2) occurs in P.

Lemma 3.3.1. *For any $C, C' \subseteq \mathbf{C}$, any $P \subseteq \mathbf{C} \times \mathbf{C}$ and any tag a, the following hold:*

$$\begin{array}{rcccl} a : C & \subseteq & a : C' & \iff & C \subseteq C' \\ P \triangleleft C & \subseteq & P \triangleleft C' & \iff & C \subseteq C' \\ P \triangleright C & \subseteq & P \triangleright C' & \iff & C \subseteq C' \end{array}$$

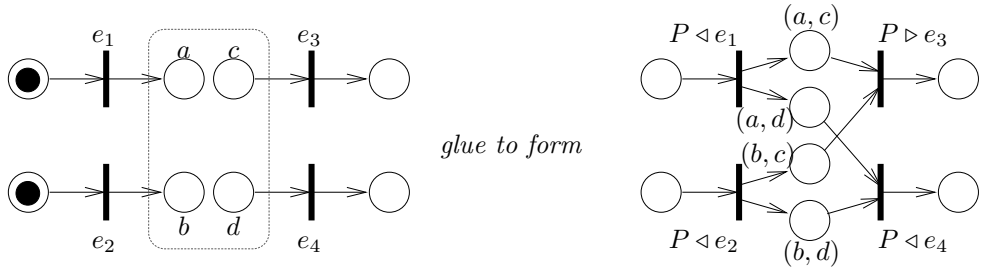
Proof. Straightforward calculation. □

The notation is extended to events to give an event $P \triangleleft e$ in the following way, recalling that gluing will only affect the control conditions used by an event and in particular not its state conditions:

$$\begin{array}{rclcrcl} ^{\mathbf{c}}(P \triangleleft e) & \triangleq & P \triangleleft ({}^{\mathbf{c}}e) & \qquad & (P \triangleleft e)^{\mathbf{c}} & \triangleq & P \triangleleft (e^{\mathbf{c}}) \\ ^{\mathbf{s}}(P \triangleleft e) & \triangleq & {}^{\mathbf{s}}e & & (P \triangleleft e)^{\mathbf{s}} & \triangleq & e^{\mathbf{s}} \end{array}$$

The notation $P \triangleright e$ is defined similarly, and it is also extended to sets of events in the obvious pointwise manner. For any marking $M = (C, \sigma)$, we will write $P \triangleleft M$ for $(P \triangleleft C, \sigma)$ and similarly write $P \triangleright M$ for $(P \triangleright C, \sigma)$.

To give an example, consider the gluings $P \triangleleft C_1$ and $P \triangleright C_2$ where $C_1 = \{a, b\}$ and $C_2 = \{c, d\}$ are joined at $P = C_1 \times C_2$. Applying $P \triangleleft C_1$ to the left net and $P \triangleright C_2$ to the right net below, this indicates how gluing is used to sequentially compose embedded nets:



The operations of gluing and tagging affect only the control flow of events, not their effect on the marking of state conditions.

Lemma 3.3.2. *Let N be an embedded net with control conditions $\mathbf{C}$. Suppose that $P \subseteq \mathbf{C} \times \mathbf{C}$. For any marking M of N and tag a:*

- $M \xrightarrow{e}\!\!\!\!\rightarrow M'$ *iff* $a : M \xrightarrow{a:e}\!\!\!\!\rightarrow a : M'$.
- $M \xrightarrow{e}\!\!\!\!\rightarrow M'$ *iff* $P \triangleleft M \xrightarrow{P \triangleleft e}\!\!\!\!\rightarrow P \triangleleft M'$, *and*
- $M \xrightarrow{e}\!\!\!\!\rightarrow M'$ *iff* $P \triangleright M \xrightarrow{P \triangleright e}\!\!\!\!\rightarrow P \triangleright M'$.

Furthermore:

- *if* $a : M \xrightarrow{a:e}\!\!\!\!\rightarrow M_1'$ *then* $M_1' = a : M'$ *for some* M',
- *if* $P \triangleleft M \xrightarrow{P \triangleleft e}\!\!\!\!\rightarrow M_1'$ *then* $M_1' = P \triangleleft M'$ *for some* M', *and*
- *if* $P \triangleright M \xrightarrow{P \triangleright e}\!\!\!\!\rightarrow M_2'$ *then* $M_2' = P \triangleright M'$ *for some* M'.

Proof. The first and fourth items are straightforward to prove. The remaining properties may be shown using the following easily-demonstrated equations, along with their counterparts for $\triangleright$, for any subset of control conditions C:

1. $C = \emptyset$ iff $P \triangleleft C = \emptyset$,
2. $P \triangleleft (C \setminus C') = (P \triangleleft C) \setminus (P \triangleleft C')$,
3. $P \triangleleft (C \cup C') = (P \triangleleft C) \cup (P \triangleleft C')$, and
4. $P \triangleleft (C \cap C') = (P \triangleleft C) \cap (P \triangleleft C')$. □

Semantics of processes

The net semantics that we now give for closed terms is defined by induction on the *size* of terms, given in the obvious way. The reason why it is not given by induction on terms themselves is that the semantics of `resource` w `do` t `od` is given according to the semantics of $[r/w]t$ for all resources r.

Nil process The nil process has no events, $\mathrm{Ev}(\varepsilon) = \emptyset$, and just one condition which is both initial and terminal:

$$\mathrm{Ic}(\varepsilon) = \mathrm{Tc}(\varepsilon) = \{(\mathsf{i}, \mathsf{t})\}.$$

Heap action Let α be an action and recall that $\mathcal{A}[\![\alpha]\!]$ is a set of pairs such that if $(D_1, D_2) \in \mathcal{A}[\![\alpha]\!]$ and D_1 is a subheap of the initial state then the occurrence of α may change the values held in locations within this subheap from those recorded in D_1 to those recorded in D_2.

Let $\mathsf{act}_{(C,C')}(D_1, D_2)$ denote an event e with

$$^{\mathbf{C}}e = C \quad e^{\mathbf{C}} = C' \quad ^{\mathbf{D}}e = D_1 \quad e^{\mathbf{D}} = D_2$$

and all other components empty, *i.e.* $^{\mathbf{L}}e = e^{\mathbf{L}} = {}^{\mathbf{R}}e = e^{\mathbf{R}} = {}^{\mathbf{N}}e = e^{\mathbf{N}} = \emptyset$. For an action α, we define:

$$\begin{array}{rcl} \mathrm{Ic}(\alpha) & \triangleq & \{\mathsf{i}\} \\ \mathrm{Tc}(\alpha) & \triangleq & \{\mathsf{t}\} \\ \mathrm{Ev}(\alpha) & \triangleq & \{\mathsf{act}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(D_1, D_2) \mid (D_1, D_2) \in \mathcal{A}[\![\alpha]\!]\}. \end{array}$$

There is an event representing each way that α can occur. The occurrence of any of these changes the marking of control conditions from the initial marking $\{\mathsf{i}\}$ to the terminal marking $\{\mathsf{t}\}$.

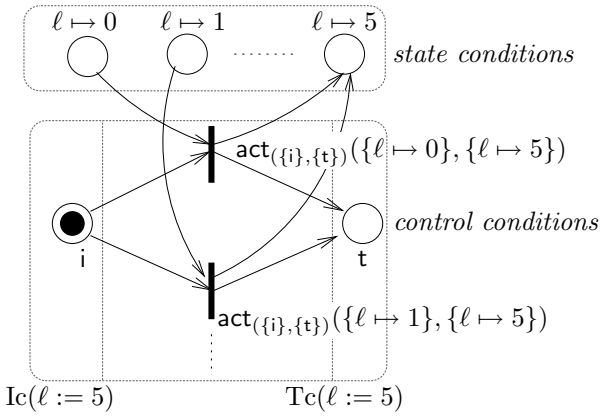
Example 3.3.1 ($\mathcal{N}[\![\ell := 5]\!]$)**.** *Recall that*

$$\mathcal{A}[\![\ell := 5]\!] = \{(\{\ell \mapsto v\}, \{\ell \mapsto 5\}) \mid v \in \mathrm{Val}\},$$

so

$$\mathrm{Ev}(\ell := 5) = \{\mathsf{act}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(\{\ell \mapsto v\}, \{\ell \mapsto 5\}) \mid v \in \mathrm{Val}\}.$$

The definitions give the net $\mathcal{N}[\![\ell := 5]\!]$*:*



So, no matter what value ℓ *initially holds, providing that there is some such value (i.e.* ℓ *is current), the condition* $\ell \mapsto 5$ *will become marked.*

Allocation and deallocation The command $\mathtt{alloc}(\ell)$ activates, by making current and assigning an arbitrary value to, a non-current location and sets ℓ to point at it. For symmetry, $\mathtt{dealloc}(\ell)$ deactivates the current location pointed to by ℓ.

We begin by defining two further event notations. First, the notation $\mathsf{alloc}_{(C,C')}(\ell, v, \ell', v')$ represents the event e such that $^{\mathbf{C}}e = C$ and $e^{\mathbf{C}} = C'$ and

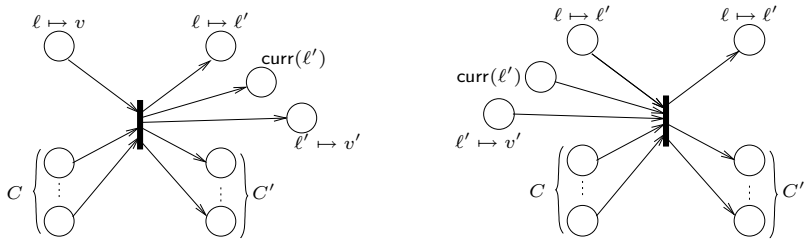
$$^{\mathbf{D}}e = \{\ell \mapsto v\} \qquad e^{\mathbf{D}} = \{\ell \mapsto \ell', \ell' \mapsto v'\} \qquad ^{\mathbf{L}}e = \emptyset \qquad e^{\mathbf{L}} = \{\mathsf{curr}(\ell')\},$$

and otherwise empty components, which changes ℓ' from being non-current to current, gives it value v' and changes the value held at ℓ from v to ℓ'. If the condition $\mathsf{curr}(\ell')$ is marked before the event takes place, contact occurs, so the event has concession only if the location ℓ' is not initially current. Second, $\mathsf{dealloc}_{(C,C')}(\ell, \ell', v')$ is the event e such that ${}^{\mathbf{C}}e = C$ and $e^{\mathbf{C}} = C'$ and

$$ {}^{\mathbf{D}}e = \{\ell \mapsto \ell', \ell' \mapsto v'\} \quad e^{\mathbf{D}} = \{\ell \mapsto \ell'\} \quad {}^{\mathbf{L}}e = \{\mathsf{curr}(\ell')\}, $$

which does the converse of allocation. The location ℓ is left with a dangling pointer to ℓ'. The two events may be drawn as:

$\mathsf{alloc}_{(C,C')}(\ell, v, \ell', v')$: $\mathsf{dealloc}_{(C,C')}(\ell, \ell', v')$:



The semantics of allocation is given by:

$$\begin{array}{rcl} \mathrm{Ic}(\texttt{alloc}(\ell)) & \triangleq & \{\mathsf{i}\} \\ \mathrm{Tc}(\texttt{alloc}(\ell)) & \triangleq & \{\mathsf{t}\} \\ \mathrm{Ev}(\texttt{alloc}(\ell)) & \triangleq & \{\mathsf{alloc}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(\ell, v, \ell', v') \mid \ell' \in \mathrm{Loc} \text{ and } \ v, v' \in \mathrm{Val}\}. \end{array}$$

Note that there is an event present for every value that ℓ might initially hold and every value that ℓ' might be assumed to take initially.

The semantics of disposal is given by:

$$\begin{array}{rcl} \mathrm{Ic}(\texttt{dealloc}(\ell)) & \triangleq & \{\mathsf{i}\} \\ \mathrm{Tc}(\texttt{dealloc}(\ell)) & \triangleq & \{\mathsf{t}\} \\ \mathrm{Ev}(\texttt{dealloc}(\ell)) & \triangleq & \{\mathsf{dealloc}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(\ell, \ell', v') \mid \ell' \in \mathrm{Loc} \text{ and } \ v' \in \mathrm{Val}\}. \end{array}$$

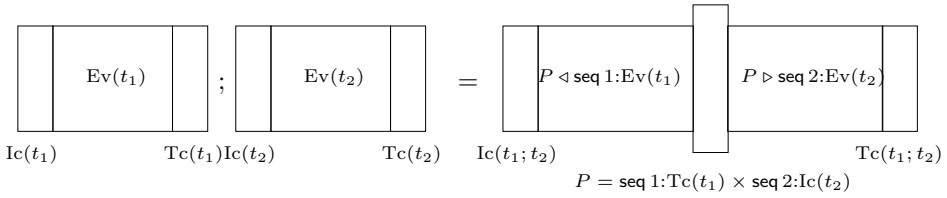
Sequential composition The sequential composition of terms involves gluing the terminal marking of the net for t_1 to the initial marking of the net for t_2. The operation is therefore performed on the set

$$P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2).$$

Following the intuitive account earlier, we take

$$\begin{array}{rcl} \mathrm{Ic}(t_1; t_2) & \triangleq & P \triangleleft \mathsf{seq}\,1{:}\mathrm{Ic}(t_1) \\ \mathrm{Tc}(t_1; t_2) & \triangleq & P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2) \\ \mathrm{Ev}(t_1; t_2) & \triangleq & (P \triangleleft \mathsf{seq}\,1{:}\mathrm{Ev}(t_1)) \cup (P \triangleright \mathsf{seq}\,2{:}\mathrm{Ev}(t_2)). \end{array}$$

The formation of the sequential composition on control conditions may be drawn schematically as:



Note that this diagram is slightly simplified since the sets $\mathrm{Ic}(t_1)$ and $\mathrm{Tc}(t_1)$ might not be disjoint, and similarly for t_2.

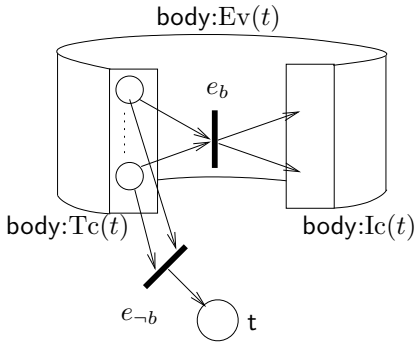
Parallel composition The control flow of the parallel composition of processes is autonomous; interaction occurs only through the state, so we have a form of 'disjoint union' of nets albeit with interaction allowed on shared state conditions. We therefore force the events of the two processes to work on disjoint sets of control conditions by giving them different tags:

$$\begin{aligned}
\mathrm{Ic}(t_1 \parallel t_2) &\triangleq \mathsf{par}\,1{:}\mathrm{Ic}(t_1) \cup \mathsf{par}\,2{:}\mathrm{Ic}(t_2)\\
\mathrm{Tc}(t_1 \parallel t_2) &\triangleq \mathsf{par}\,1{:}\mathrm{Tc}(t_1) \cup \mathsf{par}\,2{:}\mathrm{Tc}(t_2)\\
\mathrm{Ev}(t_1 \parallel t_2) &\triangleq \mathsf{par}\,1{:}\mathrm{Ev}(t_1) \cup \mathsf{par}\,2{:}\mathrm{Ev}(t_2).
\end{aligned}$$

Iteration To form the net for `while` b `do` t `od`, we tag the net for t as the 'body' of the loop and add events exhibiting the satisfaction of b from the terminal conditions of the body to its initial conditions along with events exhibiting the failure of b from the terminal conditions of the body to a newly introduced terminal condition, `t`. The net is in its initial control state when the body has terminated.

$$\begin{aligned}
\mathrm{Ic}(\texttt{while}\ b\ \texttt{do}\ t\ \texttt{od}) &\triangleq \mathsf{body}{:}\mathrm{Tc}(t)\\
\mathrm{Tc}(\texttt{while}\ b\ \texttt{do}\ t\ \texttt{od}) &\triangleq \{\mathsf{t}\}\\
\mathrm{Ev}(\texttt{while}\ b\ \texttt{do}\ t\ \texttt{od}) &\triangleq \mathsf{body}{:}\mathrm{Ev}(t)\\
&\quad \cup \{\mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t),\mathsf{body}:\mathrm{Ic}(t))}(D_1, D_1) \mid (D_1, D_1) \in \mathcal{A}[\![b]\!]\}\\
&\quad \cup \{\mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t),\{\mathsf{t}\})}(D_0, D_0) \mid (D_0, D_0) \in \mathcal{A}[\![\neg b]\!]\}
\end{aligned}$$

The loop formed can be visualized in the following way (in which we only present one event, e_b, for the boolean b and one event, $e_{\neg b}$, for the boolean $\neg b$):

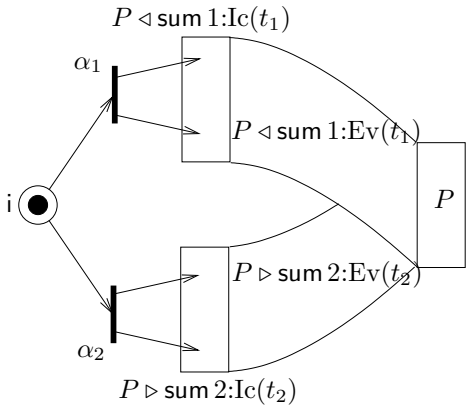


Guarded sum Let t be the term $\alpha_1.t_1 + \alpha_2.t_2$. The sum is formed by prefixing the actions onto the tagged nets representing the terms and then gluing the sets of

terminal conditions. Let $P = (\mathsf{sum}\ 1{:}\mathrm{Tc}(t_1)) \times (\mathsf{sum}\ 2{:}\mathrm{Tc}(t_2))$. Define:

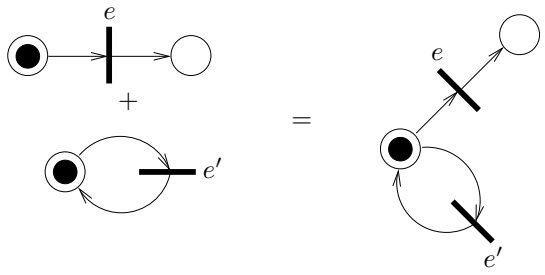
$$\begin{array}{rcl}
\mathrm{Ic}(t) & \triangleq & \{\mathsf{i}\} \\
\mathrm{Tc}(t) & \triangleq & P \\
\mathrm{Ev}(t) & \triangleq & \{\mathsf{act}_{(\{\mathsf{i}\},P \lhd \mathsf{sum}\ 1:\mathrm{Ic}(t_1))}(D_1, D_2) \mid (D_1, D_2) \in \mathcal{A}[\![\alpha_1]\!]\} \\
& & \cup\ \{\mathsf{act}_{(\{\mathsf{i}\},P \rhd \mathsf{sum}\ 2:\mathrm{Ic}(t_2))}(D_1, D_2) \mid (D_1, D_2) \in \mathcal{A}[\![\alpha_2]\!]\} \\
& & \cup\ P \lhd (\mathsf{sum}\ 1{:}\mathrm{Ev}(t_1))\ \cup\ P \rhd (\mathsf{sum}\ 2{:}\mathrm{Ev}(t_2)).
\end{array}$$

The net may be pictured schematically as follows, in which we have drawn only one representative event for each of α_1 and α_2, and have elided the effect of these events on state conditions.



On a technical point, one may wonder why the syntax of the language requires that sums possess guards. This is seemingly curious since the category of safe Petri nets, which intuitively underlies a category of embedded nets, has a coproduct construction. The definition of the unguarded sum $\mathcal{N}[\![t_1 + t_2]\!]$ arising from this construction would be similar to that above, apart from also gluing at the set $\mathsf{sum}\ 1{:}\mathrm{Ic}(t_1) \times \mathsf{sum}\ 2{:}\mathrm{Ic}(t_2)$.

However, as remarked in Section 5 of [Win87], there are cases where the coproduct of nets does not coincide with the usual interpretation of nondeterministic sum. In Section 3.3 of [Win86], this is explained as the occurrence net unfolding (the 'behaviour') of the coproduct of two nets not being equal to the coproduct of their respective unfoldings. To repeat an example given there, letting + represent coproduct in the category of safe nets, we have:



Consequently, using this coproduct as a definition of general sum, the (finite) runs of the net representing $\alpha + (\texttt{while true do}\ \alpha')$ would consist of some finite number

of executions of α' followed, possibly, by one of α. Quite clearly, this does not correspond to the normal understanding of nondeterminism presented in the transition semantics.

The restriction of processes to only use guarded sums allows us to recover the standard interpretation of sums (hence allowing the standard structural operational rule for sums). As stated in [Win87, Win86], another alternative would be to ensure that no event has a postcondition inside the initial conditions of the net. This would necessitate a different semantics for `while` loops, possibly along the lines of [vGV87] which would unfold one iteration of the loop.

Critical regions The net for `with` r `do` t `od` begins with an event that activates the body t by marking its initial control conditions. The event can proceed only if the resource r is initially available, and its occurrence makes the resource unavailable. The events of `with` r `do` t `od` also include the events of t and an event that releases the resource when the body has terminated. We force the events of the body t to be distinct from the other events by tagging them with body.

Formally, we introduce the following notations for critical region events:

$$\begin{array}{rl} \mathsf{acq}_{(C,C')}(r)\text{:} & {}^{\mathbf{R}}e = \{r\} \\ \mathsf{rel}_{(C,C')}(r)\text{:} & e^{\mathbf{R}} = \{r\} \end{array}$$

These all have ${}^{\mathbf{C}}e = C$ and $e^{\mathbf{C}} = C'$, and the components other than those listed are empty. We define

$$\begin{array}{rcl} \mathrm{Ic}(\texttt{with } r \texttt{ do } t \texttt{ od}) & \triangleq & \{\mathsf{i}\} \\ \mathrm{Tc}(\texttt{with } r \texttt{ do } t \texttt{ od}) & \triangleq & \{\mathsf{t}\} \\ \mathrm{Ev}(\texttt{with } r \texttt{ do } t \texttt{ od}) & \triangleq & \{\mathsf{acq}_{(\{\mathsf{i}\},\mathsf{body}:\mathrm{Ic}(t))}(r)\} \cup \mathsf{body}{:}\mathrm{Ev}(t) \\ & & \cup\ \{\mathsf{rel}_{(\mathsf{body}:\mathrm{Tc}(t),\{\mathsf{t}\})}(r)\}. \end{array}$$

The net semantics of the special term `rel` r, charged with releasing the resource r, introduced when giving the transition semantics, is:

$$\begin{array}{rcl} \mathrm{Ic}(\texttt{rel } r) & \triangleq & \{\mathsf{i}\} \\ \mathrm{Tc}(\texttt{rel } r) & \triangleq & \{\mathsf{t}\} \\ \mathrm{Ev}(\texttt{rel } r) & \triangleq & \mathsf{rel}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(r). \end{array}$$

Local resource The semantics of `resource` w `do` t `od` involves nondeterministically choosing some non-current resource, say r, to be used locally for w within t. All the resources free in t are assumed to be initially current in addition to the resources made current by other processes. When the process $[r/w]t$ terminates, the resource r is made available for other processes to use again.

We introduce the following notations for choosing the local resource and for making it non-current.

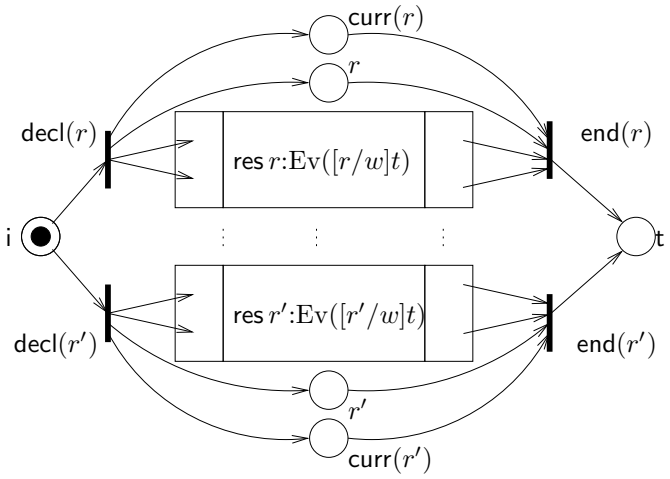
$$\begin{array}{rl} \mathsf{decl}_{(C,C')}(r)\text{:} & e^{\mathbf{R}} = \{r\} \text{ and } e^{\mathbf{N}} = \{\mathsf{curr}(r)\} \\ \mathsf{end}_{(C,C')}(r)\text{:} & {}^{\mathbf{R}}e = \{r\} \text{ and } {}^{\mathbf{N}}e = \{\mathsf{curr}(r)\} \end{array}$$

These all have ${}^{\mathbf{c}}e = C$ and $e^{\mathbf{c}} = C'$, and the components other than those listed are empty. Observe that the event $\mathsf{decl}_{(C,C')}(r)$ will avoid contact, and thus be able to occur, only if the resource r is initially non-current.

The net $\mathcal{N}[\![\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}]\!]$ is defined as:

$$\begin{array}{rcl}
\mathrm{Ic}(\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}) & \triangleq & \{\mathsf{i}\} \\
\mathrm{Tc}(\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}) & \triangleq & \{\mathsf{t}\}. \\
\mathrm{Ev}(\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}) & \triangleq & \{\mathsf{decl}_{(\{\mathsf{i}\},\mathsf{res}\,r:\mathrm{Ic}([r/w]t))}(r) \mid r \in \mathrm{Res} \setminus \mathrm{res}(t)\} \\
 & & \cup \bigcup \{\mathsf{res}\,r{:}\mathrm{Ev}([r/w]t) \mid r \in \mathrm{Res} \setminus \mathrm{res}(t)\} \\
 & & \cup \{\mathsf{end}_{(\mathsf{res}\,r:\mathrm{Tc}([r/w]t),\{\mathsf{t}\})}(r) \mid r \in \mathrm{Res} \setminus \mathrm{res}(t)\}
\end{array}$$

The net formed can be drawn as:



As such, the semantics of resource variable binding is a representation of the non-deterministic choice of resource to be selected to be used for the variable. Only one resource shall be chosen for the variable, and it will initially have been non-current due to the constraint in the token game for nets (contact) that for an event e to have concession in a marking M it must be the case that $M \setminus {}^\bullet e \cap e^\bullet = \emptyset$. Note that the semantics is invariant under alpha-equivalence.

The semantics of the special term `end` r, introduced in the transition semantics, is simply an event that releases makes the resource r non-current. Its initial control condition is i and its terminal control condition is t.

$$\begin{array}{rcl}
\mathrm{Ic}(\texttt{end}\ r) & \triangleq & \{\mathsf{i}\} \\
\mathrm{Tc}(\texttt{end}\ r) & \triangleq & \{\mathsf{t}\} \\
\mathrm{Ev}(\texttt{end}\ r) & \triangleq & \{\mathsf{end}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(r)\}
\end{array}$$

Note that we do not give a semantics to `with` w `do` t `od`, only to the term `with` r `do` t `od`, since substitutions are used in the semantics for the term `resource` w `do` t `od` above.

Used conditions

It was mentioned earlier that the semantics of any closed term could be restricted to the set of control conditions that actually occurred as a pre- or postcondition to some event

$$
\begin{array}{rcl}
\mathrm{Cond}(\varepsilon) & \triangleq & \{(\mathsf{i},\mathsf{t})\} \\
\mathrm{Cond}(\alpha) & \triangleq & \{(\mathsf{i},\mathsf{t})\} \\
\mathrm{Cond}(\texttt{alloc}(\ell)) & \triangleq & \{(\mathsf{i},\mathsf{t})\} \\
\mathrm{Cond}(\texttt{dealloc}(\ell)) & \triangleq & \{(\mathsf{i},\mathsf{t})\} \\
\mathrm{Cond}(t_1;t_2) & \triangleq & \quad (\mathsf{seq}\,1{:}\mathrm{Cond}(t_1) \cup \mathsf{seq}\,2{:}\mathrm{Cond}(t_2)) \\
 & & \setminus(\mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \cup \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)) \\
 & & \cup(\mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)) \\
\mathrm{Cond}(t_1 \parallel t_2) & \triangleq & \mathsf{par}\,1{:}\mathrm{Cond}(t_1) \cup \mathsf{par}\,2{:}\mathrm{Cond}(t_2) \\
\mathrm{Cond}(\alpha_1.t_1 + \alpha_2.t_2) & \triangleq & \{\mathsf{i}\} \cup (\mathsf{sum}\,1{:}\mathrm{Cond}(t_1) \cup \mathsf{sum}\,2{:}\mathrm{Cond}(t_2)) \\
 & & \setminus(\mathsf{sum}\,1{:}\mathrm{Tc}(t_1) \cup \mathsf{sum}\,2{:}\mathrm{Tc}(t_2)) \\
 & & \cup(\mathsf{sum}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{sum}\,2{:}\mathrm{Tc}(t_2)) \\
\mathrm{Cond}(\texttt{while}\ b\ \texttt{do}\ t\ \texttt{od}) & \triangleq & \{\mathsf{t}\} \cup \mathsf{body}{:}\mathrm{Cond}(t) \\
\mathrm{Cond}(\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}) & \triangleq & \{\mathsf{i},\mathsf{t}\} \cup \bigcup\{\mathsf{res}\,r{:}\mathrm{Cond}([r/w]t) \mid r \in \mathrm{Res} \setminus \mathrm{res}(t)\} \\
\mathrm{Cond}(\texttt{with}\ r\ \texttt{do}\ t\ \texttt{od}) & \triangleq & \{\mathsf{i},\mathsf{t}\} \cup \mathsf{body}{:}\mathrm{Cond}(t)
\end{array}
$$

Figure 3.4: Used conditions, $\mathrm{Cond}(t)$

or were initial or terminal. For reference, this set, written $\mathrm{Cond}(t)$, is defined inductively on the size of closed terms in Figure 3.4. To show that this definition has the required property, that $\mathrm{Cond}(t)$ is precisely the set of conditions used by events in $\mathrm{Ev}(t)$ with the initial and terminal conditions of t, we need a lemma about the structure of embedded nets representing terms:

Lemma 3.3.3. *For any closed term t:*

$$
\begin{array}{rl}
\forall c \in \mathrm{Ic}(t): & c \in \mathrm{Tc}(t) \text{ or } \exists e \in \mathrm{Ev}(t) : c \in {}^{\mathbf{c}}e \\
\forall c \in \mathrm{Tc}(t): & c \in \mathrm{Ic}(t) \text{ or } \exists e \in \mathrm{Ev}(t) : c \in e^{\mathbf{c}}
\end{array}
$$

Proof. The proof is by induction on the size of terms. The first property is similar to the second, so we shall only show the first. The only two interesting cases are where $t = t_1;t_2$ or $t = \alpha_1.t_1 + \alpha_2.t_2$, and the first of these two cases is representative of the other.

Suppose that $c \in \mathrm{Ic}(t_1;t_2)$. Let $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$. From the definition of $\mathrm{Ic}(t_1;t_2)$ as $P \triangleleft \mathsf{seq}\,1{:}\mathrm{Ic}(t_1)$, we have either $c = \mathsf{seq}\,1{:}c_1$ for $c_1 \in \mathrm{Ic}(t_1) \setminus \mathrm{Tc}(t_1)$ or $c = (\mathsf{seq}\,1{:}c_1, \mathsf{seq}\,2{:}c_2)$ for $c_1 \in \mathrm{Ic}(t_1) \cap \mathrm{Tc}(t_1)$ and $c_2 \in \mathrm{Ic}(t_2)$.

If $c = \mathsf{seq}\,1{:}b_1$, since $c_1 \notin \mathrm{Tc}(t_1)$, from the induction hypothesis there must exist $e_1 \in \mathrm{Ev}(t_1)$ such that $c \in {}^{\mathbf{c}}e_1$. Hence $\mathsf{seq}\,1{:}c_1 \in {}^{\mathbf{c}}(P \triangleleft \mathsf{seq}\,1{:}e_1)$, and this is an event in $\mathrm{Ev}(t_1;t_2)$.

If $c = (\mathsf{seq}\,1{:}c_1, \mathsf{seq}\,2{:}c_2)$ then either $c \in \mathrm{Tc}(t_1;t_2)$ and the case is complete or $c_2 \notin \mathrm{Tc}(t_2)$ from the definition of $\mathrm{Tc}(t_1;t_2)$. Therefore, from the induction hypothesis applied to t_2, there exists $e_2 \in \mathrm{Ev}(t_2)$ such that $c_2 \in {}^{\mathbf{c}}e_2$. Hence $c \in {}^{\mathbf{c}}(P \triangleright \mathsf{seq}\,2{:}e_2)$, and this is an event in $\mathrm{Ev}(t_1;t_2)$. □

We are now able to prove the main result about $\mathrm{Cond}(t)$:

Proposition 3.1. *For any closed term t and control condition c:*

$c \in \mathrm{Cond}(t)$ *iff either* $c \in \mathrm{Ic}(t) \cup \mathrm{Tc}(t)$ *or there exists* $e \in \mathrm{Ev}(t)$ *such that* $c \in {}^{\mathbf{C}}e^{\mathbf{C}}$.

Proof. The proof is by induction on the size of terms. Again, the interesting cases are for $t = t_1; t_2$ and $t = \alpha_1.t_1 + \alpha_2.t_2$, and the sequential composition is representative.

It is easy to see that $\mathrm{Cond}(t_1; t_2)$ contains all the conditions in $\mathrm{Ic}(t_1; t_2)$ and $\mathrm{Tc}(t_1; t_2)$ and pre- and postconditions of events in $\mathrm{Ev}(t_1; t_2)$.

Suppose that $c \in \mathrm{Cond}(t_1; t_2)$. Either $c \in (\mathsf{seq}\,1{:}\mathrm{Cond}(t_1) \cup \mathsf{seq}\,2{:}\mathrm{Cond}(t_2)) \setminus (\mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \cup \mathsf{seq}\,2{:}\mathrm{Ic}(t_2))$ or $c \in P$ where $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$.

In the first case, without loss of generality, suppose that $c = \mathsf{seq}\,1{:}c_1$ for $c_1 \in \mathrm{Cond}(t_1)$. By assumption, $c_1 \notin \mathrm{Tc}(t_1)$. By induction, either $c_1 \in \mathrm{Ic}(t_1)$ or there exists $e_1 \in \mathrm{Ev}(t_1)$ such that $c_1 \in {}^{\mathbf{C}}e^{\mathbf{C}}$. If $c_1 \in \mathrm{Ic}(t_1)$ then $\mathsf{seq}\,1{:}c_1 \in \mathrm{Ic}(t_1; t_2)$ since $\mathrm{Ic}(t_1; t_2) = P \lhd \mathsf{seq}\,1{:}\mathrm{Ic}(t_1)$ and $c \notin \mathrm{Tc}(t_1)$. If $c_1 \in {}^{\mathbf{C}}e_1{}^{\mathbf{C}}$ then $\mathsf{seq}\,1{:}c_1 \in {}^{\mathbf{C}}(P \lhd \mathsf{seq}\,1{:}e_1)^{\mathbf{C}}$, and $P \lhd \mathsf{seq}\,1{:}e_1$ is an event in $\mathrm{Ev}(t_1; t_2)$ by definition.

In the second case, suppose that $c = (\mathsf{seq}\,1{:}c_1, \mathsf{seq}\,2{:}c_2)$ for $c_1 \in \mathrm{Tc}(t_1)$ and $c_2 \in \mathrm{Ic}(t_2)$. By Lemma 3.3.3, either $c_1 \in \mathrm{Ic}(t_1)$ or there exists $e_1 \in \mathrm{Ev}(t_1)$ such that $c_1 \in e_1{}^{\mathbf{C}}$. In the first case, $c \in (P \lhd \mathsf{seq}\,1{:}\mathrm{Ic}(t_1)) = \mathrm{Ic}(t_1; t_2)$. In the second case, $c \in (P \lhd \mathsf{seq}\,1{:}e_1)^{\mathbf{C}}$, and $(P \lhd \mathsf{seq}\,1{:}e_1)$ is an event in $\mathrm{Ev}(t_1; t_2)$ by definition. □

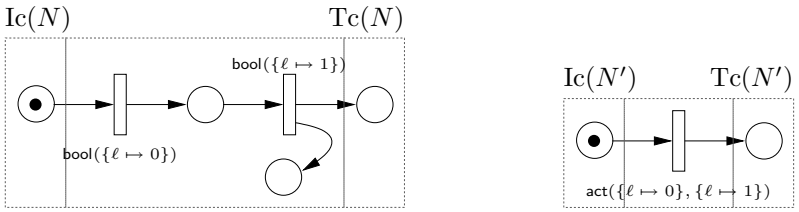
3.4 Control net

Now that we have defined the semantics of terms as nets, we need to prove that the nets formed have the expected behaviour. For example, when we later come to consider separation logic, we will need to know how runs of the net $\mathcal{N}[\![t_1; t_2]\!]$ can be formed from runs of $\mathcal{N}[\![t_1]\!]$ followed by runs of $\mathcal{N}[\![t_2]\!]$.

Often when we consider the behaviour of embedded nets, we need to ignore the particular markings of state conditions encountered. For example, an important property that the net semantics of a term t possesses is that in any reachable marking, if all of the terminal conditions of t are marked then no other condition is marked. Formally:

Terminal marking property: For any marking of state conditions σ, any marking (C, σ') reachable from $(\mathrm{Ic}(t), \sigma)$ in $\mathcal{N}[\![t]\!]$ such that $\mathrm{Tc}(t) \subseteq C$ satisfies $C = \mathrm{Tc}(t)$.

This is essential in being able to prove that an event of t_1 in the net $\mathcal{N}[\![t_1; t_2]\!]$ cannot occur after some event of t_2 has occurred. The terminal marking property is naturally proved by induction on the size of terms. When attempting the proof, a sticking-point is encountered when the parallel composition is considered since the induction hypotheses are too weak: all the runs of the net $\mathcal{N}[\![t_1 \parallel t_2]\!]$ from state σ need not be captured by $\mathcal{N}[\![t_1]\!]$ and $\mathcal{N}[\![t_2]\!]$ running from suitable initial states. To see this, consider the two nets in Figure 3.5. These two nets satisfy what would be the induction hypotheses for t_1 and t_2 (though note that the net N does not in fact represent any real term — the problem is that the induction hypothesis is too weak, not that the property fails to hold). As usual, the events of the nets are drawn with their labels and we neglect to draw the state conditions. Both of these nets satisfy the terminal marking property given above. The first net, N, only satisfies it because any initial state that allows the event labelled $\mathsf{bool}(\{\ell \mapsto 0\})$ to occur precludes the subsequent occurrence of the event labelled $\mathsf{bool}(\{\ell \mapsto 1\})$. The second net N' has the effect of changing the value held at ℓ from 0 to 1. When the nets N and N' are placed in parallel, the terminal marking property is not satisfied: this can be seen by considering the run in which the first event of N occurs, then the event of N', and finally the second event of N.



(a) Net N (b) Net N'

Figure 3.5: Nets satisfying the terminal marking property

To get around this problem, we can observe that the terminal marking property for $\mathcal{N}[\![t]\!]$ above remains valid if we consider only the *control* part of the net — in considering runs of $\mathcal{N}[\![t]\!]$, we can completely ignore the markings of state conditions, assuming them to be completely arbitrary at all times. As such, examples like the net N in Figure 3.5a, which represents no actual term, will be seen not to satisfy the new, strengthened property: the old property, that every reachable marking of control conditions containing the terminal conditions of N was equal to the terminal marking, critically relied on the encountered markings of *state* conditions.

More explicitly, when considering control properties such as that above, we will wish to consider the *control net* associated with a closed term. The control net is formed in just the same way as the net $\mathcal{N}[\![t]\!]$ apart from the following points:

- $\mathcal{C}[\![t]\!]$ has no state conditions, only control conditions. We also restrict to $\mathrm{Cond}(t)$, the control conditions that are actually used by the net, though this is of little importance.

- In the net $\mathcal{N}[\![t]\!]$, events are regarded as tuples of their pre-control, pre-state, post-control and post-state conditions. The events of $\mathcal{C}[\![t]\!]$ are also tuples of pre-control, pre-state, post-control and post-state conditions. The preconditions of events in $\mathcal{C}[\![t]\!]$ are the pre-control conditions of the corresponding event in $\mathcal{N}[\![t]\!]$, and their postconditions are the post-control conditions of the corresponding event in $\mathcal{N}[\![t]\!]$.

- The events of $\mathcal{C}[\![t]\!]$, denoted $\mathrm{Ev}_*(t)$, are defined in the same inductive way as the set of events $\mathrm{Ev}(t)$ is in the net $\mathcal{N}[\![t]\!]$. The only difference is that we ensure that every action has at least one event by adding into the definition of $\mathrm{Ev}_*(\alpha)$, for any action α, an event which we write $\mathsf{act}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(*,*)$. This event has one condition, i, and one condition, t. Formally:

$$\mathrm{Ev}_*(\alpha) = \mathrm{Ev}(\alpha) \cup \{\mathsf{act}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(*,*)\}.$$

- The other constructions of the events $\mathrm{Ev}_*(t)$ are exactly the same as they were for $\mathcal{N}[\![t]\!]$. The additional event in $\mathrm{Ev}_*(\alpha)$ passes through the inductive constructions described for $\mathcal{N}[\![t]\!]$, so for example the net $\mathcal{C}[\![\alpha_1;\alpha_2]\!]$ has two events which are not present in $\mathcal{N}[\![t]\!]$, namely:

$$\mathsf{act}_{(\{\mathsf{seq\ 1:i}\},\{(\mathsf{seq\ 1:t,seq\ 2:i})\})}(*,*) \qquad \mathsf{act}_{(\{(\mathsf{seq\ 1:t,seq\ 2:i})\},\{\mathsf{seq\ 2:t}\})}(*,*)$$

- The net $\mathcal{C}[\![t]\!]$ is a *labelled* net. Every event e in $\mathrm{Ev}_*(t)$ has a label $|e|$, which is just the corresponding label from the operational semantics. The label of any event $\mathsf{act}_{(C,C')}(*,*)$ is $\mathsf{act}(\emptyset,\emptyset)$. This gives rise to a labelling function $|-|$.

For a closed term t, the control net is defined to be the labelled net

$$\mathcal{C}[\![t]\!] \triangleq (\mathrm{Cond}(t), \mathrm{Ev}_*(t), {}^{\mathbf{C}}(-), (-)^{\mathbf{C}}, \mathrm{Ic}(t), |-|).$$

Note that the control net $\mathcal{C}[\![t]\!]$ need not be extensional: there might be two events, with different labels, in the net with the same pre- and post-control conditions.

Since the net $\mathcal{N}[\![t]\!]$ is extensional and the set of pre- and post-state conditions of any event can be inferred from its label, it is clearest to regard the events of a control net as a three-tuple $e = ({}^{\mathbf{C}}e, e^{\mathbf{C}}, |e|)$. We extend the earlier notations to this setting, so that for example $P \triangleleft ({}^{\mathbf{C}}e, e^{\mathbf{C}}, |e|) = (P \triangleleft {}^{\mathbf{C}}e, P \triangleleft e^{\mathbf{C}}, |e|)$. Note that, just as when performed on events of $\mathcal{N}[\![t]\!]$ the operations did not affect the pre- or post-state conditions of the event, the operations do not affect the labels of events in $\mathcal{C}[\![t]\!]$.

The net $\mathcal{C}[\![t]\!]$ simulates the operation of the net $\mathcal{N}[\![t]\!]$ on control conditions. That is:

Lemma 3.4.1. *For any C, C', σ and σ', if $(C, \sigma) \xrightarrow{e}\!\!\!\!\rightarrow (C', \sigma')$ in $\mathcal{N}[\![t]\!]$ then $C \xrightarrow{e}\!\!\!\!\rightarrow C'$ in $\mathcal{C}[\![t]\!]$. Hence, for any σ_0, if (C, σ) is reachable from $(\mathrm{Ic}(t), \sigma_0)$ in $\mathcal{N}[\![t]\!]$ then C is reachable from $\mathrm{Ic}(t)$ in $\mathcal{C}[\![t]\!]$.*

Proof. The first part is immediate from the definition of the token games in the two nets. The second part follows a straightforward induction on the length of path to C. □

The terminal marking property for the net $\mathcal{N}[\![t]\!]$ therefore follows from the following property for $\mathcal{C}[\![t]\!]$:

> Any marking C reachable from $\mathrm{Ic}(t)$ in $\mathcal{C}[\![t]\!]$ such that $\mathrm{Tc}(t) \subseteq C$ satisfies $C = \mathrm{Tc}(t)$.

However, before we can prove properties such as this, we still need to establish some facts about the structure of $\mathcal{C}[\![t]\!]$ and understand how runs of $\mathcal{C}[\![t]\!]$ are related to runs of the nets of subterms of t. The details, though necessary, are quite technical, and the reader may wish to ignore many of the detailed proofs or, indeed, skip straight to the key result: Theorem 3.4. It is important, however, that we show that the techniques for introducing the net semantics are amenable to such an analysis.

3.5 Structural properties

We begin with some fairly straightforward properties about the initial and terminal markings of the nets formed and show that the sets of pre- and postconditions of each event are nonempty. The first and second items of the lemma below could even be seen as part of the definition of embedded nets since non-emptiness is necessary for the constructions above to result in nets with the expected behaviours. With the final property, they can be used to show that no event has concession in the terminal marking of the net.

Lemma 3.5.1. *For any closed term t and event $e \in \mathrm{Ev}_*(t)$:*

1. $\mathrm{Ic}(t) \neq \emptyset$ *and* $\mathrm{Tc}(t) \neq \emptyset$,
2. ${}^{\mathbf{C}}e \neq \emptyset$ *and* $e^{\mathbf{C}} \neq \emptyset$, *and*
3. ${}^{\mathbf{C}}e \cap \mathrm{Tc}(t) = \emptyset$.

Proof. A simple induction on the size of terms. □

The following very simple property, that any event occurring from the initial marking of a net has a precondition in the set of initial conditions (and the corresponding property that any event into the terminal marking of the net has a postcondition inside the terminal conditions), follows immediately from the previous lemma. It will be used frequently; for instance, to show that in the net $\mathcal{N}[\![t_1;t_2]\!]$ if e_1 is an event from $\mathcal{N}[\![t_1]\!]$ and e_2 is an event from $\mathcal{N}[\![t_2]\!]$ and e_2 immediately follows e_1 in some run, then there is a control condition that occurs in both the postconditions of e_1 and the preconditions of e_2 and therefore the events are not independent. This property is used in Theorem 5.5.

Lemma 3.5.2. *For any closed term t, event $e \in \mathrm{Ev}_*(t)$ and marking C of $\mathcal{C}[\![t]\!]$:*

- *If $\mathrm{Ic}(t) \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} C$ in $\mathcal{C}[\![t]\!]$ then ${}^\bullet e \cap \mathrm{Ic}(t) \neq \emptyset$.*
- *If $C \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} \mathrm{Tc}(t)$ in $\mathcal{C}[\![t]\!]$ then $e^\bullet \cap \mathrm{Tc}(t) \neq \emptyset$.*

Proof. From Lemma 3.5.1, every event e in the net $\mathcal{N}[\![t]\!]$ has non-empty pre-control conditions and non-empty post-control conditions. The result follows immediately. □

It will be useful to know that the initial conditions of a term are equal to the terminal conditions of a term if, and only if, the term is equivalent to ε.

Lemma 3.5.3. *For any closed term t,*

$$\mathrm{Ic}(t) = \mathrm{Tc}(t) \iff t \equiv \varepsilon.$$

Proof. The key observation is that $t \equiv \varepsilon$ iff either $t = \varepsilon$ or if there exist t_1 and t_2 such that $t_1 \equiv \varepsilon$ and $t_2 \equiv \varepsilon$ and either $t = t_1;t_2$ or $t = t_1 \parallel t_2$. The result follows by a straightforward induction on terms. □

We return briefly to the terminal marking property described above. This property, which will turn out to be very important in understanding the behaviour of the nets, expresses part of the special nature of the terminal marking of conditions: that no other part of the process remains active when the terminal marking is reached.

Definition 3.5.1 (Well-termination)**.** *Say that a control net $\mathcal{C}[\![t]\!]$ is* well-terminating *if, for any marking C that is reachable from $\mathrm{Ic}(t)$, if $\mathrm{Tc}(t) \subseteq C$ then $\mathrm{Tc}(t) = C$.*

Here we arrive at a slightly awkward technical issue: When characterizing the runs of the net $\mathcal{C}[\![t]\!]$ in terms of runs of the nets representing the subterms of t (or, more accurately, in terms of nets representing terms of smaller size), we shall assume that the nets of the subterms are well-terminating. Some care is necessary since the proof that, for any closed term t, the net $\mathcal{C}[\![t]\!]$ is well-terminating itself requires understanding of the markings reachable in the net $\mathcal{C}[\![t]\!]$. To resolve this apparent 'circularity', whereby the characterizations require that nets representing terms are well-terminating and the proof of well-termination depends on the characterizations of the runs, when proving the properties required of the net $\mathcal{C}[\![t]\!]$ required to show that the net is well-terminating we shall assume that the nets representing the subterms of t are well-terminating. We shall *then* prove that any net $\mathcal{C}[\![t]\!]$ is well-terminating, allowing us to use elsewhere the properties relating runs of the net $\mathcal{C}[\![t]\!]$ to the runs of the nets of subterms of t. In effect, we will be proving well-termination and the structural properties simultaneously, by induction on the (size of) terms.

3.6 Runs of nets

Now that we have a handle on properties like termination, we can to prove that the nets formed properly represent the normal meaning of terms. In this section, we will show that the runs of the nets are as we would expect; these results shall go on to form the basis of a correspondence result with the earlier operational semantics.

The technique that we use to relate the runs of the net for a term t to the runs of the nets of its subterms is to establish a suitably strong invariant relating the markings arising before and after the occurrence of any event present in $\mathcal{C}[\![t]\!]$ and then perform an induction on the length of the run.

Sequential composition

For sequential composition, we begin by proving the following lemma characterizing the runs of the net $\mathcal{C}[\![t_1]\!]$.

Lemma 3.6.1. *Let* $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$. *Assume that* $\mathcal{C}[\![t_1]\!]$ *and* $\mathcal{C}[\![t_2]\!]$ *are well-terminating (Definition 3.5.1), and consider the net* $\mathcal{C}[\![t_1;t_2]\!]$. *For any event* $e \in \mathrm{Ev}_*(t_1;t_2)$*:*

- *Suppose that* C_1 *is reachable from* $\mathrm{Ic}(t_1)$ *in* $\mathcal{C}[\![t_1]\!]$. *If* $P \triangleleft \mathsf{seq}\,1{:}C_1 \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ *in* $\mathcal{C}[\![t_1;t_2]\!]$ *then either* $C_1 = \mathrm{Tc}(t_1)$ *or there exist* C_1' *and* e_1 *such that* $C_1 \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1'$ *in* $\mathcal{C}[\![t_1]\!]$ *and* $C' = P \triangleleft \mathsf{seq}\,1{:}C_1'$ *and* $e = P \triangleleft \mathsf{seq}\,1{:}e_1$.

- *Suppose that* C_2 *is reachable from* $\mathrm{Ic}(t_2)$ *in* $\mathcal{C}[\![t_2]\!]$. *If* $P \triangleright \mathsf{seq}\,2{:}C_2 \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ *in* $\mathcal{C}[\![t_1;t_2]\!]$ *then there exist* C_2' *and* e_2 *such that* $C_2 \stackrel{e_2}{\longrightarrow\!\!\!\!\!\rightarrow} C_2'$ *in* $\mathcal{C}[\![t_2]\!]$ *and* $C' = P \triangleright \mathsf{seq}\,2{:}C_2'$ *and* $e = P \triangleright \mathsf{seq}\,2{:}e_2$.

Proof. We show only the first item; the second is easier (requiring Lemma 3.5.1 to show that no event of t_1 has concession in the marking P).

Suppose that C_1 is reachable from $\mathrm{Ic}(t_1)$ in $\mathcal{C}[\![t_1]\!]$ and that $P \triangleleft \mathsf{seq}\,1{:}C_1 \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ in $\mathcal{C}[\![t_1;t_2]\!]$. From the definition of $\mathrm{Ev}_*(t_1;t_2)$, there are two cases: either $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ for some $e_1 \in \mathrm{Ev}_*(t_1)$ or $e = P \triangleright \mathsf{seq}\,2{:}e_2$ for some $e_2 \in \mathrm{Ev}_*(t_2)$.

First, suppose that $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ for $e_1 \in \mathrm{Ev}_*(t_1)$. From Lemma 3.3.2, we obtain $C' = P \triangleleft \mathsf{seq}\,1{:}C_1'$ for some C_1' and furthermore that $C_1 \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1'$, as required.

Now suppose that $e = P \triangleright \mathsf{seq}\,2{:}e_2$ for some $e_2 \in \mathrm{Ev}_*(t_2)$. By Lemma 3.5.1, there exists $c_2 \in {}^{\mathrm{C}}e_2$. If $c_2 \notin \mathrm{Ic}(t_2)$ then $\mathsf{seq}\,2{:}c_2 \in P \triangleleft \mathsf{seq}\,1{:}C_1$, but this is not possible according to the definition. Hence we must have $c_2 \in \mathrm{Ic}(t_2)$ and therefore $(\mathsf{seq}\,1{:}c_1, \mathsf{seq}\,2{:}c_2) \in {}^{\mathrm{C}}e$ for all $c_1 \in \mathrm{Tc}(t_1)$. It follows from e having concession in $P \triangleleft \mathsf{seq}\,1{:}C_1$ that $c_1 \in C_1$ for all $c_1 \in \mathrm{Tc}(t_1)$. The net $\mathcal{C}[\![t_1]\!]$ is assumed to be well-terminating, so $C_1 = \mathrm{Tc}(t_1)$ as required. □

Using this result, it can be shown that any state reached in $\mathcal{C}[\![t_1;t_2]\!]$ is reached either as a run of $\mathcal{C}[\![t_1]\!]$ or as a run of $\mathcal{C}[\![t_1]\!]$ to a terminal marking followed by a run of $\mathcal{C}[\![t_2]\!]$. Here, we extend the notations for gluing and tagging to sequences of events in the obvious way.

Lemma 3.6.2. *Suppose that the nets* $\mathcal{C}[\![t_1]\!]$ *and* $\mathcal{C}[\![t_2]\!]$ *are well-terminating. If* $\mathrm{Ic}(t_1;t_2) \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} C$ *in* $\mathcal{C}[\![t_1;t_2]\!]$ *then either:*

- *there exist* C_1 *and* π_1 *such that* $C = P \triangleleft \mathsf{seq}\,1{:}C_1$ *and* $\pi = P \triangleleft \mathsf{seq}\,1{:}\pi_1$ *and* $\mathrm{Ic}(t_1) \stackrel{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ *in* $\mathcal{C}[\![t_1]\!]$, *or*

- *there exist* C_2, π_1 *and* π_2 *such that* $C = P \triangleright \mathsf{seq}\,2{:}C_2$ *and* $\pi = (P \triangleleft \mathsf{seq}\,1{:}\pi_1) \cdot (P \triangleright \mathsf{seq}\,2{:}\pi_2)$ *and* $\mathrm{Ic}(t_1) \xrightarrow{\pi_1}\!\!\!\!\rightarrow \mathrm{Tc}(t_1)$ *in* $\mathcal{C}[\![t_1]\!]$ *and* $\mathrm{Ic}(t_2) \xrightarrow{\pi_2}\!\!\!\!\rightarrow C_2$ *in* $\mathcal{C}[\![t_2]\!]$,

where $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$.

Proof. A straightforward induction on the length of π using Lemma 3.6.1. □

The above lemma can be extended straightforwardly using Lemma 3.4.1 to obtain the following result involving states, using the fact that the operations of prefixing and tagging do not affect the action of events on state conditions:

Lemma 3.6.3. *Suppose that the nets* $\mathcal{C}[\![t_1]\!]$ *and* $\mathcal{C}[\![t_2]\!]$ *are well-terminating. If* $(\mathrm{Ic}(t_1;t_2),\sigma_0) \xrightarrow{\pi}\!\!\!\!\rightarrow (C,\sigma)$ *in* $\mathcal{N}[\![t_1;t_2]\!]$ *then either:*

- *there exist* C_1 *and* π_1 *such that* $C = P{\triangleleft}\mathsf{seq}\,1{:}C_1$ *and* $\pi = P{\triangleleft}\mathsf{seq}\,1{:}\pi_1$ *and* $(\mathrm{Ic}(t_1),\sigma_0) \xrightarrow{\pi_1}\!\!\!\!\rightarrow (C_1,\sigma)$ *in* $\mathcal{N}[\![t_1]\!]$, *or*
- *there exist* C_2, σ', π_1 *and* π_2 *such that* $C = P \triangleright \mathsf{seq}\,2{:}C_2$ *and* $\pi = (P \triangleleft \mathsf{seq}\,1{:}\pi_1) \cdot (P \triangleright \mathsf{seq}\,2{:}\pi_2)$ *and* $(\mathrm{Ic}(t_1),\sigma_0) \xrightarrow{\pi_1}\!\!\!\!\rightarrow (\mathrm{Tc}(t_1),\sigma')$ *in* $\mathcal{N}[\![t_1]\!]$ *and* $(\mathrm{Ic}(t_2),\sigma') \xrightarrow{\pi_2}\!\!\!\!\rightarrow (C_2,\sigma)$ *in* $\mathcal{N}[\![t_2]\!]$,

where $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$.

The converse result, that runs of the nets $\mathcal{N}[\![t_1]\!]$ and $\mathcal{N}[\![t_2]\!]$, with appropriate intermediate states, give rise to runs of the net $\mathcal{N}[\![t_1;t_2]\!]$ is an immediate consequence of Lemma 3.3.2

We shall now pass quickly through the other constructs of the language. The main detail is in the statements of the lemmas. The reader may wish to pass to Section 3.7 if now broadly content with how runs of nets are characterized.

Parallel composition

We now characterize the runs of the net $\mathcal{C}[\![t_1 \parallel t_2]\!]$. As for the sequential composition, the first step is to obtain a characterization of how single events might occur. The proof of the lemma is an easy analysis of the two forms of event in $\mathcal{C}[\![t_1 \parallel t_2]\!]$.

Lemma 3.6.4. *If* $\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2 \xrightarrow{e}\!\!\!\!\rightarrow C'$ *in* $\mathcal{C}[\![t_1 \parallel t_2]\!]$ *then either:*

1. *there exist* e_1 *and* C_1 *such that* $e = \mathsf{par}\,1{:}e_1$ *and* $C' = \mathsf{par}\,1{:}C_1' \cup \mathsf{par}\,2{:}C_2$ *and* $C_1 \xrightarrow{e_1}\!\!\!\!\rightarrow C_1'$ *in* $\mathcal{C}[\![t_1]\!]$, *or*
2. *there exist* e_2 *and* C_2 *such that* $e = \mathsf{par}\,2{:}e_2$ *and* $C' = \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2'$ *and* $C_2 \xrightarrow{e_2}\!\!\!\!\rightarrow C_2'$ *in* $\mathcal{C}[\![t_2]\!]$.

With this lemma, we can obtain the required characterization.

Lemma 3.6.5. *Let* π *be a sequence of events such that* $\mathrm{Ic}(t_1 \parallel t_2) \xrightarrow{\pi}\!\!\!\!\rightarrow C$ *in* $\mathcal{C}[\![t_1 \parallel t_2]\!]$. *Every event of* π *is of the form* $\mathsf{par}\,1{:}e_1$ *or* $\mathsf{par}\,2{:}e_2$*; let* π_1 *represent* π *restricted to events of the form* $\mathsf{par}\,1{:}e_1$ *and* π_2 *represent* π *restricted to events of the form* $\mathsf{par}\,2{:}e_2$. *There exist* C_1 *and* C_2 *such that* $C = \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2$ *and* $\mathrm{Ic}(t_1) \xrightarrow{\pi_1}\!\!\!\!\rightarrow C_1$ *in* $\mathcal{C}[\![t_1]\!]$ *and* $\mathrm{Ic}(t_2) \xrightarrow{\pi_2}\!\!\!\!\rightarrow C_2$ *in* $\mathcal{C}[\![t_2]\!]$.

Proof. A simple induction on the length of π applying the previous lemma, noting that $\mathrm{Ic}(t_1 \parallel t_2) = \mathsf{par}\,1{:}\mathrm{Ic}(t_1) \cup \mathsf{par}\,2{:}\mathrm{Ic}(t_2)$. □

Note that this lemma characterizes the markings reachable in the control net for $t_1 \parallel t_2$ directly in terms of the markings reachable in the control nets for t_1 and t_2. For the reasons outlined above, we do not obtain an analogous result for the net $\mathcal{N}[\![t_1 \parallel t_2]\!]$ running from a particular initial state, so there is no analogue of Lemma 3.6.3 here.

Nondeterministic sum

As before, we first characterize how single events can occur in markings of the net $\mathcal{C}[\![\alpha_1.t_1 + \alpha_2.t_2]\!]$.

Lemma 3.6.6. *Let* $P = \mathsf{sum}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{sum}\,2{:}\mathrm{Tc}(t_2)$ *and let* $t = \alpha_1.t_1 + \alpha_2.t_2$. *For any event* $e \in \mathrm{Ev}_*(t)$*:*

- *If* $\mathrm{Ic}(t) \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ *in* $\mathcal{C}[\![t]\!]$ *then either* $e = \mathsf{act}_{(\{\mathsf{i}\},P \lhd \mathsf{sum}\,1:\mathrm{Ic}(t_1))}(D_1, D_2)$ *for some* $(D_1, D_2) \in \mathcal{A}[\![\alpha_1]\!]$ *and* $C' = P \lhd \mathsf{sum}\,1{:}\mathrm{Ic}(t_1)$,
  or $e = \mathsf{act}_{(\{\mathsf{i}\},P \rhd \mathsf{sum}\,2:\mathrm{Ic}(t_2))}(D_1, D_2)$ *for some* $(D_1, D_2) \in \mathcal{A}[\![\alpha_2]\!]$ *and* $C' = P \rhd \mathsf{sum}\,2{:}\mathrm{Ic}(t_2)$.
- *If* $P \lhd \mathsf{sum}\,1{:}C_1 \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ *in* $\mathcal{C}[\![t]\!]$ *then there exist* e_1 *and* C_1' *such that* $e = P \lhd \mathsf{sum}\,1{:}e_1$ *and* $C' = P \lhd \mathsf{sum}\,1{:}C_1'$ *and* $C_1 \overset{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1'$ *in* $\mathcal{C}[\![t_1]\!]$.
- *If* $P \rhd \mathsf{sum}\,2{:}C_2 \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ *in* $\mathcal{C}[\![t]\!]$ *then there exist* e_2 *and* C_2' *such that* $e = P \rhd \mathsf{sum}\,2{:}e_2$ *and* $C' = P \rhd \mathsf{sum}\,2{:}C_2'$ *and* $C_2 \overset{e_2}{\longrightarrow\!\!\!\!\!\rightarrow} C_2'$ *in* $\mathcal{C}[\![t_2]\!]$.

We can now describe the runs of the net $\mathcal{C}[\![\alpha_1.t_1 + \alpha_2.t_2]\!]$ by a simple induction on the length of run.

Lemma 3.6.7. *Let* $P = \mathsf{sum}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{sum}\,2{:}\mathrm{Tc}(t_2)$ *and let* $t = \alpha_1.t_1 + \alpha_2.t_2$. *Suppose that* $\mathrm{Ic}(t) \overset{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} C$ *in* $\mathcal{C}[\![t]\!]$. *Either* π *is empty or:*

1. *there exist* D_1, D_2, C_1 *and* π_1 *such that* $(D_1, D_2) \in \mathcal{A}[\![\alpha_1]\!]$ *and*
$$\pi = \mathsf{act}_{(\{\mathsf{i}\},P \lhd \mathsf{sum}\,1:\mathrm{Ic}(t_1))}(D_1, D_2) \cdot (P \lhd \mathsf{sum}\,1{:}\pi_1)$$
and $C = P \lhd \mathsf{sum}\,1{:}C_1$ *and* $\mathrm{Ic}(t_1) \overset{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ *in* $\mathcal{C}[\![t_1]\!]$, *or*
2. *there exist* D_1, D_2, C_2 *and* π_2 *such that* $(D_1, D_2) \in \mathcal{A}[\![\alpha_2]\!]$ *and*
$$\pi = \mathsf{act}_{(\{\mathsf{i}\},P \rhd \mathsf{sum}\,2:\mathrm{Ic}(t_2))}(D_1, D_2) \cdot (P \rhd \mathsf{sum}\,2{:}\pi_2)$$
and $C = P \rhd \mathsf{sum}\,2{:}C_2$ *and* $\mathrm{Ic}(t_2) \overset{\pi_2}{\longrightarrow\!\!\!\!\!\rightarrow} C_2$ *in* $\mathcal{C}[\![t_2]\!]$.

Iteration

To obtain a concise account of what events can occur in markings of the net `while` b `do` t_0 `od`, we assume that the net $\mathcal{C}[\![t_0]\!]$ is well-terminating.

Lemma 3.6.8. *Let* $t =$ `while` b `do` t_0 `od` *and suppose that* $\mathcal{C}[\![t_0]\!]$ *is well-terminating. For any event* $e \in \mathrm{Ev}_*(t_0)$*:*

- *If* $\mathrm{Ic}(t) = \mathsf{body}{:}\mathrm{Tc}(t_0) \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ *in* $\mathcal{C}[\![t]\!]$ *then either*
  $C' = \mathsf{body}{:}\mathrm{Ic}(t_0)$ *and* $e = \mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathsf{body}:\mathrm{Ic}(t_0))}(D_0, D_0)$ *for some* D_0 *such that* $(D_0, D_0) \in \mathcal{A}[\![b]\!]$, *or*
  $C' = \mathrm{Tc}(t)$ *and* $e = \mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathrm{Tc}(t))}(D_0, D_0)$ *for some* D_0 *such that* $(D_0, D_0) \in \mathcal{A}[\![\neg b]\!]$.

- *If* C_0 *is reachable from* $\mathrm{Ic}(t_0)$ *in* $\mathcal{C}[\![t_0]\!]$ *and* $C_0 \neq \mathrm{Tc}(t_0)$ *and* $\mathsf{body}{:}C_0 \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ *in* $\mathcal{C}[\![t]\!]$ *then there exists* $e_0 \in \mathrm{Ev}_*(t_0)$ *such that* $e = \mathsf{body}{:}e_0$ *and* $C_0 \stackrel{e_0}{\longrightarrow\!\!\!\!\!\rightarrow} C_0'$ *in* $\mathcal{C}[\![t_0]\!]$ *for some* C_0' *such that* $C' = \mathsf{body}{:}C_0'$.

Proof. A straightforward analysis of the events present in the net $\mathcal{C}[\![t]\!]$. The second part makes use of the fact that $\mathcal{C}[\![t_0]\!]$ is well-terminating to show that $\mathrm{Tc}(t_0) \not\subseteq C_0$ since $C_0 \neq \mathrm{Tc}(t_0)$, thereby ruling out the possibility of an event representing one of the tests of the boolean b occurring. □

We now show that any run of the net $\mathcal{C}[\![\texttt{while}\ b\ \texttt{do}\ t_0]\!]$ is either empty or consists of some number of positive tests of b followed by runs of t_0, possibly followed either by a negative test of b or a positive test of b followed by a run of t_0.

Lemma 3.6.9. *Let* $t = \texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od}$ *and suppose that* $\mathcal{C}[\![t_0]\!]$ *is well-terminating. Suppose that* π *is a sequence of events of* $\mathcal{C}[\![t]\!]$ *such that* $\mathrm{Ic}(t) \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} C$ *in* $\mathcal{C}[\![t]\!]$. *Either:*

- π *is empty,*
- *there exist* $D_1 \ldots D_n$ *satisfying* $(D_i, D_i) \in \mathcal{A}[\![b]\!]$ *and there exist* $\pi_1, \ldots, \pi_n$ *such that*

$$\begin{aligned}\pi \;=\; & \mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathsf{body}:\mathrm{Ic}(t_0))}(D_1, D_1) \cdot (\mathsf{body}{:}\pi_1) \cdot \\ & \ldots \\ & \cdot\mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathsf{body}:\mathrm{Ic}(t_0))}(D_n, D_n) \cdot (\mathsf{body}{:}\pi_n)\end{aligned}$$

  and $\mathrm{Ic}(t_0) \stackrel{\pi_i}{\longrightarrow\!\!\!\!\!\rightarrow} \mathrm{Tc}(t_0)$ *in* $\mathcal{C}[\![t_0]\!]$ *for all* $i < n$ *and* $\mathrm{Ic}(t_0) \stackrel{\pi_n}{\longrightarrow\!\!\!\!\!\rightarrow} C_0$ *for some* C_0 *such that* $C = \mathsf{body}{:}C_0$, *or*

- $C = \mathrm{Tc}(t)$ *and there exist* $D_1 \ldots D_n$ *satisfying* $(D_i, D_i) \in \mathcal{A}[\![b]\!]$, *there exists* D_0 *such that* $D_0 \in \mathcal{A}[\![\neg b]\!]$ *and there exist* $\pi_1, \ldots, \pi_n$ *such that*

$$\begin{aligned}\pi \;=\; & \mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathsf{body}:\mathrm{Ic}(t_0))}(D_1, D_1) \cdot (\mathsf{body}{:}\pi_1) \cdot \\ & \ldots \\ & \cdot\mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathsf{body}:\mathrm{Ic}(t_0))}(D_n, D_n) \cdot (\mathsf{body}{:}\pi_n) \\ & \cdot\mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathrm{Tc}(t))}(D_0, D_0)\end{aligned}$$

  and $\mathrm{Ic}(t_0) \stackrel{\pi_i}{\longrightarrow\!\!\!\!\!\rightarrow} \mathrm{Tc}(t_0)$ *in* $\mathcal{C}[\![t_0]\!]$ *for all* $i \leq n$.

Proof. Induction on the length of π using Lemma 3.6.8. □

Though its statement perhaps appears a little long, the lemma above is useful and expresses precisely that the while loop behaves as expected. It can be applied, for example, with Lemma 3.4.1, to show that any run to a terminal marking of the net $\mathcal{N}[\![\texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od}]\!]$ from an initial state σ gives rise to a series of runs of the net $\mathcal{N}[\![t_0]\!]$ running from appropriate intermediate states that satisfy b, followed by a run of $\mathcal{N}[\![t_0]\!]$ to a state that satisfies $\neg b$.

Resource

For the `resource` w `do` t_0 `od` construct, we must relate the occurrence of events in this net to events in $[r/w]t_0$, dependent on the choice of resource r that the variable w might take.

Lemma 3.6.10. *Let* $t = \texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}$ *and suppose that* $\mathcal{C}[\![[r/w]t_0]\!]$ *is well-terminating for every* $r \in \mathrm{Res}$*. For any event* $e \in \mathrm{Ev}_*(t)$*:*

- *if* $\mathrm{Ic}(t) \xrightarrow{e}\!\!\!\!\rightarrow C'$ *then there exists* $r \in \mathrm{Res} \setminus \mathrm{res}(t_0)$ *such that* $C' = \mathsf{res}\, r{:}\mathrm{Ic}([r/w]t_0)$ *and* $e = \mathsf{decl}_{(\mathrm{Ic}(t),\mathsf{res}\, r:\mathrm{Ic}([r/w]t_0))}(r)$*,*
- *if* C_0 *is reachable from* $\mathrm{Ic}([r/w]t_0)$ *in* $\mathcal{C}[\![[r/w]t_0]\!]$ *and* $\mathsf{res}\, r{:}C_0 \xrightarrow{e}\!\!\!\!\rightarrow C'$ *in* $\mathcal{C}[\![t]\!]$ *but* $C_0 \neq \mathrm{Tc}([r/w]t_0)$ *then there exists* e_0 *such that* $e = \mathsf{res}\, r{:}e_0$ *and there exists* C_0' *such that* $C' = \mathsf{res}\, r{:}C_0'$ *and* $C_0 \xrightarrow{e_0}\!\!\!\!\rightarrow C_0'$ *in* $\mathcal{C}[\![[r/w]t_0]\!]$*, and*
- *if* $\mathsf{res}\, r{:}\mathrm{Tc}([r/w]t_0) \xrightarrow{e}\!\!\!\!\rightarrow C'$ *in* $\mathcal{C}[\![t]\!]$ *then* $e = \mathsf{end}_{(\mathsf{res}\, r:\mathrm{Tc}([r/w]t_0),\mathrm{Tc}(t))}(r)$ *and* $C' = \mathrm{Tc}(t)$*.*

Proof. Follows a simple analysis of the events in $\mathrm{Ev}_*(\texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od})$ alongside the observation that $\mathsf{res}\, r{:}\mathrm{Tc}(t_0) \not\subseteq \mathsf{res}\, r{:}C_0$ if $C_0 \neq \mathrm{Tc}(t_0)$ due to well-termination of $\mathcal{C}[\![[r/w]t_0]\!]$ and that no event of t_0 has concession in $\mathsf{res}\, r{:}\mathrm{Tc}(t_0)$ due to Lemma 3.5.1. □

We can now characterize the runs of the net $\texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}$ by inductively applying the previous lemma.

Lemma 3.6.11. *Let* $t = \texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}$ *and suppose that* $\mathcal{C}[\![[r/w]t_0]\!]$ *is well-terminating for every* $r \in \mathrm{Res}$*. Suppose that* π *is a sequence of events* π *such that* $\mathrm{Ic}(t) \xrightarrow{\pi}\!\!\!\!\rightarrow C$ *in* $\mathcal{C}[\![t]\!]$*. Either* π *is empty or there exists a resource* $r \in \mathrm{Res} \setminus \mathrm{res}(t_0)$*, a sequence* π_0 *and marking of control conditions* C_0 *such that* $\mathrm{Ic}([r/w]t_0) \xrightarrow{\pi_0}\!\!\!\!\rightarrow C_0$ *in* $\mathcal{C}[\![[r/w]t_0]\!]$ *and either:*

- $\pi = \mathsf{decl}_{(\mathrm{Ic}(t),\mathsf{res}\, r:\mathrm{Ic}([r/w]t_0))}(r) \cdot (\mathsf{res}\, r{:}\pi_0)$ *and* $C = \mathsf{res}\, r{:}C_0$*, or*
- $\pi = \mathsf{decl}_{(\mathrm{Ic}(t),\mathsf{res}\, r:\mathrm{Ic}([r/w]t_0))}(r) \cdot (\mathsf{res}\, r{:}\pi_0) \cdot \mathsf{end}_{(\mathsf{res}\, r:\mathrm{Tc}([r/w]t_0),\mathrm{Tc}(t))}(r)$ *and* $C = \mathrm{Tc}(t)$*.*

Critical regions

The final construct to be considered, $\texttt{with}\ r\ \texttt{do}\ t_0\ \texttt{od}$, is relatively straightforward. The occurrence of events in $\mathcal{C}[\![t_0]\!]$ is characterized as follows.

Lemma 3.6.12. *Let* $t = \texttt{with}\ w\ \texttt{do}\ t_0\ \texttt{od}$ *and suppose that* $\mathcal{C}[\![t_0]\!]$ *is well-terminating. For any event* $e \in \mathrm{Ev}_*(t)$*:*

- *if* $\mathrm{Ic}(t) \xrightarrow{e}\!\!\!\!\rightarrow C'$ *then* $e = \mathsf{acq}_{(\mathrm{Ic}(t),\mathsf{body}:\mathrm{Ic}(t_0))}(r)$ *and* $C' = \mathsf{body}{:}\mathrm{Ic}(t_0)$*;*
- *if* C_0 *is reachable from* $\mathrm{Ic}(t_0)$ *in* $\mathcal{C}[\![t_0]\!]$ *and* $\mathsf{body}{:}C_0 \xrightarrow{e}\!\!\!\!\rightarrow C'$ *but* $C_0 \neq \mathrm{Tc}(t_0)$ *then there exists an event* e_0 *and marking of control conditions* C_0' *such that* $e = \mathsf{body}{:}e_0$ *and* $C' = \mathsf{body}{:}C_0'$ *and* $C_0 \xrightarrow{e_0}\!\!\!\!\rightarrow C_0'$ *in* $\mathcal{C}[\![t_0]\!]$*; and*
- *if* $\mathsf{body}{:}\mathrm{Tc}(t_0) \xrightarrow{e}\!\!\!\!\rightarrow C'$ *then* $e = \mathsf{end}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathrm{Tc}(t))}(r)$ *and* $C' = \mathrm{Tc}(t)$*.*

Proof. Like Lemma 3.6.10, this follows a simple analysis of the events in $\mathrm{Ev}_*(\texttt{with}\ r\ \texttt{do}\ t_0\ \texttt{od})$ and uses the assumption that the net $\mathcal{C}[\![t_0]\!]$ is well-terminating. □

Using this lemma, a straightforward induction characterizes the paths of the control net $\mathcal{C}[\![\texttt{with}\ r\ \texttt{do}\ t_0\ \texttt{od}]\!]$ as follows:

Lemma 3.6.13. *Let* $t = \texttt{with}\ w\ \texttt{do}\ t_0\ \texttt{od}$ *and suppose that* $\mathcal{C}[\![t_0]\!]$ *is well-terminating. Suppose that* π *is a path such that* $\mathrm{Ic}(t) \xrightarrow{\pi}\!\!\!\!\rightarrow C$ *in* $\mathcal{C}[\![t]\!]$*. Either* π *is empty or there exist a path* π_0 *and a marking of control conditions* C_0 *such that* $\mathrm{Ic}(t_0) \xrightarrow{\pi_0}\!\!\!\!\rightarrow C_0$ *in* $\mathcal{C}[\![t_0]\!]$ *and either:*

- $\pi = \mathsf{acq}_{(\mathrm{Ic}(t),\mathsf{body}:\mathrm{Ic}(t_0))}(r) \cdot (\mathsf{body}{:}\pi_0)$ *and* $C = \mathsf{body}{:}C_0$, *or*
- $\pi = \mathsf{acq}_{(\mathrm{Ic}(t),\mathsf{body}:\mathrm{Ic}(t_0))}(r) \cdot (\mathsf{body}{:}\pi_0) \cdot \mathsf{end}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathrm{Tc}(t))}(r)$ *and* $C = \mathrm{Tc}(t)$.

3.7 Well-termination

We are now able to show that the net semantics of any term is well-terminating in the sense of Definition 3.5.1.

Lemma 3.7.1. *For any closed term t, the net $\mathcal{C}[\![t]\!]$ is well-terminating.*

Proof. The proof proceeds by induction on the size of terms. We shall show only two interesting cases for the term t.

Parallel composition: Suppose that C is reachable in $\mathcal{C}[\![t_1 \parallel t_2]\!]$ from $\mathrm{Ic}(t_1 \parallel t_2)$. By induction, the nets $\mathcal{C}[\![t_1]\!]$ and $\mathcal{C}[\![t_2]\!]$ are well-terminating. According to Lemma 3.6.5, we therefore have $C = \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,1{:}C_2$ for C_1 reachable from $\mathrm{Ic}(t_1)$ in $\mathcal{C}[\![t_1]\!]$ and C_2 reachable from $\mathrm{Ic}(t_2)$ in $\mathcal{C}[\![t_2]\!]$. Suppose that $\mathrm{Tc}(t_1 \parallel t_2) \subseteq C$. We must have $\mathrm{Tc}(t_1) \subseteq C_1$ and $\mathrm{Tc}(t_2) \subseteq C_2$. Since, by induction, the nets $\mathcal{C}[\![t_1]\!]$ and $\mathcal{C}[\![t_2]\!]$ are well-terminating, we have $C_1 = \mathrm{Tc}(t_1)$ and $C_2 = \mathrm{Tc}(t_2)$ and hence $C = \mathrm{Tc}(t_1 \parallel t_2)$.

Sequential composition: Suppose that C is reachable in $\mathcal{C}[\![t_1; t_2]\!]$ from $\mathrm{Ic}(t_1; t_2)$. There exists a path $\mathrm{Ic}(t_1; t_2) \xrightarrow{\pi} C$. Let $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$. By induction, the nets $\mathcal{C}[\![t_1]\!]$ and $\mathcal{C}[\![t_2]\!]$ are well-terminating, so according to Lemma 3.6.2 there are two cases for the path π.

$C = P \triangleleft \mathsf{seq}\,1{:}C_1$ for some $C_1 \neq \mathrm{Tc}(t_1)$ and $\pi = P \triangleleft \mathsf{seq}\,1{:}\pi_1$ for some π_1 such that $\mathcal{C}[\![t_1]\!]{:}\mathrm{Ic}(t_1) \xrightarrow{\pi_1} C_1$. Recall that $\mathrm{Tc}(t_1; t_2) = P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$. We shall show that $\mathrm{Tc}(t_1; t_2) \not\subseteq P \triangleleft \mathsf{seq}\,1{:}C_1$. Suppose, for contradiction, that $\mathrm{Tc}(t_1; t_2) \subseteq P \triangleleft \mathsf{seq}\,1{:}C_1$. It is easy to show that $\mathrm{Tc}(t_2) \subseteq \mathrm{Ic}(t_2)$ since otherwise $P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2) \not\subseteq P \triangleleft \mathsf{seq}\,1{:}C_1$. Since $\mathcal{C}[\![t_2]\!]$ is well-terminating, we have $\mathrm{Tc}(t_2) = \mathrm{Ic}(t_2)$. We cannot have $\mathrm{Tc}(t_1) \subseteq C_1$ since otherwise $C_1 = \mathrm{Tc}(t_1)$ by well-termination of $\mathcal{C}[\![t_1]\!]$, contradicting the earlier assumption, so there exists $b_1 \in \mathrm{Tc}(t_1)$ such that $b_1 \notin C_1$. It follows that $(\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2) \in P$ but $(\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2) \notin P \triangleleft \mathsf{seq}\,1{:}C_1$. However, since $b_2 \in \mathrm{Tc}(t_2)$ we have $(\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2) \in P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2) = \mathrm{Tc}(t_1; t_2)$. It follows immediately that $\mathrm{Tc}(t_1; t_2) \not\subseteq C$, giving the required contradiction.

$C = P \triangleright \mathsf{seq}\,2{:}C_2$ for some C_2 and $\pi = (P \triangleleft \mathsf{seq}\,1{:}\pi_1) \cdot (P \triangleright \mathsf{seq}\,2{:}\pi_2)$ for some π_1, π_2 such that $\mathcal{C}[\![t_1]\!]{:}\mathrm{Ic}(t_1) \xrightarrow{\pi_1} \mathrm{Tc}(t_2)$ and $\mathcal{C}[\![t_2]\!]{:}\mathrm{Ic}(t_2) \xrightarrow{\pi_2} C_2$.
Suppose that $\mathrm{Tc}(t_1; t_2) \subseteq C$. We have, by definition,

$$P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2) \subseteq P \triangleright \mathsf{seq}\,2{:}C_2.$$

By Lemma 3.3.1, it follows that $\mathrm{Tc}(t_2) \subseteq C_2$ and therefore $C_2 = \mathrm{Tc}(t_2)$ by well-termination of $\mathcal{C}[\![t_2]\!]$. It follows immediately that $C = \mathrm{Tc}(t_1; t_2)$, as required to complete the case. □

Since we have now shown that the control net of any closed term is well-terminating, in what follows we can ignore the assumption in Lemmas 3.6.2, 3.6.5 *etc.* that the control nets of subterms are well-terminating when we apply these lemmas.

3.8 Safety

We have now established that the net $\mathcal{C}[\![t]\!]$ is well-terminating for any term t and have characterized the reachable markings of the net $\mathcal{C}[\![t]\!]$ in terms of the reachable markings of nets of terms of smaller size. With this understanding of the reachable markings, we are able to show that the net $\mathcal{C}[\![t]\!]$ is a safe net. This is in contrast with the net $\mathcal{N}[\![t]\!]$ running from a marking of state conditions; we saw earlier that the net $\mathcal{N}[\![t]\!]$ need not be safe since, for example, it might use contact to inhibit allocation of an already-current location. The proof that $\mathcal{C}[\![t]\!]$ is safe will rely on the constructions forming the control net preserving safety.

Proposition 3.2. *The net $\mathcal{C}[\![t]\!]$ is a safe labelled Petri net.*

Proof. By induction on the size of terms. We consider three cases for t; the rest follow the same pattern.

$\underline{t = t_1 \parallel t_2}$: Let C be a reachable marking in $\mathcal{C}[\![t_1 \parallel t_2]\!]$ and suppose, for contradiction, that there exists $e \in \mathrm{Ev}_*(t_1 \parallel t_2)$ such that ${}^{\mathbf{C}}e \subseteq C$ but $C \setminus {}^{\mathbf{C}}e \cap e^{\mathbf{C}} \neq \emptyset$. Without loss of generality, since $e \in \mathrm{Ev}_*(t_1 \parallel t_2)$, suppose that $e = \mathsf{par}\,1{:}e_1$ for some $e_1 \in \mathrm{Ev}_*(t_1)$. From Lemma 3.6.5, we have $C = \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2$ for some C_1 and C_2 such that $\mathcal{C}[\![t_1]\!]{:}\mathrm{Ic}(t_1) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ and $\mathcal{C}[\![t_2]\!]{:}\mathrm{Ic}(t_2) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_2$. We must have ${}^{\mathbf{C}}e_1 \subseteq C$ but $C_1 \setminus {}^{\mathbf{C}}e_1 \cap e_1{}^{\mathbf{C}} \neq \emptyset$. Hence the net $\mathcal{C}[\![t_1]\!]$ is not safe, contradicting the induction hypothesis.

$\underline{t = t_1; t_2}$: Let C be a reachable marking in $\mathcal{C}[\![t_1 \parallel t_2]\!]$ and suppose, for contradiction, that there exists $e \in \mathrm{Ev}_*(t_1 \parallel t_2)$ such that ${}^{\mathbf{C}}e \subseteq C$ but $C \setminus {}^{\mathbf{C}}e \cap e^{\mathbf{C}} \neq \emptyset$. Let $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$. There are two cases for e.

First, if $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ for some e_1 in $\mathrm{Ev}_*(t_1)$, it follows from the fact that ${}^{\mathbf{C}}e \subseteq C$ and Lemma 3.6.2 that $C = P \triangleleft \mathsf{seq}\,1{:}C_1$ for some C_1 such that $\mathcal{C}[\![t_1]\!]{:}\mathrm{Ic}(t_1) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$. Consequently, from Lemma 3.3.1, we must have ${}^{\mathbf{C}}e_1 \subseteq C_1$ and $C_1 \setminus {}^{\mathbf{C}}e_1 \cap e_1{}^{\mathbf{C}} \neq \emptyset$. Thus $\mathcal{C}[\![t_1]\!]$ is not safe, contradicting the induction hypothesis.

Second, if $e = P \triangleright \mathsf{seq}\,2{:}e_2$ for some e_2 in $\mathrm{Ev}_*(t_2)$, it follows from the fact that ${}^{\mathbf{C}}e \subseteq C$ and Lemma 3.6.2 that $C = P \triangleright \mathsf{seq}\,2{:}C_2$ for some C_2 such that $\mathcal{C}[\![t_2]\!]{:}\mathrm{Ic}(t_2) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_2$. As before, it follows from Lemma 3.3.1 that ${}^{\mathbf{C}}e_2 \subseteq C_2$ and $C_2{}^{\mathbf{C}}e_2 \cap e_2{}^{\mathbf{C}} \neq \emptyset$. Thus $\mathcal{C}[\![t_2]\!]$ is not safe, contradicting the induction hypothesis.

$\underline{t = \texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od}}$: Let C be a reachable marking in $\mathcal{C}[\![\texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od}]\!]$ and suppose, for contradiction, that there exists $e \in \mathrm{Ev}_*(\texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od})$ such that ${}^{\mathbf{C}}e \subseteq C$ but $C \setminus {}^{\mathbf{C}}e \cap e^{\mathbf{C}} \neq \emptyset$. According to Lemma 3.6.9, there are three cases for the marking C.

First, if $C = \mathsf{body}{:}\mathrm{Tc}(t_0)$ (*i.e.* $C = \mathrm{Ic}(\texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od})$), it cannot be the case that $e = \mathsf{body}{:}e_0$ for some $e_0 \in \mathrm{Ev}_*(t_0)$. Otherwise, since ${}^{\mathbf{C}}e \subseteq C$, from Lemma 3.5.1 (2) there would exist a condition c such that $c \in {}^{\mathbf{C}}e$ and $c \in \mathrm{Tc}(t_0)$, contradicting Lemma 3.5.1 (3). Therefore, from the definition of $\mathrm{Ev}_*(\texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od})$, we must have ${}^{\mathbf{C}}e = \mathsf{body}{:}\mathrm{Tc}(t_0)$ and so $C \setminus {}^{\mathbf{C}}e = \emptyset$. It is therefore trivially the case that $C \setminus {}^{\mathbf{C}}e \cap e^{\mathbf{C}} = \emptyset$.

Second, we might have $C = \mathsf{body}{:}C_0$ for some $C_0 \neq \mathrm{Tc}(t_0)$ such that $\mathcal{C}[\![t_0]\!]{:}\mathrm{Ic}(t_0) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_0$. From Lemma 3.7.1, the net $\mathcal{C}[\![t]\!]$ is well-terminating and therefore $\mathrm{Tc}(t_0) \not\subseteq C_0$. From the definition of $\mathrm{Ev}_*(\texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od})$, there must therefore exist $e_0 \in \mathrm{Ev}_*(t_0)$

such that $e = \mathsf{body}{:}e_0$. From this, it is easy to see that the net $\mathcal{C}[\![t_0]\!]$ is not safe, contradicting the induction hypothesis.

The final case is $C = \{\mathtt{t}\}$. There is clearly no event in $\mathrm{Ev}_*(\mathtt{while}\ b\ \mathtt{do}\ t\ \mathtt{od})$ such that ${}^{\mathrm{c}}e \subseteq \{\mathtt{t}\}$. □

3.9 Terminality

It will sometimes be useful to regard the net $\mathcal{C}[\![t]\!]$ as just a (labelled) safe net, for example to apply constructions defined in the category of safe nets such as pullbacks. Since safe nets have an initial marking, in doing so we lose no information on the initial conditions of the net $\mathcal{C}[\![t]\!]$. However, by avoiding adding extra structure to safe nets, we do lose the explicit representation of the terminal marking. We require an alternative characterization of the terminal marking of the net $\mathcal{C}[\![t]\!]$.

Lemma 3.9.1. *For any closed term t, the marking $\mathrm{Tc}(t)$ is the unique marking reachable from all markings reachable from $\mathrm{Ic}(t)$ in $\mathcal{C}[\![t]\!]$ for which there exists no $e \in \mathrm{Ev}_*(t)$ with concession in $\mathrm{Tc}(t)$.*

Proof. The proof proceeds by induction on the size of terms. We show only only two representative cases for t.

$\underline{t = t_1 \parallel t_2}$: Suppose that C is reachable from $\mathrm{Ic}(t_1 \parallel t_2)$ in $\mathcal{C}[\![t_1 \parallel t_2]\!]$. According to Lemma 3.6.5, we have $C = \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2$ for some control markings C_1 and C_2 such that $\mathcal{C}[\![t_1]\!]{:}\mathrm{Ic}(t_1) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ and $\mathcal{C}[\![t_2]\!]{:}\mathrm{Ic}(t_2) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_2$. By induction, the markings $\mathrm{Tc}(t_1)$ and $\mathrm{Tc}(t_2)$ are reachable from C_1 and C_2 in $\mathcal{C}[\![t_1]\!]$ and $\mathcal{C}[\![t_2]\!]$, respectively. It is easy to see from that that $\mathcal{C}[\![t_1 \parallel t_2]\!]{:}C \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} \mathsf{par}\,1{:}\mathrm{Tc}(t_1) \cup \mathsf{par}\,2{:}\mathrm{Tc}(t_2)$.

To see that no event has concession in $\mathrm{Tc}(t_1 \parallel t_2)$, suppose for contradiction that $e \in \mathrm{Ev}_*(t_1 \parallel t_2)$ is an event with concession in $\mathrm{Tc}(t_1 \parallel t_2)$. Without loss of generality, suppose that $e = \mathsf{par}\,1{:}e_1$ for some $e_1 \in \mathrm{Ev}_*(t_1)$. Since $\mathrm{Tc}(t_1 \parallel t_2) = \mathsf{par}\,1{:}\mathrm{Tc}(t_1) \cup \mathsf{par}\,2{:}\mathrm{Tc}(t_2)$, we clearly have $\mathrm{Tc}(t_1) \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow}$, contradicting the induction hypothesis for t_1.

Now suppose that there exists a marking C reachable from $\mathrm{Ic}(t_1 \parallel t_2)$ in $\mathcal{C}[\![t_1 \parallel t_2]\!]$ in which no event has concession. We shall show that $C = \mathrm{Tc}(t_1 \parallel t_2)$, demonstrating the required uniqueness property. By Lemma 3.6.5, there exist C_1 and C_2 such that $\mathcal{C}[\![t_1]\!]{:}\mathrm{Ic}(t_1) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ and $\mathcal{C}[\![t_2]\!]{:}\mathrm{Ic}(t_2) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} C_2$ and $C = \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2$. Suppose that $C_1 \neq \mathrm{Tc}(t_1)$. Then, by induction, there exists e_1 such that $C_1 \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow}$ and hence $\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2 \stackrel{\mathsf{par}\,1:e_1}{\longrightarrow\!\!\!\!\!\rightarrow}$, contradicting the assumption that no event has concession in C. We therefore have $C_1 = \mathrm{Tc}(t_1)$ and a similar argument shows $C_2 = \mathrm{Tc}(t_2)$, and so $C = \mathsf{par}\,1{:}\mathrm{Tc}(t_1) \cup \mathsf{par}\,2{:}\mathrm{Tc}(t_2) = \mathrm{Tc}(t_1 \parallel t_2)$ as required.

$\underline{t = t_1; t_2}$: Let $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$. We first show that the marking $\mathrm{Tc}(t_1; t_2) = P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$ is reachable from any marking C reachable from $\mathrm{Ic}(t_1; t_2) = P \triangleleft \mathsf{seq}\,1{:}\mathrm{Ic}(t_1)$.

By Lemma 3.6.2, since C is reachable from $\mathrm{Ic}(t_1; t_2)$, either $C = P \triangleleft \mathsf{seq}\,1{:}C_1$ for some C_1 reachable from $\mathrm{Ic}(t_1)$ in $\mathcal{C}[\![t_1]\!]$ or $C = P \triangleright \mathsf{seq}\,2{:}C_2$ for some C_2 reachable from $\mathrm{Ic}(t_2)$ in $\mathcal{C}[\![t_2]\!]$.

In the first case, by induction $\mathrm{Tc}(t_1)$ is reachable from C_1 so $P \triangleleft \mathsf{seq}\,1{:}\mathrm{Tc}(t_1)$ is reachable from C according to Lemma 3.3.2. Also by induction, $\mathrm{Tc}(t_2)$ is reachable from $\mathrm{Ic}(t_2)$ and hence $P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$ is reachable from $P \triangleright \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$ in $\mathcal{C}[\![t_1; t_2]\!]$.

Since $P \triangleleft \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) = P \triangleright \mathsf{seq}\,2{:}\mathrm{Ic}(t_2) = P$ and $\mathrm{Tc}(t_1;t_2) = P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$, it follows that $\mathrm{Tc}(t_1;t_2)$ is reachable from C in $\mathcal{C}[\![t_1;t_2]\!]$.

In the second case, by induction $\mathrm{Tc}(t_2)$ is reachable from C_2 and hence $P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$ is reachable from $P \triangleright \mathsf{seq}\,2{:}C_2$, thus demonstrating that $\mathrm{Tc}(t_1;t_2)$ is reachable from any reachable marking in $\mathcal{C}[\![t_1;t_2]\!]$.

Now suppose, for contradiction, that an event e has concession in $\mathrm{Tc}(t_1;t_2)$. There are two kinds of event present in $\mathcal{C}[\![t_1;t_2]\!]$. It cannot be the case that $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ for some $e_1 \in \mathrm{Ev}_*(t_1)$, since according to Lemma 3.5.1 there would exist $c \in {}^\bullet e \setminus \mathrm{Tc}(t_1)$, and hence $\mathsf{seq}\,1{:}c \in {}^\bullet e$. It is easy to see that $\mathsf{seq}\,1{:}c \notin P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$, making it impossible for the event e to have concession. It must therefore be the case that $e = P \triangleright \mathsf{seq}\,2{:}e_2$ for some $e_2 \in \mathrm{Ev}_*(t_2)$. We have $P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2) \xrightarrow{P \triangleright \mathsf{seq}\,2:e_2}\!\!\!\!\!\rightarrow$, so by Lemma 3.3.2 we also have $\mathrm{Tc}(t_2) \xrightarrow{e_2}\!\!\!\!\!\rightarrow$, contradicting the induction hypothesis for t_2.

Considering the uniqueness of $\mathrm{Tc}(t_1;t_2)$, suppose that C is reachable from $\mathrm{Ic}(t_1;t_2)$ and that no event has concession in C. There are two cases to consider for C according to Lemma 3.6.2; we shall show that in each case $C = \mathrm{Tc}(t_1;t_2)$.

First, if $C = P \triangleleft \mathsf{seq}\,1{:}C_1$ for some C_1 such that $\mathcal{C}[\![t_1]\!]{:}\mathrm{Ic}(t_1) \xrightarrow{*}\!\!\!\!\!\rightarrow C_1$ then $C_1 = \mathrm{Tc}(t_1)$ since otherwise the induction hypothesis would yield an event e_1 such that $C_1 \xrightarrow{e_1}\!\!\!\!\!\rightarrow$ and hence $C \xrightarrow{P \triangleleft \mathsf{seq}\,1:e_1}\!\!\!\!\!\rightarrow$. We therefore have

$$C = P \triangleleft \mathsf{seq}\,1{:}C_1 = P \triangleleft \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) = P = P \triangleright \mathsf{seq}\,2{:}\mathrm{Ic}(t_2).$$

If $\mathrm{Ic}(t_2) \neq \mathrm{Tc}(t_2)$ then by induction there exists an event e_2 such that $\mathrm{Ic}(t_2) \xrightarrow{e_2}\!\!\!\!\!\rightarrow$ and hence $C \xrightarrow{P \triangleright \mathsf{seq}\,2:e_2}\!\!\!\!\!\rightarrow$ by Lemma 3.3.2, contradicting the assumption that no event has concession in C. Hence, in this case, we must have $C = \mathrm{Tc}(t_1;t_2)$.

In the second case, there exists C_2 such that $C = P \triangleright \mathsf{seq}\,2{:}C_2$ and $\mathcal{C}[\![t_2]\!]{:}\mathrm{Ic}(t_2) \xrightarrow{*}\!\!\!\!\!\rightarrow C_2$. If $C_2 \neq \mathrm{Tc}(t_2)$, by induction there exists e_2 such that $C_2 \xrightarrow{e_2}\!\!\!\!\!\rightarrow$ and hence, by Lemma 3.3.2, we have $C \xrightarrow{P \triangleright \mathsf{seq}\,2:e_2}\!\!\!\!\!\rightarrow$. This contradicts the assumption that no event has concession in C, so we must have $C_2 = \mathrm{Tc}(t_2)$ and hence $C = P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2) = \mathrm{Tc}(t_1;t_2)$ as required. □

3.10 Preservation of consistent markings

We have now developed quite a comprehensive understanding of the net $\mathcal{C}[\![t]\!]$. This turns out to be necessary to show that any marking of state conditions $\sigma = (D, L, R, N)$ reachable in $\mathcal{N}[\![t]\!]$ from a consistent initial marking of state conditions σ_0 is itself consistent. The challenge here will be showing that if $r \in R$ then $\mathsf{curr}(r) \in N$, which shall require some understanding of the nature of the critical regions present in our semantics; the other requirements for consistency are straightforwardly shown to be preserved through the occurrence of the events present in $\mathcal{N}[\![t]\!]$.

We shall first show that any release of a resource is dependent on the prior acquisition of that resource: for any sequence π and any resource there exists an injection f that associates any occurrence of a release event to a prior occurrence of an acquisition event of that resource, and between the two occurrences there are no other actions on that resource.

Lemma 3.10.1. *Let π be a sequence of events, $\pi = (e_1, \ldots, e_n)$. For any closed term t, resource r and marking of control conditions C such that $\mathcal{C}[\![t]\!]{:}\mathrm{Ic}(t) \xrightarrow{\pi}\!\!\!\!\!\rightarrow C$, there exists a partial function $f{:}\mathbb{N} \rightharpoonup \mathbb{N}$ satisfying, for all $i, j \in \mathbb{N}$:*

- *f is injective,*
- *if there exist sets of control conditions C_1, C_2 such that $e_i = \mathsf{rel}_{(C_1,C_2)}(r)$ then $f(i)$ defined, and*
- *if $f(i)$ defined then $f(i) < i$ and there exist sets of control conditions C_1, C_2 such that $e_{f(i)} = \mathsf{acq}_{(C_1,C_2)}(r)$.*

Moreover, if there exist markings of state conditions $\sigma_0, \ldots, \sigma_n$ and markings of control conditions $C_0, \cdots, C_n$ such that σ_i is consistent for all $0 \leq i < n$ and $\mathcal{N}[\![t]\!] : (C_{i-1}, \sigma_{i-1}) \xrightarrow{e_i} (C_i, \sigma_i)$ for all i with $0 < i \leq n$ and $C_0 = \mathrm{Ic}(t)$, then there exists an f satisfying the above constraints and such that, for all k with $f(i) < k < i$, there exist no C' and C'' such that either $e_k = \mathsf{acq}_{(C',C'')}(r)$ or $e_k = \mathsf{rel}_{(C',C'')}(r)$.

Proof. The first part of the lemma about runs of $\mathcal{C}[\![t]\!]$ is shown, using the control properties of sequences established above, straightforwardly by induction on the size of terms.

The second part of the lemma follows from the first part along with the observation that if $e_i = \mathsf{acq}_{(C_i,C_i')}(r)$ and $e_j = \mathsf{acq}_{(C_j,C_j')}(r)$ for $i < j$ then there must exist k such that $i < k < j$ and $e_k = \mathsf{rel}_{(C_k,C_k')}(r)$. Otherwise, the resource r would not be available in state σ_{j-1}. This relies on consistency of all σ_k. □

We are now able to show that the nets formed preserve the consistency of the markings of state conditions.

Proposition 3.3 (Preservation of consistent markings)**.** *For any closed term t and sequence of events π', if $(\mathrm{Ic}(t), \sigma_0) \xrightarrow{\pi'} (C, \sigma)$ in the net $\mathcal{N}[\![t]\!]$ and the marking σ_0 of state conditions is consistent then σ is consistent.*

Proof. It is easy to see that the events present in that net $\mathcal{N}[\![t]\!]$ are all of one of the following forms:

$$\begin{array}{llll} \mathsf{act}_{(C,C')}(D, D') & \mathsf{alloc}_{(C,C')}(\ell, v, \ell', v') & \mathsf{decl}_{(C,C')}(r) & \mathsf{acq}_{(C,C')}(r) \\ & \mathsf{dealloc}_{(C,C')}(\ell, \ell', v') & \mathsf{end}_{(C,C')}(r) & \mathsf{rel}_{(C,C')}(r) \end{array}$$

It is readily shown that each form of event preserves the consistency of the marking of state conditions, apart from showing that if $r \in \sigma$ then $\mathsf{curr}(r) \in \sigma$.

Suppose, for contradiction, that π' is a path such that $\mathcal{N}[\![t]\!] : (\mathrm{Ic}(t), \sigma_0) \xrightarrow{\pi'}{}^{*} (C, \sigma)$ and that $r \in \sigma$ but $\mathsf{curr}(r) \notin \sigma$. Assume, furthermore, and without loss of generality, that any other marking of state conditions σ' along π' has the property that if $r \in \sigma$ then $\mathsf{curr}(r) \in \sigma$, and is therefore consistent. It must be the case that $\pi' = \pi \cdot \mathsf{rel}_{(K_1,K_1')}(r)$ for some K_1, K_1' and π. By Lemma 3.10.1, there exist K_2, K_2', π_1 and π_2 such that $\pi = \pi_1 \cdot \mathsf{acq}_{(K_2,K_2')}(r) \cdot \pi_2$ and no event in π_2 is an $\mathsf{acq}(r)$ or $\mathsf{rel}(r)$ event. Let (C_1, σ_1) be the marking obtained by the path $\mathcal{N}[\![t]\!] : (\mathrm{Ic}(t), \sigma_0) \xrightarrow{\pi_1}{}^{*} (C_1, \sigma_1)$. We must have $r \in \sigma_1$, and by assumption $\mathsf{curr}(r) \in \sigma_1$. It can be seen that we must have $\mathsf{curr}(r) \in \sigma'$ and $r \notin \sigma'$ for all states σ' reached along $\mathsf{acq}_{(D_2,D_2')}(r) \cdot \pi_2$ from (C_1, σ_1) since no $\mathsf{end}(r)$ event can have concession in such markings. Consequently, we must have $\mathsf{curr}(r) \in \sigma_2$ for σ_2 obtained by following the path $\mathcal{N}[\![t]\!] : (\mathrm{Ic}(t), \sigma_0) \xrightarrow{\pi} (C_2, \sigma_2)$, and therefore $\mathsf{curr}(r) \in \sigma$ — a contradiction, as required. □

It will also be useful to know that the structure of processes ensures that any resource initially current remains current through the execution of the net. The same property working backwards from the terminal marking of the net also holds.

Lemma 3.10.2. *Let σ, σ' be a consistent markings of state conditions. For any markings of control conditions C, C':*

1. *If $(\mathrm{Ic}(t), \sigma) \xrightarrow{*} (C', \sigma')$ in $\mathcal{N}[\![t]\!]$ and $\mathsf{curr}(r) \in \sigma$ then $\mathsf{curr}(r) \in \sigma'$.*

2. *If $(C, \sigma) \xrightarrow{*} (\mathrm{Tc}(t), \sigma')$ in $\mathcal{N}[\![t]\!]$ and $\mathsf{curr}(r) \in \sigma'$ then $\mathsf{curr}(r) \in \sigma$.*

Proof. We shall only show (1) since (2) is similar. An induction on the size of terms using the control properties above gives the following:

- If there exists a sequence π such that

$$\mathcal{C}[\![t]\!] : \mathrm{Ic}(t) \xrightarrow{\pi \cdot \mathsf{end}_{(C_1, C_2)}(r)} C$$

  for some C_1, C_2 then there exists an event $\mathsf{decl}_{(C_1', C_2')}(r)$ in π for some C_1', C_2'.

Let π' be a sequence $\mathcal{N}[\![t]\!] : (\mathrm{Ic}(t), \sigma) \xrightarrow{\pi'} (C', \sigma')$ and assume that $\mathsf{curr}(r) \in \sigma$. Without loss of generality, suppose that (C', σ') is the earliest marking along π' from $(\mathrm{Ic}(t), \sigma)$ such that $\mathsf{curr}(r) \notin \sigma'$; otherwise, we can take the initial segment of π' with this property. Examination of the events given by our semantics reveals that the last event in π' is an $\mathsf{end}_{(C_1, C_2)}(r)$ event, since otherwise $\mathsf{curr}(r)$ is not in the state prior to σ'. Now, applying the result above informs that there is an event $\mathsf{decl}_{(C_1', C_2')}(r)$ in π' and this must occur before $\mathsf{end}_{(C_1, C_2)}(r)$. Now, the event $\mathsf{decl}_{(C_1', C_2')}(r)$ can only occur in a marking σ_0 of state conditions such that $\mathsf{curr}(r) \notin \sigma_0$, but this contradicts our assumption that σ' was the first marking of state conditions reachable along π' from $(\mathrm{Ic}(t), \sigma)$ with $\mathsf{curr}(r) \notin \sigma'$. □

3.11 Net equivalence and correspondence

As well as the net semantics for the programming language, we gave a corresponding operational semantics. The proof of their correspondence is helped by having a form of bisimulation on nets that is a congruence on terms. Standard bisimulation on the net $\mathcal{N}[\![t]\!]$ is not a congruence: it involves the initial marking of state conditions σ_0 and relates configurations (C, σ) comprising a marking of control conditions with a marking of state conditions.

Example 3.11.1. *Let $\sigma_0 = \{\ell \mapsto 0\}$ and consider the nets $\mathcal{N}[\![\ell = 1]\!]$ and $\mathcal{N}[\![\ell = 2]\!]$. Recalling that the action $\ell = 1$ is a boolean guard that can proceed only if ℓ holds value 1, and similarly for $\ell = 2$, neither net has an event that can occur from its initial marking of control conditions in the state σ_0. We therefore have*

$$(\mathcal{N}[\![\ell = 1]\!], \mathrm{Ic}(\ell = 1), \sigma_0) \text{ bisimilar to } (\mathcal{N}[\![\ell = 2]\!], \mathrm{Ic}(\ell = 2), \sigma_0).$$

If we place the processes in parallel with $\ell := 1$ which assigns value 1 to location ℓ, the resulting processes are not bisimilar. That is, the two nets

$$(\mathcal{N}[\![\ell = 1 \parallel \ell := 1]\!], \mathrm{Ic}(\ell = 1 \parallel \ell := 1), \sigma_0)$$
$$(\mathcal{N}[\![\ell = 2 \parallel \ell := 1]\!], \mathrm{Ic}(\ell = 2 \parallel \ell := 1), \sigma_0).$$

are not bisimilar.

Instead, we shall consider an equivalence on control nets. An equivalence on the nets $\mathcal{C}[\![t]\!]$ and $\mathcal{C}[\![t']\!]$ shall be exhibited by a span of **Pom**-open morphisms from a safe net N to $\mathcal{C}[\![t_1]\!]$ and $\mathcal{C}[\![t']\!]$ as described in Section 2.4. It shall therefore be a **Pom**-open map bisimulation, so a (strong) history preserving bisimulation on nets [NW96] — see Section 2.4. Essentially, this can be thought-of as a form of bisimulation that preserves independence.

The proof that the net and operational semantics correspond is interesting but unfortunately rather technical. We therefore present the full development in Appendix E. Since many readers will not be familiar with open map bisimulation on nets, we shall weaken the result presented in the appendix to give a more accessible version based on standard bisimulation here:

Theorem 3.4. *Let t_0 be a closed term and σ_0 be a consistent state. There exists a relation R between markings (C, σ) of the net $\mathcal{N}[\![t_0]\!]$ from σ_0 and configurations $\langle t, \sigma\rangle$ of the operational semantics such that $R((\mathrm{Ic}(t_0), \sigma_0), \langle t_0, \sigma_0\rangle)$ and whenever $R((C, \sigma), \langle t, \sigma\rangle)$:*

- *if $(C, \sigma) \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} (C', \sigma')$ then there exists t' such that $\langle t, \sigma\rangle \xrightarrow{|e|} \langle t', \sigma'\rangle$ and $R((C', \sigma'), \langle t', \sigma'\rangle)$, and*
- *if $\langle t, \sigma\rangle \xrightarrow{\lambda} \langle t', \sigma'\rangle$ then there exist e and C' such that $(C, \sigma) \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} (C', \sigma')$ and $|e| = \lambda$.*

Furthermore, R respects terminality and states in the sense that:
if $R((C, \sigma), \langle t, \sigma'\rangle)$ then $\sigma = \sigma'$ and if $C = \mathrm{Tc}(t_0)$ then $t \equiv \varepsilon$.

Proof. Immediate from Theorem E.2. □

In Appendix E, we show that this result means that the net semantics is adequate with respect to the operational semantics.

Chapter 4

Concurrent separation logic

As discussed in the introduction, concurrent separation logic provides a system for reasoning about concurrent heap-manipulating programs. At the core of the logic is the notion of *separation of ownership*, which can be thought of as meaning that parallel processes each own a separate part of the heap and may only access the part that they own. This allows the unconstrained forms of interaction described in the introduction due to aliasing to be discounted, allowing a compositional logic to be developed.

We now give an account of separation logic focusing on ownership. A semantics for the logic will be given by defining an interpretation of validity for judgements of concurrent separation logic. The rules for the judgement $\vdash \{\varphi\}t\{\psi\}$ derivable according to the logic will be proved sound (though an example of their incompleteness will be shown).

For a fuller introduction of separation logic than that to be presented here, we refer the reader to [O'H07]. On a related point, the logic that we consider is be slightly different from that in [O'H07] in that we shall not have program variables inside the language. Instead, as in the net semantics presented earlier, programs directly access heap locations. In examples, we assume the existence of conveniently-named locations to make them readable. It would be very easy to adapt this work to deal with variables, but we shall refrain from adapting them here since they introduce untidiness into the logic that is orthogonal to the issues that we wish to investigate.

4.1 Heap formulae

Concurrent separation logic establishes partial correctness assertions about concurrent heap-manipulating programs; that whenever a given program running from a heap satisfying a *heap formula* φ terminates, the resulting heap satisfies a heap formula ψ. The semantics of the heap logic arises as an instance of the logic of Bunched Implications [OP99, Pym02]. At its core are the associated notions of heap composition and the separating conjunction of heap formulae. Two heaps may be composed if they are defined over disjoint sets of locations:

$$D_1 \cdot D_2 \triangleq D_1 \cup D_2 \text{ if } \mathrm{dom}(D_1) \cap \mathrm{dom}(D_2) = \emptyset.$$

A heap satisfies the separating conjunction $\varphi_1 \star \varphi_2$ if it can be split into two parts, one satisfying φ_1 and the other φ_2:

$$\begin{array}{lll} D \models \varphi_1 \star \varphi_2 & \text{iff} & \text{there exist } D_1, D_2 \text{ such that } D_1 \cdot D_2 \text{ defined and} \\ & & D = D_1 \cdot D_2 \text{ and } \; D_1 \models \varphi_1 \text{ and } \; D_2 \models \varphi_2. \end{array}$$

The full syntax and semantics of the heap logic is given in Figures 4.2 and 4.1. The heap logic augments the standard logical connectives $\wedge$, $\vee$ and $\rightarrow$, negation $\neg$ and truth values $\top$ and $\bot$ with primitives for describing the structure of the heap. We adopt the usual binding precedences, and $\star$ binds more tightly the standard logical connectives. In addition to the separating conjunction, there is a formula empty to assert that the heap is empty (has no locations in its domain). There is also a formula $e_{\mathsf{loc}} \mapsto e'$ to assert that the heap consists of a single location represented by the location expression e_{loc} that holds a value represented by the expression e'. According to the syntax of the logic, there are two kinds of expression: location expressions e_{loc}, which are either a fixed location or a variable of the logic representing a location, ranged over by x_{loc}, and value expressions e_{val}, which may be either a constant value or a variable representing a value of arbitrary kind, ranged over by x_{val}. The logic has existential and universal quantifiers which bind variables, giving rise to the standard definition of *closed* formula as being a formula with no free variables. By restricting the syntax of expressions in this way, we rule out meaningless formulae such as

$$v \mapsto v'$$

for a value v that is not a location, and do not have to give a semantics to them. We define the shorthand notation $\ell \mapsto \_$ for $\exists x_{\mathsf{val}}.(\ell \mapsto x_{\mathsf{val}})$.

Example 4.1.1. *Let $D = \{\ell \mapsto 0, \ell' \mapsto 1\}$. We have $D \models \ell \mapsto 0 \star \top$ since $D = \{\ell \mapsto 0\} \cdot \{\ell' \mapsto 1\}$ and $\{\ell \mapsto 0\} \models \ell \mapsto 0$ and $\{\ell' \mapsto 1\} \models \top$. From the semantics, we therefore have $D \models \exists x_{\mathsf{loc}}.(x_{\mathsf{loc}} \mapsto 0 \star \top)$.*

We do not, however, have $D \models \ell \mapsto 0$ since the semantics of this assertion would require that $D = \{\ell \mapsto 0\}$. The heap logic is therefore not intuitionistic in the sense of [Rey02, Rey00, O'H07].

Unlike the heap logic presented in [Bro07], we do not allow arithmetic on memory locations; this is just to simplify the presentation, and such arithmetic could easily be added. Alongside the definition of satisfaction of a formula given in Figure 4.2, there is the associated definition of validity, written $\models \varphi$, meaning that $D \models \varphi$ for all heaps D. Logical entailment is written as $\varphi \implies \psi$, equivalent to $\models \varphi \rightarrow \psi$.

4.2 Ownership

Concurrent separation logic as presented in [O'H07] is a system of proof rules forming a judgement. In preparation for the formal definition of validity for judgements, $\Gamma \models \{\varphi\}t\{\psi\}$, to be presented in Chapter 5 and the subsequent proof of soundness, we present the intuition for what the key judgement of concurrent separation logic, $\Gamma \vdash \{\varphi\}t\{\psi\}$, means. Here, φ and ψ are formulae of the heap logic, and Γ is a *environment* of *resource invariants* of the form $r_1 : \chi_1, \cdots, r_n : \chi_n$, associating heap formulae χ_i called *invariants* with resource constants r_i.

> *In any run from a heap satisfying φ and the environment of invariants Γ, the process t never accesses locations that it does not own, and if the process t terminates then it does so in a heap satisfying ψ and the invariants Γ.*

Central to this understanding is the notion of *ownership*, which we capture formally in the Chapter 5. Initially the process t is considered to *own* that part of the heap which satisfies φ, and accordingly to *own* the locations in that subheap.

Variables:	x	::=	x_{loc}	Location variable
		|	x_{val}	Value variable
Location expressions:	e_{loc}	::=	x_{loc}	Location variable
		|	ℓ	Location, $\ell \in \mathrm{Loc}$
Expressions:	e	::=	e_{loc}	Location expression
		|	x_{val}	Value variable
		|	v	Value, $v \in \mathrm{Val}$
Formulae:	φ	::=	$e_{\mathsf{loc}} \mapsto e$	heap location
		|	$\varphi \star \varphi$	separating conjunction
		|	empty	empty heap
		|	$\varphi \wedge \varphi$	conjunction
		|	$\varphi \vee \varphi$	disjunction
		|	$\varphi \rightarrow \varphi$	implication
		|	$\neg\varphi$	negation
		|	$\exists x.\varphi$	existential quantification
		|	$\forall x.\varphi$	universal quantification
		|	$e = e$	equality
		|	$\top$	true
		|	$\bot$	false

Figure 4.1: Syntax of the heap logic

Semantics of closed formulæ:

$D \models \ell \mapsto v$	iff	$D = \{\ell \mapsto v\}$
$D \models \varphi_1 \star \varphi_2$	iff	there exist D_1, D_2 such that $D_1 \cdot D_2$ defined and $D = D_1 \cdot D_2$ and $D_1 \models \varphi_1$ and $D_2 \models \varphi_2$
$D \models \mathsf{empty}$	iff	$D = \emptyset$
$D \models \varphi_1 \wedge \varphi_2$	iff	$D \models \varphi_1$ and $D \models \varphi_2$
$D \models \varphi_1 \vee \varphi_2$	iff	$D \models \varphi_1$ or $D \models \varphi_2$
$D \models \varphi_1 \rightarrow \varphi_2$	iff	$D \models \varphi_1$ implies $D \models \varphi_2$
$D \models \neg\varphi$	iff	not $D \models \varphi$
$D \models \exists x_{\mathsf{loc}}.\varphi$	iff	there exists $\ell \in \mathrm{Loc}$ such that $D \models [\ell/x_{\mathsf{loc}}]\varphi$
$D \models \exists x_{\mathsf{val}}.\varphi$	iff	there exists $v \in \mathrm{Val}$ such that $D \models [v/x_{\mathsf{val}}]\varphi$
$D \models \forall x_{\mathsf{loc}}.\varphi$	iff	for all $\ell \in \mathrm{Loc}$: $D \models [\ell/x_{\mathsf{loc}}]\varphi$
$D \models \forall x_{\mathsf{val}}.\varphi$	iff	for all $v \in \mathrm{Val}$: $D \models [v/x_{\mathsf{val}}]\varphi$
$D \models v = v'$	iff	$v = v'$
$D \models \top$	always	
$D \models \bot$	never	

Figure 4.2: Semantics of the heap logic

Ownership plays a key role in making the judgements of concurrent separation logic compositional: a judgement $\Gamma \vdash \{\varphi\}t\{\psi\}$ should hold even if other (unknown) processes are to execute in the same heap. It is therefore necessary to make certain assumptions about the ways in which these other processes might interact with the process t. This is achieved through ownership, by assuming that each process owns, throughout its execution, a part of the heap disjoint from that owned by other processes. The part of the heap that each process owns must not be accessed by any other process, and moreover a process must not access locations that it does not own. Essentially, this gives rise to a form of rely-guarantee reasoning, taken further in [VP07], where the judgement relies on the fact the other processes to execute do not access the locations that t is said to own and, dually, guarantees that t only accesses locations that it owns.

There is some subtlety, to be teased-out in our model, due to the fact that as t runs the locations it owns may change. This will be through the acquisition and release of resources, when it will gain and relinquish ownership of the locations used in justifying their invariants. Ownership will also be affected by the allocation of resources, when it will gain ownership of the newly-allocated location, and by deallocation.

The inference rules of concurrent separation logic are presented in Figures 4.3 and 4.4 in the style of [Bro07]. The only significant difference between the two systems is that we omit the rules for auxiliary variables and for existential quantification. Both are omitted for simplicity since they are peripheral to the focus of our work.

As a first example, the rule for heap actions (L-Act) would allow the judgement

$$\Gamma \vdash \{\ell \mapsto 0\}\ell := 1\{\ell \mapsto 1\}.$$

When considering this judgement, we see the process as owning the part of the heap, and this part satisfies the formula $\ell \mapsto 0$. The part owned by the process therefore consists of the location ℓ which holds value 0. Following execution of the assignment, the part of the heap that we see as owned by the process satisfies $\ell \mapsto 1$. It should be emphasized that ownership is not assumed to be actually present in the semantics of processes; it is overlaid to understand the judgements made by the logic.

Following the account above, the assignment will not access any location that we do not see it as owning. The judgement

$$\Gamma \vdash \{\ell \mapsto 0 \star k \mapsto 0\}\ell := 1\{\ell \mapsto 1 \star k \mapsto 0\}$$

is therefore valid since the location k will not be accessed. More generally, for any formula φ the judgement

$$\Gamma \vdash \{\ell \mapsto 0 \star \varphi\}\ell := 1\{\ell \mapsto 1 \star \varphi\}$$

is valid since the part of the heap satisfying φ is disjoint from the part of the heap satisfying $\ell \mapsto 0$ according to the semantics of the separating conjunction $\star$, and the earlier judgement $\Gamma \vdash \{\ell \mapsto 0\}\ell := 1\{\ell \mapsto 1\}$ ensures that the process does not access any locations used to satisfy φ. It follows that the execution of the process does not affect the holding of φ, so whenever the process terminates the resulting state satisfies $\ell \mapsto 1 \star \varphi$. This is generalized to form the important *frame rule* (L-Frame), which depends critically upon the hypothesis that the process can only ever access locations that we see it as owning, called the *ownership hypothesis*.

To see how the proof system would collapse if we were to allow the ownership hypothesis to be broken, suppose that we have the judgement

$$\Gamma \vdash \{\top\}\ell := 2\{\top\}.$$

(L-Nil) : $$\Gamma \vdash \{\varphi\}\varepsilon\{\varphi\}$$

(L-Act) : $$\frac{\begin{array}{c}\text{for all } D \models \varphi \text{ and } (D_1, D_2) \in \mathcal{A}[\![\alpha]\!] : \\ \mathrm{dom}(D_1) \subseteq \mathrm{dom}(D) \\ \text{and} \\ D_1 \subseteq D \text{ implies } (D \setminus D_1) \cup D_2 \models \psi\end{array}}{\Gamma \vdash \{\varphi\}\alpha\{\psi\}}$$

(L-Alloc) : $$\Gamma \vdash \{\ell \mapsto \_\}\texttt{alloc}(\ell)\{\exists x_{\mathsf{loc}}.(\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_)\}$$

(L-Dealloc) : $$\Gamma \vdash \{\exists x_{\mathsf{loc}}.(\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_)\}\texttt{dealloc}(\ell)\{\exists x_{\mathsf{loc}}.(\ell \mapsto x_{\mathsf{loc}})\}$$

(L-Seq) : $$\frac{\Gamma \vdash \{\varphi\}t_1\{\varphi'\} \qquad \Gamma \vdash \{\varphi'\}t_2\{\psi\}}{\Gamma \vdash \{\varphi\}t_1; t_2\{\psi\}}$$

(L-Sum) : $$\frac{\begin{array}{cc}\Gamma \vdash \{\varphi\}\alpha_1\{\varphi_1\} & \Gamma \vdash \{\varphi\}\alpha_2\{\varphi_2\} \\ \Gamma \vdash \{\varphi_1\}t_1\{\psi\} & \Gamma \vdash \{\varphi_2\}t_2\{\psi\}\end{array}}{\Gamma \vdash \{\varphi\}\alpha_1.t_1 + \alpha_2.t_2\{\psi\}}$$

(L-While) : $$\frac{\begin{array}{cc}\Gamma \vdash \{\varphi\}b\{\varphi'\} & \Gamma \vdash \{\varphi\}\neg b\{\psi\} \\ \multicolumn{2}{c}{\Gamma \vdash \{\varphi'\}t\{\varphi\}}\end{array}}{\Gamma \vdash \{\varphi\}\texttt{while } b \texttt{ do } t \texttt{ od}\{\psi\}}$$

(L-Res) : $$\frac{\Gamma, r{:}\chi \vdash \{\varphi\}[r/w]t\{\psi\}}{\Gamma \vdash \{\varphi \star \chi\}\texttt{resource } w \texttt{ do } t \texttt{ od}\{\psi \star \chi\}} \left(\begin{array}{l}\chi \text{ precise} — \text{see p67} \\ r \notin \mathrm{dom}(\Gamma)\end{array}\right)$$

(L-CR) : $$\frac{\Gamma, r{:}\chi \vdash \{\varphi \star \chi\}t\{\psi \star \chi\}}{\Gamma, r{:}\chi \vdash \{\varphi\}\texttt{with } r \texttt{ do } t \texttt{ od}\{\psi\}}$$

(L-Par) : $$\frac{\Gamma \vdash \{\varphi_1\}t_1\{\psi_1\} \qquad \Gamma \vdash \{\varphi_2\}t_2\{\psi_2\}}{\Gamma \vdash \{\varphi_1 \star \varphi_2\}t_1 \parallel t_2\{\psi_1 \star \psi_2\}}$$

Figure 4.3: Syntax-directed rules of concurrent separation logic

(L-Frame) : $$\frac{\Gamma \vdash \{\varphi\}t\{\psi\}}{\Gamma \vdash \{\varphi \star \varphi'\}t\{\psi \star \varphi'\}}$$

(L-Consequence) : $$\frac{\varphi \implies \varphi' \qquad \Gamma \vdash \{\varphi'\}t\{\psi'\} \qquad \psi' \implies \psi}{\Gamma \vdash \{\varphi\}t\{\psi\}}$$

(L-Conjunction) : $$\frac{\Gamma \vdash \{\varphi_1\}t\{\psi_1\} \qquad \Gamma \vdash \{\varphi_2\}t\{\psi_2\}}{\Gamma \vdash \{\varphi_1 \wedge \varphi_2\}t\{\psi_1 \wedge \psi_2\}}$$

(L-Disjunction) : $$\frac{\Gamma \vdash \{\varphi_1\}t\{\psi_1\} \qquad \Gamma \vdash \{\varphi_2\}t\{\psi_2\}}{\Gamma \vdash \{\varphi_1 \vee \varphi_2\}t\{\psi_1 \vee \psi_2\}}$$

(L-Expansion) : $$\frac{\Gamma \vdash \{\varphi\}t\{\psi\}}{\Gamma, \Gamma' \vdash \{\varphi\}t\{\psi\}}$$

(L-Contraction) : $$\frac{\Gamma, \Gamma' \vdash \{\varphi\}t\{\psi\}}{\Gamma \vdash \{\varphi\}t\{\psi\}}(\mathrm{res}(t) \subseteq \mathrm{dom}(\Gamma))$$

Figure 4.4: Logical rules of concurrent separation logic

This breaks the ownership hypothesis since the process might initially own no part of the heap (since the empty heap satisfies the heap assertion $\top$) but accesses the location ℓ. Using the frame rule, we could then derive

$$\Gamma \vdash \{\top \star \ell \mapsto 0\}\ell := 2\{\top \star \ell \mapsto 0\}.$$

This clearly spurious judgement implies that, running from a heap in which part of that owned by the process is the location ℓ holding value 0, whenever the assignment completes, part of the resulting heap owned by the process is the location ℓ holding value 0.

As an aside, note that the judgement

$$\Gamma \vdash \{\bot\}\ell := 1\{\bot\}$$

is derivable using the rule (L-Act). This is consistent with the informal account earlier, which only applies to runs from states of which the part owned by the process initially satisfies $\bot$; there are, of course, no such states.

Another instance of the separating conjunction is seen in the rule for parallel composition, (L-Par):

$$\frac{\Gamma \vdash \{\varphi_1\}t_1\{\psi_1\} \qquad \Gamma \vdash \{\varphi_2\}t_2\{\psi_2\}}{\Gamma \vdash \{\varphi_1 \star \varphi_2\}t_1 \parallel t_2\{\psi_1 \star \psi_2\}}$$

Informally, the rule is sound because the part of the initial heap that is owned by the process $t_1 \parallel t_2$ can be split into two parts, one part satisfying φ_1 owned by t_1 and the other satisfying φ_2 owned by t_2; as the processes execute the subheaps that we see each as owning remain disjoint from each other and end up separately satisfying ψ_1 and ψ_2.

It is vital to the soundness of the rule (L-Par) that the logic enforces the requirement that processes only act on locations that they own. If this requirement were not imposed, so that the judgement

$$\Gamma \vdash \{\top\}\ell := 2\{\top\}$$

were derivable, then the rule for parallel composition could be applied with the other judgement above to conclude that

$$\Gamma \vdash \{\ell \mapsto 0 \star \top\}\ell := 1 \parallel \ell := 2\{\ell \mapsto 1 \star \top\}.$$

This flawed assertion would imply that whenever the process $\ell := 1 \parallel \ell := 2$ runs from a state in which ℓ holds value 0, the location ℓ always holds value 1 in the resulting state, which is obviously wrong.

To allow the logic to make judgements beyond those applicable to the almost 'disjointly concurrent' programs outlined so far, further interaction is allowed through a system of *invariants*. The judgement environment Γ records a formula called an invariant for each resource in its domain, which contains all the resources occurring in the term. The intuition (as presented in [O'H07]) is that whenever a resource r with an invariant χ is available, there is part of the heap unowned by any other process and protected by the resource that satisfies χ. In such a situation, we shall say that the locations used to satisfy χ are 'owned' by the invariant for r. Processes may gain ownership of these locations, and thereby the right to access them, by entering a critical region protected by the resource. When the process leaves the critical region, the invariant must be restored and the ownership of the locations used to satisfy the invariant is relinquished. This is reflected in the rule (L-CR). As an example, we have the following derivation:

$$\frac{\dfrac{}{r{:}\ell \mapsto 0 \vdash \{\ell' \mapsto \_ \star \ell \mapsto 0\}\ell' := \ell\{\ell' \mapsto 0 \star \ell \mapsto 0\}}\ (\text{L-Act})}{r{:}\ell \mapsto 0 \vdash \{\ell' \mapsto \_\}\texttt{with}\ r\ \texttt{do}\ \ell' := \ell\ \texttt{od}\{\ell' \mapsto 0\}}\ (\text{L-CR})$$

The process initially owns the location ℓ' and the location ℓ is protected by the resource r. We reason about the process inside the critical region running from a state with ownership of the locations governed by the invariant in addition to those that it owned before entering the critical region since no other process can be operating on them; that is, we reason about $\ell' := \ell$ with locations ℓ and ℓ' owned by the process. However, when the process leaves the critical region, ownership of the locations used to satisfy the invariant is lost, indicated by the conclusion $\ell' \mapsto 0$ in the judgement rather than $\ell' \mapsto 0 \star \ell \mapsto 0$.

An invariant is required to be a *precise* heap logic formula.

Definition 4.2.1 (Precision)**.** *A heap logic formula χ is* precise *if for any heap D there is at most one subheap $D_0 \subseteq D$ such that $D_0 \models \chi$.*

We leave discussion of the role of precision to the conclusion, though it might be seen to be of use since it identifies uniquely the part of the heap that is owned by the invariant if the resource is available. Formally, Γ ranges over finite partial functions from resources to precise heap formulae. We write $\mathrm{dom}(\Gamma)$ for the set of resources on which Γ is defined, and write Γ, Γ' for the union of the two partial functions, defined only if $\mathrm{dom}(\Gamma) \cap \mathrm{dom}(\Gamma') = \emptyset$. We write $r{:}\chi$ for the singleton environment taking resource r to χ, and we allow ourselves to write $r{:}\chi \in \Gamma$ if $\Gamma(r) = \chi$.

The final remark to be made on the rules of the logic is that (L-Res) allows invariants to be established for newly declared resources. We reason about the closed term $[r/w]t$, for an arbitrary 'fresh' resource r; it is sufficient to consider only one such resource, as shall be seen in Lemma 5.3.8. The resource r is known not to occur in the domain of Γ and hence does not occur in the term t thanks to the following lemma, proved straightforwardly by induction on the judgement.

Lemma 4.2.1. *If $\Gamma \vdash \{\varphi\}t\{\psi\}$ then $\mathrm{res}(t) \subseteq \mathrm{dom}(\Gamma)$.*

4.3 Examples

For the reader unfamiliar with concurrent separation logic, we now give some more substantial examples to show the kind of process that can be verified. The most important example is the second one, which shows how the logic allows the transfer of ownership between processes.

Message passing example

First, we shall implement a simple form of channel for synchronous message passing. The implementation will make use of *conditional critical regions*, having the syntax

$$\texttt{with}\ r\ \texttt{when}\ b\ \texttt{do}\ t\ \texttt{od}.$$

We shall see this as an abbreviation for the following process, which repeatedly acquires the resource r until the boolean b holds:

```
flag := true;              %% local variable
while flag = true do       %% repeat until flag set
  with r do                %% acquire resource
    if b then              %% set repeat flag to false and run body
      flag := false;
      t
    else ε fi              %% try again
  od
od
```

Here, we have written $\texttt{if}\ b\ \texttt{then}\ t\ \texttt{else}\ t'\ \texttt{fi}$ to represent the guarded sum $b.t + \neg b.t'$. The location denoted *flag* should be local to the process being considered. The best way to implement this would be to define a local variable if we had variables in the language. Since we do not, we regard the conditional critical region as being parameterized by the choice of this location

$$\texttt{with}\ r\ \texttt{when}\ b\ \texttt{do}\ t\ \texttt{od}_{\mathit{flag}}.$$

In [O'H07], conditional critical regions are used in place of the critical regions used here. The normal interpretation of conditional critical regions is slightly different from that above in that the process waits until both the resource is available and the boolean holds before entering the conditional critical region, whereas the process above enters the critical region and then checks the boolean — normally called a 'busy-wait'. For the purpose of reasoning about partial correctness, it is entirely sufficient to regard the two implementations as being the same.

Moving on to the implementation of the synchronous channel, a channel will be implemented using a collection of locations recording its status. Access to these locations will be protected by conditional critical regions depending on the resource c. The locations protected by the resource are *free* to indicate whether or not the channel is in use, *val* to record the value being transmitted, and *received* to indicate that the value has been received. The invariant to describe the channel is:

$$\mathit{Chan} = \left(\begin{array}{cccccc} & \mathit{free} \mapsto \texttt{true} & \star & \mathit{received} \mapsto \texttt{false} & \star & \mathit{val} \mapsto \_ \\ \vee & \mathit{free} \mapsto \texttt{false} & \star & \mathit{received} \mapsto \_ & \star & \mathit{val} \mapsto \_ \end{array} \right)$$

The first process that we shall define shall be the sender, which outputs a value v on the channel. The channel must be free before the process can send the message. Upon updating the channel, the process waits until the status of the channel is updated by the receiver before it resumes. We annotate the code with the heap logic assertions that can be derived to hold of the heap owned by the process at each point.

$\{\mathit{flag1} \mapsto \_\}$
$\texttt{with}\ c\ \texttt{when}\ \mathit{free} = \texttt{true}\ \texttt{do}$
  $\{\mathit{flag1} \mapsto \_ \star \mathit{free} \mapsto \texttt{true} \star \mathit{received} \mapsto \texttt{false} \star \mathit{val} \mapsto \_\}$
  %% write value and record that the channel is in use
  $\mathit{val} := v;$
  $\mathit{free} := \texttt{false}$
  $\{\mathit{flag1} \mapsto \_ \star \mathit{free} \mapsto \texttt{false} \star \mathit{received} \mapsto \texttt{false} \star \mathit{val} \mapsto \_\}$
  $\{\mathit{flag1} \mapsto \_ \star \mathit{Chan}\}$
$\texttt{od}_{\mathit{flag1}};$
$\{\mathit{flag1} \mapsto \_\}$
%% wait for value to be received
$\texttt{with}\ c\ \texttt{when}\ \mathit{received} = \texttt{true}\ \texttt{do}$
  $\{\mathit{flag1} \mapsto \_ \star \mathit{free} \mapsto \texttt{false} \star \mathit{received} \mapsto \texttt{true} \star \mathit{val} \mapsto \_\}$
  %% restore channel to unused state
  $\mathit{received} := \texttt{false};$
  $\mathit{free} := \texttt{true}$
  $\{\mathit{flag1} \mapsto \_ \star \mathit{free} \mapsto \texttt{true} \star \mathit{received} \mapsto \texttt{false} \star \mathit{val} \mapsto \_\}$
  $\{\mathit{flag1} \mapsto \_ \star \mathit{Chan}\}$
$\texttt{od}_{\mathit{flag1}}$
$\{\mathit{flag1} \mapsto \_\}$

The process that receives on the channel is somewhat simpler. It waits for the location called *free* to be set to false, indicating that a process is ready to send to the channel, and

then reads the value sent to the location *input*.

$\{input \mapsto \_ \star flag2 \mapsto \_\}$
`%% wait until sender sets free to false`
`with` c `when` $free = \texttt{false}$ `do`
  $\{input \mapsto \_ \star flag2 \mapsto \_ \star free \mapsto \texttt{false} \star received \mapsto \texttt{false} \star val \mapsto \_\}$
  `%% get channel input and set received flag`
  $input := val;$
  $received := \texttt{true}$
  $\{input \mapsto \_ \star flag2 \mapsto \_ \star free \mapsto \texttt{false} \star received \mapsto \texttt{true} \star val \mapsto \_\}$
  $\{input \mapsto \_ \star flag2 \mapsto \_ \star Chan\}$
$\texttt{od}_{flag2}$
$\{input \mapsto \_ \star flag2 \mapsto \_\}$

The two processes can be placed in parallel to yield the judgement

$$c{:}Chan \vdash \begin{array}{l}\{flag1 \mapsto \_ \star flag2 \mapsto \_ \star input \mapsto \_\} \\ \quad Sender \parallel Receiver \\ \{flag1 \mapsto \_ \star flag2 \mapsto \_ \star input \mapsto \_\}\end{array}$$

This example establishes that whenever the resource protecting the channel is available (*i.e.* whenever neither the sender nor the receiver are inside a critical region), the values held in locations for the channel are consistent. It also demonstrates that the processes only access these locations when inside their critical regions.

What if we wanted to establish other properties of the channel? For example, how can we show that the location *input* always ends up holding value v? Since this is a global judgement, that requires understanding of both the receiver and the sender, it has to be reflected in the invariant for the channel. The new invariant would be

$$Chan_v = \left(\begin{array}{llllll} & free \mapsto \texttt{true} & \star & received \mapsto \texttt{false} & \star & val \mapsto \_ \\ \vee & free \mapsto \texttt{false} & \star & received \mapsto \_ & \star & val \mapsto v \end{array} \right)$$

With this invariant, the proofs above can be modified straightforwardly to give the judgement

$$c{:}Chan_v \vdash \begin{array}{l}\{flag1 \mapsto \_ \star flag2 \mapsto \_ \star input \mapsto \_\} \\ \quad Sender \parallel Receiver \\ \{flag1 \mapsto \_ \star flag2 \mapsto \_ \star input \mapsto v\}\end{array} .$$

Of course, in itself, this is not an awfully useful kind of judgement. The principle will be extended later to see how, in fact, *ownership* can be transferred over channels.

Transfer of ownership

The notion of ownership is subtle since the collection of locations that a process owns may change as the process evolves. As seen in the rule (L-Alloc), the intuitive reading is that, after an allocation event has taken place, the process owns the newly-current location. Similarly, deallocation of a location leads to loss of ownership. For example, it is possible to make the judgement

$$\Gamma \vdash \{\ell \mapsto \_\}\texttt{alloc}(\ell)\{\exists x_{\mathsf{loc}}.\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_\}.$$

If the new location were ℓ' which initially held value v, this would mean that in the the (fragment of the) resulting heap $\{\ell \mapsto \ell', \ell' \mapsto v\}$, the locations ℓ and ℓ' would be owned

by the process. Consequently, the action $[\ell] := 0$ which assigns 0 to the location pointed to by ℓ resulting in the heap $\{\ell \mapsto \ell', \ell' \mapsto 0\}$ allows the judgement

$$\Gamma \vdash \{\exists x_{\mathsf{loc}}.\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_\}[\ell] := 0\{\exists x_{\mathsf{loc}}.\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto 0\}$$

by (L-Act) since both locations would be owned by the process. The rule (L-Seq) can now be applied to obtain

$$\Gamma \vdash \{\ell \mapsto \_\}\mathtt{alloc}(\ell); [\ell] := 0\{\exists x_{\mathsf{loc}}.\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto 0\},$$

indicating that the process has ownership of the location ℓ', seen in the ability to write to ℓ', once it has been allocated.

A rather more subtle point is that the rules allow ownership of locations to be transferred through invariants. Consider the invariant χ defined as $\ell' \mapsto 0 \vee (\ell' \mapsto 1 \star \ell \mapsto 0)$. If the resource is available, the invariant is satisfied: it either protects the location ℓ', which has value 0, or it protects location ℓ', which has value 1, as well as location ℓ. A process can acquire ownership of ℓ across a critical region by changing the value of ℓ' from 1 to 0 and may leave ownership of ℓ inside the invariant by changing the value of ℓ' from 0 to 1. This gives rise to separation logic being able to support *daring* concurrency in the sense of [O'H07], where concurrent processes can access some common location whilst outside critical regions but without causing a race to occur.

Assume, for example, that the process owns location ℓ. The only way in which the invariant χ can be satisfied disjointly from the locations that the process owns is for ℓ' to hold value 0. That is, we have

$$\ell \mapsto 0 \star (\ell' \mapsto 0 \vee (\ell' \mapsto 1 \star \ell \mapsto 0)) \implies \ell \mapsto 0 \star \ell' \mapsto 0$$

which is implicitly used in the instance of the rule (L-Consequence) below. Consequently, as the process enters a critical region protected by r, it gains ownership of location ℓ'. If the process sets the value of ℓ' to 1, when the process leaves the critical region it must restore the invariant to the resource, and so relinquish ownership of both ℓ' and ℓ. This is seen in the derivation of the following judgement, in which we take $\Gamma = r{:}\chi$.

$$\dfrac{\dfrac{\dfrac{}{\Gamma \vdash \{\ell \mapsto 0 \star \ell' \mapsto 0\}\ell' := 1\{\ell \mapsto 0 \star \ell' \mapsto 1\}}\,(\text{L-Act})}{\Gamma \vdash \{\ell \mapsto 0 \star \chi\}\ell' := 1\{\mathsf{empty} \star \chi\}}\,(\text{L-Consequence})}{\Gamma \vdash \{\ell \mapsto 0\}\mathtt{with}\ r\ \mathtt{do}\ \ell' := 1\ \mathtt{od}\{\mathsf{empty}\}}\,(\text{L-CR})$$

It is also possible to acquire ownership of locations through an invariant. Let the action `diverge` have the same semantics as that of the boolean guard `false`, which is an action that can never occur *i.e.* the process is stuck. We have the following derivation:

$$\dfrac{\dfrac{\begin{array}{rrrr}\Gamma \vdash & \{\chi\} & \ell' = 0 & \{\ell' \mapsto 0\} \\ \Gamma \vdash & \{\ell' \mapsto 0\}\ \mathtt{diverge} & & \{\ell' \mapsto 0 \star \ell \mapsto 0\} \\ \Gamma \vdash & \{\chi\} & \ell' = 1 & \{\ell' \mapsto 1 \star \ell \mapsto 0\} \\ \Gamma \vdash & \{\ell' \mapsto 1 \star \ell \mapsto 0\} & \ell' := 0 & \{\ell' \mapsto 0 \star \ell \mapsto 0\}\end{array}}{\Gamma \vdash \{\chi\}(\ell' = 0.\mathtt{diverge}) + (\ell' = 1.\ell' := 0)\{\ell \mapsto 0 \star \ell' \mapsto 0\}}\,(\text{L-Sum})}{\Gamma \vdash \{\mathsf{empty} \star \chi\}(\ell' = 0.\mathtt{diverge}) + (\ell' = 1.\ell' := 0)\{\ell \mapsto 0 \star \chi\}}\,(\text{L-Conseq.})}{\Gamma \vdash \{\mathsf{empty}\}\mathtt{with}\ r\ \mathtt{do}\ (\ell' = 0.\mathtt{diverge}) + (\ell' = 1.\ell' := 0)\ \mathtt{od}\{\ell \mapsto 0\}}\,(\text{L-CR})$$

The undischarged hypotheses at the top of the derivation are all proved by the rule (L-Act). Let t_0 denote the process $(\ell' = 0.\mathtt{diverge}) + (\ell' = 1.\ell' := 0)$. Observe that the process `with` r `do` t_0 `od` is considered to own no part of the initial heap. As the

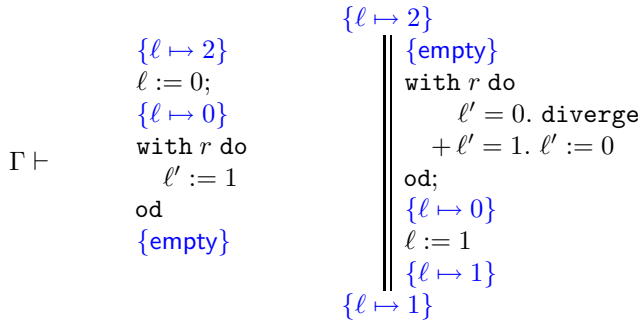


Figure 4.5: Program transferring ownership

process enters the critical region, it is considered to take ownership of the part of the heap satisfying the invariant for r, namely $\chi = \ell' \mapsto 0 \vee (\ell' \mapsto 1 \star \ell \mapsto 0)$. There are two ways in which χ might be satisfied:

1. It may be that the process gains ownership of the location ℓ' which holds value 0. In this case, only the guard $\ell' = 0$ of t_0 can pass, so the process must evolve to `diverge` and therefore never terminates.

2. The process might have taken control of the locations ℓ, holding value 0, and ℓ', holding value 1. Inside the critical region, only the guard $\ell' = 1$ can pass so the process t_0 can be seen to change the value of ℓ' from 1 to 0. The only way that the invariant χ can then be satisfied is by the location ℓ' holding 0, so ownership of ℓ' is lost as the process leaves the critical region. Importantly, the process retains ownership of location ℓ.

Either way, whenever the process terminates, it owns location ℓ since it can only terminate in case (2).

Using the derivations given above, in Figure 4.5 we give an example of ownership of ℓ, as exhibited by the right to write to ℓ, being transferred (we have annotated internal assertions arising from the proofs above inside the program). We also see that in any terminating run of this process, it must be the case that the process on the left terminates strictly before the process on the right begins.

Ownership transmission

Before we move on to the formal semantics of the logic, we shall go through one more example of the logic at work, combining the two examples above to show how ownership may be transferred across synchronous channels.

This time, rather than holding a value, processes pass a reference to a location along a channel along with ownership of the location pointed to. This example can be generalized to allow the introduction of channels that allow transfer of any kind of structure, such as a list, rather than just a single location.

The invariant to represent a channel capable of transferring ownership of a single location is as follows: $Chan' =$

$$\left(\begin{array}{llllll} & \mathit{free} \mapsto \texttt{true} & \star & \mathit{received} \mapsto \texttt{false} & \star & \mathit{val} \mapsto \_ \\ \vee & \mathit{free} \mapsto \texttt{false} & \star & \mathit{received} \mapsto \texttt{false} & \star & \exists x_{\mathsf{loc}}.(\mathit{val} \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_) \\ \vee & \mathit{free} \mapsto \texttt{false} & \star & \mathit{received} \mapsto \texttt{true} & \star & \mathit{val} \mapsto \_ \end{array}\right)$$

$\{flag1 \mapsto \_ \star p \mapsto \_\}$
$\texttt{alloc}(p)$;
$\{flag1 \mapsto \_ \star \exists x_{\mathsf{loc}}.(p \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_)\}$
$\texttt{with}\ c\ \texttt{when}\ free = \texttt{true}\ \texttt{do}$
  $\{flag1 \mapsto \_ \star \exists x_{\mathsf{loc}}.(p \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_) \star$
    $free \mapsto \texttt{true} \star received \mapsto \texttt{false} \star val \mapsto \_\}$
  %% write value and record that the channel is in use
  $val := p$;
  $free := \texttt{false}$
  $\{flag1 \mapsto \_ \star \exists x_{\mathsf{loc}}.(p \mapsto x_{\mathsf{loc}} \star val \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_) \star$
    $free \mapsto \texttt{false} \star received \mapsto \texttt{false}\}$
  $\{flag1 \mapsto \_ \star p \mapsto \_ \star Chan\}$
$\texttt{od}_{flag1}$;
$\{flag1 \mapsto \_ \star p \mapsto \_\}$
%% wait for value to be received
$\texttt{with}\ c\ \texttt{when}\ received = \texttt{true}\ \texttt{do}$
  $\{flag1 \mapsto \_ \star p \mapsto \_ \star free \mapsto \texttt{false} \star received \mapsto \texttt{true} \star val \mapsto \_\}$
  %% restore channel to unused state
  $received := \texttt{false}$;
  $free := \texttt{true}$
  $\{flag1 \mapsto \_ \star p \mapsto \_ \star free \mapsto \texttt{true} \star received \mapsto \texttt{false} \star val \mapsto \_\}$
  $\{flag1 \mapsto \_ \star p \mapsto \_ \star Chan\}$
$\texttt{od}_{flag1}$
$\{flag1 \mapsto \_ \star p \mapsto \_\}$

Figure 4.6: Process sending ownership over a channel

The first disjunct represents no value being sent along the channel. The second disjunct represents a sender having assigned a pointer to the location represented by x_{loc} to *val* but the message not yet having been received. The final disjunct represents the case where the receiver has received the message, allowing the sender to resume. Note how ownership of the location pointed to by *val* will be transferred to the invariant in proceeding from the first disjunct to the second and how it will be transferred from the invariant in proceeding from the second disjunct to the third.

The proofs of the earlier *Sender* and *Receiver* processes can be modified to give the following judgements with respect to the environment $c{:}Chan'$. The sender process is modified as shown in Figure 4.6, first allocating a location and then sending it along the channel. Recall that *flag1* is a location used to implement the conditional critical region.

The proof for the receiver process, presented in Figure 4.7 is changed to account for acquiring ownership of the location passed along the channel prior to its deallocation. Together, the two processes demonstrate how a location can be allocated by the sender, its ownership transferred, and then it be deallocated by the receiver.

$\{input \mapsto \_ \star flag2 \mapsto \_\}$
%% wait until sender sets free to false
with c when $free = \texttt{false} \ \wedge \ \texttt{received} = \texttt{false}$ do
  $\{input \mapsto \_ \star flag2 \mapsto \_$
    $\star free \mapsto \texttt{false} \star received \mapsto \texttt{false} \star \exists x_{\mathsf{loc}}.(val \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_)\}$
  %% get channel input and set received flag
  $input := val$;
  $received := \texttt{true}$
  $\{\exists x_{\mathsf{loc}}.(input \mapsto x_{\mathsf{loc}} \star val \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_) \star flag2 \mapsto \_$
    $\star free \mapsto \texttt{false} \star received \mapsto \texttt{true}\}$
  $\{\exists x_{\mathsf{loc}}.(input \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_) \star flag2 \mapsto \_ \star Chan\}$
$\mathtt{od}_{flag2}$;
$\{\exists x_{\mathsf{loc}}.(input \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_) \star flag2 \mapsto \_\}$
$\mathtt{dealloc}(input)$
$\{input \mapsto \_ \star flag2 \mapsto \_\}$

Figure 4.7: Process receiving ownership over a channel

Chapter 5

Net semantics of separation logic

We now progress to give a formal interpretation of the rules presented in the previous chapter. The key results are:

- Theorem 5.2 on page 94, where the rules of concurrent separation logic are proved sound with respect to the model to be developed,
- Corollary 5.3 on page 94, where the soundness proof is connected to the standard net semantics,
- Proposition 5.4 on page 95, where it is shown that proved processes avoid faults such as accessing non-allocated memory locations, and
- Theorem 5.5 on page 97, where the separation of proved processes is captured.

The key idea is that the judgement $\Gamma \vdash \{\varphi\}t\{\psi\}$ is robust against the operation of other 'external' processes (which have themselves been subject to a judgement in the logic) on the state, so that the rule for parallel composition is valid. From the account presented earlier, external processes may act on the heap providing they do not access the locations that we see as 'owned' by the process t, and they may act to acquire and release resources providing they respect the invariants in Γ. External processes may also make non-current resources current through the instantiation of a local resource variable. The semantics of judgements (*i.e.* the model with respect to which validity shall be considered) must therefore keep a record of how each current location in the heap and each current resource is owned: whether the process being considered is allowed to access the location, whether the location forms part of an invariant protected by a resource, or whether external processes might act on that location, along with a similar record for resources. The semantics of judgements will include *interference events* to represent such forms of action by external processes. We shall see shortly how the events of the process correspond to interference events and shall specify precisely how they affect the distribution of ownership, but we focus first on the interference events themselves.

5.1 Interference nets

Capturing the requirements above, we construct an *interference net* with respect to the environment Γ to represent the execution of suitable external processes proved against Γ. This involves creating mutually exclusive *ownership conditions* (conditions in the Petri net sense) which we denote $\omega_{\mathsf{proc}}(\ell)$, $\omega_{\mathsf{inv}}(\ell)$ and $\omega_{\mathsf{oth}}(\ell)$ for each location ℓ. The intuition is that $\omega_{\mathsf{proc}}(\ell)$ is marked if ℓ is owned by the process, $\omega_{\mathsf{inv}}(\ell)$ is marked if ℓ is used to

satisfy the invariant for an available resource, and $\omega_{\mathsf{oth}}(\ell)$ is marked if ℓ is current but owned by another process.

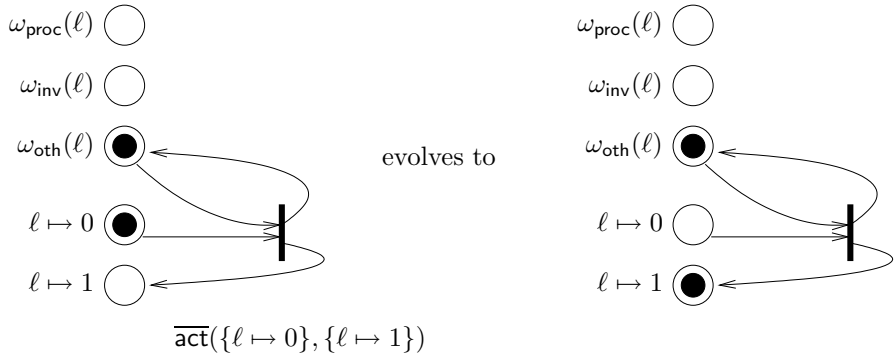
To give an example, suppose that we have the judgement

$$\Gamma \vdash \{k \mapsto 1\}k := 0\{k \mapsto 0\}.$$

The proof can be composed with the judgement $\Gamma \vdash \{\ell \mapsto 0\}\ell := 0\{\ell \mapsto 1\}$ to obtain

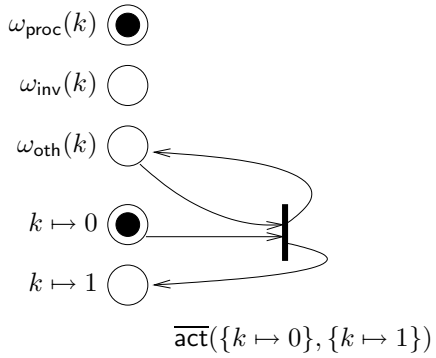
$$\Gamma \vdash \{k \mapsto 1 \star \ell \mapsto 0\}k := 0 \parallel \ell := 1\{k \mapsto 0 \star \ell \mapsto 1\}.$$

The first proof, that the assignment $k := 0$ changes the value at k from 1 to 0, must take into account the possibility that the values held at other locations may change. In particular, it must take into account the possibility that the value at ℓ (not to equal k) changes from 0 to 1. (It is, of course, obvious that the assignment to ℓ does not affect the judgement about the assignment to k; we shall come to a more interesting example shortly.) We therefore reason about the net $\mathcal{N}[\![k := 0]\!]$ in the presence of the following *interference event* which changes the value held at ℓ from 0 to 1:



Notably, the above event requires that the location ℓ is owned by an external process, *i.e.* that the condition $\omega_{\mathsf{oth}}(\ell)$ is marked.

Since we do not know *a priori* with which other judgements $\Gamma \vdash \{k \mapsto 1\}k := 0\{k \mapsto 0\}$ may be composed, there are interference events present in the net for all the forms of interference permissible according to the notion of ownership. For instance, the interference event where another process changes the value of k from 0 to 1



is present in the net. However, the judgement asserts that k is owned by the process, so this interference event (and indeed any other interference event that affects k) will not be able to occur because the condition $\omega_{\mathsf{proc}}(k)$ will be marked, not $\omega_{\mathsf{oth}}(k)$.

As mentioned above, we introduce interference events to mimic the action of external processes on resources. The notion of ownership is therefore extended in this setting to

resources, for example so that an external process cannot be allowed to release a resource held by the current process. It is important to make a distinction between resources in the domain of the environment Γ, called *open* resources, and those that are not, called *closed* resources: *open resources* have invariants associated with them, so the ownership of the heap is affected by events that acquire or release them as presented earlier in the last chapter; this is not the case for closed resources. *Closed resources* are those resources made current to instantiate a local resource variable. They may either be used by the process being considered if it declared the resource or be used by some external process if some external process declared the resource. We shall introduce conditions $\omega_{\mathsf{proc}}(r)$, $\omega_{\mathsf{inv}}(r)$ and $\omega_{\mathsf{oth}}(r)$ for each resource r. The condition $\omega_{\mathsf{proc}}(r)$ will be marked if either the resource is closed and was made current by the process or if the resource is open and is held by the process. The condition $\omega_{\mathsf{inv}}(r)$ will be marked if r is open and available. The condition $\omega_{\mathsf{oth}}(r)$ will be marked if either the resource is closed and was made current by an external process or if the resource is both open and the external process holds it.

Interference and critical regions

We now consider the effect of interference on a process that modifies part of the heap protected by an open resource. Let the invariant for the resource r in the environment Γ be $k \mapsto 0 \vee k \mapsto 1$. We shall explain why the Hoare triple

$$\{\ell \mapsto 0\}\texttt{with}\ r\ \texttt{do}\ k := 1\ \texttt{od}; \texttt{with}\ r\ \texttt{do}\ l := k\ \texttt{od}\{\ell \mapsto 1\}$$

is *not* valid with respect to Γ.

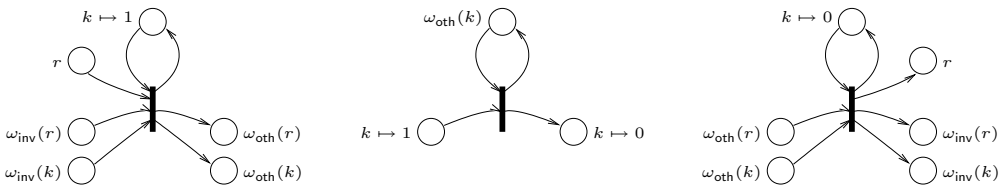
Informally, the reason for this is that some external process might acquire the resource r between the two critical regions and change the value of k to 0. In more detail, suppose that the process starts in a state in which the resource r is available. The location k will be protected by the invariant, so the condition $\omega_{\mathsf{inv}}(k)$ will be marked. Following the account above, since the resource r is open and available, the condition $\omega_{\mathsf{inv}}(r)$ will also be marked. As the process enters the critical region, it gains ownership of r and k resulting in $\omega_{\mathsf{proc}}(r)$ and $\omega_{\mathsf{proc}}(k)$ becoming marked instead of $\omega_{\mathsf{inv}}(r)$ and $\omega_{\mathsf{inv}}(k)$. The process changes the value at k to 1 and then leaves the critical region. As it does so, it records that it no longer owns the resource r by switching the marking of ownership conditions back to $\omega_{\mathsf{inv}}(r)$ and $\omega_{\mathsf{inv}}(k)$. Hence, between the two critical regions, the marking is:

$$\{\ell \mapsto 0, k \mapsto 1, r, \omega_{\mathsf{proc}}(\ell), \omega_{\mathsf{inv}}(k), \omega_{\mathsf{inv}}(r)\}.$$

The account above explains that external processes might enter and leave critical regions. For example, we might later wish to compose this judgement with the (derivable) judgement

$$\Gamma \vdash \{\mathsf{empty}\}\texttt{with}\ r\ \texttt{do}\ k := 0\ \texttt{od}\{\mathsf{empty}\}.$$

To mimic the behaviour of this process, the interference net contains the following interference events:



(The interference net contains events to represent *all* forms of possible interference; these shall be specified in the following section.) The first of these events represents that the second process above, of which we were not specifically aware when reasoning about the first process above, might acquire the resource r. In doing so, this other process takes ownership of the location k since the invariant is satisfied through k. The process also takes ownership of the resource r. The second event represents that the other process can change the value of k providing it owns the location k. The third event represents that the other process can release the resource r providing it restores the invariant and returns ownership of the locations satisfying the invariant.

We can see that these three events in succession have concession from the intermediate marking given above. Their effect is to change the value of k from 1 to 0. It is clear that the second critical region of the process that we were originally considering

$$\texttt{with}\ r\ \texttt{do}\ \ell := k\ \texttt{od}$$

now results in the value 0 being copied to ℓ, invalidating the original Hoare triple.

5.2 Synchronization and the ownership net

We now formally specify the interference net and show how this is used to form an *ownership net*. The ownership net models how processes, both the process being considered and external processes, affect the distribution of ownership as they run.

Both the interference net and the ownership net augment the original net semantics with ownership conditions. The set of ownership conditions is denoted $\mathbf{W}$:

$$\begin{aligned}\mathbf{W} \triangleq\ & \{\omega_{\mathsf{proc}}(\ell), \omega_{\mathsf{inv}}(\ell), \omega_{\mathsf{oth}}(\ell) \mid \ell \in \mathrm{Loc}\} \\ & \cup \{\omega_{\mathsf{proc}}(r), \omega_{\mathsf{inv}}(r), \omega_{\mathsf{oth}}(r) \mid r \in \mathrm{Res}\}.\end{aligned}$$

We use W to range over markings of ownership conditions and introduce the notations ${}^{\mathbf{w}}e$ and $e^{\mathbf{w}}$ for the sets of pre-ownership conditions of e and post-ownership conditions of e, respectively. For a set of locations L, we define the notation

$$\omega_{\mathsf{proc}}(L) \triangleq \{\omega_{\mathsf{proc}}(\ell) \mid \ell \in L\},$$

and define $\omega_{\mathsf{inv}}(L)$ and $\omega_{\mathsf{oth}}(L)$ similarly. Only certain markings of ownership conditions are consistent with a state σ:

Definition 5.2.1 (Consistent marking)**.** *The marking of state and ownership conditions* (σ, W) *is* consistent *if:*

1. σ *is a consistent state in* $\mathcal{N}[\![t]\!]$*,*
2. *for each* $z \in \mathrm{Loc} \cup \mathrm{Res}$*, at most one of* $\{\omega_{\mathsf{proc}}(z), \omega_{\mathsf{inv}}(z), \omega_{\mathsf{oth}}(z)\}$ *is marked,*
3. *for each* $z \in \mathrm{Loc} \cup \mathrm{Res}$*, the ownership condition* $\mathsf{curr}(z)$ *is in* σ *iff precisely one of* $\{\omega_{\mathsf{proc}}(z), \omega_{\mathsf{inv}}(z), \omega_{\mathsf{oth}}(z)\}$ *is in* W*,*
4. *if* $r \in \mathrm{dom}(\Gamma)$ *and* $r \in R$ *then* $\omega_{\mathsf{inv}}(r) \in W$*,*
5. *if* $r \in \mathrm{dom}(\Gamma)$ *and* $r \notin R$ *then either* $\omega_{\mathsf{proc}}(r) \in W$ *or* $\omega_{\mathsf{oth}}(r) \in W$*, and*
6. *if* $\mathsf{curr}(r) \in \sigma$ *and* $r \notin \mathrm{dom}(\Gamma)$ *then either* $\omega_{\mathsf{proc}}(r) \in W$ *or* $\omega_{\mathsf{oth}}(r) \in W$*.*

A marking of the control, state and ownership conditions (C, σ, W) *is consistent if* (σ, W) *is consistent.*

It shall be shown in Proposition 5.1 on page 83 that consistency of markings is preserved. Requirements (2) and (3) assert that W is essentially a function from the set of current locations and resources to ownership. Requirement (4) states that any available open resource is owned as an invariant: it can be accessed either by the process being considered or by an external process, and there is an invariant associated with r. Requirement (5) states that any unavailable open resource is either held by the process or by an external process. Requirement (6) asserts that any closed resource is owned either by the current process or by an external process.

Table 5.1 defines a number of notations for events corresponding to the permitted interference described. To summarize, there will be interference events to represent the following kinds of action by external processes:

- $\overline{\mathsf{act}}(D_1, D_2)$: Arbitrary action on the heap (excluding allocation or deallocation) owned by external processes.
- $\overline{\mathsf{alloc}}(\ell, v, \ell', v')$: Allocation of a new location ℓ' by an external process, storing the result in the location ℓ. The location ℓ must be owned by an external process. Ownership of the new location ℓ' is taken by the external process.
- $\overline{\mathsf{dealloc}}(\ell, v, \ell', v')$: Disposal of the location ℓ' pointed to by ℓ. Both locations are initially owned by external processes, so $\omega_{\mathsf{oth}}(\ell)$ and $\omega_{\mathsf{oth}}(\ell')$ are preconditions to the event.
- $\overline{\mathsf{decl}}(r)$: Declaration of a resource r. The condition $\mathsf{curr}(r)$ is marked by the event, so the resource was not initially current. Ownership of r is taken by the external process, so $\omega_{\mathsf{oth}}(r)$ is in the postconditions of the event.
- $\overline{\mathsf{end}}(r)$: End of scope of a resource r, only permissible if the resource was initially declared by an external process and therefore $\omega_{\mathsf{oth}}(r)$ is marked.
- $\overline{\mathsf{acq}}(r)$: For a closed resource r, the external process may acquire the resource if it is not local to the process being considered and therefore $\omega_{\mathsf{oth}}(r)$ is marked.
- $\overline{\mathsf{rel}}(r)$: For a closed resource r, the external process may release the resource if it is not local to the process being considered and therefore $\omega_{\mathsf{oth}}(r)$ is marked.
- $\overline{\mathsf{acq}}(r, D_0)$: For an open resource r with an invariant χ in Γ, if $D_0 \models \chi$ and D_0 is part of the current heap then ownership of the locations in the domain of D_0 is changed from being protected by the resource to being owned by the external process, *i.e.* unmarking $\omega_{\mathsf{inv}}(\ell)$ and marking $\omega_{\mathsf{oth}}(\ell)$ for each location $\ell \in \mathrm{dom}(D_0)$. The ownership of r also changes from $\omega_{\mathsf{inv}}(r)$ being marked to $\omega_{\mathsf{oth}}(r)$ being marked.
- $\overline{\mathsf{rel}}(r, D_0)$: The corresponding release action.

Definition 5.2.2 (Interference net)**.** *The interference net for* Γ *has conditions* $\mathbf{S}$*, the state conditions, and* $\mathbf{W}$*, the ownership conditions. It has the following events called* interference events*:*

- $\overline{\mathsf{act}}(D_1, D_2)$ *for all* D_1 *and* D_2 *forming partial functions with the same domain*
- $\overline{\mathsf{alloc}}(\ell, v, \ell', v')$ *and* $\overline{\mathsf{dealloc}}(\ell, \ell', v')$ *for all locations* ℓ *and* ℓ' *and values* v *and* v'

u	*Preconditions* ${}^{\mathbf{s}}u$	${}^{\mathbf{w}}u$	*Postconditions* $u^{\mathbf{s}}$	$u^{\mathbf{w}}$
$\overline{\mathsf{act}}(D_1, D_2)$	D_1	$\omega_{\mathsf{oth}}(\mathrm{dom}(D_1))$	D_2	$\omega_{\mathsf{oth}}(\mathrm{dom}(D_2))$
$\overline{\mathsf{alloc}}(\ell, v, \ell', v')$	$\{\ell \mapsto v\}$	$\{\omega_{\mathsf{oth}}(\ell)\}$	$\{\mathsf{curr}(\ell')\} \cup$ $\{\ell \mapsto \ell', \ell' \mapsto v'\}$	$\{\omega_{\mathsf{oth}}(\ell), \omega_{\mathsf{oth}}(\ell')\}$
$\overline{\mathsf{dealloc}}(\ell, \ell', v')$	$\{\mathsf{curr}(\ell')\} \cup$ $\{\ell \mapsto \ell', \ell' \mapsto v'\}$	$\{\omega_{\mathsf{oth}}(\ell), \omega_{\mathsf{oth}}(\ell')\}$	$\{\ell \mapsto \ell'\}$	$\{\omega_{\mathsf{oth}}(\ell)\}$
$\overline{\mathsf{decl}}(r)$	$\{\}$	$\{\}$	$\{\mathsf{curr}(r), r\}$	$\{\omega_{\mathsf{oth}}(r)\}$
$\overline{\mathsf{end}}(r)$	$\{\mathsf{curr}(r), r\}$	$\{\omega_{\mathsf{oth}}(r)\}$	$\{\}$	$\{\}$
$\overline{\mathsf{acq}}(r)$	$\{r\}$	$\{\omega_{\mathsf{oth}}(r)\}$	$\{\}$	$\{\omega_{\mathsf{oth}}(r)\}$
$\overline{\mathsf{rel}}(r)$	$\{\}$	$\{\omega_{\mathsf{oth}}(r)\}$	$\{r\}$	$\{\omega_{\mathsf{oth}}(r)\}$
$\overline{\mathsf{acq}}(r, D_0)$	$D_0 \cup \{r\}$	$\omega_{\mathsf{inv}}(\mathrm{dom}(D_0)) \cup$ $\{\omega_{\mathsf{inv}}(r)\}$	D_0	$\omega_{\mathsf{oth}}(\mathrm{dom}(D_0)) \cup$ $\{\omega_{\mathsf{oth}}(r)\}$
$\overline{\mathsf{rel}}(r, D_0)$	D_0	$\{\omega_{\mathsf{oth}}(r)\} \cup$ $\omega_{\mathsf{oth}}(\mathrm{dom}(D_0))$	$D_0 \cup \{r\}$	$\{\omega_{\mathsf{inv}}(r)\} \cup$ $\omega_{\mathsf{inv}}(\mathrm{dom}(D_0))$

Table 5.1: Notation for interference events

- $\overline{\mathsf{decl}}(r)$ *and* $\overline{\mathsf{end}}(r)$ *for all closed resources* r
- $\overline{\mathsf{acq}}(r)$ *and* $\overline{\mathsf{rel}}(r)$ *for all closed resources* r
- $\overline{\mathsf{acq}}(r, D_0)$ *and* $\overline{\mathsf{rel}}(r, D_0)$ *for all* $r \in \mathrm{dom}(\Gamma)$ *and* D_0 *such that* $D_0 \models \chi$*, for* χ *the unique formula such that* $r{:}\chi \in \Gamma$

We use the symbol u *to range over interference events.*

The interference events illustrate how the ownership of locations is *dynamic* and how this constrains the possible forms of interference. The rule for parallel composition requires that the behaviour of the process being reasoned about itself conforms to these constraints, allowing its action to be seen as interference when reasoning about the other process. This requirement may be captured by *synchronizing* the events of the process with those from the interference net. Sycnhronization of events is a general operation, introduced in [Win82]. In the case of the framework for concurrent separation logic here, synchronization is restricted to complementary events:

- The process event $\mathsf{act}_{(C,C')}(D, D')$ synchronizes with $\overline{\mathsf{act}}(D, D')$
- The process event $\mathsf{alloc}_{(C,C')}(\ell, v, \ell', v')$ synchronizes with $\overline{\mathsf{alloc}}(\ell, v, \ell', v')$
- The process event $\mathsf{dealloc}_{(C,C')}(\ell, \ell', v')$ synchronizes with $\overline{\mathsf{dealloc}}(\ell, \ell', v')$
- The process event $\mathsf{decl}_{(C,C')}(r)$ synchronizes with $\overline{\mathsf{decl}}(r)$
- The process event $\mathsf{end}_{(C,C')}(r)$ synchronizes with $\overline{\mathsf{end}}(r)$
- The process event $\mathsf{acq}_{(C,C')}(r)$ synchronizes with $\overline{\mathsf{acq}}(r)$ for any closed resource r, *i.e.* for any $r \notin \mathrm{dom}(\Gamma)$

- The process event $\mathsf{rel}_{(C,C')}(r)$ synchronizes with $\overline{\mathsf{rel}}(r)$ for any closed resource r

- If r is an open resource with $r{:}\chi \in \Gamma$, the process event $\mathsf{acq}_{(C,C')}(r)$ synchronizes with every $\overline{\mathsf{acq}}(r, D_0)$ such that $D_0 \models \chi$. Similarly, $\mathsf{rel}_{(C,C')}(r)$ synchronizes with every $\overline{\mathsf{rel}}(r, D_0)$ such that $D_0 \models \chi$.

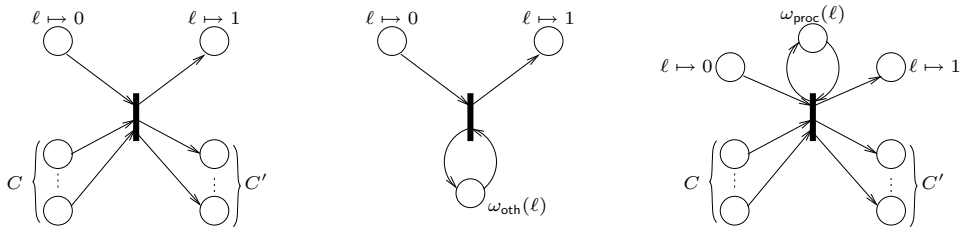
Suppose that two events synchronize, e from the process and u from the interference net. The event u is the event that would fire in the net for the other parallel process to simulate the event e; it is its dual. Let $e \cdot u$ be the event formed by taking the union of the pre- and postconditions of e and u, other than using $\omega_{\mathsf{proc}}(\ell)$ in place of $\omega_{\mathsf{oth}}(\ell)$ and similarly $\omega_{\mathsf{proc}}(r)$ in place of $\omega_{\mathsf{oth}}(r)$.

$$\begin{array}{rcl} {}^\bullet(e \cdot u) & \triangleq & \{b \mid b \in {}^\bullet e \cup {}^\bullet u \text{ and } \nexists z.b = \omega_{\mathsf{oth}}(z)\} \cup \{\omega_{\mathsf{proc}}(z) \mid \omega_{\mathsf{oth}}(z) \in {}^\bullet u\} \\ (e \cdot u)^\bullet & \triangleq & \{b \mid b \in e^\bullet \cup u^\bullet \text{ and } \nexists z.b = \omega_{\mathsf{oth}}(z)\} \cup \{\omega_{\mathsf{proc}}(z) \mid \omega_{\mathsf{oth}}(z) \in u^\bullet\} \end{array}$$

Example 5.2.1 (Synchronization of heap actions)**.** *Define the events e and u as*

$$e = \mathsf{act}_{(C,C')}(\{\ell \mapsto 0\}, \{\ell \mapsto 1\}) \qquad u = \overline{\mathsf{act}}(\{\ell \mapsto 0\}, \{\ell \mapsto 1\}).$$

The events e and u are drawn as follows along with the result of synchronizing them, $e \cdot u$:



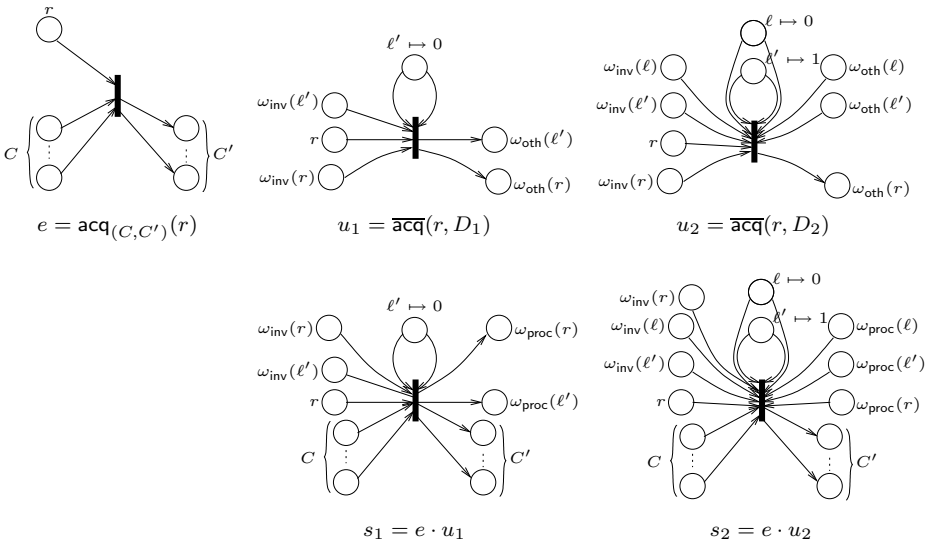
$e = \mathsf{act}_{(C,C')}(\{\ell \mapsto 0\}, \{\ell \mapsto 1\})$ $u = \overline{\mathsf{act}}(\{\ell \mapsto 0\}, \{\ell \mapsto 1\})$ $e \cdot u$

The event e is an event inside the process net, with pre-control conditions C and post-control conditions C', that changes the value of ℓ from 0 to 1. It synchronizes with only one event, u, which performs the corresponding interference action. For the event u to occur, the condition $\omega_{\mathsf{oth}}(\ell)$ must be marked i.e. the location ℓ must be seen as owned by an 'external' process. The event formed by synchronizing e and u is $e \cdot u$, which requires the location ℓ to be owned by the current process for it to occur.

Example 5.2.2 (Synchronization of critical regions)**.** *Assume that e (drawn below) is an event inside the process net, with pre-control conditions C and post-control conditions C', that acquires the open resource r.*

Suppose that the invariant for r is $\ell' \mapsto 0 \vee (\ell' \mapsto 1 \star \ell \mapsto 0)$. There are two heaps, $D_1 = \{\ell' \mapsto 0\}$ and $D_2 = \{\ell' \mapsto 1, \ell \mapsto 0\}$ that satisfy this formula. There are correspondingly two interference events u_1 and u_2 that synchronize with e: the event u_1 acquires the resource r and transfers the ownership of ℓ' and r to the external process from the invariant, whereas the event u_2 acquires the resource r and transfers ownership of ℓ, ℓ' and r to the external process from the invariant. The event u_1 requires that the heap initially has value 0 at ℓ'; the event u_2 requires that the heap initially has value 1 at ℓ' and 0 at ℓ.

The synchronized events $e \cdot u_1$ and $e \cdot u_2$ are similar, transferring ownership from the invariant to the process being considered.



Even though there are two synchronized events corresponding to e in the example above, namely $e \cdot u_1$ and $e \cdot u_2$, in Lemma 5.2.1 we shall see that the precision of invariants means that in any reachable marking only one of these can occur. First, though, we shall give the formal definition of the nets formed with interfererence and synchronized events.

The semantics of judgements made using the rules of concurrent separation logic will consider a net $\mathcal{W}[\![t]\!]_\Gamma$ with interference events to represent external processes running and synchronized events to represent the process t.

Definition 5.2.3 (Ownership net)**.** *The ownership net for t in Γ, denoted $\mathcal{W}[\![t]\!]_\Gamma$, is the net formed with the previous definitions of control conditions* **C***, state conditions* **S** *and ownership conditions* **W***, and events:*

- *Every event u from the interference net for Γ, and*
- *Every event $e \cdot u$ where e is an event of $\mathcal{N}[\![t]\!]$ and u from the interference net such that e and u synchronize.*

We shall continue to use the symbol e to refer to any kind of event in ownership nets, but shall reserve the symbol s for those events known in particular to be synchronized events.

Behavioural properties of ownership nets

We now turn to some behavioural properties of ownership nets. We begin with Lemma 5.2.1, an important property meaning that, because invariants are precise formulae, there is never any choice in runs of the ownership net of how ownership is transferred once it is known which corresponding event of the net $\mathcal{N}[\![t]\!]$ is to occur. We then relate the behaviour of ownership nets to the original nets in Lemmas 5.2.2 and 5.2.3, which rely on an important characterization of markings (called *violating markings*) in which the extra requirements due to ownership prevent the occurrence of the a synchronized event.

A consequence of requiring invariants to be precise is that at most one of the synchronized events corresponding to an event in $\mathcal{N}[\![t]\!]$ may be enabled in any marking of the ownership net $\mathcal{W}[\![t]\!]_\Gamma$.

Lemma 5.2.1. *For any marking σ of state conditions, let (C, σ, W) and (C', σ, W') be consistent markings of the net $\mathcal{W}[\![t]\!]_\Gamma$. For any event e in $\mathcal{N}[\![t]\!]$ and any interference*

events u and u' in $\mathcal{W}\,[\![t]\!]_\Gamma$, if $e \cdot u$ has concession in (C, σ, W) and $e \cdot u'$ has concession in (C', σ, W') then $u = u'$.

Proof. Straightforwardly seen to follow from precision by an analysis of the possible forms of the event e. □

The occurrence of a synchronized event $e \cdot u$ in a marking (C, σ, W) of the net $\mathcal{W}\,[\![t]\!]_\Gamma$ clearly gives rise to the occurrence of the event e in $\mathcal{N}\,[\![t]\!]$ in the marking (C, σ) and hence to the occurrence of e in $\mathcal{C}\,[\![t]\!]$ in the marking C. The earlier results describing the behaviour of $\mathcal{C}\,[\![t]\!]$ in terms of the behaviour of the nets representing its subterms can therefore be applied to the net $\mathcal{W}\,[\![t]\!]_\Gamma$.

Lemma 5.2.2. *If $M = (C, \sigma, W)$ and $M' = (C', \sigma', W')$ are markings of $\mathcal{W}\,[\![t]\!]_\Gamma$ and $M \xrightarrow{e} M'$ then either e is an interference event and $C = C'$ or $e = e_1 \cdot u$ for an event e_1 of $\mathcal{N}\,[\![t]\!]$ and an interference event u and $(C, \sigma) \xrightarrow{e_1} (C', \sigma')$ in $\mathcal{N}\,[\![t]\!]$.*

Proof. The events of $\mathcal{W}\,[\![t]\!]_\Gamma$ are, by definition, only interference events or synchronized events. If e is an interference event, $C = C'$ because ${}^{\mathbf{C}}e = \emptyset$ and $e^{\mathbf{C}} = \emptyset$. For a synchronized event $e_1 \cdot u$, observe that ${}^{\mathbf{C}}(e_1 \cdot u) = {}^{\mathbf{C}}e_1$ and that $(e_1 \cdot u)^{\mathbf{C}} = {e_1}^{\mathbf{C}}$, and similarly for ${}^{\mathbf{L}}e_1$, ${}^{\mathbf{R}}e_1$, ${}^{\mathbf{N}}e_1$, ${e_1}^{\mathbf{L}}$, ${e_1}^{\mathbf{R}}$ and ${e_1}^{\mathbf{N}}$. The only cases where either ${}^{\mathbf{D}}(e_1 \cdot u) \neq {}^{\mathbf{D}}e_1$ or $(e_1 \cdot u)^{\mathbf{D}} \neq {e_1}^{\mathbf{D}}$ are acquisition or release of an open resource, but in these cases ${}^{\mathbf{D}}e_1 = \emptyset = {e_1}^{\mathbf{D}}$ and ${}^{\mathbf{D}}(e_1 \cdot u) = (e_1 \cdot u)^{\mathbf{D}}$. The result follows as a straightforward calculation. □

It is now straightforward to show that the consistency of markings is preserved by the events in $\mathcal{W}\,[\![t]\!]_\Gamma$. In fact, the proof is slightly easier than that for the net $\mathcal{N}\,[\![t]\!]$ presented in Proposition 3.3 due to the additional requirements imposed by ownership.

Proposition 5.1 (Preservation of consistent markings)**.** *For any closed term t and environment Γ, if $\mathcal{W}\,[\![t]\!]_\Gamma : (\mathrm{Ic}(t), \sigma_0, W_0) \xrightarrow{*} (C, \sigma, W)$ and (σ_0, W_0) is consistent then (σ, W) is consistent.*

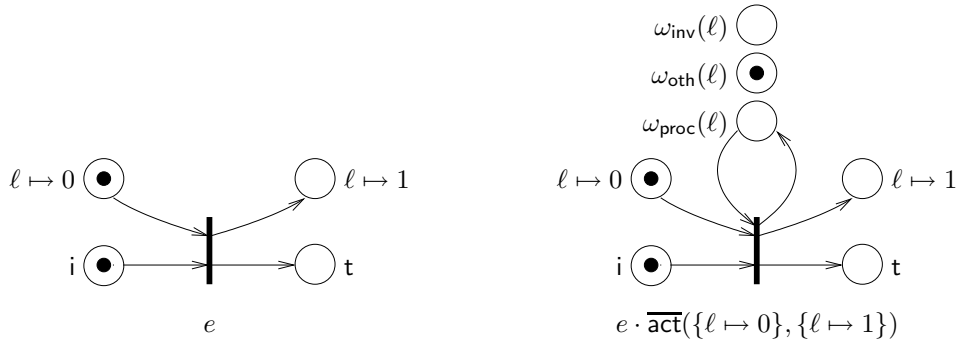
Proof. A straightforward analysis of the possible cases of the events in $\mathcal{W}\,[\![t]\!]_\Gamma$. □

The formulation of the ownership net permits a fundamental understanding of when a process acts in a way that cannot be seen as any form of interference from the perspective of another process; that is, when the process has violated its guarantees.

Definition 5.2.4 (Violating marking)**.** *Let (C, σ, W) be a consistent marking of $\mathcal{W}\,[\![t]\!]_\Gamma$. We say that M is* violating *if there exists an event e of $\mathcal{N}\,[\![t]\!]$ that has concession in marking (C, σ) but there is no event u from the interference net that synchronizes with e such that $e \cdot u$ has concession in (C, σ, W).*

We shall give two examples of violating markings. The first shall be an example of action on an unowned location and the second shows how release of an open resource will cause a violation if the invariant is not restored.

Example 5.2.3. *Let (σ, W) be a consistent state with heap component $\{\ell \mapsto 0\}$ and $\omega_{\mathsf{oth}}(\ell) \in W$. The event $e = \mathsf{act}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(\{\ell \mapsto 0\}, \{\ell \mapsto 1\})$, drawn below, has concession in the marking $(\mathrm{Ic}(\ell := 1), \sigma)$ of $\mathcal{N}\,[\![\ell := 1]\!]$ but the only corresponding synchronized event $e \cdot \overline{\mathsf{act}}(\{\ell \mapsto 0\}, \{\ell \mapsto 1\})$, also drawn below, does not have concession in the marking $(\mathrm{Ic}(\ell := 1), \sigma, W)$ of $\mathcal{W}\,[\![t]\!]_\Gamma$. The marking is therefore violating, representing the fact that the process acts on a location that it does not own.*



Example 5.2.4. *Let r be an open resource with the invariant $\chi = \ell' \mapsto 0 \vee (\ell' \mapsto 1 \star \ell \mapsto 0)$ and let (C, σ, W) be a consistent marking of $\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}$ with $\{\ell \mapsto 1, \ell' \mapsto 1\} \subseteq \sigma$ and $\omega_{\mathsf{proc}}(\ell), \omega_{\mathsf{proc}}(\ell') \in W$. Suppose further that the event $e = \mathsf{rel}_{(C_1,C_2)}(r)$ has concession in (C, σ) in the net $\mathcal{N}[\![t]\!]$. The only two interference events in $\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}$ that synchronize with e are*

$$\begin{aligned} u_1 &= \overline{\mathsf{rel}}(r, \{\ell' \mapsto 0\}) \\ u_2 &= \overline{\mathsf{rel}}(r, \{\ell' \mapsto 1, \ell \mapsto 0\}), \end{aligned}$$

corresponding to the two ways in which χ can be satisfied. The invariant is not satisfied in the heap component of σ, so the preconditions of the two events

$$\begin{aligned} {}^\bullet(e \cdot u_1) &= C \cup \{\ell' \mapsto 0, \omega_{\mathsf{proc}}(\ell')\} \\ {}^\bullet(e \cdot u_2) &= C \cup \{\ell' \mapsto 1, \ell \mapsto 0, \omega_{\mathsf{proc}}(\ell'), \omega_{\mathsf{proc}}(\ell)\} \end{aligned}$$

are not contained in the marking (C, σ, W), which is therefore therefore a violating marking because there was no part of the owned heap that satisfied the invariant yet the resource was released.

For any environment of invariants Γ, if no violating marking is ever encountered, the behaviour of $\mathcal{W}[\![t]\!]_\Gamma$ encapsulates all that of $\mathcal{N}[\![t]\!]$.

Lemma 5.2.3. *For any consistent marking (C, σ, W) of the net $\mathcal{W}[\![t]\!]_\Gamma$ and any event $e \in \mathrm{Ev}(t)$, if $(C, \sigma) \xrightarrow{e} (C', \sigma')$ in $\mathcal{N}[\![t]\!]$ then either (C, σ, W) is violating or there exists a marking of ownership conditions W' and an interference event u that synchronizes with e such that $(C, \sigma, W) \xrightarrow{e \cdot u} (C', \sigma', W')$ in $\mathcal{W}[\![t]\!]_\Gamma$.*

Proof. Immediate from the definition of violating marking and the fact that, for any e and u that synchronize and any state σ

$${}^{\mathbf{C}}(e \cdot u) = {}^{\mathbf{C}}e \qquad (e \cdot u)^{\mathbf{C}} = e^{\mathbf{C}} \qquad \sigma \setminus {}^{\mathbf{S}}(e \cdot u) \cup (e \cdot u)^{\mathbf{S}} = \sigma \setminus {}^{\mathbf{S}}e \cup e^{\mathbf{S}}$$

which is easily proved by inspection of the forms that $e \cdot u$ may take. □

5.3 Soundness and validity

Proving soundness of the logic deriving $\Gamma \vdash \{\varphi\} t \{\psi\}$ will involve showing that, whenever such a judgement is derivable, it is valid: that any run of $\mathcal{W}[\![t]\!]_\Gamma$ from a state in which the process owns part of the heap that satisfies φ and the invariants are satisfied is such that no violating marking is reachable and that whenever a terminal marking is reached, the part of the heap owned by the process satisfies ψ.

The key to the proof of soundness is to show that any run of the net $\mathcal{W}[\![t_1 \parallel t_2]\!]_\Gamma$ running from an appropriate initial state can be obtained by combining runs of $\mathcal{W}[\![t_1]\!]_\Gamma$ and $\mathcal{W}[\![t_2]\!]_\Gamma$ running from suitable initial states. For example, suppose that we derive the judgement $\emptyset \vdash \{\ell \mapsto 0 \star k \mapsto 0\}\ell := 1 \parallel k := 1\{\ell \mapsto 1 \star k \mapsto 1\}$ from $\emptyset \vdash \{\ell \mapsto 0\}\ell := 1\{\ell \mapsto 1\}$ and $\emptyset \vdash \{k \mapsto 0\}k := 1\{k \mapsto 1\}$. The runs that we will consider for validity of the judgement made about $\ell := 1 \parallel k := 1$ will assign ownership of ℓ and k to the parallel process. Soundness shall be shown by rule induction, so the induction hypotheses shall be the validity of the judgements made for $\ell := 1$ and $k := 1$. Consequently, we shall need to relate runs of the process $\ell := 1 \parallel k := 1$ from states in which the process owns ℓ and k to runs of $\ell := 1$ from states in which the process owns ℓ and runs of $k := 1$ from states in which the process owns k.

The rule for parallel composition permits the view that the ownership of the heap is initially split between the two processes, so that what one process owns is seen as owned by an external process by the other.

Definition 5.3.1 (Ownership split)**.** *Let W be a marking of ownership conditions. Markings of ownership conditions W_1 and W_2 form an* ownership split *of W if for all $z \in \mathrm{Loc} \cup \mathrm{Res}$:*

- $\omega_{\mathsf{oth}}(z) \in W$ *iff* $\omega_{\mathsf{oth}}(z) \in W_1$ *and* $\omega_{\mathsf{oth}}(z) \in W_2$,
- $\omega_{\mathsf{inv}}(z) \in W$ *iff* $\omega_{\mathsf{inv}}(z) \in W_1$ *and* $\omega_{\mathsf{inv}}(z) \in W_2$, *and*
- $\omega_{\mathsf{proc}}(z) \in W$ *iff either* $\omega_{\mathsf{proc}}(z) \in W_1$ *and* $\omega_{\mathsf{oth}}(z) \in W_2$, *or* $\omega_{\mathsf{proc}}(z) \in W_2$ *and* $\omega_{\mathsf{oth}}(z) \in W_1$.

If W_1 and W_2 form an ownership split of W, then fewer locations and resources are owned by the process in W_1 than in W, and similarly for W_2. As one would expect, a process can act in the same way without causing a violation if it owns more, and more interference can occur if the process owns less. This is the essence of the frame property referred to earlier.

Lemma 5.3.1. *Consider consistent markings of the net $\mathcal{W}[\![t]\!]_\Gamma$. Let W_1 and W_2 form an ownership split of W.*

- *For any synchronized event $s = e \cdot u$, if $(C, \sigma, W_1) \xrightarrow{s}\!\!\!\!\rightarrow (C', \sigma', W_1')$ then there exist W' and W_2' such that $(C, \sigma, W) \xrightarrow{s}\!\!\!\!\rightarrow (C', \sigma', W')$ and $(C, \sigma, W_2) \xrightarrow{u}\!\!\!\!\rightarrow (C, \sigma', W_2')$, and furthermore W_1' and W_2' form an ownership split of W'.*
- *For any interference event u, if $(C, \sigma, W) \xrightarrow{u}\!\!\!\!\rightarrow (C, \sigma', W')$ then there exist W_1' and W_2' such that $(C, \sigma, W_1) \xrightarrow{u}\!\!\!\!\rightarrow (C, \sigma, W_1')$ and $(C, \sigma, W_2) \xrightarrow{u}\!\!\!\!\rightarrow (C, \sigma', W_2')$, and furthermore W_1' and W_2' form an ownership split of W'.*

Proof. Straightforward to verify by going through the possible forms that s and u may take. □

Following Brookes' lead, we are now able to prove the key lemma upon which the proof of soundness lies. The effect of this lemma is that the terminal states of parallel processes may be determined simply by observing the terminal markings of the net of each parallel process running in isolation if we split the ownership of the initial state correctly. The technique is inspired by Brookes' proof, though it is also very different. Brookes' proof is based on a more abstract, less direct consideration of a *local enabling relation* which,

essentially, defines the semantics over partial representations of the heap. We do not define such an additional semantics: instead, we use the embellishment of the original semantics for processes with interference and ownership to obtain the proof almost automatically.

For convenience, the lemma is stated without intimating the particular event that takes place on the net transition relation.

Lemma 5.3.2 (Parallel decomposition)**.** *Let $M = (\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2, \sigma, W)$ be a consistent marking of the net $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$, and let W_1 and W_2 form an ownership split of W. The markings $M_1 = (C_1, \sigma, W_1)$ and $M_2 = (C_2, \sigma, W_2)$ are consistent, and furthermore:*

- *If the marking M is violating in $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$ then either M_1 is violating in $\mathcal{W}\,[\![t_1]\!]_\Gamma$ or M_2 is violating in $\mathcal{W}\,[\![t_2]\!]_\Gamma$.*
- *If neither M_1 nor M_2 is violating and $(\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2, \sigma, W) \twoheadrightarrow (\mathsf{par}\,1{:}C_1' \cup \mathsf{par}\,2{:}C_2', \sigma', W')$ in $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$ then there exist W_1' and W_2' forming an ownership split of W' such that $(C_1, \sigma, W_1) \twoheadrightarrow (C_1', \sigma', W_1')$ in $\mathcal{W}\,[\![t_1]\!]_\Gamma$ and $(C_2, \sigma, W_2) \twoheadrightarrow (C_2', \sigma', W_2')$ in $\mathcal{W}\,[\![t_2]\!]_\Gamma$.*

Proof. It is straightforward from Definition 5.2.1 to see that M_i is a consistent marking for both $i \in \{1, 2\}$.

1. Suppose that the marking M is violating in $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$. Without loss of generality, assume that this is because there exists an event $\mathsf{par}\,1{:}e_1$ of $\mathcal{N}\,[\![t_1 \parallel t_2]\!]$ that has concession in marking $(\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2, \sigma)$ but there is no event interference event u such that $\mathsf{par}\,1{:}e_1$ synchronizes with u and $(\mathsf{par}\,1{:}e_1) \cdot u$ has concession in M. Assume, for contradiction, that the marking M_1 is non-violating in $\mathcal{W}\,[\![t_1]\!]_\Gamma$. The event e_1 has concession in marking (C_1, σ) of $\mathcal{N}\,[\![t_1]\!]$ by the first part of Lemma 3.3.2, so there must exist u_1 an interference event of $\mathcal{W}\,[\![t_1]\!]_\Gamma$ such that $e_1 \cdot u_1$ has concession in M_1. The interference events of $\mathcal{W}\,[\![t_1]\!]_\Gamma$ are precisely the interference events of $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$ and the tagging of control conditions has no effect on whether events may synchronize, so the event $(\mathsf{par}\,1{:}e_1) \cdot u_1$ is in $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$. From Lemmas 5.3.1 and 3.3.2, the event $\mathsf{par}\,1{:}e_1 \cdot u_1$ has concession in marking M, which is therefore not violating — a contradiction.

2. It is a straightforward consequence of Lemma 5.3.1 that the second property holds if the transition $(\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2, \sigma, W) \twoheadrightarrow (\mathsf{par}\,1{:}C_1' \cup \mathsf{par}\,2{:}C_2', \sigma', W')$ is induced by the occurrence of an interference event. Suppose instead that it is induced by a synchronized event. Without loss of generality, suppose that in $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$ we have $M \overset{(\mathsf{par}\,1:e_1)\cdot u}{\twoheadrightarrow} M'$ for $M' = (\mathsf{par}\,1{:}C_1' \cup \mathsf{par}\,2{:}C_2', \sigma', W')$, for some event e_1 in $\mathcal{N}\,[\![t_1]\!]$. We shall show that $M_1 \overset{e_1 \cdot u}{\twoheadrightarrow} (C_1, \sigma', W_1')$ in $\mathcal{W}\,[\![t_1]\!]_\Gamma$ and $M_2 \overset{u}{\twoheadrightarrow} (C_2', \sigma', W_2')$ in $\mathcal{W}\,[\![t_2]\!]_\Gamma$ for some W_1', W_2' such that W_1' and W_2' form an ownership split of W'. Since we have $M \overset{(\mathsf{par}\,1:e_1)\cdot u}{\twoheadrightarrow} M'$ in $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$, it is easy to see that we have $(\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2, \sigma) \overset{\mathsf{par}\,1:e_1}{\twoheadrightarrow} (\mathsf{par}\,1{:}C_1' \cup \mathsf{par}\,2{:}C_2', \sigma')$ in $\mathcal{N}\,[\![t_1 \parallel t_2]\!]$ and $C_2 = C_2'$. Hence in $\mathcal{N}\,[\![t_1]\!]$ we have $(C_1, \sigma) \overset{e_1}{\twoheadrightarrow} (C_1', \sigma')$. By assumption, the marking (C_1, σ, W_1) is not a violating marking of $\mathcal{W}\,[\![t_1]\!]_\Gamma$, so there exists an interference event u_1 that synchronizes with e_1 such that $(C_1, \sigma, W_1) \overset{e_1 \cdot u_1}{\twoheadrightarrow} (C_1', \sigma', W_1'')$ for some W_1'' in $\mathcal{W}\,[\![t_1]\!]_\Gamma$, so in $\mathcal{W}\,[\![t_1 \parallel t_2]\!]_\Gamma$ we therefore have $(\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2, \sigma, W_1) \overset{(\mathsf{par}\,1:e_1)\cdot u_1}{\twoheadrightarrow} (\mathsf{par}\,1{:}C_1' \cup \mathsf{par}\,2{:}C_2, \sigma', W_1'')$. By Lemma 5.2.1, we have $u_1 = u$ and therefore $W_1'' = W_1$ because the occurrence of an event in a marking yields a unique marking. Now, by Lemma

5.3.1 there exist W'' and W_2' such that W_1' and W_2' form an ownership split of W'' and $(\mathsf{par}\ 1{:}C_1 \cup \mathsf{par}\ 2{:}C_2, \sigma, W) \overset{(\mathsf{par}\ 1:e_1)\cdot u}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{par}\ 1{:}C_1' \cup \mathsf{par}\ 2{:}C_2, \sigma', W'')$ and $(\mathsf{par}\ 1{:}C_1 \cup \mathsf{par}\ 2{:}C_2, \sigma, W_2) \overset{u}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{par}\ 1{:}C_1 \cup \mathsf{par}\ 2{:}C_2, \sigma', W_2')$. The occurrence of an event in a marking leads to a unique marking, so $W'' = W'$. It is easy to see that $(C_1, \sigma, W_1) \overset{e_1\cdot u}{\longrightarrow\!\!\!\!\!\rightarrow} (C_1', \sigma', W_1')$ in $\mathcal{W}[\![t_1]\!]_\Gamma$ and that $(C_2, \sigma, W_2) \overset{u}{\longrightarrow\!\!\!\!\!\rightarrow} (C_2, \sigma, W_2')$ in $\mathcal{W}[\![t_2]\!]_\Gamma$, so the proof is complete. □

The ownership semantics described above has been carefully defined to explicitly take into account the intuitions behind the rule for parallel composition, resulting in the short proof of the parallel decomposition lemma above. The remaining complexity in the proof of soundness lies in the rule for establishing an invariant associated with a resource:

$$(\text{L-Res}) : \frac{\Gamma, r{:}\chi \vdash \{\varphi\}[r/w]t\{\psi\}}{\Gamma \vdash \{\varphi \star \chi\}\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}\{\psi \star \chi\}} \left(\begin{array}{l} \chi \text{ precise} \\ r \notin \mathrm{dom}(\Gamma) \end{array} \right)$$

It is quite easy to see why this rule follows the intuitive semantics for judgements presented above: Any complete run of the net $\mathcal{W}[\![\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}]\!]_\Gamma$ to a terminal marking from a state with the heap owned by the process initially satisfying $\varphi \star \chi$ can be seen, in conjunction with Lemma 3.6.11, as consisting first of an event that declares a fresh resource r current, then a run of $\mathcal{W}[\![[r/w]t]\!]_\Gamma$, followed by an event that makes r non-current (thereby excluding the possibility that the run starts or ends with an interference event). The run of $\mathcal{W}[\![[r/w]t]\!]_\Gamma$ from a state where the part of the heap that the process owns satisfies $\varphi \star \chi$ is simulated by a run of $\mathcal{W}[\![[r/w]t]\!]_{\Gamma, r:\chi}$ along which the locations satisfying χ are owned by the invariant χ in an environment where r is an open resource. The run of $\mathcal{W}[\![[r/w]t]\!]_{\Gamma, r:\chi}$ obtained has no interference on the resource r or the locations that it protects and r is available in the terminal state of the run. Assuming the validity of the judgement $\Gamma, r{:}\chi \vdash \{\varphi\}[r/w]t\{\psi\}$, the resulting state owned by the process is therefore seen to satisfy the formula $\varphi \star \chi$. Similarly, if there were a reachable marking in $\mathcal{W}[\![\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}]\!]_\Gamma$ where the process accesses a location or resource that it does not own then there would be a reachable marking in $\mathcal{W}[\![[r/w]t]\!]_{\Gamma, r:\chi}$ where the process accesses an unowned location or resource. The more formal presentation of this intuition follows. However, before doing so, we introduce a little notation that will be of significant use later. The first is a notation for all the available resources.

Definition 5.3.2 (Conjunction of available resource invariants, $\mathrm{inv}(\Gamma, R)$)**.** *Write* $\mathrm{inv}(\Gamma, R)$ *for the formula* $\chi_1 \star \ldots \star \chi_n$ *formed as the separating conjunction of the invariants of all the* available*, according to* R*, open resources. It is defined by induction on the size of the domain of* Γ*:*

$$\begin{aligned} \mathrm{inv}(\emptyset, R) &\triangleq \mathsf{empty} \\ \mathrm{inv}((\Gamma, r{:}\chi), R) &\triangleq \begin{cases} \mathrm{inv}(\Gamma, R), & \text{if } r \notin R \\ \chi \star \mathrm{inv}(\Gamma, R), & \text{if } r \in R. \end{cases} \end{aligned}$$

The second is notation for restricting the heap to its parts that are owned by the current process, protected by invariants for available resources, and owned by other processes.

Definition 5.3.3 (Heap restriction by ownership)**.** *Define the notations*

$$\begin{aligned} D \restriction_W \mathsf{proc} &\triangleq \{\ell \mapsto v \in D \mid \omega_{\mathsf{proc}}(\ell) \in W\} \\ D \restriction_W \mathsf{inv} &\triangleq \{\ell \mapsto v \in D \mid \omega_{\mathsf{inv}}(\ell) \in W\} \\ D \restriction_W \mathsf{oth} &\triangleq \{\ell \mapsto v \in D \mid \omega_{\mathsf{oth}}(\ell) \in W\} \end{aligned}$$

to represent the heap at locations owned by the process, invariants and other processes, respectively.

In the states that we consider, we often expect and correspondingly demonstrate that $D \restriction_W \mathsf{inv} \models \mathrm{inv}(\Gamma, R)$.

Technical results on resources

On first reading, if happy with the intuitive explanation above, the reader may wish to pass quickly through this rather technical account to Definition 5.3.5. We shall begin by explicitly characterizing the runs of the net $\mathcal{W}[\![\texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}]\!]_\Gamma$, analogously to Lemma 3.6.11 which characterizes the runs of $\mathcal{C}[\![\texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}]\!]$. Its formal statement is rather long, but it says nothing more than that any run comprises some number of interference events, then an event that declares w to be r for some resource r, then a run of $\mathcal{W}[\![[r/w]t_0]\!]_\Gamma$, and finally an event that records the end of the declaration of w followed by some interference events.

Lemma 5.3.3. *Suppose that σ_0 and W_0 form a consistent marking of state and ownership conditions and let $t = \texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}$. For a resource r, define the synchronized events*

$$\begin{aligned} s_r &= \mathsf{decl}_{(\{\mathrm{i}\},\mathsf{res}\, r{:}\mathrm{Ic}([r/w]t_0))}(r) \cdot \overline{\mathsf{decl}}(r) \\ s'_r &= \mathsf{end}_{(\mathsf{res}\, r{:}\mathrm{Tc}([r/w]t_0),\{\mathrm{t}\})}(r) \cdot \overline{\mathsf{end}}(r) \end{aligned}$$

For any path π such that $\mathcal{W}[\![t]\!]_\Gamma : (\mathrm{Ic}(t), \sigma_0, W_0) \xrightarrow{\pi}\!\!\!\!\rightarrow (C, \sigma, W)$, either:

- *$C = \mathrm{Ic}(t)$ and π consists only of interference events, or*
- *there exist $r, C', \sigma', W', \pi_0$ and π_1 such that π_0 comprises only interference events, $C = \mathsf{res}\, r{:}C'$ and*

$$\pi = \pi_0 \cdot s_r \cdot (\mathsf{res}\, r{:}\pi_1)$$

and

$$\begin{aligned} \mathcal{W}[\![t]\!]_\Gamma &: \quad (\mathrm{Ic}(t), \sigma_0, W_0) \xrightarrow{\pi_0 \cdot s_r}\!\!\!\!\rightarrow (\mathsf{res}\, r{:}\mathrm{Ic}([r/w]t_0), \sigma', W') \quad \textit{and} \\ \mathcal{W}[\![[r/w]t_0]\!]_\Gamma &: \quad (\mathrm{Ic}([r/w]t_0), \sigma', W') \xrightarrow{\pi_1}\!\!\!\!\rightarrow (C', \sigma, W), \qquad\qquad\qquad \textit{or} \end{aligned}$$

- *$C = \mathrm{Tc}(t)$ and there exist $r, \sigma', \sigma'', W', W'', \pi_0, \pi_1, \pi_2$ such that π_0 and π_2 comprise only interference events,*

$$\pi = \pi_0 \cdot s_r \cdot (\mathsf{res}\, r{:}\pi_1) \cdot s'_r \cdot \pi_2,$$

and

$$\begin{aligned} \mathcal{W}[\![t]\!]_\Gamma &: \quad (\mathrm{Ic}(t), \sigma_0, W_0) \xrightarrow{\pi_0 \cdot s_r}\!\!\!\!\rightarrow (\mathsf{res}\, r{:}\mathrm{Ic}([r/w]t_0), \sigma', W'), \\ \mathcal{W}[\![[r/w]t_0]\!]_\Gamma &: \quad (\mathrm{Ic}([r/w]t_0), \sigma', W') \xrightarrow{\pi_1}\!\!\!\!\rightarrow (\mathrm{Tc}([r/w]t_0), \sigma'', W''), \quad \textit{and} \\ \mathcal{W}[\![t]\!]_\Gamma &: \quad (\mathsf{res}\, r{:}\mathrm{Tc}([r/w]t_0), \sigma'', W'') \xrightarrow{s'_r \cdot \pi_2}\!\!\!\!\rightarrow (\mathrm{Tc}(t), \sigma, W). \end{aligned}$$

Proof. A straightforward argument that follows from Lemmas 3.7.1, 3.6.11, 3.4.1 and 5.2.2. □

It can be shown, as a consequence of the preceding lemma, that during any run of the net following the declaration event, the resource r chosen for w is owned by the process until it is made non-current at the end of the variable w's scope.

Lemma 5.3.4. *Let* $t = \texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}$. *If* $(\mathsf{res}\, r{:}C', \sigma, W)$ *is reachable from* $(\mathrm{Ic}(t), \sigma_0, W_0)$, *which is a consistent marking of* $\mathcal{W}[\![t]\!]_\Gamma$, *then* $\omega_{\mathsf{proc}}(r) \in W$.

Proof. Let $\pi \cdot e$ be the path $\mathcal{W}[\![t]\!]_\Gamma : (\mathrm{Ic}(t), \sigma_0, W_0) \stackrel{\pi \cdot e}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{res}\, r{:}C', \sigma, W)$. Suppose, for contradiction, that $\omega_{\mathsf{proc}}(r) \notin W$. Without loss of generality, we may assume that $(\mathsf{res}\, r{:}C', \sigma, W)$ is the earliest marking along $\pi \cdot e$ such that $\omega_{\mathsf{proc}}(r) \notin W$.

According to the second clause of Lemma 5.3.3, there exists a path π_1 such that

$$\mathcal{W}[\![[r/w]t_0]\!]_\Gamma : (\mathrm{Ic}([r/w]t_0), \sigma', W') \stackrel{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} (C', \sigma, W)$$

and there exists a path π_0 such that

$$\mathcal{W}[\![t]\!]_\Gamma : (\mathrm{Ic}(t), \sigma_0, W_0) \stackrel{\pi_0 \cdot s_r}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{res}\, r{:}\mathrm{Ic}([r/w]t_0), \sigma', W')$$

for s_r the synchronized event that declares the new local resource r as defined in Lemma 5.3.3. Hence $\mathsf{curr}(r) \in \sigma'$ and $\omega_{\mathsf{proc}}(r) \in W'$. Furthermore, $\pi \cdot e = \pi_0 \cdot s_r \cdot (\mathsf{res}\, r{:}\pi_1)$. It follows that there exist an event e_1 and a path π_1' such that $e = \mathsf{res}\, r{:}e_1$ and $\pi_1 = \pi_1' \cdot e_1$ and

$$\mathcal{W}[\![[r/w]t_0]\!]_\Gamma : (\mathrm{Ic}([r/w]t_0), \sigma', W') \stackrel{\pi_1'}{\longrightarrow\!\!\!\!\!\rightarrow} (C'', \sigma'', W'') \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} (C', \sigma, W).$$

The resource r does not occur in the domain of Γ since the event s_r occurs in a consistent, by Proposition 5.1, marking of $\mathcal{W}[\![t]\!]_\Gamma$. Since we have $\omega_{\mathsf{proc}}(r) \in W''$ but $\omega_{\mathsf{proc}}(r) \notin W$, the only form of event that e_1 can take is that of a synchronized event that ends the resource r; that is, there exist C_1 and C_1' such that $e_1 = \mathsf{end}_{(C_1, C_1')}(r) \cdot \overline{\mathsf{end}}(r)$. Let π_1'' be the path obtained by stripping away the interference events from π_1'; it is easy to see that

$$\mathcal{C}[\![[r/w]t_0]\!] : \mathrm{Ic}([r/w]t_0) \stackrel{\pi_1''}{\longrightarrow\!\!\!\!\!\rightarrow} C'' \stackrel{\mathsf{end}_{(C_1, C_1')}(r)}{\longrightarrow\!\!\!\!\!\rightarrow} C'.$$

It can be shown, in a manner similar to that in Section 3.10, that there must exist an event that declares the resource r inside π_1'', so there exist C_2, C_2' such that $\mathsf{decl}_{(C_2, C_2')}(r)$ is in π_1''. It follows that the event $\mathsf{decl}_{(C_2, C_2')}(r) \cdot \overline{\mathsf{end}}(r)$ is in π_1. For this event to have concession, the condition $\omega_{\mathsf{proc}}(r)$ cannot be marked, but we earlier assumed that $(\mathsf{res}\, r{:}C', \sigma, W)$ was the earliest marking on which $\omega_{\mathsf{proc}}(r)$ was not marked, so we arrive at the required contradiction. □

When considering validity of the judgement

$$\Gamma \vdash \{\varphi \star \chi\}\texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}\{\psi \star \chi\},$$

we will have by induction validity of $\Gamma, r{:}\chi \vdash \{\varphi\}[r/w]t_0\{\psi\}$. A marking of the net $\mathcal{W}[\![t_0]\!]_\Gamma$ can be converted to a marking of $\mathcal{W}[\![t_0]\!]_{\Gamma, r:\chi}$ by, if r is available, regarding ownership of the locations satisfying the invariant χ as being owned by the invariant rather than by the process.

Definition 5.3.4. *Suppose that* χ *is a precise heap formula. Suppose that the consistent marking* $M = (C, (D, L, R, N), W)$ *of* $\mathcal{W}[\![t]\!]_\Gamma$ *is such that if* $r \in R$ *then there exists (necessarily unique)* $D_0 \subseteq D \restriction_W \mathsf{proc}$ *such that* $D_0 \models \chi$. *Define the projection of* M *into the net* $\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}$ *to be*

$$\mathsf{toInv}_r^\chi(M) \triangleq (C, (D, L, R, N), W'),$$

where:

- *if* $r \notin R$: $W' = W$
- *if* $r \in R$:

$$\begin{array}{rcl} W' & = & \{\omega_{\mathsf{oth}}(\ell) \ \ | \omega_{\mathsf{oth}}(\ell) \in W\} \\ & \cup & \{\omega_{\mathsf{oth}}(r') \ | \omega_{\mathsf{oth}}(r') \in W\} \\ & \cup & \{\omega_{\mathsf{inv}}(\ell) \ \ | \omega_{\mathsf{inv}}(\ell) \in W \ \textit{or} \ \ \ell \in \mathrm{dom}(D_0)\} \\ & \cup & \{\omega_{\mathsf{inv}}(r') \ | \omega_{\mathsf{inv}}(r') \in W \ \textit{or} \ \ r' = r\} \\ & \cup & \{\omega_{\mathsf{proc}}(\ell) \ | \omega_{\mathsf{proc}}(\ell) \in W \ \textit{and} \ \ \ell \notin \mathrm{dom}(D_0)\} \\ & \cup & \{\omega_{\mathsf{proc}}(r') | \omega_{\mathsf{proc}}(r') \in W \ \textit{and} \ \ r' \neq r\} \end{array}$$

It is clear that if M is a consistent marking of $\mathcal{W}[\![t]\!]_\Gamma$ then $\mathsf{tolnv}_r^\chi(M)$ is a consistent marking of $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$. They key lemma representing the account above is that behaviour in the net where a resource is closed is simulated by the net where the resource is open.

Lemma 5.3.5. *Let r be a resource such that $r \notin \mathrm{dom}(\Gamma)$ and let χ be a precise heap logic formula. Let $M = (C, (D, L, R, N), W)$ be a consistent marking of $\mathcal{W}[\![t]\!]_\Gamma$ such that:*

- $\omega_{\mathsf{proc}}(r) \in W$,
- $D \upharpoonright_W \mathsf{inv} \models \mathrm{inv}(\Gamma, R)$, *and*
- *if $r \in R$ then there exists $D_0 \subseteq D \upharpoonright_W \mathsf{proc}$ such that $D_0 \models \chi$.*

Then

1. *If M is a violating marking in $\mathcal{W}[\![t]\!]_\Gamma$ then $\mathsf{tolnv}_r^\chi(M)$ is a violating marking in $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$.*
2. *For any event u of $\mathcal{W}[\![t]\!]_\Gamma$ that is an interference event, if M is a non-violating marking and $M \xrightarrow{u}\!\!\!\!\rightarrow M'$ where $M' = (C', (D', L', R', N'), W')$ and $\omega_{\mathsf{proc}}(r) \in W'$ then:*
   - $\mathsf{tolnv}_r^\chi(M) \xrightarrow{u}\!\!\!\!\rightarrow \mathsf{tolnv}_r^\chi(M')$ *in* $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$ *and:*
     - $D' \upharpoonright_{W'} \mathsf{inv} \models \mathrm{inv}(\Gamma, R')$
     - *if $r \in R'$ then there exists $D_0 \subseteq D' \upharpoonright_{W'} \mathsf{proc}$ such that $D_0 \models \chi$.*
3. *For any synchronized event $s = e_1 \cdot u$ of $\mathcal{W}[\![t]\!]_\Gamma$, if M is not a violating marking and $M \xrightarrow{s}\!\!\!\!\rightarrow M'$ where $M' = (C', (D', L', R', N'), W')$ and $\omega_{\mathsf{proc}}(r) \in W'$ then either:*
   - $\mathsf{tolnv}_r^\chi(M)$ *is violating in* $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$, *or*
   - *there exists u' such that* $\mathsf{tolnv}_r^\chi(M) \xrightarrow{e_1 \cdot u'}\!\!\!\!\rightarrow \mathsf{tolnv}_r^\chi(M')$ *in* $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$ *and:*
     - $D' \upharpoonright_{W'} \mathsf{inv} \models \mathrm{inv}(\Gamma, R')$
     - *if $r \in R'$ then there exists $D_0 \subseteq D' \upharpoonright_{W'} \mathsf{proc}$ such that $D_0 \models \chi$.*

Proof. We show (1), (2) and (3) in turn.

1. Let $\sigma = (D, L, R, N)$ and suppose that $M = (C, \sigma, W)$ is a violating marking of $\mathcal{W}[\![t]\!]_\Gamma$. There exists an event e such that $\mathcal{N}[\![t]\!] : (C, \sigma) \xrightarrow{e}\!\!\!\!\rightarrow$ but there is no interference event u that synchronizes with e in $\mathcal{W}[\![t]\!]_\Gamma$ such that $\mathcal{W}[\![t]\!]_\Gamma : (C, \sigma, W) \xrightarrow{e \cdot u}\!\!\!\!\rightarrow$. We wish to show that there is no u' that synchronizes with e in $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$ that has concession in $\mathsf{tolnv}_r^\chi(M)$.

For contradiction, suppose that there exists such a u'. There are only two events that u' might be that are not in $\mathcal{W}[\![t]\!]_\Gamma$. These are $\overline{\mathsf{acq}}(r, D_0)$ and $\overline{\mathsf{rel}}(r, D_0)$. In these two cases, however, the event e synchronizes with $\overline{\mathsf{acq}}(r)$ and $\overline{\mathsf{rel}}(r)$, respectively, which are the corresponding interference events since the resource $r \notin \mathrm{dom}(\Gamma)$, and the synchronized event clearly has concession in M.

2. The event u cannot be equal to $\overline{\mathsf{acq}}(r), \overline{\mathsf{rel}}(r), \overline{\mathsf{decl}}(r)$ or $\overline{\mathsf{end}}(r)$ since, by assumption, $\omega_{\mathsf{proc}}(r) \in W$. The result is easily seen to follow from a case analysis of the possible other forms that u might take.

3. The case is shown by an analysis of the forms that e_1 and u may take. The important cases will be for action, where it will be shown that the marking $\mathsf{tolnv}^\chi_r(M)$ being non-violating implies that the process does not act on the locations in D_0 if r is available, and for release of the resource r, where it will be shown that $\mathsf{tolnv}^\chi_r(M)$ being non-violating ensures that there is part of the heap that satisfies the invariant. All the other cases are straightforward. In more detail:

$e_1 = \mathsf{act}_{(C_1,C_2)}(D_1, D_2)$ and $u = \overline{\mathsf{act}}(D_1, D_2)$: We have $R = R'$. Let $L_1 = \mathrm{dom}(D_1)$. The definition of actions requires that D_1 and D_2 have the same domain, so $\mathrm{dom}(D_1) = \mathrm{dom}(D_2)$. Suppose that $\mathsf{tolnv}^\chi_r(M)$ is non-violating in $\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}$. First, assume that $r \notin R$. Since $\mathsf{tolnv}^\chi_r(M)$ is non-violating, there is no $\ell \in L_1$ such that $\omega_{\mathsf{inv}}(\ell) \in W$. Hence $D' \restriction_{W'} \mathsf{inv} = D \restriction_W \mathsf{inv}$ and therefore $D' \restriction_{W'} \mathsf{inv} \models \mathrm{inv}(\Gamma, R')$, as required to complete the case. Now assume that $r \in R$. By assumption, there exists $D_0 \subseteq D$ such that $D_0 \models \chi$. Since $\mathsf{tolnv}^\chi_r(M)$ is non-violating, there is no $\ell \in L_1$ such that $\omega_{\mathsf{inv}}(\ell) \in W$ or $\ell \in \mathrm{dom}(D_0)$. It follows immediately that $D \restriction_W \mathsf{inv} = D' \restriction_{W'} \mathsf{inv}$, and so $D' \restriction_{W'} \mathsf{inv} \models \mathrm{inv}(\Gamma, R')$, and that $D_0 \subseteq D'$ as required to complete the case.

$e_1 = \mathsf{rel}_{(C_1,C_2)}(r')$ and $u = \overline{\mathsf{rel}}(r')$: The case is straightforward unless $r' = r$. Since we have $M \xrightarrow{e_1 \cdot u} M'$ in $\mathcal{W}[\![t]\!]_\Gamma$, it is easy to see that $(C, (D, L, R, N)) \xrightarrow{e_1} (C', (D', L', R', N'))$ in $\mathcal{N}[\![t]\!]$. The marking $\mathsf{tolnv}^\chi_r(M)$ is a non-violating marking of $\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}$, so there exists an interference event u_1 such that $\mathsf{tolnv}^\chi_r(M) \xrightarrow{e_1 \cdot u_1} \mathsf{tolnv}^\chi_r(M')$. Since the resource r is open in $\Gamma, r{:}\chi$ and u_1 synchronizes with e_1, there must exist D_0 such that $u_1 = \overline{\mathsf{rel}}(r, D_0)$ and $D_0 \models \chi$. Since the event $e_1 \cdot u_1$ has concession in $\mathsf{tolnv}^\chi_r(M)$, we have $D_0 \subseteq D$ and $\omega_{\mathsf{proc}}(\mathrm{dom}(D_0)) \subseteq W$. We also have $R' = R \cup \{r\}$ and $D' = D$ and $W' = W$. It follows that $D' \restriction_{W'} \mathsf{inv} = D \restriction_W \mathsf{inv}$, so $D' \restriction_{W'} \mathsf{inv} \models \mathrm{inv}(\Gamma, R')$, completing the case. □

Validity and soundness

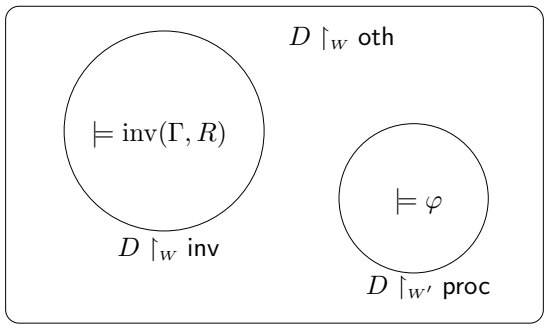
All the pieces are now in place for us to turn to validity. We shall say that a state σ with an ownership marking W satisfies the formula φ and the invariants in Γ if the heap restricted to the owned locations satisfies φ and the invariants are met for all the available resources. The rest of the heap, seen as owned by external processes, is unconstrained.

Definition 5.3.5. *Let $\sigma = (D, L, R, N)$. A marking (C, σ, W) of $\mathcal{W}[\![t]\!]_\Gamma$* satisfies φ in Γ *if:*

- *the marking (C, σ, W) is consistent,*
- *$D \restriction_W \mathsf{proc} \models \varphi$, and*

- $D \upharpoonright_W \mathsf{inv} \models \mathrm{inv}(\Gamma, R)$.

Graphically, we have:



We now attach a notion of validity to judgements $\Gamma \vdash \{\varphi\}t\{\psi\}$. It shall assert that no violating marking is ever reached and that whenever the process t runs to completion from a state where the part of the heap that it owns satisfies φ then the part of the resulting heap that it owns satisfies ψ.

Definition 5.3.6 (Validity)**.** *Let t be a closed term. Define $\Gamma \models \{\varphi\}t\{\psi\}$ if, for any σ and W such that the marking $(\mathrm{Ic}(t), \sigma, W)$ satisfies φ in Γ:*

- *any marking reachable in $\mathcal{W}[\![t]\!]_\Gamma$ from $(\mathrm{Ic}(t), \sigma, W)$ is non-violating, and*
- *for any σ' and W', if the terminal marking $(\mathrm{Tc}(t), \sigma', W')$ is reachable in $\mathcal{W}[\![t]\!]_\Gamma$ from the initial marking $(\mathrm{Ic}(t), \sigma, W)$ then $(\mathrm{Tc}(t), \sigma', W')$ satisfies ψ in Γ.*

It is useful to note that the occurrence of an interference event does not affect whether a marking satisfies φ in Γ or whether it is violating. Visually, this is because enabled interference events only connect to parts of the heap owned by 'other' processes. Consequently, when considering validity it is unnecessary to account for runs of the net $\mathcal{W}[\![t]\!]_\Gamma$ that start or end with an interference event.

Lemma 5.3.6. *Let M be a consistent marking of $\mathcal{W}[\![t]\!]_\Gamma$ that satisfies φ in Γ and is non-violating. If u is an interference event and $M \xrightarrow{u}\!\!\!\!\rightarrow M'$ then M' satisfies φ in Γ and is non-violating.*

Proof. Straightforward from the definition of satisfaction of φ in Γ by considering the possible forms of u. □

In the rule (L-Res) which allows invariants to be established, only one resource constant is considered for substitution for the variable, whereas the semantics allows *any* resource to be chosen. The following lemma shows that this is sufficient; that judgements are unaffected by the choice of resource. In the terminology of nominal sets [GP01], this states that the property is *equivariant*.

Lemma 5.3.7. *For any two resources r and r', define the operation $(r\ r') \cdot t$ on terms as swapping any occurrences of r and r' in t. Define an operation $(r\ r') \cdot \Gamma$ on environments similarly, interchanging any occurrences of r and r' in Γ. Then*

$$\Gamma \vdash \{\varphi\}t\{\psi\} \iff (r\ r') \cdot \Gamma \vdash \{\varphi\}(r\ r') \cdot t\{\psi\}.$$

Proof. An induction on the proof tree of the judgement $\vdash$ shows that

$$\Gamma \vdash \{\varphi\}t\{\psi\} \implies \forall r, r'.\ (r\ r') \cdot \Gamma \vdash \{\varphi\}(r\ r') \cdot t\{\psi\},$$

from which the result follows (the reverse direction as a consequence of $(r\ r') \cdot (r\ r') \cdot \Gamma = \Gamma$ and $(r\ r') \cdot (r\ r') \cdot t = t$).

The only interesting case in the proof is for $\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}$, so suppose that $\Gamma \vdash \{\varphi \star \chi\}\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}\{\psi \star \chi\}$ because there exists $r_0 \notin \mathrm{dom}(\Gamma)$ such that $\Gamma, r_0{:}\chi \vdash \{\varphi\}[r_0/w]t\{\psi\}$. By induction, $(r\ r') \cdot (\Gamma, r_0{:}\chi) \vdash \{\varphi\}(r\ r') \cdot [r_0/w]t\{\psi\}$. Suppose that $r_0 = r$ (or, symmetrically, $r_0 = r'$) — if neither equality holds, the proof is easier. We have $(r\ r') \cdot (\Gamma, r_0{:}\chi) = ((r\ r') \cdot \Gamma), r'{:}\chi$ and $(r\ r') \cdot [r_0/w]t = [r'/w](r\ r') \cdot t$. Since $r' \notin \mathrm{dom}((r\ r') \cdot \Gamma)$ because $r = r_0 \notin \mathrm{dom}(\Gamma)$, we can apply the rule (L-Res) to obtain

$$(r\ r') \cdot \Gamma \vdash \{\varphi \star \chi\}(r\ r') \cdot t\{\psi \star \chi\},$$

as required. □

The insensitivity of the logic to permutation of resources is matched in the validity of judgements.

Lemma 5.3.8. *For any resources r and r'*

$$\Gamma \models \{\varphi\}t\{\psi\} \iff (r\ r') \cdot \Gamma \models \{\varphi\}(r\ r') \cdot t\{\psi\}.$$

Proof. The net $\mathcal{W}[\![t]\!]_\Gamma$ is clearly isomorphic to $\mathcal{W}[\![(r\ r') \cdot t]\!]_{(r\ r') \cdot \Gamma}$ through interchanging the conditions

$$r \leftrightarrow r' \qquad \mathsf{curr}(r) \leftrightarrow \mathsf{curr}(r')$$
$$\omega_{\mathsf{proc}}(r) \leftrightarrow \omega_{\mathsf{proc}}(r') \qquad \omega_{\mathsf{inv}}(r) \leftrightarrow \omega_{\mathsf{inv}}(r') \qquad \omega_{\mathsf{oth}}(r) \leftrightarrow \omega_{\mathsf{oth}}(r').$$

The result follows from the definition of validity being insensitive to such permutations. □

We are now in a position where we the rules of concurrent separation logic can be proved sound. Before we do so, we note that the use of the rules (L-Contraction) and (L-Expansion) can be eliminated from the derivation of any judgement. It will therefore be unnecessary to consider them in the proof of soundness.

Lemma 5.3.9. *Let $\vdash_0$ denote the judgement formed with the rules of separation logic defined in Figures 4.3 and 4.4 excluding the rules* (L-Contraction) *and* (L-Expansion)*. Then $\Gamma \vdash \{\varphi\}t\{\psi\}$ iff $\Gamma \vdash_0 \{\varphi\}t\{\psi\}$.*

Proof. The 'if' direction is an immediate consequence of the fact that any proof in the system $\vdash_0$ translates immediately into a proof in the system $\vdash$.

The 'only if' direction follows from the following two properties:

1. If $\Gamma, r{:}\chi \vdash_0 \{\varphi\}t\{\psi\}$ and $r \notin \mathrm{res}(t)$ then $\Gamma \vdash_0 \{\varphi\}t\{\psi\}$.
2. If $\Gamma \vdash_0 \{\varphi\}t\{\psi\}$ and Γ, Γ' well-defined then $\Gamma, \Gamma' \vdash_0 \{\varphi\}t\{\psi\}$.

Property (1) is shown by a straightforward induction on the judgement $\vdash_0$. Property (2) is shown similarly, with all cases being straightforward apart from that for (L-Res), which we now show:

Suppose that $\Gamma \vdash_0 \{\varphi \star \chi\}\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}\{\psi \star \chi\}$ because $\Gamma, r{:}\chi \vdash_0 \{\varphi\}[r/w]t\{\psi\}$ for some resource r such that $r \notin \mathrm{dom}(\Gamma)$.

If $r \notin \mathrm{dom}(\Gamma')$ then $\Gamma, r{:}\chi, \Gamma'$ is well-defined and, by induction, $\Gamma, r{:}\chi, \Gamma' \vdash \{\varphi\}[r/w]t\{\psi\}$. According to the rule (L-Res), we may conclude that $\Gamma, \Gamma' \vdash \{\varphi \star \chi\}\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}\{\psi \star \chi\}$, as required.

If $r \in \mathrm{dom}(\Gamma')$, there exists an environment Γ'_0 and formula χ' such that $\Gamma' = \Gamma'_0, r{:}\chi'$. Let r' be a resource such that $r' \neq r$ and $r' \notin \mathrm{dom}(\Gamma, \Gamma'_0)$ — such a resource must exist because environments are finite. By induction, the judgement $\Gamma, r{:}\chi, r'{:}\chi', \Gamma'_0 \vdash \{\varphi\}[r/w]t\{\psi\}$ is derivable. Hence $\Gamma, r'{:}\chi', \Gamma_0 \vdash \{\varphi \star \chi\}\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}\{\psi \star \chi\}$. By Lemma 5.3.7, $(r\ r') \cdot (\Gamma, r'{:}\chi', \Gamma'_0) \vdash \{\varphi \star \chi\}(r\ r') \cdot \texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}\{\psi \star \chi\}$. Since $r, r' \notin \mathrm{dom}(\Gamma, \Gamma'_0)$ and $\mathrm{res}(\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}) \subseteq \mathrm{dom}(\Gamma)$ by Lemma 4.2.1, we may conclude that

$$\Gamma, r{:}\chi', \Gamma'_0 \vdash \{\varphi \star \chi\}\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}\{\psi \star \chi\},$$

as required. □

Key result

Theorem 5.2 (Soundness)**.** *For any closed term t, if $\Gamma \vdash \{\varphi\}t\{\psi\}$ then $\Gamma \models \{\varphi\}t\{\psi\}$.*

We defer the proof of soundness until Section 5.7, when it shall be shown alongside two other important properties that we now proceed to define: that all runs are race-free and do not have any 'faults' such as accessing a non-current location. The following result connects the definition of validity to the execution of processes without interference or ownership.

Corollary 5.3 (Connection)**.** *Let t be a closed term with $\mathrm{res}(t) = \emptyset$ and let $\sigma = (D, L, \emptyset, \emptyset)$ be a consistent marking of state conditions for which $D \models \varphi$. If $\emptyset \models \{\varphi\}t\{\psi\}$ then whenever a terminal marking $(\mathrm{Tc}(t), \sigma')$ is reachable from $(\mathrm{Ic}(t), \sigma)$ in $\mathcal{N}[\![t]\!]$, the resulting heap D' satisfies ψ, where $\sigma' = (D', L', R', N')$.*

Proof. A consequence of soundness and Lemma 5.2.3. □

5.4 Fault-avoidance

When introducing the net semantics, it was mentioned that the logic will ensure that processes, running from suitable initial states, only access current locations. The syntax of the language ensures that processes only access current resources and that they are never blocked when releasing a resource through it already being available. We shall now prove that processes avoid such 'faults', in which we shall say that an event e is *control-enabled* in a marking C of control conditions if there exists a marking C' such that $\mathcal{C}[\![t]\!] : C \xrightarrow{e}\!\!\!\!\rightarrow C'$.

Definition 5.4.1 (Fault)**.** *There is a* fault *in a marking $M = (C, \sigma)$ of the net $\mathcal{N}[\![t]\!]$ if there exists a control-enabled event e in $\mathcal{N}[\![t]\!]$ with ${}^{\mathrm{C}}e = C_1$ and $e^{\mathrm{C}} = C_2$ for some C_1, C_2 such that either:*

1. *there exist D, D' such that $e = \mathsf{act}_{(C_1,C_2)}(D, D')$ and there exists $\ell \in \mathrm{dom}(D)$ with $\mathsf{curr}(\ell) \notin \sigma$,*
2. *there exist ℓ, v, ℓ', v' such that $e = \mathsf{alloc}_{(C_1,C_2)}(\ell, v, \ell', v')$ and $\mathsf{curr}(\ell) \notin \sigma$,*
3. *there exist ℓ, v, ℓ', v' such that $e = \mathsf{dealloc}_{(C_1,C_2)}(\ell, v, \ell')$ and either $\mathsf{curr}(\ell) \notin \sigma$ or $\mathsf{curr}(\ell') \notin \sigma$,*

4. *there exists* r *such that either* $e = \mathsf{acq}_{(C_1,C_2)}(r)$ *or* $e = \mathsf{rel}_{(C_1,C_2)}(r)$ *and* $\mathsf{curr}(r) \notin \sigma$, *or*

5. *there exists* r *such that* $e = \mathsf{rel}_{(C_1,C_2)}(r)$ *and* $r \in \sigma$.

This definition also applies to markings (C, σ, W) of $\mathcal{W}[\![t]\!]_\Gamma$ by ignoring the marking of ownership conditions W and considering synchronized events $e \cdot u$.

Proposition 5.4 (Fault-avoidance)**.** *Suppose that* $\Gamma \vdash \{\varphi\}t\{\psi\}$ *and that the initial marking* $(\mathrm{Ic}(t), \sigma_0, W_0)$ *satisfies* φ *in* Γ. *If* (C, σ, W) *is reachable from* $(\mathrm{Ic}(t), \sigma_0, W_0)$ *then* (C, σ, W) *is fault-free.*

Proof. Deferred until Section 5.7. □

From this result and Lemma 5.2.3, clearly if $\emptyset \vdash \{\varphi\}t\{\psi\}$ then no fault is reachable from an initial marking of $\mathcal{N}[\![t]\!]$ if the heap initially satisfies φ.

5.5 Separation

As mentioned in the introduction, the logic discriminates between the parallel composition of processes and their interleaved expansion. In Brookes' trace semantics [Bro07], this was accounted for by making the notion of a *race* primitive within the semantics: when forming the parallel composition of processes, if two processes concurrently write to the same location, a special 'race' action occurs and the trace proceeds no further. The net model retains information on the concurrency of actions, so our approach when defining the semantics can be been different; we shall not regard a race as 'catastrophic' and do not embellish our semantics with special race states. Instead, we shall prove, using the semantics directly, that races do not occur for proved processes running from suitable initial states. This is a form of 'well-typed programs do not go wrong', only here we mean well-*proved* programs.

Generally, a race can be said to occur when two interacting heap actions occur concurrently. Recall that a heap action is represented in the net semantics by a set of events, with common pre- and post-control conditions, representing each way in which the action can affect the heap. According to the net model, two actions might be scheduled to run concurrently if their events do not overlap on their pre- or post-control conditions. In such a situation, where ${}^\bullet e_1{}^\bullet \cap {}^\bullet e_2{}^\bullet = \emptyset$, we shall say that e_1 and e_2 are *control-independent*.

One way of capturing the race freedom of a process running from an initial state is to show that there is no reachable marking in the net where two control-independent events are control-enabled but access a common heap location. We, however, shall prove a result based on the *behaviour* of processes; that whenever two events are control-independent and can occur, then they are independent overall, and in particular on the shared state. In fact, the situation is more subtle since, for example, two concurrent processes might be ready to allocate a new heap location. In this situation, the two processes cannot allocate the same location, and this manifests itself in the race-freedom property.

Definition 5.5.1 (Separation of synchronized events)**.** *Let* M *be a marking of* $\mathcal{W}[\![t]\!]_\Gamma$ *and let* $s_1 = e_1 \cdot u_1$ *and* $s_2 = e_2 \cdot u_2$ *be control-independent synchronized events of* $\mathcal{W}[\![t]\!]_\Gamma$. *The* separation property *of* s_1 *and* s_2 *at* M *is defined as:*

1. *If* $M \xrightarrow{s_1}\!\!\!\!\rightarrow M_1$ *and* $M \xrightarrow{s_2}\!\!\!\!\rightarrow M_2$ *and* s_1 *and* s_2 *are not independent then either:*

- *s_1 and s_2 compete to allocate the same location:*
  $e_1 = \mathsf{alloc}_{(C_1,C_1')}(\ell, v, \ell', v')$ *and* $e_2 = \mathsf{alloc}_{(C_2,C_2')}(k, w, \ell', w')$ *for some* $\ell, \ell', k, v, v', w'$*;*
- *s_1 and s_2 compete to make the same resource current:*
  $e_1 = \mathsf{decl}_{(C_1,C_1')}(r)$ *and* $e_2 = \mathsf{decl}_{(C_2,C_2')}(r)$ *for some r; or*
- *s_1 and s_2 compete to acquire the same resource:*
  $e_1 = \mathsf{acq}_{(C_1,C_1')}(r)$ *and* $e_2 = \mathsf{acq}_{(C_2,C_2')}(r)$ *for some r.*

2. *If $M \xrightarrow{s_1} \!\!\!\!\rightarrow\, M_1 \xrightarrow{s_2} \!\!\!\!\rightarrow\, M_2$ and s_1 and s_2 are not independent then either:*

- *s_1 deallocates a location that s_2 allocates:*
  $e_1 = \mathsf{dealloc}_{(C_1,C_2)}(\ell, v, \ell', v')$ *and* $e_2 = \mathsf{alloc}_{(C_2,C_2')}(k, w, \ell', w')$ *for some* $\ell, \ell', k, v, v', w'$*;*
- *s_1 makes a resource non-current that s_2 makes current:*
  $e_1 = \mathsf{end}_{(C_1,C_1')}(r)$ *and* $e_2 = \mathsf{decl}_{(C_2,C_2')}(r)$ *for some r; or*
- *s_1 releases a resource that s_2 takes:*
  $e_1 = \mathsf{rel}_{(C_1,C_1')}(r)$ *and* $e_2 = \mathsf{acq}_{(C_2,C_2')}(r)$ *for some r.*

3. *The symmetric statement for $M \xrightarrow{s_2} \!\!\!\!\rightarrow\, M_2 \xrightarrow{s_1} \!\!\!\!\rightarrow\, M_1$.*

This is a formulation of Dijkstra's principle of separation.

The first part of the property above tells us how the enabled events of parallel processes *conflict* with each other in a state: the way in which one parallel process can prevent the other acting in a particular way on the global state. The second part dictates how the event occurrences of parallel processes *causally depend* on each other: the way in which the ability of one process to affect the global state in a particular way is dependent on events of the other process.

Importantly, whenever the two events s_1 and s_2 arise from heap actions, they neither conflict nor causally depend on each other. This is our net analogue of race freedom. Theorem 5.5 shows that processes proved by the logic are race-free when running from suitable initial states. We shall make use of the following rather technical lemmas in the proof.

Firstly, in Lemma 5.3.8, we showed that validity was unaffected by switching resources round in an environment and a term. An analogous result for the separation property holding in any reachable marking also holds, proved by considering the same bijection as used in the proof of Lemma 5.3.8.

Lemma 5.5.1 (Equivariance of reachability and the separation property)**.** *For any environment Γ, term t, markings M and M', sequence of events π and resources r, r':*

$$\mathcal{W}\,[\![t]\!]_\Gamma : M_0 \xrightarrow{\pi} \!\!\!\!\rightarrow\, M \quad \textit{iff} \quad \mathcal{W}\,[\![(r\ r') \cdot t]\!]_{(r\ r')\cdot\Gamma} : (r\ r') \cdot M_0 \xrightarrow{(r\ r')\cdot\pi} \!\!\!\!\rightarrow\, (r\ r') \cdot M$$

Furthermore, M satisfies the separation property iff $(r\ r') \cdot M$ satisfies the separation property.

For a synchronized event s and an interference event u, define the separation property for s and u at M similarly to that above, recalling that any synchronized event is trivially control-independent from any interference event because ${}^{\mathbf{c}}u^{\mathbf{c}} = \emptyset$ for any interference event u. It is always the case that a synchronized event and an interference event satisfy the separation property in any consistent marking.

Lemma 5.5.2. *If M is a consistent marking of $\mathcal{W}\,[\![t]\!]_\Gamma$ and s is a synchronized event and u is an interference event then s and u satisfy the separation property in M.*

Proof. A simple case analysis of the events s and u. □

The following lemma relates independence from an interference event to independence from any corresponding synchronized event and will be used in the proof of the separation property for parallel compositions. Recall that we write eIe' if e and e' are independent.

Lemma 5.5.3. *Let s be any synchronized event of $\mathcal{W}[\![t]\!]_\Gamma$ and u be an interference event of $\mathcal{W}[\![t]\!]_\Gamma$. Suppose that M is a consistent marking in which they both have concession. If e_1 is an event of $\mathcal{N}[\![t]\!]$ that synchronizes with u and sIu and s is control-independent from e_1 then $sI(e_1 \cdot u)$.*

Proof. It is easy to see that the preconditions of $e_1 \cdot u$ are simply the preconditions of u along with the pre-control conditions of e_1 apart from replacing $\omega_{\mathsf{oth}}(\ell)$ with $\omega_{\mathsf{proc}}(\ell)$ and replacing $\omega_{\mathsf{oth}}(r)$ with $\omega_{\mathsf{proc}}(r)$. The postconditions of $e_1 \cdot u$ are obtained similarly.

Suppose, for contradiction, that $\neg(sI(e_1 \cdot u))$. Since sIu and s is control-independent of e_1, it follows that there must exist $z \in \mathrm{Loc} \cup \mathrm{Res}$ such that $\omega_{\mathsf{proc}}(z) \in {}^\bullet s^\bullet \cap {}^\bullet(e_1 \cdot u)^\bullet$. From the definition of synchronization, we therefore have $\omega_{\mathsf{oth}}(z) \in {}^\bullet u^\bullet$. The proof is completed by analysis of the cases for how $\omega_{\mathsf{proc}}(z) \in {}^\bullet s^\bullet$; we shall show only one illustrative case, that where z is a location ℓ such that $\omega_{\mathsf{proc}}(\ell) \in {}^\bullet s$ but $\omega_{\mathsf{proc}}(\ell) \notin s^\bullet$.

In this case, the event s must either deallocate the location ℓ or must release a resource r with $r \in \mathrm{dom}(\Gamma)$ and ℓ forms part of the heap used to satisfy the invariant for r. As the event s has concession in M, we have $\omega_{\mathsf{proc}}(\ell) \in M$. By assumption, u has concession in M and $\omega_{\mathsf{oth}}(\ell) \in {}^\bullet u^\bullet$. We cannot have $\omega_{\mathsf{oth}}(\ell) \in {}^\bullet u$ since $\omega_{\mathsf{proc}}(\ell) \in M$, so $\omega_{\mathsf{oth}}(\ell) \in u^\bullet$. Therefore, the event u is an interference event that either allocates the location ℓ or acquires an open resource r and ℓ is part of the heap that satisfies the invariant for r. If u is an event that acquires r, it must be the case that $\omega_{\mathsf{inv}}(\ell) \in {}^\bullet u$ so $\omega_{\mathsf{inv}}(\ell) \in M$, contradicting that M is a consistent marking with $\omega_{\mathsf{proc}}(\ell) \in M$. Consequently, u must in fact be an event that allocates the location ℓ, so therefore $\mathsf{curr}(\ell) \notin M$. We then arrive at another contradiction since it must then be the case that $\omega_{\mathsf{proc}}(\ell) \notin M$ because M is consistent. □

We may now show that the separation property does indeed hold for any two events s_1 and s_2 in $\mathcal{W}[\![t]\!]_\Gamma$ for any term t and environment Γ such that $\Gamma \vdash \{\varphi\}t\{\psi\}$ in any marking $M = (C, \sigma, W)$ reachable from an initial marking of t that satisfies φ in Γ. The proof is most interesting in the case where $t = t_1 \parallel t_2$ and s_1 is an event of t_1 and s_2 is an event of t_2. The case proceeds by establishing, as in Theorem 5.2, that there exists an ownership split W_1 and W_2 of W for which s_1 has concession in (C_1, σ, W_1), where C_1 is the marking of control conditions in C for t_1, and there exist e_2 and u_2 such that $s_2 = (\mathsf{par}\,2{:}e_2) \cdot u_2$ and u_2 also has concession in the marking (C_1, σ, W_1) of $\mathcal{W}[\![t_1]\!]_\Gamma$. By Lemma 5.5.2, the separation property therefore holds for s_1 and u_2 in the marking (C_1, σ, W_1). It follows that the separation property holds for s_1 and s_2 in M since, by Lemma 5.5.3, if the events s_1 and u_2 are independent then so are s_1 and s_2.

Theorem 5.5 (Separation)**.** *Suppose that $\Gamma \vdash \{\varphi\}t\{\psi\}$ and that $(\mathrm{Ic}(t), \sigma_0, W_0)$ satisfies φ in Γ. For any events s_1 and s_2 in $\mathcal{W}[\![t]\!]_\Gamma$ and any marking (C, σ, W) reachable from $(\mathrm{Ic}(t), \sigma_0, W_0)$, the separation property holds for s_1 and s_2 at (C, σ, W).*

Proof. Deferred until Section 5.7. □

The result can be applied, using Lemma 5.2.3 and the observation that $e_1 \cdot u_1 I e_2 \cdot u_2$ implies that $e_1 I e_2$, to obtain a similar result for the net semantics of terms without ownership.

Corollary 5.6. *Let t be a closed term. Suppose that $\emptyset \vdash \{\varphi\}t\{\psi\}$ and that $\sigma_0 = (D_0, L_0, \emptyset, \emptyset)$ is a state for which $D_0 \models \varphi$. If M is a marking reachable from $(\mathrm{Ic}(t), \sigma_0)$ in $\mathcal{N}[\![t]\!]$ and e_1 and e_2 are control-independent events then:*

- *If $M \xrightarrow{e_1} M_1 \xrightarrow{e_2} M'$ then either e_1 and e_2 are independent or e_1 releases a resource or a location that e_2 correspondingly takes or allocates, or e_1 makes non-current a resource that e_2 makes current.*
- *If $M \xrightarrow{e_1} M_1$ and $M \xrightarrow{e_2} M_2$ then either e_1 and e_2 are independent or e_1 and e_2 compete either to make current the same resource, acquire the same resource or to allocate the same location.*

5.6 Incompleteness

The separation result highlights an important form of possible interaction between concurrent processes.[1] Observe that, although there is neither conflict nor causal dependence arising from heap events (and hence the processes are race-free in the sense of Brookes), there may be interaction through the occurrence of allocation and deallocation events. It is therefore possible to give judgements for parallel processes that interact without using critical regions. Suppose, for example, that we have a heap

$$D = \{\ell_0 \mapsto \ell_1, \ell_1 \mapsto 1, \ell_2 \mapsto 2, \ell_3 \mapsto 3, \ell_4 \mapsto 4\}.$$

For any processes t_1 and t_2 such that t_1 does not deallocate ℓ_1, if we place the process

$$t_1; \mathtt{dealloc}(\ell_0)$$

in parallel with

$$\mathtt{alloc}(\ell_2); \quad \mathtt{while}\ \ell_2 \neq \&\ell_1\ \mathtt{do}\ \mathtt{alloc}(\ell_2)\ \mathtt{od}; \quad t_2,$$

the process t_2 can only take place once t_1 has terminated. This arises from the fact that the loop in the second process will only exit when location ℓ_1 is allocated by the command $\mathtt{alloc}(\ell_2)$; this can only occur once $\mathtt{dealloc}(\ell_0)$ makes ℓ_1 non-current and therefore available for allocation by $\mathtt{alloc}(\ell_2)$. Denote this process $\mathrm{seq}(t_1, t_2)$.

We can use this to show that concurrent separation logic is incomplete with respect to our definition of validity: Let t_1 be the assignment $\ell_3 := 1$ and t_2 be $\ell_3 := 2$. Define the formula

$$\delta \triangleq \ell_0 \mapsto \ell_1 \star \ell_1 \mapsto \_ \star \ell_2 \mapsto \_ \star \ell_3 \mapsto \_ \star \ell_4 \mapsto \_.$$

We have $\emptyset \models \{\delta\}\mathrm{seq}(t_1, t_2)\{\ell_3 \mapsto 2 \star \top\}$ since, whenever $\mathrm{seq}(t_1, t_2)$ terminates, the assignment $\ell_3 := 2$ always occurs after the assignment $\ell_3 := 1$. The separation property holds in any marking reachable from any heap initially satisfying δ. Since the location ℓ_3 is assigned-to on both sides of the parallel composition, it can be seen that there is no way to derive

$$\emptyset \vdash \{\delta\}\mathrm{seq}(t_1, t_2)\{\ell_3 \mapsto 2 \star \top\},$$

so the logic is incomplete, even for processes satisfying the separation property.

Though the example above satisfies the separation property, both components can assign a value to ℓ_3 — the process is daringly-concurrent without there being any critical regions in the process. One might wonder whether the separation property can be

[1] Thanks to Peter O'Hearn for suggesting this example.

strengthened to obtain completeness. In particular, does strengthening it to account for ownership of heap locations yield completeness? Unfortunately, the answer in this case appears to be negative: there are examples of incompleteness where neither process accesses a common heap location along any run. Let

$$\begin{array}{rcl} t_1' & = & \texttt{alloc}(\ell_3); \texttt{while}\ \ell_3 \neq \&\ell_5\ \texttt{do}\ \texttt{alloc}(\ell_3)\ \texttt{od} \\ t_2' & = & \texttt{alloc}(\ell_4); (\ell_4 = \&\ell_5).\texttt{skip} + (\ell_4 \neq \ell_5).\texttt{diverge}, \end{array}$$

for the previous definition of `diverge` and the obvious definition of `skip`. Since the location ℓ_5 is always current following termination of t_1' from D, process t_2' always diverges. We have

$$\emptyset \models \{\delta\}\mathrm{seq}(t_1', t_2')\{\bot\}.$$

However, there are no δ_1, δ_2 such that δ is logically equivalent to $\delta_1 \star \delta_2$ and $\emptyset \models \{\delta_2\}t_2'\{\bot\}$, which would be necessary if it were possible to prove $\emptyset \vdash \{\delta\}\mathrm{seq}(t_1', t_2')\{\bot\}$.

Instead, this form of incompleteness seems to stem from allowing processes that are sensitive to the particular locations to be allocated. Since any form of completeness result would be highly intricate, we shall not proceed further with this at present.

5.7 Proof of soundness, separation and fault-avoidance

It is convenient to prove soundness of the logic, the separation property and that proved processes avoid faults all at the same time. To give the proof, we must have a compositional understanding of the runs of $\mathcal{W}[\![t]\!]_\Gamma$. This arises from the earlier results on runs of $\mathcal{C}[\![t]\!]$, using the following lemma — extending Lemma 3.4.1 to ownership nets (as defined in Definition 5.2.3 on page 82).

Lemma 5.7.1. *For any $C, C', \sigma, \sigma', W$ and W', if $(C, \sigma, W) \xrightarrow{e \cdot u} (C', \sigma', W')$ in $\mathcal{W}[\![t]\!]_\Gamma$ then $C \xrightarrow{e} C'$ in $\mathcal{C}[\![t]\!]$.*

A path π of an ownership net $\mathcal{W}[\![t]\!]_\Gamma$ therefore yields a path $\hat{\pi}$ of the net $\mathcal{C}[\![t]\!]$ obtained by removing all interference events from π and replacing any synchronized event $e \cdot u$ with just e.

In the proof to come, which will we occupy the remainder of this chapter, we use the following notations for synchronized events:

$$\begin{array}{rcl} \overline{\overline{\mathsf{act}}}_{(C,C')}(D_1, D_2) & \triangleq & \mathsf{act}_{(C,C')}(D_1, D_2) \cdot \overline{\mathsf{act}}(D_1, D_2), \\ \overline{\overline{\mathsf{alloc}}}_{(C,C')}(\ell, v, \ell', v') & \triangleq & \mathsf{alloc}_{(C,C')}(\ell, v, \ell', v') \cdot \overline{\mathsf{alloc}}(\ell, v, \ell', v'), \\ \overline{\overline{\mathsf{dealloc}}}_{(C,C')}(\ell, \ell', v') & \triangleq & \mathsf{dealloc}_{(C,C')}(\ell, \ell', v') \cdot \overline{\mathsf{dealloc}}(\ell, \ell', v'), \\ \overline{\overline{\mathsf{decl}}}_{(C,C')}(r) & \triangleq & \mathsf{decl}_{(C,C')}(r) \cdot \overline{\mathsf{decl}}(r) \\ \overline{\overline{\mathsf{end}}}_{(C,C')}(r) & \triangleq & \mathsf{end}_{(C,C')}(r) \cdot \overline{\mathsf{end}}(r) \\ \overline{\overline{\mathsf{acq}}}_{(C,C')}(r) & \triangleq & \mathsf{acq}_{(C,C')}(r) \cdot \overline{\mathsf{acq}}(r) \\ \overline{\overline{\mathsf{rel}}}_{(C,C')}(r) & \triangleq & \mathsf{rel}_{(C,C')}(r) \cdot \overline{\mathsf{rel}}(r) \\ \overline{\overline{\mathsf{acq}}}_{(C,C')}(r, D_0) & \triangleq & \mathsf{acq}_{(C,C')}(r) \cdot \overline{\mathsf{acq}}(r, D_0) \\ \overline{\overline{\mathsf{rel}}}_{(C,C')}(r, D_0) & \triangleq & \mathsf{rel}_{(C,C')}(r) \cdot \overline{\mathsf{rel}}(r, D_0) \end{array}$$

With the notation now defined, we now prove the main theorem.

Theorem 5.7. *Let t be a closed term. If $\Gamma \vdash_0 \{\varphi\}t\{\psi\}$ then for any marking $M = (C, \sigma, W)$ reachable from an initial marking $M_0 = (\mathrm{Ic}(t), \sigma_0, W_0)$ that satisfies φ in Γ:*

- *there is no fault in (C, σ, W),*
- *(C, σ, W) is non-violating,*
- *if $C = \mathrm{Tc}(t)$ then (C, σ, W) satisfies ψ in Γ, and*
- *the separation property holds in (C, σ, W) for any pair of synchronized events.*

Let $\sigma = (D, L, R, N)$ and $\sigma_0 = (D_0, L_0, R_0, N_0)$. The proof will proceed by rule induction on the judgement $\vdash$. Recall that we do not need to consider runs from $(\mathrm{Ic}(t), \sigma_0, W_0)$ that start with interference events according to Lemma 5.3.6.

(L-Nil)

Suppose that $\Gamma \vdash_0 \{\varphi\}\varepsilon\{\varphi\}$. As an easy starting case, the only marking reachable from $(\mathrm{Ic}(\varepsilon), \sigma_0, W_0)$ in $\mathcal{W}[\![\varepsilon]\!]_\Gamma$ by a path that does not start or end with any interference events is, of course, $(\mathrm{Ic}(\varepsilon), \sigma_0, W_0)$. Since $\mathrm{Ev}(\varepsilon) = \emptyset$, the marking is trivially non-violating, fault-free and satisfies the separation property. We have $\mathrm{Tc}(\varepsilon) = \mathrm{Ic}(\varepsilon)$, so the third requirement is also met.

(L-Act)

Suppose that $\Gamma \vdash_0 \{\varphi\}\alpha\{\psi\}$ for some action α because for all D such that $D \models \varphi$ and $(D_1, D_2) \in \mathcal{A}[\![\alpha]\!]$ we have $\mathrm{dom}(D_1) \subseteq \mathrm{dom}(D)$ and if $D_1 \subseteq D$ then $D \setminus D_1 \cup D_2 \models \psi$. The only synchronized events in $\mathcal{W}[\![\alpha]\!]_\Gamma$ are of the form $\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(D_1, D_2)$ where $(D_1, D_2) \in \mathcal{A}[\![\alpha]\!]$, so $\mathrm{dom}(D_1) = \mathrm{dom}(D_2)$. The initial marking M_0 satisfies φ in Γ, so $D_0 \restriction_{W_0} \mathsf{proc} \models \varphi$. The first premise of the rule (L-Act) gives $\mathrm{dom}(D_1) \subseteq \mathrm{dom}(D_0)$, so a fault does not occur in the marking M_0.

To see that M_0 is non-violating, suppose that the event $\mathsf{act}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(D_1, D_2)$ has concession in $\mathcal{N}[\![\alpha]\!]$ in the marking $(\mathrm{Ic}(\alpha), \sigma_0)$. From the rule, we have $\mathrm{dom}(D_1) \subseteq \mathrm{dom}(D)$ for any D such that $D \models \varphi$. It follows that $D_1 \subseteq D_0 \restriction_{W_0} \mathsf{proc}$ since $D_0 \restriction_{W_0} \mathsf{proc} \models \varphi$ because M_0 satisfies φ in Γ. It is easy to see from this and the fact that $\mathrm{dom}(D_1) = \mathrm{dom}(D_2)$ that the event $\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(D_1, D_2)$ has concession in the marking M_0, so it is a non-violating marking.

The marking following $\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(D_1, D_2)$ in M_0 is $M' = (\mathrm{Tc}(\alpha), \sigma', W_0)$ where $\sigma' = (D', L_0, R_0, N_0)$ and $D' = D_0 \setminus D_1 \cup D_2$. In particular, since $D_1 \subseteq D_0 \restriction_{W_0} \mathsf{proc}$ and $\mathrm{dom}(D_1) = \mathrm{dom}(D_2)$, we have $D' \restriction_{W_0} \mathsf{proc} = (D_0 \restriction_{W_0} \mathsf{proc}) \setminus D_1 \cup D_2$ and $D' \restriction_{W_0} \mathsf{inv} = D_0 \restriction_{W_0} \mathsf{inv}$. From the second premise of (L-Act), we have $D' \restriction_{W_0} \mathsf{proc} \models \psi$, so the marking M' satisfies ψ in Γ. The marking M' is fault-free and non-violating because no event is control-enabled in the control marking $\mathrm{Tc}(\alpha)$.

The separation property trivially holds because there are no two synchronized events in $\mathcal{W}[\![\alpha]\!]_\Gamma$ that are control-independent.

(L-Alloc)

Suppose that $\Gamma \vdash_0 \{\ell \mapsto \_\}\mathtt{alloc}(\ell)\{\exists x_{\mathsf{loc}}.(\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_)\}$ for some location ℓ. Assume that M_0 satisfies $\ell \mapsto \_$ in Γ; it follows that $D_0 \restriction_{W_0} \mathsf{proc} = \{\ell \mapsto v_0\}$ for some $v_0 \in \mathrm{Val}$. Now, any synchronized event in $\mathcal{W}[\![\mathtt{alloc}(\ell)]\!]_\Gamma$ is equal to $\overline{\overline{\mathsf{alloc}}}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(\ell, v, \ell', v')$ for some ℓ', v, v'. The marking M_0 is consistent, so $\mathsf{curr}(\ell) \in M_0$ and therefore there is

no fault in M_0. The marking M_0 is not violating since we have $\omega_{\mathsf{proc}}(\ell) \in W_0$ because $D_0 \restriction_{W_0} \mathsf{proc} = \{\ell \mapsto v_0\}$. Observe also that the separation property trivially holds because there are no two control-independent synchronized events.

Suppose that $\overline{\overline{\mathsf{alloc}}}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(\ell, v, \ell', v_1)$ has concession in M_0. It follows that $v = v_0$ and $\mathsf{curr}(\ell') \notin L_0$. As M_0 is consistent, this implies that there is no ownership condition for ℓ' in W_0. The resulting marking M' has:

$$\begin{array}{rcl} R' & = & R_0 \\ N' & = & N_0 \\ C' & = & \mathrm{Tc}(\mathtt{alloc}(\ell)) \\ D' & = & D_0 \setminus \{\ell \mapsto v_0\} \cup \{\ell \mapsto \ell', \ell' \mapsto v_1\} \\ L' & = & L_0 \cup \{\mathsf{curr}(\ell')\} \\ W' & = & W_0 \cup \{\omega_{\mathsf{proc}}(\ell')\}. \end{array}$$

Hence $D' \restriction_{W'} \mathsf{inv} = D_0 \restriction_W \mathsf{inv}$ and $D' \restriction_{W'} \mathsf{proc} = \{\ell \mapsto \ell', \ell' \mapsto v_1\}$. Thus

$$D' \restriction_{W'} \mathsf{proc} \models \exists x_{\mathsf{loc}}.(\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_),$$

and so M' satisfies ψ in Γ as required. Again, the marking M' is fault-free and non-violating because no event has concession on its control conditions in the terminal marking.

(L-Dealloc)

Suppose that $\Gamma \vdash_0 \{\exists x_{\mathsf{loc}}.(\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_)\}\mathtt{dealloc}(\ell)\{\exists x_{\mathsf{loc}}.\ell \mapsto x_{\mathsf{loc}})\}$ for some location ℓ. Assume that M_0 satisfies $\exists x_{\mathsf{loc}}.(\ell \mapsto x_{\mathsf{loc}} \star x_{\mathsf{loc}} \mapsto \_)$ in Γ, so $D_0 \restriction_{W_0} \mathsf{proc} = \{\ell \mapsto \ell_0, \ell_0 \mapsto v_0\}$ for some location ℓ_0 and value v_0. Any synchronized event in $\mathcal{W}[\![\mathtt{dealloc}(\ell)]\!]_\Gamma$ is equal to $\overline{\overline{\mathsf{dealloc}}}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(\ell, \ell', v')$ for some ℓ' and v'. The marking M_0 is consistent, so $\mathsf{curr}(\ell_0) \in M_0$ and therefore there is no fault in M_0. Since we have $\omega_{\mathsf{proc}}(\ell), \omega_{\mathsf{proc}}(\ell') \in W_0$, the marking M_0 is non-violating. Observe also that there are no two control-independent synchronized events in the net so the separation property trivially holds in all markings.

Now, suppose that $\overline{\overline{\mathsf{dealloc}}}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(\ell, \ell', v')$ has concession in M_0. It follows that $\ell' = \ell_0$ and $v' = v_0$. The marking M' following the occurrence of the event in M has

$$\begin{array}{rcl} R' & = & R_0 \\ N' & = & N_0 \\ C' & = & \mathrm{Tc}(\mathtt{dealloc}(\ell)) \\ D' & = & D_0 \setminus \{\ell_0 \mapsto v_0\} \\ L' & = & L_0 \setminus \{\mathsf{curr}(\ell_0)\} \\ W' & = & W_0 \setminus \{\omega_{\mathsf{proc}}(\ell_0)\}. \end{array}$$

Hence $D' \restriction_{W'} \mathsf{inv} = D \restriction_W \mathsf{inv}$ and $D' \restriction_{W'} \mathsf{proc} = \{\ell \mapsto \ell_0\}$. Furthermore, $D' \restriction_{W'} \mathsf{proc} \models \exists x_{\mathsf{loc}}.(\ell \mapsto x_{\mathsf{loc}})$, and so M' satisfies ψ in Γ. As in (L-Alloc), it is trivially the case that M' is fault-free and non-violating because C' is the terminal marking of control conditions.

(L-Seq)

Suppose that $\Gamma \vdash_0 \{\varphi\}t_1; t_2\{\psi\}$ because $\Gamma \vdash_0 \{\varphi\}t_1\{\varphi'\}$ and $\Gamma \vdash_0 \{\varphi'\}t_2\{\psi\}$. Let M be reached in $\mathcal{W}[\![t_1; t_2]\!]_\Gamma$ from $(\mathrm{Ic}(t_1; t_2), \sigma_0, W_0)$ by path π. From Lemma 5.7.1, we have

$\mathcal{C}[\![t_1;t_2]\!]:\mathrm{Ic}(t_1;t_2) \xrightarrow{\hat{\pi}}\!\!\!\!\!\rightarrow C$. Let $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$. There are two cases according to the sequential path lemma, Lemma 3.6.2 (alongside the proof of well-termination, Lemma 3.7.1):

1. $C \neq P$ and there exist C_1 and π_1' such that $\hat{\pi} = P \triangleleft \mathsf{seq}\,1{:}\pi_1'$ and $C = P \triangleleft \mathsf{seq}\,1{:}C_1$ and $\mathcal{C}[\![t_1]\!]:\mathrm{Ic}(t_1) \xrightarrow{\pi_1}\!\!\!\!\!\rightarrow C_1$

2. there exist C_2, π_1' and π_2' such that $\hat{\pi} = (P \triangleleft \mathsf{seq}\,1{:}\pi_1') \cdot (P \triangleright \mathsf{seq}\,2{:}\pi_2')$ and $C = P \triangleright \mathsf{seq}\,2{:}C_2$ and $\mathcal{C}[\![t_1]\!]:\mathrm{Ic}(t_1) \xrightarrow{\pi_1'}\!\!\!\!\!\rightarrow \mathrm{Tc}(t_1)$ and $\mathcal{C}[\![t_2]\!]:\mathrm{Ic}(t_2) \xrightarrow{\pi_2'}\!\!\!\!\!\rightarrow C_2$.

First suppose that (1) is the case. By induction on the length of π, it can be shown that there must exist a sequence π_1 of events of $\mathcal{W}[\![t_1]\!]_\Gamma$ such that $\pi = P \triangleleft \mathsf{seq}\,1{:}\pi_1$ and $\mathcal{W}[\![t_1]\!]_\Gamma:(\mathrm{Ic}(t_1),\sigma_0,W_0) \xrightarrow{\pi_1}\!\!\!\!\!\rightarrow (C_1,\sigma,W)$.

If M is a violating marking then there must exist an event e of $\mathcal{N}[\![t_1;t_2]\!]$ that has concession in marking (C,σ) but there is no interference event u of $\mathcal{W}[\![t_1;t_2]\!]_\Gamma$ that synchronizes with e such that $e \cdot u$ has concession in M. It follows from Lemma 3.6.1 that there is an event e_1 of $\mathcal{N}[\![t_1]\!]$ such that $e = P \triangleleft \mathsf{seq}\,1{:}e_2$ that has concession in the marking (C_1,σ). The interference events of $\mathcal{W}[\![t_1]\!]_\Gamma$ are precisely the same as those of $\mathcal{W}[\![t_1;t_2]\!]_\Gamma$, so it follows that the marking (C_1,σ,W) is a violating marking of $\mathcal{W}[\![t_1]\!]_\Gamma$. This contradicts the induction hypothesis for the judgement $\Gamma \vdash_0 \{\varphi\}t_1\{\varphi'\}$. It can be shown similarly that there is no fault in the marking M.

It is easy to see that $C \neq \mathrm{Tc}(t_1;t_2)$ because $C \neq P$, so all that remains to show in this case is that the separation property holds. Firstly, suppose that $M \xrightarrow{s}\!\!\!\!\!\rightarrow M' \xrightarrow{s'}\!\!\!\!\!\rightarrow M''$ in $\mathcal{W}[\![t_1;t_2]\!]_\Gamma$ and that s and s' are control-independent synchronized events. It follows from Lemmas 3.6.1 and 3.5.2 that there exist synchronized events s_1 and s_1' of $\mathcal{W}[\![t_1]\!]_\Gamma$ and markings M_1' and M_1'' with $(C_1,\sigma,W) \xrightarrow{s_1}\!\!\!\!\!\rightarrow M_1' \xrightarrow{s_1'}\!\!\!\!\!\rightarrow M_1''$ and $s = P \triangleleft \mathsf{seq}\,1{:}s_1$ and $s' = P \triangleleft \mathsf{seq}\,1{:}s_1'$. This follows in particular from Lemma 3.5.2, which ensures that s' cannot correspond to an event in $\mathcal{W}[\![t_2]\!]_\Gamma$ because then there would exist a condition inside P that would cause s and s' to fail to be control-independent. If s_1 and s_1' are independent, it is easy to see that this is reflected into the independence of s and s'. Otherwise, if s_1 and s_2 are not independent, it is readily seen from the induction hypothesis for the judgement $\Gamma \vdash_0 \{\varphi\}t_1\{\varphi'\}$ and Definition 5.5.1 that s_1 and s_1' must be one of the distinguished kinds of causally dependent events (*e.g.* s_1' allocates a location that s_1 deallocates), and so s and s' also follow the same pattern. Hence the second and third requirements for the separation of M according to Definition 5.5.1 are met. The other requirements for the separation property to hold for all pairs of synchronized events at M are proved similarly.

Now suppose that the (2) is the case. Let s_2 be the synchronized event in π corresponding to the first event of π_2' unless π_2' is empty. By induction on the length of π up to, but not including s_2 (or the length of π if π_2' is empty), it can be shown that there exist a path π_1' and markings σ' and W' such that $\mathcal{W}[\![t_1]\!]_\Gamma:(\mathrm{Ic}(t_1),\sigma_0,W_0) \xrightarrow{\pi_1'}\!\!\!\!\!\rightarrow (\mathrm{Tc}(t_1),\sigma',W')$. By induction on the length of π from s_2, it can be show that there exists a path π_2' such that $\mathcal{W}[\![t_2]\!]_\Gamma:(\mathrm{Ic}(t_2),\sigma',W') \xrightarrow{\pi_2'}\!\!\!\!\!\rightarrow (C_2,\sigma,W)$.

The induction hypothesis for judgement $\Gamma \vdash_0 \{\varphi\}t_1\{\varphi'\}$ ensures that the marking $(\mathrm{Tc}(t_1),\sigma',W')$ satisfies φ' in Γ. The arguments for M being non-violating, fault-free and satisfying the separation property follow the arguments above, being a little simpler because it follows immediately from Lemma 3.6.1 that any two events occurring sequentially from M both correspond to events in $\mathcal{W}[\![t_2]\!]_\Gamma$. It follows from Lemma 3.6.1 that if

$C = \mathrm{Tc}(t_1; t_2)$ then $C_2 = \mathrm{Tc}(t_2)$. Hence, from the induction hypothesis for the judgement $\Gamma \vdash_0 \{\varphi'\}t_2\{\psi\}$, it must be the case that M satisfies ψ in Γ, as required.

(L-Sum)

Suppose that $\Gamma \vdash_0 \{\varphi\}\alpha_1.t_1 + \alpha_2.t_2\{\psi\}$ because $\Gamma \vdash_0 \{\varphi\}\alpha_i\{\varphi_i\}$ and $\Gamma \vdash_0 \{\varphi_i\}t_i\{\psi\}$ for $i = 1, 2$. In the manner of the previous case, this time using Lemma 3.6.7 which characterizes runs of $\mathcal{C}[\![\alpha_1.t_1 + \alpha_2.t_2]\!]$, it can be shown that there are three distinct cases for the run

$$\mathcal{W}[\![\alpha_1.t_1 + \alpha_2.t_2]\!]_\Gamma : (\mathrm{Ic}(\alpha_1.t_1 + \alpha_2.t_2), \sigma_0, W_0) \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} (C, \sigma, W)$$

Let P be the set of conditions where the nets for t_1 and t_2 are joined, $P = \mathsf{sum}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{sum}\,2{:}\mathrm{Tc}(t_2)$. The first case is that the path π is empty. In this case, $C = \mathrm{Ic}(\alpha_1.t_1 + \alpha_2.t_2) = \{\mathsf{i}\}$. From the definition of the events of the net, it is easy to see that the only events with concession on their control conditions are either of the form $\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\}, P \lhd \mathsf{sum}\,1:\mathrm{Ic}(t_1))}(D_1, D_1')$ for $(D_1, D_1') \in \mathcal{A}[\![\alpha_1]\!]$ or $\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\}, P \lhd \mathsf{sum}\,2:\mathrm{Ic}(t_2))}(D_2, D_2')$ for $(D_2, D_2') \in \mathcal{A}[\![\alpha_2]\!]$. It is easy to see from the induction hypotheses for the judgements $\Gamma \vdash_0 \{\varphi\}\alpha_1\{\varphi_1\}$ and $\Gamma \vdash_0 \{\varphi\}\alpha_2\{\varphi_2\}$ that the marking (C, σ_0, W_0) cannot be violating and is fault-free. The marking of control conditions C cannot possibly equal $\mathrm{Tc}(\alpha_1.t_1 + \alpha_2.t_2)$, so all that remains is to show that the separation property holds. Firstly, no two events that can occur in the initial marking are control-independent. Secondly, from Lemma 3.6.7 and Lemma 3.5.2, if s is an event such that in the net $\mathcal{W}[\![\alpha_1.t_1 + \alpha_2.t_2]\!]_\Gamma$ there is a marking M' such that

$$(\mathrm{Ic}(\alpha_1.t_1 + \alpha_2.t_2), \sigma_0, W_0) \stackrel{\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\}, P \lhd \mathsf{sum}\,1:\mathrm{Ic}(t_1))}(D_1, D_1')}{\longrightarrow\!\!\!\!\!\rightarrow} (C', \sigma', W') \stackrel{s}{\longrightarrow\!\!\!\!\!\rightarrow} M',$$

then ${}^{\mathrm{c}}s \cap P \lhd \mathsf{sum}\,1{:}\mathrm{Ic}(t_1) \neq \emptyset$. Hence s and $\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\}, P \lhd \mathsf{sum}\,1:\mathrm{Ic}(t_1))}(D_1, D_1')$ are not control-independent, so (with the symmetric argument for t_2) the separation property holds.

The second case (symmetric to the third for t_2) for the path π is that it is equal to $\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\}, P \lhd \mathsf{sum}\,1:\mathrm{Ic}(t_1))}(D_1, D_1) \cdot (P \lhd \mathsf{sum}\,1{:}\pi_1)$ for some path π_1. In this case, there exist σ_1, W_1 and C_1 such that $C = P \lhd \mathsf{sum}\,1{:}C_1$,

$$\begin{array}{rcl} \mathcal{W}[\![\alpha_1]\!]_\Gamma & : & (\mathrm{Ic}(\alpha_1), \sigma_0, W_0) \stackrel{\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\}, \{\mathsf{t}\})}(D_1, D_1')}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathrm{Tc}(\alpha_1), \sigma_1, W_1), \quad \text{and} \\ \mathcal{W}[\![t_1]\!]_\Gamma & : & (\mathrm{Ic}(t_1), \sigma_1, W_1) \stackrel{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} (C_1, \sigma, W). \end{array}$$

From the induction hypothesis for $\Gamma \vdash_0 \{\varphi\}\alpha_1\{\varphi_1\}$, the marking $(\mathrm{Ic}(t_1), \sigma_0, W_0)$ satisfies φ_1 in Γ. From the induction hypothesis for $\Gamma \vdash_0 \{\varphi_1\}t_1\{\psi\}$, the marking (C_1, σ, W) of $\mathcal{W}[\![t_1]\!]_\Gamma$ is fault-free, non-violating and satisfies the separation property. It is easy to see from this, using Lemma 3.6.7, that the marking (C, σ, W) is a fault-free, non-violating marking of $\mathcal{W}[\![(\alpha_1.t_1 + \alpha_2.t_2)]\!]_\Gamma$ that also satisfies the separation property. We have $C = \mathrm{Tc}(\alpha_1.t_1 + \alpha_2.t_2)$ iff $C_1 = \mathrm{Tc}(t_1)$ by Lemma 3.3.1. If $C_1 = \mathrm{Tc}(t_1)$ then (C_1, σ, W) satisfies ψ in Γ according to the induction hypothesis. It follows immediately that if $C = \mathrm{Tc}(\alpha_1.t_1 + \alpha_2.t_2)$ then (C, σ, W) satisfies ψ in Γ, as required to complete the case.

(L-While)

Let $t = \texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od}$. Suppose that we have $\Gamma \vdash_0 \{\varphi\}t\{\psi\}$ because:

$$\Gamma \vdash_0 \{\varphi\}b\{\varphi'\} \qquad \Gamma \vdash_0 \{\varphi\}\neg b\{\psi\} \qquad \Gamma \vdash_0 \{\varphi'\}t_0\{\varphi\}.$$

Assume that the marking M is reachable in $\mathcal{W}[\![\texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od}]\!]_\Gamma$. Recalling that $(\mathrm{Ic}(t), \sigma_0, W_0)$ is the initial marking of $\mathcal{W}[\![t]\!]_\Gamma$ being considered, as above, but now using Lemma 3.6.9, it can be shown that there exists a natural number $n \geq 0$ and $\sigma_1, \ldots, \sigma_n$ and $W_1, \ldots, W_n$ such that, for all $0 \leq i < n$:

$$\begin{array}{lcl} \mathcal{W}[\![b]\!]_\Gamma & : & (\mathrm{Ic}(b), \sigma_i, W_i) \longrightarrow\!\!\!\!\!\rightarrow (\mathrm{Tc}(b), \sigma_i, W_i) \\ \mathcal{W}[\![t_0]\!]_\Gamma & : & (\mathrm{Ic}(t_0), \sigma_i, W_i) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathrm{Tc}(t_0), \sigma_{i+1}, W_{i+1}) \end{array}$$

Furthermore, either:

1. the loop is in its initial control state: $C = \mathsf{body}{:}\mathrm{Tc}(t_0)$ and $\sigma = \sigma_n$ and $W = W_n$,
2. the loop has exited: $C = \{\mathsf{t}\}$ and

   $$\mathcal{W}[\![\neg b]\!]_\Gamma : (\mathrm{Ic}(\neg b), \sigma_n, W_n) \longrightarrow\!\!\!\!\!\rightarrow (\mathrm{Tc}(\neg b), \sigma, W),$$

   or
3. the body of the loop is being executed: $C = \mathsf{body}{:}C_0$ for some marking $C_0 \neq \mathrm{Tc}(t_0)$ of $\mathcal{W}[\![t_0]\!]_\Gamma$ with

   $$\begin{array}{lcl} \mathcal{W}[\![b]\!]_\Gamma & : & (\mathrm{Ic}(b), \sigma_n, W_n) \longrightarrow\!\!\!\!\!\rightarrow (\mathrm{Tc}(b), \sigma_n, W_n) \\ \mathcal{W}[\![t_0]\!]_\Gamma & : & (\mathrm{Ic}(t_0), \sigma_n, W_n) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} (C_0, \sigma, W) \end{array}$$

By induction on n, applying the induction hypotheses for $\Gamma \vdash_0 \{\varphi\}b\{\varphi'\}$ and $\Gamma \vdash_0 \{\varphi'\}t_0\{\varphi\}$, we may infer that σ_n with W_n satisfies φ in Γ. We now consider the cases (1), (2) and (3) separately:

1. If $C = \mathsf{body}{:}\mathrm{Tc}(t_0)$, we have $C \neq \mathrm{Tc}(t)$. We first show that the marking $M = (C, \sigma, W)$ is non-violating. Assume that the boolean b holds in $\sigma_n = \sigma$; the case for $\neg b$ is similar. According to Lemma 3.6.9, the event $\mathsf{act}_{(\mathsf{body}:\mathrm{Tc}(t_0), \mathsf{body}:\mathrm{Ic}(t_0))}(D_1, D_2)$ that has concession in marking (C, σ) of $\mathcal{N}[\![t]\!]$ but $\overline{\overline{\mathsf{act}}}_{(\mathsf{body}:\mathrm{Tc}(t_0), \mathsf{body}:\mathrm{Ic}(t_0))}(D_1, D_2)$ does not have concession in M. It follows that in $\mathcal{W}[\![b]\!]_\Gamma$ the event $\overline{\overline{\mathsf{act}}}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(D_1, D_2)$ does not have concession in marking $(\mathrm{Ic}(b), \sigma, W)$ but $\mathsf{act}_{(\{\mathsf{i}\},\{\mathsf{t}\})}(D_1, D_2)$ does have concession in marking $(\mathrm{Ic}(b), \sigma)$. Now, as the marking $(\mathrm{Ic}(b), \sigma, W)$ satisfies φ in Γ, we may conclude that a violating marking is immediately reached in $\mathcal{W}[\![b]\!]_\Gamma$, contradicting the induction hypothesis for $\Gamma \vdash_0 \{\varphi\}b\{\varphi'\}$. It can be shown similarly that the marking M is fault-free.

   All that remains is to show the separation property. It follows from Lemmas 5.2.2 and 3.6.9 that if s_1 and s_2 are synchronized events such that $M \stackrel{s_1}{\longrightarrow\!\!\!\!\!\rightarrow} M_1$ and $M \stackrel{s_2}{\longrightarrow\!\!\!\!\!\rightarrow} M_2$ then s_1 and s_2 are tests of the boolean b. Hence s_1 and s_2 are not control-independent. Furthermore, if $M \stackrel{s_1}{\longrightarrow\!\!\!\!\!\rightarrow} M_1 \stackrel{s_2}{\longrightarrow\!\!\!\!\!\rightarrow} M_2$ for synchronized events s_1 and s_2 then by Lemmas 3.6.9 and 3.5.2 we have $s_1 = \overline{\overline{\mathsf{act}}}_{(\mathsf{body}:\mathrm{Tc}(t_0), \mathsf{body}:\mathrm{Ic}(t_0))}(D_1, D_2)$ for some $(D_1, D_2) \in \mathcal{A}[\![b]\!]$ and $s_2 = \mathsf{body}{:}s_0$ for some event s_0 that can occur from the initial control marking $\mathrm{Ic}(t_0)$ of $\mathcal{W}[\![t_0]\!]_\Gamma$. By Lemma 3.5.2, there is a control condition b such that $b \in {}^{\mathrm{C}}s_0$ and $b \in \mathrm{Ic}(t_0)$. It follows that $\mathsf{body}{:}b \in {s_1}^\bullet \cap {}^\bullet s_2$, and therefore the events are not control-independent.
2. If $C = \{\mathsf{t}\}$, we have $C = \mathrm{Tc}(t)$. By Lemmas 5.2.2 and 3.6.9, no event has concession on its control conditions in this marking, so all that we must check is that the

marking M satisfies ψ in Γ. We know that $(\mathsf{body}{:}\mathrm{Tc}(t_0), \sigma_n, W_n)$ satisfies φ in Γ, and therefore $(\mathrm{Ic}(\neg b), \sigma_n, W_n)$ also does. Now,

$$\mathcal{W}[\![\neg b]\!]_\Gamma : (\mathrm{Ic}(\neg b), \sigma_n, W_n) \longrightarrow\!\!\!\!\!\rightarrow (\mathrm{Tc}(\neg b), \sigma_n, W_n)$$

by (2). The induction hypothesis for $\Gamma \vdash_0 \{\varphi\}\neg b\{\psi\}$ allows us to infer that the marking $(\mathrm{Tc}(\neg b), \sigma, W)$ satisfies ψ in Γ. Hence M also satisfies ψ in Γ since $\sigma = \sigma_n$ and $W = W_n$.

3. We have $C = \mathsf{body}{:}C_0$ where C_0 is a marking of $\mathcal{W}[\![t_0]\!]_\Gamma$ not equal to $\mathrm{Tc}(t_0)$. We know that $(\mathrm{Ic}(b), \sigma_n, W_n)$ satisfies φ in Γ, so, because

$$\mathcal{W}[\![b]\!]_\Gamma : (\mathrm{Ic}(b), \sigma_n, W_n) \longrightarrow\!\!\!\!\!\rightarrow (\mathrm{Tc}(b), \sigma_n, W_n),$$

from the induction hypothesis for $\Gamma \vdash_0 \{\varphi\}b\{\varphi'\}$ we may infer that $(\mathrm{Ic}(t_0), W_n, \sigma_n)$ satisfies φ' in Γ.

Suppose, for contradiction, that the marking $M = (C, \sigma, W)$ is violating. By definition, there is an event e that has concession in marking (C, σ) in $\mathcal{N}[\![t]\!]$ but there is no interference event u such that $e \cdot u$ has concession in M. Since e has concession in (C, σ), by Lemmas 3.6.9 and 5.2.2 there is an event e_0 with $e = \mathsf{body}{:}e_0$ which has concession in the marking (C_0, σ) of $\mathcal{N}[\![t_0]\!]$. By assumption, the marking (C_0, σ, W) is reachable from $(\mathrm{Ic}(t_0), W_n, \sigma_n)$ in $\mathcal{W}[\![t_0]\!]_\Gamma$. From the induction hypothesis, (C_0, σ, W) is non-violating, so there exists an interference event u' such that $e_0 \cdot u'$ has concession in (C_0, σ, W). Now, the interference events of $\mathcal{W}[\![t_0]\!]_\Gamma$ are equal to the interference events of $\mathcal{W}[\![t]\!]_\Gamma$, so u' is an interference event of $\mathcal{W}[\![t]\!]_\Gamma$. The events e and e_0 differ only on their control conditions and e has concession on its control conditions in M, so the event $e \cdot u'$ is present in the net $\mathcal{W}[\![t]\!]_\Gamma$ and moreover has concession in marking M. This contradicts M being a violating marking. It can be shown similarly that M is fault-free.

The separation property on synchronized events of $\mathcal{W}[\![t]\!]_\Gamma$ straightforwardly holds: Let s_1 and s_2 be synchronized events present in the net. If $M \overset{s_1}{\longrightarrow\!\!\!\!\!\rightarrow} M_1$ and $M \overset{s_2}{\longrightarrow\!\!\!\!\!\rightarrow} M_2$ then, using Lemma 3.6.9, we can see that there are events s_1' and s_2' and markings M_1' and M_2' such that $(C_0, \sigma, W) \overset{s_1}{\longrightarrow\!\!\!\!\!\rightarrow} M_1'$ and $(C_0, \sigma, W) \overset{s_2'}{\longrightarrow\!\!\!\!\!\rightarrow} M_2'$ in $\mathcal{W}[\![t_0]\!]_\Gamma$, and furthermore $s_1 = \mathsf{body}{:}s_1'$ and $s_2 = \mathsf{body}{:}s_2'$. If s_1 and s_2 are control-independent, it is clear to see that so are s_1' and s_2'. If s_1' and s_2' are independent then so are s_1 and s_2. Hence, if s_1 and s_2 are not independent then neither are s_1' and s_2', so by the induction hypothesis for $\Gamma \vdash_0 \{\varphi'\}t_0\{\varphi\}$ we see that s_1' and s_2' are of the particular forms specified in the third clause of the definition of separation, and therefore so are s_1 and s_2.

Now suppose that $M \overset{s_1}{\longrightarrow\!\!\!\!\!\rightarrow} M_1 \overset{s_2}{\longrightarrow\!\!\!\!\!\rightarrow} M_2$ in $\mathcal{W}[\![t]\!]_\Gamma$ and that s_1 and s_2 are control-independent synchronized events. Let the marking of control conditions in M_1 be C_1. It cannot be that $C_1 = \mathsf{body}{:}\mathrm{Tc}(t_0)$: We have, by assumption, $C \neq \mathsf{body}{:}\mathrm{Tc}(t_0)$ so, if it were, there would exist a control condition b with $b \in {}^{\mathrm{C}}s_1{}^{\mathrm{C}} \cap \mathsf{body}{:}\mathrm{Tc}(t_0)$. By Lemmas 5.2.2 and 3.6.9, s_2 would have to correspond to an action testing the boolean b, so ${}^{\mathrm{C}}s_2 = \mathsf{body}{:}\mathrm{Tc}(t_0)$. This contradicts the assumption that s_1 and s_2 are control independent. The remainder of the case is similar to the preceding argument.

(L-Res)

Let $t = \texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}$ and assume that $\Gamma \vdash_0 \{\varphi \star \chi\}t\{\psi \star \chi\}$ because $\Gamma, r_0{:}\chi \vdash_0 \{\varphi\}[r_0/r]t_0\{\psi\}$ for some $r_0 \notin \mathrm{dom}(\Gamma)$.

Assume that σ_0, W_0 satisfies $\varphi \star \chi$ in Γ. Recall that $\mathrm{Ic}(t) = \{\mathrm{i}\}$, and that σ_0 is equal (D_0, L_0, R_0, N_0). Let π be the path to the marking M from $(\mathrm{Ic}(t), \sigma_0, W_0)$ in $\mathcal{W}[\![t]\!]_\Gamma$. Lemma 5.3.3 on page 88 identifies the three possible cases for the path π:

The first case has $C = \mathrm{Ic}(t)$ and π is empty. The only synchronized events with concession on their control conditions are to declare a resource. It follows immediately that no fault can occur in the marking and that the marking M is non-violating because the marking is consistent. In addition, there are no two control-independent events that may occur either both in marking M or sequentially from marking M, so the separation property is also satisfied.

Now consider the second case, where $C = \mathsf{res}\, r{:}C'$ for some resource r. There exist paths π_0 and π_1 such that:

$$\pi = \pi_0 \cdot s_r \cdot (\mathsf{res}\, r{:}\pi_1)$$

and

$$\begin{aligned} \mathcal{W}[\![t]\!]_\Gamma: &\quad (\mathrm{Ic}(t), \sigma_0, W_0) \overset{\pi_0 \cdot s_r}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{res}\, r{:}\mathrm{Ic}([r/w]t_0), \sigma', W') \quad \text{and} \\ \mathcal{W}[\![[r/w]t_0]\!]_\Gamma: &\quad (\mathrm{Ic}([r/w]t_0), \sigma', W') \overset{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} (C', \sigma, W). \end{aligned}$$

In addition, the path π_0 consists only of interference events.

From Lemma 5.3.4, we have $\omega_{\mathsf{proc}}(r) \in W_1$ for every marking of ownership conditions W_1 reachable from $(\mathrm{Ic}([r/w]t_0), \sigma', W')$ along the path π_1.

The resource r is not in $\mathrm{dom}(\Gamma)$ because the event s_r has concession in a marking, which must be consistent according to Proposition 5.1, that is reachable from $(\mathrm{Ic}(t), \sigma_0, W_0)$. Since the marking $(\mathrm{Ic}(t), \sigma_0, W_0)$ satisfies $\varphi \star \chi$ in Γ and interference events preserve this property, it is therefore the case that the marking $(\mathsf{res}\, r{:}\mathrm{Ic}([r/w]t_0), \sigma', W')$ therefore satisfies φ in Γ. Let $\sigma' = (D', L', R', N')$. We have $r \in R'$. Interference and resource declaration events do not affect the heap owned by the process, so $D_0 \restriction_{W_0} \mathsf{proc} = D' \restriction_{W'} \mathsf{proc}$. Since the marking D_0 satisfies $\varphi \star \chi$ in Γ, there exists $D_r \subseteq D_0 \restriction_{W_0} \mathsf{proc}$ such that $D_r \models \chi$. All the assumptions to apply Lemma 5.3.5 are met. Since $r \in R'$, it can be seen from Definition 5.3.4 that $\mathsf{tolnv}^\chi_r(\mathrm{Ic}([r/w]t_0), \sigma', W')$ satisfies φ in $\Gamma, r{:}\chi$. By induction, the judgement $\Gamma, r_0{:}\chi \vdash_0 \{\varphi\}[r_0/w]t_0\{\psi\}$ is valid. From Lemma 5.3.8, recalling that $r \notin \mathrm{dom}(\Gamma)$, we therefore have $\Gamma, r{:}\chi \models \{\varphi\}[r/w]t_0\{\psi\}$. An induction on the length of π_1 applying Lemma 5.3.5 shows that the marking $\mathsf{tolnv}^\chi_r(C', \sigma, W)$ is reachable in $\mathcal{W}[\![[r/w]t_0]\!]_{\Gamma, r:\chi}$ from the marking $\mathsf{tolnv}^\chi_r(\mathrm{Ic}([r/w]t_0), \sigma_0, W_0)$. Hence, from the definition of validity, the marking $\mathsf{tolnv}^\chi_r(C', \sigma, W)$ is non-violating and therefore so is (C', σ, W) in $\mathcal{W}[\![[r/w]t_0]\!]_\Gamma$ by Lemma 5.3.5. It can also be seen that the marking (C', σ, W) is fault-free as a consequence of $\mathsf{tolnv}^\chi_r(C', \sigma, W)$ being fault-free. From Lemma 3.6.11, if $C' \neq \mathrm{Tc}([r/w]t_0)$ then the marking (C, σ, W) of $\mathcal{W}[\![t]\!]_\Gamma$ is non-violating and fault-free because any synchronized event s with concession on its control conditions in (C, σ, W) corresponds to an event s_0 with $s = \mathsf{res}\, r{:}s_0$ that has concession on its control conditions in (C', σ, W). If $C' = \mathrm{Tc}([r/w]t_0)$, it follows from the earlier remark that $\Gamma, r{:}\chi \models \{\varphi\}[r/w]t_0\{\psi\}$ that $\mathsf{tolnv}^\chi_r(C', \sigma, W)$ satisfies ψ in $\Gamma, r{:}\chi$. By Lemma 3.6.11, it can be seen that the only events with concession on their control conditions in this marking are equal to $\overline{\overline{\mathsf{end}}}_{(\mathsf{res}\, r:\mathrm{Tc}([r/w]t_0), \mathrm{Tc}(t))}(r, D_i)$ for some D_i such that $D_i \models \chi$. The marking is easily seen to be fault-free, so all that we must show is that it is non-violating. If the event $\mathsf{end}_{(\mathsf{res}\, r:\mathrm{Tc}([r/w]t_0), \mathrm{Tc}(t))}(r)$ has concession in (C, σ) in the net $\mathcal{N}[\![t]\!]$ then $r \in \sigma$. Let $\sigma = (D, L, R, N)$. From the definition of $\mathsf{tolnv}^\chi_r(C', \sigma, W)$, it follows that there exists $D_j \subseteq D \restriction_W \mathsf{proc}$ such that $D_j \models \chi$. It is easily seen, therefore, that the event $\overline{\overline{\mathsf{end}}}_{(\mathsf{res}\, r:\mathrm{Tc}([r/w]t_0), \mathrm{Tc}(t))}(r, D_j)$ has concession in the marking (C, σ, W), so the marking is non-violating.

We now consider the separation property for marking (C, σ, W). First suppose that there are synchronized events s and s' that both have concession in (C, σ, W). If $C =$

$\mathsf{res}\, r{:}\mathrm{Tc}([r/w]t_0)$ then, by Lemma 5.3.3, the events s and s' are not control independent. Otherwise, if $C \neq \mathsf{res}\, r{:}\mathrm{Tc}([r/w]t_0)$, observe that the separation property holds in the marking $\mathsf{tolnv}_r^\chi(C', \sigma, W)$ from the induction hypothesis for $\Gamma, r_0{:}\{\varphi\}[r_0/w]t_0\{\psi\}$ and Lemma 5.5.1 because $(r\ r_0) \cdot (\Gamma, r_0{:}\chi) = \Gamma, r{:}\chi$ and $(r\ r_0) \cdot [r_0/w]t_0 = [r/w]t_0$ since $\mathrm{fv}([r_0/w]t_0) \subseteq \mathrm{dom}(\Gamma, r_0{:}\chi)$ by Lemma 4.2.1 — in essence, because Lemma 5.5.1 allows us to infer that the separation property holds in the run of $\mathcal{W}[\![[r/w]t_0]\!]_{\Gamma, r:\chi}$ because it holds in the corresponding run of $\mathcal{W}[\![[r_0/w]t_0]\!]_{\Gamma, r_0:\chi}$ according to the induction hypothesis.

It follows straightforwardly from Lemma 5.3.3 and Lemma 5.3.5 that the marking M satisfies part (1) of the separation property. Parts (2) and (3) of the separation property, which consider the occurrence of s and s' in consecutive markings, are demonstrated similarly.

Finally, consider case the third case for the path π according to Lemma 5.3.3 on page 88, where $C = \mathrm{Tc}(t)$. Recall that there exist π_0, π_1 and π_2 such that

$$\pi = \pi_0 \cdot s_r \cdot (\mathsf{res}\, r{:}\pi_1) \cdot s'_r \cdot \pi_2,$$

and

$$\begin{aligned}
\mathcal{W}[\![t]\!]_\Gamma{:}\quad & (\mathrm{Ic}(t), \sigma_0, W_0) \overset{\pi_0 \cdot s_r}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{res}\, r{:}\mathrm{Ic}([r/w]t_0), \sigma', W'),\\
\mathcal{W}[\![[r/w]t_0]\!]_\Gamma{:}\quad & (\mathrm{Ic}([r/w]t_0), \sigma', W') \overset{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathrm{Tc}([r/w]t_0), \sigma'', W''), \quad \text{and}\\
\mathcal{W}[\![t]\!]_\Gamma{:}\quad & (\mathsf{res}\, r{:}\mathrm{Tc}([r/w]t_0), \sigma'', W'') \overset{s'_r \cdot \pi_2}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathrm{Tc}(t), \sigma, W).
\end{aligned}$$

The paths π_0 and π_2 consist only of interference events.

The marking $M = (\mathrm{Tc}(t), \sigma, W)$ is clearly non-violating and fault-free since no synchronized event in $\mathcal{W}[\![t]\!]_\Gamma$ has concession on its control conditions. For the same reason, the separation property trivially holds in this marking. All that remains is to show that M satisfies $\psi \star \chi$ in Γ.

As argued in the previous case, the marking $\mathsf{tolnv}_r^\chi(\mathrm{Tc}([r/w]t_0), \sigma'', W'')$ satisfies ψ in $\Gamma, r{:}\chi$. We have $r \in \sigma''$ since the event s'_r has concession in $(\mathrm{Tc}([r/w]t_0), \sigma'', W'')$. From the definition of $\mathsf{tolnv}_r^\chi(\mathrm{Tc}([r/w]t_0), \sigma'', W'')$, it is simple to see that $(\mathrm{Tc}([r/w]t_0), \sigma'', W'')$ satisfies $\psi \star \chi$ in Γ, as required. The disposal of the resource by the event s'_r yields a marking satisfying $\psi \star \chi$ in Γ. Since the events of π_2 are only interference events, Lemma 5.3.6 ensures that every marking along π_2 satisfies $\psi \star \chi$ in Γ. Hence the marking $(\mathrm{Tc}(t), \sigma, W)$ satisfies $\psi \star \chi$ in Γ, as required.

(L-CR)

For brevity, let $t = \texttt{with}\ w\ \texttt{do}\ t_0\ \texttt{od}$. Suppose that $\Gamma, r{:}\chi \vdash_0 \{\varphi\}t\{\psi\}$ because $\Gamma, r{:}\chi \vdash_0 \{\varphi \star \chi\}t_0\{\psi \star \chi\}$. Let π be a path such that $\mathcal{W}[\![t]\!]_\Gamma{:}(\mathrm{Ic}(t), \sigma_0, W_0) \overset{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} (C, \sigma, W)$ for some σ_0 and W_0 that satisfy φ in Γ. According to Lemma 5.3.6, we may assume that π starts with no interference events and that no interference event occurs if the terminal control conditions are marked. Just as Lemma 5.3.3 on page 88 uses Lemma 3.6.11 to characterize the runs of the net $\mathcal{W}[\![\texttt{resource}\ w\ \texttt{do}\ t_0\ \texttt{od}]\!]_\Gamma$, Lemma 3.6.13 can be used to show that π follows one of the three following cases:

1. $C = \mathrm{Ic}(t)$, $\sigma = \sigma_0$, $W = W_0$ and $\pi = ()$.
2. There exist D_1, C_0, π_0, σ' and W' such that $C = \mathsf{body}{:}C_0$ and $D_1 \models \chi$ and $\pi = s \cdot \mathsf{body}{:}\pi_0$ and
$$\begin{aligned}
\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}{:}\ & (\mathrm{Ic}(t), \sigma_0, W_0) \overset{\overline{\overline{\mathsf{acq}}}_{(\mathrm{Ic}(t), \mathsf{body}:\mathrm{Ic}(t_0))}(r, D_1)}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{body}{:}\mathrm{Ic}(t_0), \sigma', W')\\
\mathcal{W}[\![t_0]\!]_{\Gamma, r:\chi}{:}\ & (\mathrm{Ic}(t_0), \sigma', W') \overset{\pi_0}{\longrightarrow\!\!\!\!\!\rightarrow} (C_0, \sigma, W).
\end{aligned}$$

3. $C = \mathrm{Tc}(t)$ and there exist D_1, D_2, π_0, σ', σ'', W' and W'' such that $D_1 \models \chi$ and $D_2 \models \chi$, and

$$\pi = \overline{\overline{\mathsf{acq}}}_{(\mathrm{Ic}(t),\mathsf{body}:\mathrm{Ic}(t_0))}(r, D_1) \cdot \mathsf{body}{:}\pi_0 \cdot \overline{\overline{\mathsf{rel}}}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathrm{Tc}(t))}(r, D_2)$$

and

$$\begin{aligned}
&\mathcal{W}[\![t]\!]_{\Gamma,r:\chi} : (\mathrm{Ic}(t), \sigma_0, W_0) \overset{\overline{\overline{\mathsf{acq}}}_{(\mathrm{Ic}(t),\mathsf{body}:\mathrm{Ic}(t_0))}(r,D_1)}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{body}{:}\mathrm{Ic}(t_0), \sigma', W'),\\
&\mathcal{W}[\![t_0]\!]_{\Gamma,r:\chi} : (\mathrm{Ic}(t_0), \sigma', W') \overset{\pi_0}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathrm{Tc}(t_0), \sigma'', W''),\\
&\mathcal{W}[\![t]\!]_{\Gamma,r:\chi} : (\mathsf{body}{:}\mathrm{Tc}(t_0), \sigma'', W'') \overset{\overline{\overline{\mathsf{rel}}}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathrm{Tc}(t))}(r,D_2)}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathrm{Tc}(t), \sigma, W)
\end{aligned}$$

Case (1): According to Lemma 3.6.13, the only synchronized events with concession on their control conditions are of the form $\overline{\overline{\mathsf{acq}}}_{(C_1,C_2)}(r, D_1')$ for some D_1' such that $D_1' \models \chi$, so the initial marking must be fault-free. The marking $(\mathrm{Ic}(t), \sigma_0, W_0)$ is assumed to be consistent, so if $r \in R$ we must have $\omega_{\mathsf{inv}}(r) \in W$ and there exists (unique) $D_1 \subseteq D$ such that $D_1 \models \chi$ with $\omega_{\mathsf{proc}}(\mathrm{dom}(D_1)) \subseteq W$. It is straightforward to see from this that the marking is non-violating. The separation property is readily seen to be met since if two synchronized events occur sequentially from $(\mathrm{Ic}(t_0), \sigma_0, W_0)$ then they cannot be control-independent according to Lemmas 5.2.2 and 3.6.13.

Case (2): We now consider case (2), where $C = \mathsf{body}{:}C_0$, which will be the most substantial part of the proof for (L-CR). Let $\sigma_0 = (D_0, L_0, R_0, N_0)$ and $\sigma' = (D', R', N', L')$. We must have $r \in R_0$ and $R' = R_0 \setminus \{r\}$, but $D = D'$ and $L = L'$ and $N = N'$. From the definition of $\overline{\overline{\mathsf{acq}}}_{(\mathrm{Ic}(t),\mathsf{body}:\mathrm{Ic}(t_0))}(r, D_1)$, we have $\omega_{\mathsf{inv}}(\mathrm{dom}(D_1)) \subseteq W_0$ and

$$W' = W_0 \setminus \omega_{\mathsf{inv}}(\mathrm{dom}(D_1)) \cup \omega_{\mathsf{proc}}(\mathrm{dom}(D_1)).$$

Since (σ_0, W_0) satisfies φ in $\Gamma, r{:}\chi$, it follows from χ being precise that (σ', W') satisfies $\varphi \star \chi$ in $\Gamma, r{:}\chi$. It follows from the induction hypothesis that (C_0, σ, W) is a non-violating, fault-free marking of $\mathcal{W}[\![t_0]\!]_{\Gamma,r:\chi}$, and if $C_0 = \mathrm{Tc}(t_0)$ then (C_0, σ, W) satisfies $\psi \star \chi$ in $\Gamma, r{:}\chi$.

If there is a fault in the marking (C, σ, W) in the net $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$ and $C_0 \neq \mathrm{Tc}(t_0)$ then it follows from Lemmas 3.6.13 and 5.2.2 that there is a fault in the marking (C_0, σ, W), from which we derive a contradiction. If, instead, $C_0 = \mathrm{Tc}(t_0)$, then according to Lemmas 3.6.13 and 5.2.2, it is because there exists D_2 such that $\overline{\overline{\mathsf{rel}}}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathrm{Tc}(t))}(r, D_2)$ has concession on its control conditions but $r \in R$. However, a simple induction on the length of π_0, applying Lemma 3.10.1, allows us to see that $\omega_{\mathsf{proc}}(r) \in W$ and $r \notin \sigma$. It follows that the marking (C, σ, W) is fault-free.

We now show that the marking (C, σ, W) is non-violating. Let e be any event of $\mathcal{N}[\![t]\!]$ that has concession in the marking (C, σ) of $\mathcal{N}[\![t]\!]$. First suppose that $C_0 \neq \mathrm{Tc}(t_0)$. From Lemmas 3.4.1 and 3.6.13, there exists an event e_0 such that $e = \mathsf{body}{:}e_0$ and so e_0 has concession in the marking (C_0, σ) of $\mathcal{N}[\![t_0]\!]$. The marking (C_0, σ, W) is non-violating from the induction hypothesis, so there exists u_0 such that $e_0 \cdot u_0$ has concession in (C_0, σ, W). The interference events of $\mathcal{W}[\![t_0]\!]_{\Gamma,r:\chi}$ and $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$ are the same, and so $e \cdot u_0$ is an event in $\mathcal{W}[\![t]\!]_{\Gamma,r:\chi}$. Moreover, the event $e \cdot u_0$ has concession in (C, σ, W), so the marking is non-violating. Now suppose, instead, that $C_0 = \mathrm{Tc}(t_0)$. It now follows from Lemmas 3.4.1 and 3.6.13 that $e = \mathsf{rel}_{(\mathsf{body}:\mathrm{Tc}(t_0),\mathrm{Tc}(t))}(r)$. From the induction hypothesis, (C_0, σ, W) satisfies $\psi \star \chi$ in $\Gamma, r{:}\chi$. The event e has concession, so $r \in \sigma$, and there exists $D_2 \subseteq D \restriction_W \mathsf{proc}$ such that $D_2 \models \chi$ because (C, σ, W)

satisfies $\varphi \star \chi$ in Γ. It follows that the event $e \cdot \overline{\mathsf{rel}}(r, D_2)$ has concession in (C, σ, W), and therefore the marking is non-violating.

The final part of this case is to consider the separation property for the marking (C, σ, W). Let s and s' be control-independent synchronized events in $\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}$, and suppose first that both s and s' have concession in (C, σ, W). If $C_0 \neq \mathrm{Tc}(t_0)$ then, from Lemmas 5.2.2 and 3.6.13, there exist s_0 and s_0' such that $s = \mathsf{body}{:}s_0$ and $s' = \mathsf{body}{:}s_0'$. The events s_0 and s_0' have concession in (C_0, σ, W). Since the separation property holds in (C_0, σ, W) according to the induction hypothesis, either the events s_0 and s_0' are independent, in which case so are s and s', or s_0 and s_0' compete to allocate the same location, in which case so do s and s', *etc.* If, instead, $C_0 = \mathrm{Tc}(t_0)$, it follows from Lemmas 5.2.2 and 3.6.13 that the events s and s' cannot be control independent.

For the second part of the separation property, suppose that there exists a marking (C'', σ'', W'') such that $\mathcal{W}[\![t]\!]_\Gamma : (C, \sigma, W) \xrightarrow{s} (C'', \sigma'', W'') \xrightarrow{s'}$ for s and s' that are control-independent. It follows from Lemmas 5.2.2 and 3.6.13 that $C \neq \mathsf{body}{:}\mathrm{Tc}(t_0)$ and $C'' \neq \mathsf{body}{:}\mathrm{Tc}(t_0)$ since otherwise s and s' would not be control independent. According to these lemmas, there exist s_0 and s_0' such that $s = \mathsf{body}{:}s_0$ and $s' = \mathsf{body}{:}s_0'$. We have $\mathcal{W}[\![t_0]\!]_: (C_0, \sigma, W) \xrightarrow{s_0} (C_0'', \sigma'', W'') \xrightarrow{s_0'}$ for some C_0'' such that $C'' = \mathsf{body}{:}C_0''$. The separation property holds in (C_0, σ, W), so either s_0 and s_0' are independent, in which case so are s and s', or s_0 deallocates a location that s_0' allocates, *etc.* The separation property in the marking (C, σ, W) is therefore met.

Case (3): In this case, we have $C = \mathrm{Tc}(t)$ and

$$\begin{aligned}
\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}: &\quad (\mathrm{Ic}(t), \sigma_0, W_0) \xrightarrow{\overline{\overline{\mathsf{acq}}}_{(\mathrm{Ic}(t), \mathsf{body}:\mathrm{Ic}(t_0))}(r, D_1)} (\mathsf{body}{:}\mathrm{Ic}(t_0), \sigma', W'),\\
\mathcal{W}[\![t_0]\!]_{\Gamma, r:\chi}: &\quad (\mathrm{Ic}(t_0), \sigma', W') \xrightarrow{\pi_0} (\mathrm{Tc}(t_0), \sigma'', W''), \qquad \text{and}\\
\mathcal{W}[\![t]\!]_{\Gamma, r:\chi}: &\quad (\mathsf{body}{:}\mathrm{Tc}(t_0), \sigma'', W'') \xrightarrow{\overline{\overline{\mathsf{rel}}}_{(\mathsf{body}:\mathrm{Tc}(t_0), \mathrm{Tc}(t))}(r, D_2)} (\mathrm{Tc}(t), \sigma, W)
\end{aligned}$$

It follows from Lemmas 5.2.2 and 3.6.13 that no synchronized event has concession on its control conditions in the marking (C, σ, W), so the marking is non-violating, fault-free and the separation property trivially holds. All that we must show is that the marking $(\mathrm{Tc}(t), \sigma, W)$ satisfies φ in $\Gamma, r{:}\chi$. From the analysis in the previous case, the marking $(\mathrm{Tc}(t_0), \sigma'', W'')$ satisfies $\varphi \star \chi$ in $\Gamma, r{:}\chi$ and $r \notin \sigma''$. From the definition of $\overline{\overline{\mathsf{rel}}}_{(\mathsf{body}:\mathrm{Tc}(t_0), \mathrm{Tc}(t))}(r, D_2)$, we have $D_2 \models \chi$ and

$$\sigma = \sigma'' \cup \{r\} \qquad \text{and} \qquad W = W'' \setminus \omega_{\mathsf{proc}}(\mathrm{dom}(D_2)) \cup \omega_{\mathsf{inv}}(\mathrm{dom}(D_2)).$$

Note that $\sigma'' = (D, L, R \cup \{r\}, N)$. Since $D \restriction_{W''} \mathsf{proc} \models \psi \star \chi$, it follows that $D \restriction_W \mathsf{proc} \models \psi$ because the formula χ is precise. We have $D \restriction_{W''} \mathsf{inv} \models \mathrm{inv}(\Gamma, r{:}\chi,\ R'')$, so $D \restriction_W \mathsf{inv} \models \mathrm{inv}(\Gamma, r{:}\chi,\ R'') \star \chi$, and therefore $D \restriction_W \mathsf{inv} \models \mathrm{inv}(\Gamma, r{:}\chi,\ R)$. It is therefore the case that $(\mathrm{Tc}(t), \sigma, W)$ satisfies φ in $\Gamma, r{:}\chi$, as required.

(L-Par)

Suppose that $\Gamma \vdash_0 \{\varphi_1 \star \varphi_2\} t_1 \parallel t_2 \{\psi_1 \star \psi_2\}$ because $\Gamma \vdash_0 \{\varphi_1\} t_1 \{\psi_1\}$ and $\Gamma \vdash_0 \{\varphi_2\} t_2 \{\psi_2\}$. Assume that the marking $(\mathrm{Ic}(t_1 \parallel t_2), \sigma_0, W_0)$ satisfies $\varphi_1 \star \varphi_2$ in Γ. It can be seen from the definitions that there exist W_{01} and W_{02} forming an ownership split of W_0 such that the marking $(\mathrm{Ic}(t_1), \sigma_0, W_{01})$ satisfies φ_1 in Γ and $(\mathrm{Ic}(t_2), \sigma_0, W_{02})$ satisfies φ_2 in Γ. The marking (C, σ, W) is reachable from $(\mathrm{Ic}(t_1 \parallel t_2), \sigma_0, W_0)$. A simple induction along the

path to this marking (using Lemmas 3.6.5 and 5.2.2 to show that there exist C_1 and C_2 such that $C = \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2$) using Lemma 5.3.2 shows that there exist W_1 and W_2 forming an ownership split of W such that

$$\begin{array}{rcll} \mathcal{W}[\![t_1]\!]_\Gamma & : & (\mathrm{Ic}(t_1), \sigma_0, W_{01}) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} (C_1, \sigma, W_1) & \text{and} \\ \mathcal{W}[\![t_2]\!]_\Gamma & : & (\mathrm{Ic}(t_2), \sigma_0, W_{02}) \stackrel{*}{\longrightarrow\!\!\!\!\!\rightarrow} (C_2, \sigma, W_2). & \end{array}$$

From the induction hypotheses, the markings (C_1, σ, W_1) and (C_2, σ, W_2) are fault-free, non-violating, satisfy the separation property and are such that if $C_1 = \mathrm{Tc}(t_1)$ then $(\mathrm{Tc}(t_1), \sigma, W_1)$ satisfies φ_1 in Γ, and if $C_2 = \mathrm{Tc}(t_2)$ then $(\mathrm{Tc}(t_2), \sigma, W_2)$ satisfies φ_2 in Γ. It is easy to see, as a consequence, that the marking (C, σ, W) is fault-free, and, from Lemma 5.3.2, that it is non-violating. Suppose that $C = \mathrm{Tc}(t_1 \parallel t_2)$; it is easy to see that $C_1 = \mathrm{Tc}(t_1)$ and $C_2 = \mathrm{Tc}(t_2)$. It follows from the fact that W_1 and W_2 form an ownership split of W_1 and W_2 that the marking (C, σ, W) satisfies $\varphi_1 \star \varphi_2$ in Γ.

We now consider the separation property in the marking (C, σ, W). Let s_1 and s_2 be control-independent synchronized events in $\mathcal{W}[\![t_1 \parallel t_2]\!]_\Gamma$. If $s_1 = (\mathsf{par}\,1{:}e_1) \cdot u_1$ and $s_2 = (\mathsf{par}\,1{:}e_2) \cdot u_2$ for some $e_1, e_2 \in \mathrm{Ev}(t_1)$ and interference events u_1 and u_2 in $\mathcal{W}[\![t_1]\!]_\Gamma$, the result follows routinely from the induction hypothesis, and similarly if s_1 and s_2 both arise from events of $\mathcal{N}[\![t_2]\!]$. Suppose instead that there exist $e_1 \in \mathrm{Ev}(t_1)$, $e_2 \in \mathrm{Ev}(t_2)$ and interference events u_1 and u_2 such that $s_1 = (\mathsf{par}\,1{:}e_1) \cdot u_1$ and $s_2 = (\mathsf{par}\,2{:}e_2) \cdot u_2$.

First, suppose that in the net $\mathcal{W}[\![t_1 \parallel t_2]\!]_\Gamma$ we have

$$(C, \sigma, W) \stackrel{s_1}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{par}\,1{:}C_1' \cup \mathsf{par}\,2{:}C_2', \sigma', W') \stackrel{s_2}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{par}\,1{:}C_1'' \cup \mathsf{par}\,2{:}C_2'', \sigma'', W'').$$

Applying the parallel decomposition lemma (Lemma 5.3.2) twice shows that

$$(C_1, \sigma, W_1) \stackrel{e_1 \cdot u_1}{\longrightarrow\!\!\!\!\!\rightarrow} (C_1', \sigma', W_1') \stackrel{u_2}{\longrightarrow\!\!\!\!\!\rightarrow} (C_1'', \sigma'', W_1'')$$

in $\mathcal{W}[\![t_1]\!]_\Gamma$ for some W_1', W_1'' for W_1 the marking of ownership conditions obtained above. By Lemma 5.5.2, the separation property holds for $e_1 \cdot u_1$ and u_2 in (C_1, σ, W_1); consider how it might hold. If $e_1 \cdot u_1$ deallocates a location that u_2 allocates then s_1 deallocates a location that s_2 allocates, so the separation property holds for s_1 and s_2. The argument is similar for all the other cases where $e_1 \cdot u_1$ and u_2 are not independent. Suppose instead that $e_1 \cdot u_1 I u_2$. The event u_2 has concession in the marking (C_1, σ, W_1) by virtue of the fact that the occurrence of independent events in a run can be interchanged (Proposition 2.2.1). Consider the marking $(\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2, \sigma, W_1)$ of $\mathcal{W}[\![t_1 \parallel t_2]\!]_\Gamma$. The event s_1 is readily seen using Lemma 3.3.2 to have concession in this marking, as does u_2. The event $\mathsf{par}\,2{:}e_2$ is control-independent from $\mathsf{par}\,1{:}e_1$, so by Lemma 5.5.3 we have $s_1 I s_2$, as required.

Now suppose that in the net $\mathcal{W}[\![t_1 \parallel t_2]\!]_\Gamma$ we have

$$\begin{array}{rl} & (C, \sigma, W) \stackrel{s_1}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{par}\,1{:}C_1' \cup \mathsf{par}\,2{:}C_2', \sigma', W') \\ \text{and} & (C, \sigma, W) \stackrel{s_2}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{par}\,1{:}C_1'' \cup \mathsf{par}\,2{:}C_2'', \sigma'', W''). \end{array}$$

Applying the parallel decomposition lemma, we have

$$(C_1, \sigma, W_1) \stackrel{e_1 \cdot u_1}{\longrightarrow\!\!\!\!\!\rightarrow} (C_1', \sigma', W_1') \qquad \text{and} \qquad (C_1, \sigma, W_1) \stackrel{u_2}{\longrightarrow\!\!\!\!\!\rightarrow} (C_1'', \sigma'', W_1'')$$

in $\mathcal{W}[\![t_1]\!]_\Gamma$ for some W_1', W_1''. By Lemma 5.5.2, the separation property holds for $e_1 \cdot u_1$ and u_2 in (C_1, σ, W_1); consider how it might hold. If $e_1 \cdot u_1$ allocates a location that u_2 also allocates, then s_1 allocates a location that s_2 allocates, so the separation property

holds for s_1 and s_2. The argument is similar for all the other cases where $e_1 \cdot u_1$ and u_2 are not independent. Suppose instead that $e_1 \cdot u_1 I u_2$. Consider the marking ($\mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}C_2, \sigma, W_1$) of $\mathcal{W}[\![t_1 \parallel t_2]\!]_\Gamma$. The event s_1 has concession in this marking as does u_2. The event $\mathsf{par}\,2{:}e_2$ is control-independent from $\mathsf{par}\,1{:}e_1$, so by Lemma 5.5.3 we have $s_1 I s_2$, as required.

(L-Frame)

Suppose that $\Gamma \vdash_0 \{\varphi \star \varphi'\} t \{\psi \star \varphi'\}$ because $\Gamma \vdash_0 \{\varphi\} t \{\psi\}$. By the rule (L-Nil), we have $\Gamma \vdash_0 \{\varphi'\} \varepsilon \{\varphi'\}$. We therefore have $\Gamma \vdash_0 \{\varphi \star \varphi'\} t \parallel \varepsilon \{\psi \star \varphi'\}$. Let $M' = (\mathrm{Ic}(t \parallel \varepsilon), \sigma_0, W_0)$ be a marking that satisfies φ in Γ. Applying the argument above for parallel composition to the induction hypothesis, we see that every marking reachable from M' is fault-free, non-violating, satisfies the separation property and, if terminal, satisfies $\psi \star \varphi'$ in Γ. The very close relationship between $\mathcal{W}[\![t]\!]_\Gamma$ and $\mathcal{W}[\![t \parallel \varepsilon]\!]_\Gamma$, that the nets are isomorphic but for the isolated marked control condition $\mathsf{par}\,2{:}(\mathsf{i}, \mathsf{t})$ in $\mathcal{W}[\![t \parallel \varepsilon]\!]_\Gamma$, ensures that any marking reachable from $(\mathrm{Ic}(t), \sigma_0, W_0)$ in $\mathcal{W}[\![t]\!]_\Gamma$ is also non-violating, fault-free, satisfies the separation property and, if terminal, satisfies $\psi \star \varphi'$ in Γ.

(L-Conjunction)

Suppose that we have $\Gamma \vdash_0 \{\varphi_1 \wedge \varphi_2\} t \{\psi_1 \wedge \psi_2\}$ because $\Gamma \vdash_0 \{\varphi_1\} t \{\psi_1\}$ and $\Gamma \vdash_0 \{\varphi_2\} t \{\psi_2\}$. Suppose that the initial marking $(\mathrm{Ic}(t), \sigma_0, W_0)$ satisfies $\varphi_1 \wedge \varphi_2$; it therefore satisfies φ_1 and also satisfies φ_2 from the definition of satisfaction and the definition of conjunction in the heap logic. From the induction hypothesis of $\Gamma \vdash_0 \{\varphi_1\} t \{\psi_1\}$, we may conclude that every marking reachable from $M_0 = (\mathrm{Ic}(t), \sigma_0, W_0)$ in $\mathcal{W}[\![t]\!]_\Gamma$ is non-violating and fault-free, that the separation property holds for all reachable markings, and that any terminal marking satisfies ψ_1. From the induction hypothesis of $\Gamma \vdash_0 \{\varphi_2\} t \{\psi_2\}$, we may further conclude that any terminal marking reachable from $(\mathrm{Ic}(t), \sigma_0, W_0)$ satisfies ψ_2. Consequently, by the definition of a marking satisfying a formula, any terminal marking reachable from M_0 satisfies ψ_1 and ψ_2. By the definition of conjunction in the heap logic, any terminal marking reachable from M_0 therefore satisfies $\psi_1 \wedge \psi_2$.

(L-Disjunction)

Suppose that we have $\Gamma \vdash_0 \{\varphi_1 \vee \varphi_2\} t \{\psi_1 \vee \psi_2\}$ because $\Gamma \vdash_0 \{\varphi_1\} t \{\psi_1\}$ and $\Gamma \vdash_0 \{\varphi_2\} t \{\psi_2\}$. Suppose that the initial marking $M_0 = (\mathrm{Ic}(t), \sigma_0, W_0)$ satisfies $\varphi_1 \vee \varphi_2$. It therefore either satisfies φ_1 or φ_2. Suppose first that it satisfies φ_1. From the induction hypothesis for $\Gamma \vdash_0 \{\varphi_1\} t \{\psi_1\}$, we may conclude that any reachable marking is non-violating, fault-free and satisfies the separation property. Furthermore, any reachable terminal marking satisfies ψ_1 and, consequently, any reachable terminal marking satisfies $\psi_1 \vee \psi_2$. The case if M satisfies φ_2 is similar, so any terminal marking satisfies $\psi_1 \vee \psi_2$. □

Chapter 6

Refinement

When we consider concurrent programs, the atomicity assumed of their primitive actions, called their *granularity*, is of significance. For example, suppose that the concurrent program

$$\begin{array}{ll} & \ell := \ell' + 1 \\ \| & (\ell' \neq \ell).\texttt{diverge} + (\ell' = \ell).\texttt{skip} \end{array}$$

runs from the heap $\{\ell \mapsto 0, \ell' \mapsto 1\}$. Given the prior interpretations of `skip` and `diverge`, we might conclude that the program never terminates since the assignment $\ell := \ell' + 1$ maintains the property through execution that ℓ and ℓ' hold different values. Consequently, the left branch of the guarded sum is always chosen so the right-hand parallel process always diverges.

It may not, however, be reasonable to assume that the assignment is executed atomically. For instance, the processor on which the process runs might have primitive actions for copying the values held in memory locations and for incrementing them, but not for copying *and* incrementing in one clock step. The process $\ell := \ell' + 1$ might therefore be compiled to execute as $\ell := \ell'; \ell := \ell + 1$. Quite clearly, the process

$$\begin{array}{ll} & \ell := \ell'; \ell := \ell + 1 \\ \| & (\ell' \neq \ell).\texttt{diverge} + (\ell' = \ell).\texttt{skip} \end{array}$$

may terminate, so we failed to exhibit a proper degree of caution when asserting that it would fail to terminate.

An important observation in [Rey04] is that changes in the granularity of actions *are* permissible in the absence of races. The absence of races excludes interference of the form in the example above. Since processes subjected to a judgement in concurrent separation logic are race-free (when running from suitable initial states), it follows that proved processes are not susceptible to changes in granularity.

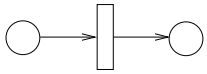
In this section, we will begin to show how the net model might be applied to provide a formal basis for this intuitive account. We will provide a framework for refinement (an operation on the semantics) that allows the granularity of actions to be changed. We then provide a constraint on refinement called *non-interference* that is sufficient to demonstrate that the refinements do not affect the big-step semantics of processes. As we shall discuss at the end of this chapter, this refinement operation may form the basis of an attempt to prove Reynolds' conjecture.

6.1 Contexts for refinement

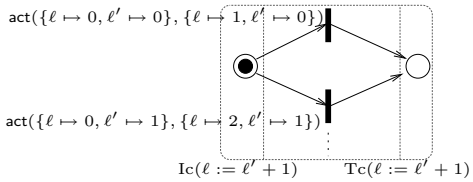
We showed earlier that programs were race-free by proving in Theorem 5.5 the separation property for proved processes. Within our net model we can provide a form of *refinement*, similar to that of [vGG89] but suited to processes executing in a shared environment, that begins to capture these ideas. The property required to apply the refinement operation may be captured directly in terms of independence, with no changes to our semantics. This concurs with the point made in [Pra86] and [CMP87] that independence models are well-suited to such refinement operations.

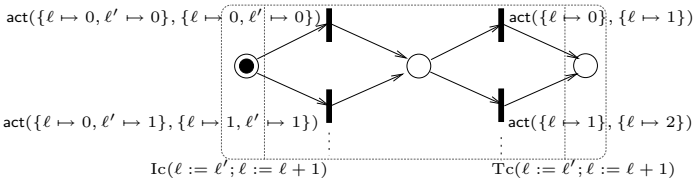
We will relate the nets representing processes with different levels of atomicity by regarding them as alternative substitutions into a *context*. We will then give a condition on substitutions led by Theorem 5.5 to show that any partial correctness assertion made for one of the nets also holds for the other.

In this section, when we draw contexts the hole will be drawn using a hollow rectangle:

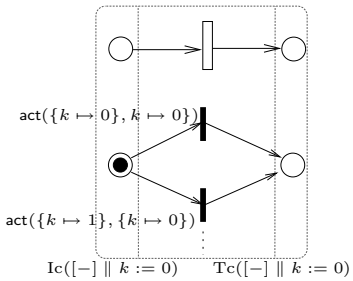


As an example, a change in granularity of the command $\ell := \ell' + 1$ to become $\ell := \ell'; \ell := \ell + 1$ in the term $\ell := \ell' + 1 \parallel k := 0$ shall be represented by considering the two nets $\mathcal{N}[\![\ell := \ell' + 1]\!]$ and $\mathcal{N}[\![\ell := \ell'; \ell := \ell + 1]\!]$





as alternative substitutions into the context representing their parallel composition with $k := 0$



(we omit the state conditions from both diagrams but do give the labels of events).

Formally, a context K will be an embedded net with a special event $[-]$ to represent the 'hole'. The event $[-]$ has no effect in the state conditions, which represent the values held in memory locations, so it has no pre-*state* or post-*state* conditions; it only has effect on control conditions (note that in this chapter we no longer include ownership conditions

or interference events). We shall require the net K and any net N substituted into the hole to be well-terminating (Lemma 3.5.1) and to satisfy all the constraints of Lemma 3.5.1. The definitions become a little technically-involved, so to simplify matters we shall also assume that ${}^\bullet[-]$ and $[-]^\bullet$ are disjoint and also that $\mathrm{Ic}(N)$ and $\mathrm{Tc}(N)$ are disjoint, though these requirements could be dropped.

Definition 6.1.1 (Context)**.** *Define a* context K *to be a embedded net with a distinguished event* $[-]$*. The event* $[-]$ *is such that* ${}^\bullet[-]^\bullet \subseteq \mathbf{C}$ *and its pre- and postconditions form disjoint, nonempty sets.*

We may now construct the net representing the substitution of a net N for the hole in a context K. We shall assume that, as in the semantics for terms, the two nets are formed with the same sets of conditions. As the nets are extensional (we regard an event simply as its set of preconditions paired with its set of postconditions), all that we need to specify is the events of the net and its initial and terminal markings of control conditions.

In the net $K[N]$, for conditions c from the context K we add a prefix to form $\mathsf{in}_K : c$, and similarly conditions from N shall be of the form $\mathsf{in}_N : c$. At the boundary of the substitution, we shall write conditions as pairs $(\mathsf{in}_K : c, \mathsf{in}_N : c')$ so that we can use the gluing operation defined earlier (on page 36).

Definition 6.1.2 (Substitution)**.** *Let* K *be a context and* N *an embedded net. Define the sets*

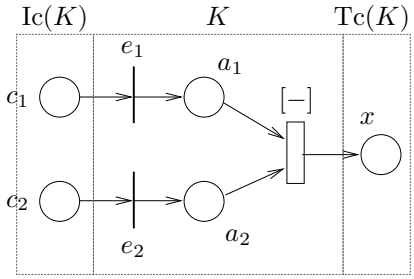
$$P_\mathsf{i} \triangleq \mathsf{in}_K : {}^\bullet[-] \times \mathsf{in}_N : \mathrm{Ic}(N) \qquad P_\mathsf{t} \triangleq \mathsf{in}_K : [-]^\bullet \times \mathsf{in}_N : \mathrm{Tc}(N).$$

The substitution $K[N]$ *is defined to be the embedded net with:*

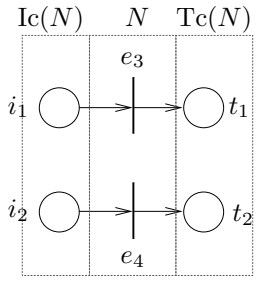
$$\begin{aligned}
\mathrm{Ev}(K[N]) &\triangleq (P_\mathsf{i} \cup P_\mathsf{t}) \triangleleft \mathsf{in}_K : (\mathrm{Ev}(K) \setminus \{[-]\}) \cup (P_\mathsf{i} \cup P_\mathsf{t}) \triangleright \mathsf{in}_N : \mathrm{Ev}(N) \\
\mathrm{Ic}(K[N]) &\triangleq (P_\mathsf{i} \cup P_\mathsf{t}) \triangleleft \mathsf{in}_K : \mathrm{Ic}(K) \\
\mathrm{Tc}(K[N]) &\triangleq (P_\mathsf{i} \cup P_\mathsf{t}) \triangleleft \mathsf{in}_K : \mathrm{Tc}(K)
\end{aligned}$$

To see the definition at work, we give the following example.

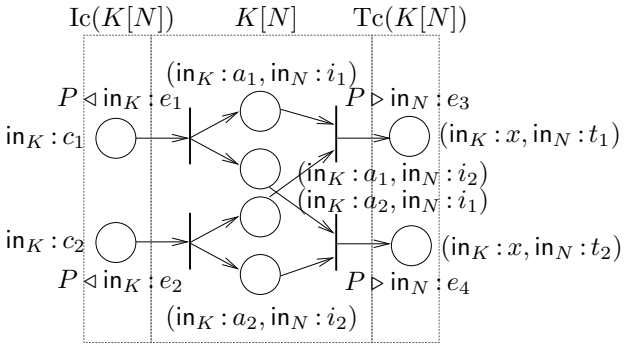
Example 6.1.1. *Let* K *be the context which has two concurrent events,* e_1 *and* e_2*, which can occur concurrently in the initial marking before the substituend is activated:*



Suppose that we insert the following net, which runs e_3 *and* e_4 *concurrently:*



We obtain the following net, representing the sequential composition of $e_1 \parallel e_2$ followed by $e_3 \parallel e_4$. We elide details of the action of events on state conditions, which is unaffected by the substitution operation.



Suppose that N_1 and N_2 are nets representing a change in granularity of an action. We wish relate runs of the net $K[N_1]$ to runs of the net $K[N_2]$.

Definition 6.1.3. *Let π be a (finite) sequence of events of the net N. Sequence π is said to be* complete from σ to σ' *if*

$$N{:}(\mathrm{Ic}(N), \sigma) \xrightarrow{\pi}\!\!\!\!\!\rightarrow (\mathrm{Tc}(N), \sigma').$$

Write $N{:}\sigma \Downarrow \sigma'$ if there exists a complete sequence from σ to σ' in N.

Using this definition, we can define a notion of complete trace equivalence $\simeq$ as:

$$N_1 \simeq N_2 \quad \text{iff} \quad (\forall \sigma, \sigma')N_1{:}\sigma \Downarrow \sigma' \iff N_2{:}\sigma \Downarrow \sigma'.$$

We wish to constrain K, N_1 and N_2 appropriately so that if $N_1 \simeq N_2$ then for all suitable initial states σ_0

$$(\forall \sigma')K[N_1]{:}\sigma_0 \Downarrow \sigma' \iff K[N_2]{:}\sigma_0 \Downarrow \sigma'.$$

As an example, the context introduced at the start of this section for $\ell := \ell' + 1$ does not give rise to nets with this property.

Example 6.1.2. *Write, in the obvious way, "$-$" for the action term that will be interpreted as forming the hole of a context. Define*

$$\begin{aligned} K &\triangleq \mathcal{N}\left[\!\left[- \parallel \ (\ell' \neq \ell).\mathtt{diverge} + (\ell' = \ell).\mathtt{skip} \right]\!\right] \\ N_1 &\triangleq \mathcal{N}\left[\!\left[\ell := \ell' + 1 \right]\!\right] \\ N_2 &\triangleq \mathcal{N}\left[\!\left[\ell := \ell'; \ell := \ell + 1 \right]\!\right]. \end{aligned}$$

We have $N_1 \simeq N_2$ and

$$K[N_1]{:}\{\ell \mapsto 0, \ell' \mapsto 1\} \not\Downarrow \{\ell \mapsto 2, \ell' \mapsto 1\}$$

but

$$K[N_2]{:}\{\ell \mapsto 0, \ell' \mapsto 1\} \Downarrow \{\ell \mapsto 2, \ell' \mapsto 1\}.$$

The problem here is that the context K interferes with the execution of N_1 and N_2. We wish to find a constraint that rules this out.

6.2 Contextual interference

Return to the general case for a substitution $K[N]$. Intuitively, if the substituend N were an atomic event, it would start running only if the conditions P_{i} were marked and P_{t} were not. There are two distinct ways in which the context K can affect the execution of N. Firstly, it might affect the marking of conditions in P_{i} or P_{t} whilst N is running. Secondly, it might change the marking of state conditions in a way that affects the execution of N. An instance of the latter form of interference is seen in the preceding example. We now define a form of constrained substitution, guided by Theorem 5.5, so that N is not subject to these forms of interference. The development is slightly different from that in [HW08a] (here, we impose 'control non-interference') in order to obtain a conceptually clearer framework.

Say that a control condition c of $K[N]$ is *internal* to N if $c = \mathsf{in}_N : c_2$ where c_2 is a pre- or a postcondition of an event of N that is not in $\mathrm{Ic}(N)$ or $\mathrm{Tc}(N)$. It is useful to classify the markings of $K[N]$ according to whether they support the occurrence of N- or K-events on the conditions P_{i} and P_{t}:

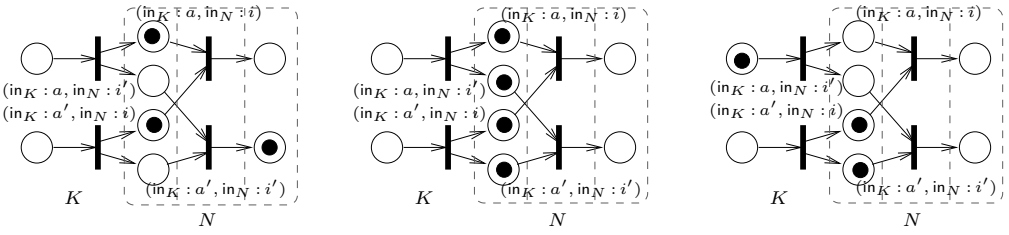
Definition 6.2.1. *A marking C of control conditions of $K[N]$ is an N-marking if for all $a, a' \in {}^\bullet[-]$, $x, x' \in [-]^\bullet$, $i \in \mathrm{Ic}(N)$ and $t \in \mathrm{Tc}(N)$:*

- *if $(\mathsf{in}_K : a, \mathsf{in}_N : i) \in C$ then $(\mathsf{in}_K : a', \mathsf{in}_N : i) \in C$, and*
- *if $(\mathsf{in}_K : x, \mathsf{in}_N : t) \in C$ then $(\mathsf{in}_K : x', \mathsf{in}_N : t) \in C$.*

A marking C of control conditions of $K[N]$ is a K-marking if there is no N-internal condition marked, and furthermore, for all $a \in {}^\bullet[-]$, $x \in [-]^\bullet$, $i, i' \in \mathrm{Ic}(N)$ and $t, t' \in \mathrm{Tc}(N)$:

- *if $(\mathsf{in}_K : a, \mathsf{in}_N : i) \in C$ then $(\mathsf{in}_K : a, \mathsf{in}_N : i') \in C$, and*
- *if $(\mathsf{in}_K : x, \mathsf{in}_N : t) \in C$ then $(\mathsf{in}_K : x, \mathsf{in}_N : t') \in C$.*

For example, the marking on the left is an N-marking, the marking on the right is a K-marking and the middle marking is both a K- and an N-marking.



It is straightforward to prove, using Lemma 3.3.2, that any K-event occurring from a K-marking gives rise to another K-marking and, similarly, N-events preserve the property of being an N-marking. Given a marking C of control conditions of $K[N]$, say that N is *active* if it is an N-marking such that $P_{\mathsf{t}} \not\subseteq C$.

The first stage in showing that there is no interference between K and N in $K[N]$ is to capture the property that the context K does not affect the control conditions in $P_{\mathsf{i}} \cup P_{\mathsf{t}}$ whilst the subnet N is active; we call this property *control non-interference*. In an analogous manner to the definition in Section 3.4 of the control net of a term, we denote by $\mathcal{C}(N)$ the (labelled) net obtained by stripping away the state conditions from

the embedded net N. This net represents the control features of N and will be used to describe its control properties. As in Lemma 3.4.1, it is easy to see that

$$N{:}(C,\sigma) \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} (C',\sigma) \implies \mathcal{C}(N){:}C \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'.$$

We shall say that the net $K[N]$ represents a *control-noninterfering* substitution if whenever the net $\mathcal{C}(N)$ is active in $\mathcal{C}(K[N])$ and an event from $\mathcal{C}[\![K]\!]$ has concession, the event is disjoint from the conditions in $P_{\mathsf{i}} \cup P_{\mathsf{t}}$. In addition, it must be the case that the subnet N is initialized only if none of its terminal conditions are marked.

Definition 6.2.2. *The net $\mathcal{C}(K[N])$ is a control-noninterfering substitution if, for any marking C reachable from $\mathrm{Ic}(K[N])$ in $\mathcal{C}(K[N])$, the following two properties are satisfied:*

- *if N is active in C, any event e that is an event of $\mathcal{C}(K)$ (i.e. there exists $e_0 \neq [-]$ in $\mathcal{C}(K)$ such that $e = (P_{\mathsf{i}} \cup P_{\mathsf{t}}) \triangleleft \mathsf{in}_K : e_0$) and has concession in C satisfies*

$$^{\mathbf{C}}e^{\mathbf{C}} \cap (P_{\mathsf{i}} \cup P_{\mathsf{t}}) = \emptyset$$

- *if $P_{\mathsf{i}} \subseteq C$ then $P_{\mathsf{t}} \cap C = \emptyset$*

We can now express when the net $K[N]$ is a non-interfering substitution.

Definition 6.2.3. *For a given marking of state conditions σ, we say that $K[N]$ is a* non-interfering *substitution if $\mathcal{C}(K[N])$ is a control-noninterfering substitution and, for all markings M reachable from $(\mathrm{Ic}(K[N]),\sigma)$:*

> *if $M \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} M_1 \stackrel{e_2}{\longrightarrow\!\!\!\!\!\rightarrow} M'$, one of e_1 and e_2 is from N and the other is from K and N is active in M and M_1, then e_1 and e_2 are independent.*

Note that the definition of non-interfering substitution is from a particular initial marking of state conditions. This corresponds to the race freedom result, which asserts that processes are race-free only from particular states.

We wish to show that if $K[N_1]$ and $K[N_2]$ are non-interfering substitutions from some initial state then their sets of reachable terminal states are equal. The reason why this is so is that the independence obtained from the non-interference property ensures that the events in any complete run of $K[N_1]$ can be reordered to give a run consisting only of K events, followed by a sequence of events forming a run of N_1, then K events, and so on. We now proceed to formalize this. The details are quite technical, so the reader may wish to skip directly to Theorem 6.1.

A sequence of events $\pi = (e_1,\ldots,e_n)$ considered from a marking M gives rise to markings $M_1,\ldots,M_n$ such that $M \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} M_1 \ldots \stackrel{e_n}{\longrightarrow\!\!\!\!\!\rightarrow} M_n$. To describe the structure of such sequences, we shall say that π from marking M is of the form $\Pi_1 \cdot \Pi_2$ if there exist π_1 and π_2 such that $\pi = \pi_1 \cdot \pi_2$, where $\cdot$ denotes the concatenation of sequences, and π_1 is of form Π_1 from marking M and π_2 is of form Π_2 from the marking obtained by following π_1 from M. Sequence π is of form Π^* if it is the concatenation of a finite number of sequences, each of form Π.

Throughout the remainder of this section, when we consider the substitution $K[N]$ let P_{i} and P_{t} be defined as in Definition 6.1.2:

$$\begin{aligned} P_{\mathsf{i}} &\triangleq \mathsf{in}_K : {}^{\bullet}[-] \times \mathsf{in}_N : \mathrm{Ic}(N) \\ P_{\mathsf{t}} &\triangleq \mathsf{in}_K : [-]^{\bullet} \times \mathsf{in}_N : \mathrm{Tc}(N). \end{aligned}$$

Any reachable marking of control conditions of the net $K[N]$ can be partitioned into two sets: conditions that occur solely within K and conditions that are either N-internal or in P_{i} or P_{t}. Formally, a condition c is a *K-condition* if $c = \mathsf{in}_K : c_1$ for some condition c_1 of K not in ${}^\bullet[-]^\bullet$. A condition c is an *N-condition* if either $c \in P_{\mathsf{i}} \cup P_{\mathsf{t}}$ or $c = \mathsf{in}_N : c_2$ for some condition c_2 of N not in $\mathrm{Ic}(N) \cup \mathrm{Tc}(N)$. Recall that we call $\mathsf{in}_N : c_2$ an N-internal condition. It is easy to see that from the marking $\mathrm{Ic}(K[N])$ in $\mathcal{C}(K[N])$ only K- or N-control conditions may be marked: If C is a reachable marking of $\mathcal{C}(K[N])$, we have $C = C_N \cup C_K$ for some marking C_N of N-conditions and some marking C_K of K-conditions. We shall frequently use the notation (C_N, C_K) for a marking of control conditions, where C_N comprises only N-conditions and C_K comprises only K-conditions.

When considering a substitution $K[N]$, we shall refer to an event e as being an N-event if it is equal to $(P_{\mathsf{i}} \cup P_{\mathsf{t}}) \triangleright \mathsf{in}_N : e_2$ for some e_2 in N. Otherwise, it is a K-event. A little care is necessary since an event in the net $K[N]$ might arise from both K and N if there are events e of N and $e' \neq [-]$ of K with the same effect on state conditions such that:

$$\begin{array}{rclcrcl} {}^{\mathbf{c}}e & = & \mathrm{Ic}(N), & & e^{\mathbf{c}} & = & \mathrm{Tc}(N) \\ {}^{\mathbf{c}}e' & = & {}^\bullet[-], & & e'^{\mathbf{c}} & = & [-]^\bullet. \end{array}$$

Throughout the remainder of this section, for simplicity we shall require that the substitution $K[N]$ has no such events. This restriction may be lifted with little effect on the development so-far by allowing the net formed to be non-extensional, or by considering this as a special case when demonstrating properties of the net $K[N]$

We now define operations to relate markings of the nets N, K and $K[N]$.

From a marking of control conditions (C_N, C_K) of the net $K[N]$, we can extract markings of control conditions for the nets N and K. We shall define $\rho_N(C_N)$ to be the marking of N obtained from (C_N, C_K), which is not dependent on the marking C_K of K-conditions, and $\rho_K(C_N, C_K)$ for the marking of K obtained from (C_N, C_K), which is dependent on the marking of N-conditions (namely, the marking of N-conditions in $P_{\mathsf{i}} \cup P_{\mathsf{t}}$).

For a marking C of the context K, we define $\theta_K(C)$ to be the corresponding marking of $K[N]$. For a marking C' of the net N, we shall define $\theta_N(C')$ to be the marking of N-conditions in the net $\mathcal{C}[\![K[N]]\!]$ corresponding to C'.

Definition 6.2.4. *Let $K[N]$ be any substitution. For any marking C_N of N-conditions and C_K of K-conditions, define*

$$\begin{array}{rcrl} \rho_K(C_N, C_K) & \triangleq & \{ & a \in {}^\bullet[-] \mid \forall i \in \mathrm{Ic}(N).(\mathsf{in}_K : a, \mathsf{in}_N : i) \in C_N \} \\ & & \cup\{ & x \in [-]^\bullet \mid \forall t \in \mathrm{Tc}(N).(\mathsf{in}_K : x, \mathsf{in}_N : t) \in C_N \} \\ & & \cup\{ & c \notin {}^\bullet[-] \cup [-]^\bullet \mid \mathsf{in}_K : c \in C_K \} \\ \\ \rho_N(C_N) & \triangleq & \{ & i \in \mathrm{Ic}(N) \mid \forall a \in {}^\bullet[-].(\mathsf{in}_K : a, \mathsf{in}_N : i) \in C_N \} \\ & & \cup\{ & t \in \mathrm{Tc}(N) \mid \forall x \in [-]^\bullet.(\mathsf{in}_K : x, \mathsf{in}_N : t) \in C_N \} \\ & & \cup\{ & c \notin \mathrm{Ic}(N) \cup \mathrm{Tc}(N) \mid \mathsf{in}_N : c \in C_N \}. \end{array}$$

For any marking C of control conditions of the net K and marking C' of control conditions of the net N, define

$$\begin{array}{rcl} \theta_K(C) & \triangleq & P \triangleleft \mathsf{in}_K : C \\ \theta_N(C) & \triangleq & P \triangleright \mathsf{in}_N : C. \end{array}$$

For an event e of $K[N]$, define $\rho_N(e) = e'$ for the unique e' such that $e = (P_\mathsf{i} \cup P_\mathsf{t}) \triangleright \mathsf{in}_N : e'$. For an event e of N, define $\theta_N(e) = (P_\mathsf{i} \cup P_\mathsf{t}) \triangleright \mathsf{in}_N : e$. Define $\rho_K(e)$ and $\theta_K(e)$ similarly, apart from having $\theta_K([-])$ undefined.

Lemma 6.2.1. *For any marking C of control conditions of K, the marking $\theta_K(C)$ is a K-marking in $K[N]$. For any marking C' of control conditions of N, the marking $\theta_N(C')$ is an N-marking in $K[N]$.*

Proof. Immediate from the definitions. □

It is clear that ρ_N and θ_N form a bijection between N-events and $\mathrm{Ev}(N)$. It is also clear that ρ_K and θ_K form a bijection between K-events and $\mathrm{Ev}(K) \setminus \{[-]\}$. On markings, the situation is a little more intricate:

Lemma 6.2.2. *Let $K[N]$ be a substitution. For any marking of control conditions $C_K \cup C_N$ of $K[N]$ that is a K-marking and any marking C of control conditions of K:*

$$\theta_K(\rho_K(C_K \cup C_N)) = C_K \cup C_N \quad \textit{and} \quad \rho_K(\theta_K(C)) = C.$$

For any marking of control conditions $C_K \cup C_N$ of $K[N]$ that is an N-marking and any marking C of control conditions of N:

$$\theta_N(\rho_N(C_N)) = C_N \quad \textit{and} \quad \rho_N(\theta_N(C)) = C.$$

Proof. First, let C be any marking of control conditions of K. We shall show that $\rho_K(\theta_K(C)) = C$. Let c be any control condition of the net K. We earlier assumed that ${}^\bullet[-] \cap [-]^\bullet = \emptyset$, so there are three distinct cases: $c \notin {}^\bullet[-]^\bullet$, $c \in {}^\bullet[-]$ or $c \in [-]^\bullet$. The first case is straightforward since the operation of θ_K on such conditions is to add a '$\mathsf{in}_K :$'-tag which is removed by ρ_K. Now consider $c \in {}^\bullet[-]$; the case for $c \in [-]^\bullet$ will be similar. By the definition of θ_K, since $\mathrm{Ic}(N)$ is nonempty (by Lemma 3.5.1):

$$c \in C \qquad \text{iff} \qquad \forall i \in \mathrm{Ic}(N).(\mathsf{in}_K : c, \mathsf{in}_N : i) \in \theta_K(C).$$

From the definition of ρ_K, we have $\forall i \in \mathrm{Ic}(N).(\mathsf{in}_K : c, \mathsf{in}_N : i) \in \theta_K(C)$ iff $c \in \rho_K(\theta_K(C))$. So $c \in C$ iff $c \in \rho_K(\theta_K(C))$.

Now suppose that (C_K, C_N) is a K-marking of the substitution $K[N]$. Let c be any condition of the net $K[N]$. There are three distinct possible cases: $c \notin P_\mathsf{i} \cup P_\mathsf{t}$, $c \in P_\mathsf{i}$ or $c \in P_\mathsf{t}$. First, suppose that $c \notin P_\mathsf{i} \cup P_\mathsf{t}$:

$$\begin{array}{lll} c \in C_N \cup C_K & \text{iff} \quad c \in C_K & \text{(def. of } K\text{-marking)} \\ & \text{iff} \quad \exists c_1.(c_1 \in \rho_K(C_N, C_K) \text{ and } c = \mathsf{in}_K : c_1) & \text{(def. of } \rho_K) \\ & \text{iff} \quad c \in \theta_K(\rho_K(C_N \cup C_K)) & \text{(def. of } \theta_K) \end{array}$$

Now suppose that $c \in P_\mathsf{i}$, so $c = (\mathsf{in}_K : a, \mathsf{in}_N : i)$ for some $a \in {}^\bullet[-]$ and $i \in \mathrm{Ic}(N)$:

$$\begin{array}{lll} c \in C_N \cup C_K & \text{iff} \quad \forall i' \in \mathrm{Ic}(N).\ (\mathsf{in}_K : a, \mathsf{in}_N : i') \in C_N \cup C_K & \\ & & \text{(def. of } K\text{-marking)} \\ & \text{iff} \quad a \in \rho_K(C_N \cup C_K) & \text{(def. of } \rho_K) \\ & \text{iff} \quad c \in \theta_K(\rho_K(C_N \cup C_K)) & \text{(def. of } \theta_K) \end{array}$$

We have a similar analysis if $c \in P_\mathsf{t}$. Hence $(C_K, C_N) = \theta_K(\rho_K(C_K, C_N))$.

For any marking of control conditions C of the net N and any N-marking (C_K, C_N),

$$\theta_N(\rho_N(C_N)) = C_N \quad \text{and} \quad \rho_N(\theta_N(C)) = C$$

are shown similarly, this time with the first analysis considering conditions in $\mathrm{Ic}(N)$, $\mathrm{Tc}(N)$ and conditions not in either set. □

The operations also relate the behaviour of the nets K, N and $K[N]$. We first show how behaviour in $\mathcal{C}(K[N])$ gives rise to behaviour in $\mathcal{C}(K)$ and $\mathcal{C}(N)$. For our purposes, it will be sufficient to consider only the cases where K-events occur in K-markings and N-events occur in N-markings, and not for example K-events occurring in N-markings.

Lemma 6.2.3. *Let (C_K, C_N) and (C'_K, C'_N) be markings of $\mathcal{C}(K[N])$. Suppose that e is an event such that $(C_K, C_N) \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} (C'_K, C'_N)$.*

1. *If e is a K-event and (C_K, C_N) and (C'_K, C'_N) are K-markings then $\rho_K(C_K, C_N) \overset{\rho_K(e)}{\longrightarrow\!\!\!\!\!\rightarrow} \rho_K(C'_K, C'_N)$ in $\mathcal{C}(K)$.*
2. *If e is an N-event and (C_K, C_N) and (C'_K, C'_N) are N-markings then $C_K = C'_K$ and $\rho_N(C_N) \overset{\rho_N(e)}{\longrightarrow\!\!\!\!\!\rightarrow} \rho_N(C'_N)$ in $\mathcal{C}(N)$.*

Proof. First consider (1). The event e is a K-event, so there is an event e_1 of K such that $e_1 \neq [-]$ and $e = (P_{\mathsf{i}} \cup P_{\mathsf{t}}) \lhd \mathsf{in}_K : e_1$. We have

$$(C_N, C_K) \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} (C'_N, C'_K)$$

in $\mathcal{C}(K[N])$. By Lemma 6.2.2, we have $\theta_K(\rho_K(C_N, C_K)) = (C_N, C_K)$ and $\theta_K(\rho_K(C'_N, C'_K)) = (C'_N, C'_K)$. From the definition of θ_K, we therefore have

$$(P_{\mathsf{i}} \cup P_{\mathsf{t}}) \lhd \mathsf{in}_K : \rho_K(C_N, C_K) \overset{(P_{\mathsf{i}} \cup P_{\mathsf{t}}) \lhd \mathsf{in}_K : e_1}{\longrightarrow\!\!\!\!\!\rightarrow} (P_{\mathsf{i}} \cup P_{\mathsf{t}}) \lhd \mathsf{in}_K : \rho_K(C'_N, C'_K).$$

Using Lemma 3.3.2, we may therefore conclude that

$$\rho_K(C_N, C_K) \overset{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} \rho_K(C'_N, C'_K)$$

in $\mathcal{C}(K)$. The proof of (2) is similar. □

Next, we show how behaviour in K and N gives rise to behaviour in $K[N]$.

Lemma 6.2.4. *1. Let C and C' be markings of control conditions of K. If $C \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ in $\mathcal{C}(K)$ then $\theta_K(C) \overset{\theta_K(e)}{\longrightarrow\!\!\!\!\!\rightarrow} \theta_K(C')$ in $\mathcal{C}(K[N])$.*

2. Now let C and C' be markings of control conditions of N. If $C \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ in $\mathcal{C}(N)$ then $(\theta_N(C), C_K) \overset{\theta_N(e)}{\longrightarrow\!\!\!\!\!\rightarrow} (\theta_N(C'), C_K)$ in $K[N]$ for any marking C_K of K-conditions.

Proof. First consider (1). Suppose that $C \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ in $\mathcal{C}(K)$ for some event $e \neq [-]$. By Lemma 3.3.2, we have

$$(P_{\mathsf{i}} \cup P_{\mathsf{t}}) \lhd \mathsf{in}_K : C \overset{(P_{\mathsf{i}} \cup P_{\mathsf{t}}) \lhd \mathsf{in}_K : e}{\longrightarrow\!\!\!\!\!\rightarrow} (P_{\mathsf{i}} \cup P_{\mathsf{t}}) \lhd \mathsf{in}_K : C'$$

in $\mathcal{C}(K[N])$. Since $\theta_K(C) = (P_{\mathsf{i}} \cup P_{\mathsf{t}}) \lhd \mathsf{in}_K : C$, and similarly for C' and e, we therefore have

$$\theta_K(C) \overset{\theta_K(e)}{\longrightarrow\!\!\!\!\!\rightarrow} \theta_K(C'),$$

as required. The proof of (2) is similar. □

We are now able to characterize the runs of the net $\mathcal{C}(K[N])$ when a control non-interfering substitution is formed, thereby gaining an understanding of the control flow of the net $K[N]$ when a non-interfering substitution is formed.

Lemma 6.2.5. *Let $\mathcal{C}(K[N])$ be a control non-interfering substitution. Any complete sequence π from $\mathrm{Ic}(K[N])$ in $\mathcal{C}(K[N])$ is of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^*$, where:*

- *Π_0 ranges over sequences consisting of K-events between K-markings.*
- *Π_1 ranges over nonempty sequences π_1 of K- and N-events between N-markings. If $C_N \cup C_K$ and $C'_N \cup C'_K$ are the initial and final markings of π_1, respectively, then $C_N = P_{\mathsf{i}}$ and $C'_N = P_{\mathsf{t}}$. The first event of π_1 is an N-event and the final event of π_1 is also an N-event. Every marking reached along π_1 is N-active apart from the final marking.*

Proof. We first show that any sequence π in $\mathcal{C}(N)$ from $\mathrm{Ic}(K[N])$ is of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^*$ or $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^* \cdot \Pi'_1$ by induction on the length of sequence, where a sequence π_1 is of form Π'_1 if:

- it is a nonempty sequence of K- and N-events between N-markings,
- every marking reached along π_1 is N-active, and
- if (C_N, C_K, σ) is the initial marking of π_1 then $C_N = P_{\mathsf{i}}$, and the first event of π_1 is an N-event.

The base case of the induction is trivial since $\mathrm{Ic}(K[N])$ is a K-marking. For the inductive case, suppose that $\pi \cdot e$ is a path such that

$$\mathcal{C}(K[N]){:}\mathrm{Ic}(K[N]) \xrightarrow{\pi}\!\!\!\!\rightarrow (C_N, C_K) \xrightarrow{e}\!\!\!\!\rightarrow (C'_N, C'_K).$$

By induction, the path π is either of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^*$ or $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^* \cdot \Pi'_1$.

First suppose that π is of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^*$. The marking (C_N, C_K) is therefore a K-marking and K-events preserve K-markings, so, if e is a K-event, the path $\pi \cdot e$ is also of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^*$. Now suppose that e is an N-event; there exists e_0 such that $e = (P_{\mathsf{i}} \cup P_{\mathsf{t}}) \triangleright \mathsf{in}_N : e_0$. The event e_0 has a precondition not in $\mathrm{Tc}(N)$ according to Lemma 3.5.1. Hence e has a precondition c, and this cannot be N-internal since (C_N, C_K) is a K-marking. According to the definition of ${}^\bullet e$ as $(P_{\mathsf{i}} \cup P_{\mathsf{t}}) \triangleright \mathsf{in}_N : {}^\bullet e_0$, it must therefore be the case that $c = (\mathsf{in}_N : i, \mathsf{in}_K : a)$ for some $i \in \mathrm{Ic}(N)$ and $a \in {}^\bullet[-]$. Again according to the definition of ${}^\bullet e$, we have $(\mathsf{in}_N : i, \mathsf{in}_K : a') \in {}^\bullet e$ for all $a' \in {}^\bullet[-]$. Recalling that ${}^\bullet e \subseteq (C_N, C_K)$ and that (C_N, C_K) is a K-marking, we must therefore have $P_{\mathsf{i}} \subseteq (C_N, C_K)$. We assumed that $K[N]$ was control non-interfering, so in fact $C_N = P_{\mathsf{i}}$. It follows that (C_N, C_K) is also an N-marking, and N-events preserve the property of being an N-marking. If the marking (C'_N, C'_K) is an N-active marking, the path $\pi \cdot e$ is therefore of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^* \cdot \Pi'_1$. Otherwise, if the marking (C'_N, C'_K) is not N-active, we have $P_{\mathsf{t}} \subseteq C'_N$. By Lemma 6.2.3, in the net $\mathcal{C}(N)$ we have $\mathrm{Ic}(N) \xrightarrow{e_0}\!\!\!\!\rightarrow \rho_N(C'_N)$. It follows from the requirement that N should be well-terminating (Definition 3.5.1) that we must therefore have $C'_N = P_{\mathsf{t}}$. The path $\pi \cdot e$ is then readily seen to be of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^*$.

Now suppose that π is of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^* \cdot \Pi'_1$. The marking (C_N, C_K) is therefore an N-active marking. First suppose that e is a K-event. We have ${}^\bullet e^\bullet \cap (P_{\mathsf{i}} \cup P_{\mathsf{t}}) = \emptyset$ since $K[N]$ is control non-interfering. It follows that the marking $C_N = C'_N$ and therefore (C'_N, C'_K) is N-active. The path $\pi \cdot e$ is therefore of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^* \cdot \Pi'_1$. Now suppose that e is an N-event. The marking (C'_N, C'_K) is an N-marking since N-events preserve N-markings. If (C'_N, C'_K) is N-active, the path $\pi \cdot e$ is easily seen to of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^* \cdot \Pi'_1$. Suppose instead that (C'_N, C'_K) is not N-active. Let π_1 be the path for which there exists π_0 such that $\pi = \pi_0 \cdot \pi_1$ and π_0 is of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^*$ and π_1

is of the form Π_1'. By stripping away the K-events from π_1, using Lemma 6.2.3 it is easy to see that we obtain a path in $\mathcal{C}(N)$ from $\mathrm{Ic}(N)$ to C_N'. By definition, since (C_N', C_K') is not N-active, we must have $P_\mathsf{t} \subseteq C_N$. From well-termination of N, we therefore have $C_N' = P_\mathsf{t}$ and so the path $\pi \cdot e$ will be of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)^*$.

We conclude the proof by showing that any complete run π of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0) \cdot \Pi_1'$ is also of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)$. Let (C_N, C_K) be the marking of control conditions reached by π. We have $\mathrm{Tc}(K[N]) = (C_N, C_K)$ because the run is complete, and this is a K-marking. It is easy to see that there are four cases for C_N that could make (C_N, C_K) both an N- and a K-marking, *viz.*

$$C_N = P_\mathsf{i} \qquad C_N = P_\mathsf{t} \qquad C_N = P_\mathsf{i} \cup P_\mathsf{t} \qquad C_N = \emptyset.$$

We saw earlier that $\rho_N(C_N)$ is reachable from $\mathrm{Ic}(N)$ in N. Since the net N is assumed to be well-terminating, we must therefore have $C_N \neq P_\mathsf{i} \cup P_\mathsf{t}$. From the requirements on N in Lemma 3.5.1, it is also easy to see that $C_N \neq \emptyset$. If we had $C_N = P_\mathsf{i}$, we would also have ${}^\bullet[-] \subseteq \mathrm{Tc}(K)$, contradicting the requirements for K in Lemma 3.5.1. It follows that $C_N = P_\mathsf{t}$ and therefore that the path π is of the form $\Pi_0 \cdot (\Pi_1 \cdot \Pi_0)$, thus completing the proof. □

Having now dealt with the control structure of contexts, we return to the idea that, given a net $K[N_1]$ which is a non-interfering substitution from state σ, the events in any sequence may be reordered in a way that ensures that events of N_1 occur consecutively and form a "complete run" of the net N_1. As $N_1 \simeq N_2$, the net $K[N_2]$ will therefore have a path between the same sets of state conditions.

To formalize this, let π be any sequential run of a non-interfering substitution $K[N]$ from marking M. The set $\mathcal{P}_{K[N]}(\pi, M)$ is defined to be the least set of sequences from marking M of $K[N]$ closed under the operation of swapping consecutive independent events that contains the sequence π. It is easy to see that if $M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ and $\pi' \in \mathcal{P}_{K[N]}(\pi, M)$ then $M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ for any paths π and π'. Define the order $\prec$ on $\mathcal{P}_{K[N]}(\pi, M)$ as follows:

Definition 6.2.5. *Let $\pi, \pi' \in \mathcal{P}_{K[N]}(\pi_0, M)$. Define $\prec$ to be the transitive closure of $\prec_1$, where $\pi \prec_1 \pi'$ iff there exist sequences π_1 and π_2, an N-event e and a K-event e' such that eIe' and $\pi = \pi_1 \cdot e \cdot e' \cdot \pi_2$ and $\pi' = \pi_1 \cdot e' \cdot e \cdot \pi_2$.*

It is clear that the order $\prec$ is well-founded since any sequence is, by definition, of finite length.

Definition 6.2.6. *Say that a sequence π of $K[N]$ from marking M is* N-complete *if $M = (P_\mathsf{i}, C_K, \sigma)$ for some C_K and σ, every event of π is an N-event, and*

$$(P_\mathsf{i}, C_K, \sigma) \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} (P_\mathsf{t}, C_K, \sigma').$$

Lemma 6.2.6. *Let $K[N]$ be a non-interfering substitution from state σ_0 and let $M_0 = (\mathrm{Ic}(K[N]), \sigma_0)$. Suppose that π_0 is a complete sequence of $K[N]$ from M_0. The $\prec$-minimal elements of $\mathcal{P}_{K[N]}(\pi_0, M_0)$ are of the form*

$$\Pi_0 \cdot (\Pi_N \cdot \Pi_0)^*,$$

where Π_N matches N-complete paths and Π_0 is as in Lemma 6.2.5.

Proof. Suppose that π is a $\prec$-minimal element of $\mathcal{P}_{K[N]}(\pi_0, M_0)$ but not of the form above. The sequence π is of the form of Lemma 6.2.5 because π is a complete path of $K[N]$. Consequently, there are π_1, π_2 and π_3 such that $\pi = \pi_1 \cdot \pi_2 \cdot \pi_3$ and $\pi_2 = (e \cdot e')$ where e is a K-event and e' is an N-event. Furthermore, the marking M_1 such that $M_0 \stackrel{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} M_1$ is N-active. Now, from the definition of non-interfering substitution, the events e and e' are independent. Hence the sequence $\pi_1 \cdot e' \cdot e \cdot \pi_3$ is in $\mathcal{P}_{K[N]}(\pi_0, M_0)$ and is beneath π, contradicting its minimality. □

This gives us the ability to prove the key result by induction on paths of $K[N_1]$.

Theorem 6.1. *If $K[N_1]$ and $K[N_2]$ are non-interfering substitutions from σ_0 and $N_1 \simeq N_2$ then, for all states σ':*

$$K[N_1]{:}\sigma_0 \Downarrow \sigma' \qquad \Longleftrightarrow \qquad K[N_2]{:}\sigma_0 \Downarrow \sigma'.$$

Proof. Suppose that π is a complete sequence of $K[N_1]$ from σ_0 to σ'. We shall show that, for all $\pi_1 \in \mathcal{P}_{K[N_1]}(\pi, (\mathrm{Ic}(K[N_1]), \sigma_0))$, if π_1 is a complete sequence from σ_0 to σ' then there exists a complete sequence π_2 of $K[N_2]$ from σ_0 to σ'. The proof shall proceed by induction on the well-founded order $\prec$. In particular $\pi \in \mathcal{P}_{K[N_1]}(\pi, (\mathrm{Ic}(K[N_1]), \sigma_0))$, so, with the symmetric proof for the other direction, this will complete the proof of the required property.

π_1 minimal: The sequence π_1 is minimal within $\mathcal{P}_{K[N_1]}(\pi, (\mathrm{Ic}(K[N_1]), \sigma_0))$, so, by Lemma 6.2.6, there exists an $n \in \mathbb{N}$ such that there exist sequences $\pi_0, \pi_{01}, \pi_{11}, \ldots \pi_{0n}, \pi_{1n}$ with

$$\pi_1 = \pi_0 \cdot \pi_{11} \cdot \pi_{01} \ldots \pi_{1n} \cdot \pi_{0n}.$$

Furthermore, for each $i \leq n$, the sequence π_{0i} is of the form Π_0 defined in Lemma 6.2.5, as is the sequence π_0; and, for each $i \leq n$, the sequence π_{1i} is of the form Π_{N_1}, which matches N_1-complete subpaths of $K[N_1]$ as defined in Definition 6.2.6. Define:

$$\begin{array}{llll} P_{\mathsf{i}}^{(1)} & \triangleq & \mathsf{in}_K : {}^{\bullet}[-] \times \mathsf{in}_{N_1} : \mathrm{Ic}(N_1) & P_{\mathsf{i}}^{(2)} \triangleq \mathsf{in}_K : {}^{\bullet}[-] \times \mathsf{in}_{N_2} : \mathrm{Ic}(N_2) \\ P_{\mathsf{t}}^{(1)} & \triangleq & \mathsf{in}_K : [-]^{\bullet} \times \mathsf{in}_{N_1} : \mathrm{Tc}(N_1) & P_{\mathsf{t}}^{(2)} \triangleq \mathsf{in}_K : [-]^{\bullet} \times \mathsf{in}_{N_2} : \mathrm{Tc}(N_2). \end{array}$$

Let $\rho_K^{(1)}$ be ρ_K from Definition 6.2.4 for $K[N_1]$ and let $\rho_K^{(2)}$ be ρ_K from Definition 6.2.4 for $K[N_2]$, and similarly for $\rho_{N_1}^{(1)}$, $\rho_{N_2}^{(2)}$, $\theta_K^{(1)}$, *etc.* We shall show, by induction on n, that if π_1 is a sequence of this form in $K[N_1]$ from $(\mathrm{Ic}(K[N_1]), \sigma_0)$ to the marking (C_1', σ') then there exists a path π_2 from $(\mathrm{Ic}(K[N_2]), \sigma_0)$ to (C_2', σ') for some C_2' such that $\rho_K^{(1)}(C_1') = \rho_K^{(2)}(C_2')$.

- $n = 0$: Then π_1 is of the form Π_0. Let $\pi_1 = (e_1 \cdot \ldots \cdot e_m)$ and suppose that in $K[N_1]$ we have

$$(\mathrm{Ic}(K[N_1]), \sigma_0) \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} (C_1, \sigma_1) \stackrel{e_2}{\longrightarrow\!\!\!\!\!\rightarrow} \ldots \stackrel{e_m}{\longrightarrow\!\!\!\!\!\rightarrow} (C_m, \sigma_m).$$

By assumption, π_1 is a sequence from $(\mathrm{Ic}(K[N_1]), \sigma_0)$ to (C_1', σ'), so $C_1' = C_m$ and $\sigma' = \sigma_m$. Now, $\mathrm{Ic}(K[N_1])$ is a K-marking, and, since π_1 is of the form Π_0, for every i such that $0 < i \leq m$, the marking C_i is a K-marking and e_i is a K-event. By Lemma 6.2.3, in the net K we have

$$(\rho_K^{(1)}(\mathrm{Ic}(K[N_1])), \sigma) \stackrel{\rho_K^{(1)}(e_1)}{\longrightarrow\!\!\!\!\!\rightarrow} (\rho_K^{(1)}(C_1), \sigma_1) \stackrel{\rho_K^{(1)}(e_2)}{\longrightarrow\!\!\!\!\!\rightarrow} \ldots \stackrel{\rho_K^{(1)}(e_m)}{\longrightarrow\!\!\!\!\!\rightarrow} (\rho_K^{(1)}(C_m), \sigma_m).$$

In the net $K[N_2]$, by Lemma 6.2.4, we therefore have

$$\begin{array}{lll}(\theta_K^{(2)}\rho_K^{(1)}(\mathrm{Ic}(K[N_1])),\sigma) & \overset{\theta_K^{(2)}\rho_K^{(1)}(e_1)}{\longrightarrow\!\!\!\!\!\rightarrow} & (\theta_K^{(2)}\rho_K^{(1)}(C_1),\sigma_1)\\ & \overset{\theta_K^{(2)}\rho_K^{(1)}(e_2)}{\longrightarrow\!\!\!\!\!\rightarrow} & \ldots\\ & \overset{\theta_K^{(2)}\rho_K^{(1)}(e_m)}{\longrightarrow\!\!\!\!\!\rightarrow} & (\theta_K^{(2)}\rho_K^{(1)}(C_m),\sigma_m).\end{array}$$

Let $C_2' = \theta_K^{(2)}\rho_K^{(1)}(C_m)$. From Lemma 6.2.1, $\theta_K^{(2)}$ generates K-markings of $K[N_2]$ from markings of K. By Lemma 6.2.2, we therefore have $\rho_K^{(2)}(C_2') = \rho_K^{(1)}(C_1')$ since $C_1' = C_m$, which is a K-marking. It is an easy calculation to show that $\rho_K^{(1)}(\mathrm{Ic}(K[N_1])) = \mathrm{Ic}(K)$ and $\theta_K^{(2)}(\mathrm{Ic}(K)) = \mathrm{Ic}(K[N_2])$. There therefore exists a path from the marking $(\mathrm{Ic}(K[N_2]),\sigma_0)$ to (C_2',σ_m) in $K[N_2]$ and $\rho_K^{(2)}(C_2') = \rho_K^{(1)}(C_1')$, which is all that is required since $\sigma_m = \sigma'$.

- $n > 0$: Assume that $\pi_1 = \pi_{11}\cdot\pi_{12}\cdot\pi_{13}$ for some sequence π_{11} of form $\Pi_0\cdot(\Pi_{N_1}\cdot\Pi_0)^{n-1}$, some sequence π_{12} of form Π_{N_1} and some sequence π_{13} of form Π_0. Let (C_1',σ') be the marking obtained by following π_1 from $(\mathrm{Ic}(K[N_1]),\sigma_0)$ in $K[N_1]$. We wish to show that there is a path π_2 of $K[N_2]$ from $(\mathrm{Ic}(K[N_2]),\sigma_0)$ to (C_2',σ') for some C_2' such that $\rho_K^{(1)}(C_1') = \rho_K^{(2)}(C_2')$.

  Let (C_{11},σ_1) be the marking obtained by following path π_{11} from $(\mathrm{Ic}(K[N_1]),\sigma_0)$. Since π_{12} follows π_{11} and π_{12} is of form Π_{N_1}, it must be the case that $C_{11} = (P_{\mathsf{i}}^{(1)}, C_K)$ for some marking C_K of K-conditions.

  By induction, there is a path π_{21} in $K[N_2]$ from $(\mathrm{Ic}(K[N_2]),\sigma_0)$ to (C_{21},σ_1) for some C_{21} such that $\rho_K^{(1)}(C_{11}) = \rho_K^{(2)}(C_{21})$. Now, $C_{11} = (P_{\mathsf{i}}^{(1)}, C_K)$, so $\rho_K^{(1)}(C_{11}) = {}^\bullet[-]\cup\{c \mid \mathsf{in}_K : c \in C_K\}$. From the definition of $\rho_K^{(2)}$, we must therefore have $C_{21} = (P_{\mathsf{i}}^{(2)}, C_K)$. Hence

$$(\mathrm{Ic}(K[N_2]),\sigma_0) \overset{\pi_{21}}{\longrightarrow\!\!\!\!\!\rightarrow} (P_{\mathsf{i}}^{(2)}, C_K, \sigma_1).$$

  Suppose that in $K[N_1]$ we have $(P_{\mathsf{i}}^{(1)}, C_K, \sigma_1) \overset{\pi_{12}}{\longrightarrow\!\!\!\!\!\rightarrow} (C_{N_1}', C_K', \sigma_2)$. Since π_{12} is of the form Π_{N_1}, it is an N_1-complete path, so $C_{N_1}' = P_{\mathsf{t}}^{(1)}$. The events of π_{12} are all N_1-events. Using Lemma 6.2.3, a simple induction shows that $C_K = C_K'$ and that there is a path from $(\rho_{N_1}^{(1)}(P_{\mathsf{i}}^{(1)}),\sigma_1)$ to $(\rho_{N_1}^{(1)}(P_{\mathsf{t}}^{(1)}),\sigma_2)$ in N_1. Observe that $\rho_{N_1}^{(1)}(P_{\mathsf{i}}^{(1)}) = \mathrm{Ic}(N_1)$ and $\rho_{N_1}^{(1)}(P_{\mathsf{t}}^{(1)}) = \mathrm{Tc}(N_1)$, so $N_1{:}\sigma_1 \Downarrow \sigma_2$. As $N_1 \simeq N_2$, there is therefore a path of N_2 from $(\mathrm{Ic}(N_2),\sigma_1)$ to $(\mathrm{Tc}(N_2),\sigma_2)$. By Lemma 6.2.4, a simple induction on the length of this sequence shows that there is a sequence π_{22} from $(\theta_{N_2}^{(2)}(\mathrm{Ic}(N_2)), C_K, \sigma_1)$ to $(\theta_{N_2}^{(2)}(\mathrm{Tc}(N_2)), C_K, \sigma_2)$ in $K[N_2]$. Observe that $\theta_{N_2}^{(2)}(\mathrm{Ic}(N_2)) = P_{\mathsf{i}}^{(2)}$ and $\theta_{N_2}^{(2)}(\mathrm{Tc}(N_2)) = P_{\mathsf{t}}^{(2)}$, so

$$(P_{\mathsf{i}}^{(2)}, C_K, \sigma_1) \overset{\pi_{22}}{\longrightarrow\!\!\!\!\!\rightarrow} (P_{\mathsf{t}}^{(2)}, C_K, \sigma_2).$$

  As π_{13} follows path π_{12} in π_1, the sequence π_{13} is from $(P_{\mathsf{t}}^{(1)}, C_K, \sigma_2)$ to (C_1',σ') and contains only K-events because it is of form Π_0. Using Lemma 6.2.3, a simple induction on the length of π_{13} shows that there is a path from $(\rho_K^{(1)}(P_{\mathsf{t}}^{(1)}, C_K),\sigma_2)$ to $(\rho_K^{(1)}(C_1'),\sigma')$ in K. A simple induction on the length of this path using Lemma

6.2.4 shows that there is a path π_{23} of $K[N_2]$ such that $(\theta_K^{(2)}\rho_K^{(1)}(P_{\mathsf{t}}^{(1)}, C_K), \sigma_2) \xrightarrow{\pi_{23}} (\theta_K^{(2)}\rho_K^{(1)}(C_1'), \sigma')$. From the definition of $\rho_K^{(1)}$, we have $\rho_K^{(1)}(P_{\mathsf{t}}^{(1)}, C_K) = [-]^\bullet \cup \{c \mid \mathsf{in}_K : c \in C_K\}$. From the definition of $\theta_K^{(2)}$, we have $\theta_K^{(2)}([-]^\bullet \cup \{c \mid \mathsf{in}_K : c \in C_K\}) = P_{\mathsf{t}}^{(2)} \cup C_K$. Hence

$$(P_{\mathsf{t}}^{(2)}, C_K, \sigma_2) \xrightarrow{\pi_{23}} (\theta_K^{(2)}\rho_K^{(1)}(C_1'), \sigma').$$

Take $C_2' = (\theta_K^{(2)}\rho_K^{(1)}(C_1'), \sigma')$. By Lemma 6.2.2, we have $\rho_K^{(2)}(C_2') = \rho_K^{(1)}(C_1')$. Consequently, the path $\pi_2 = \pi_{21} \cdot \pi_{22} \cdot \pi_{23}$ satisfies

$$(\mathrm{Ic}(K[N_2]), \sigma_0) \xrightarrow{\pi_2} (C_2', \sigma'),$$

for some C_2' such that $\rho_K(C_1') = \rho_K(C_2')$, which is all that is required to complete this inner induction.

Now, recall that π_1 is a complete sequence of $K[N_1]$, so

$$(\mathrm{Ic}(K[N_1]), \sigma_0) \xrightarrow{\pi_1} (\mathrm{Tc}(K[N_1]), \sigma').$$

From the immediately preceding induction, there exists a path π_2 of $K[N_2]$ such that $(\mathrm{Ic}(K[N_2]), \sigma_0) \xrightarrow{\pi_2} (C_2', \sigma')$ for some C_2' s.t. $\rho_K^{(1)}(\mathrm{Tc}(K[N_1])) = \rho_K^{(2)}(C_2')$. Now, clearly $\rho_K^{(1)}(\mathrm{Tc}(K[N_1])) = \mathrm{Tc}(K)$ by the definitions of ρ and $K[N_1]$. Hence $\mathrm{Tc}(K) = \rho_K^{(2)}(C_2')$, so by Lemma 6.2.2 we have $\theta_K^{(2)}(\mathrm{Tc}(K)) = C_2'$. The definition of $K[N_2]$ and θ gives $\theta_K^{(2)}(\mathrm{Tc}(K)) = \mathrm{Tc}(K[N_2])$. Hence

$$(\mathrm{Ic}(K[N_2]), \sigma_0) \xrightarrow{\pi_2} (\mathrm{Tc}(K[N_2]), \sigma'),$$

as required.

π_1 not minimal: Suppose that the path π_1 is not minimal and that π_1 is a complete path of $K[N_1]$ with $(\mathrm{Ic}(K[N_1]), \sigma_0) \xrightarrow{\pi_1} (\mathrm{Tc}(K[N_1]), \sigma')$. It is easy to see that the order $\prec$ is irreflexive, so there exists a path π_1' such that $\pi_1' \prec_1 \pi_1$. Hence there exist paths π_2 and π_3 and a K-event e and an N-event e' such that $\pi_1 = \pi_2 \cdot e \cdot e' \cdot \pi_3$ and $\pi_1' = \pi_2 \cdot e' \cdot e \cdot \pi_3$. Furthermore, the events e and e' are independent, so π_1' must also be a path $(\mathrm{Ic}(K[N_1]), \sigma_0) \xrightarrow{\pi_1'} (\mathrm{Tc}(K[N_1]), \sigma')$. By induction, there exists a path $(\mathrm{Ic}(K[N_2]), \sigma_0) \xrightarrow{\pi_2'} (\mathrm{Tc}(K[N_2]), \sigma')$, as required to complete the case.

Hence, if $K[N_1]{:}\sigma_0 \Downarrow \sigma'$, there exists a path

$$K[N_1]{:}(\mathrm{Ic}(K[N_1]), \sigma_0) \xrightarrow{\pi} (\mathrm{Tc}(K[N_1]), \sigma').$$

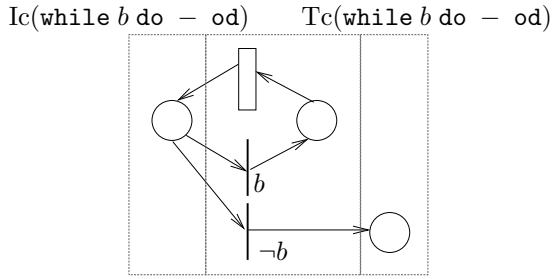
From what we have just shown, since $\pi \in \mathcal{P}_{K[N_1]}(\pi, \sigma_0)$, we have a path

$$K[N_2]{:}(\mathrm{Ic}(K[N_2]), \sigma_0) \xrightarrow{\pi_2} (\mathrm{Tc}(K[N_2]), \sigma'),$$

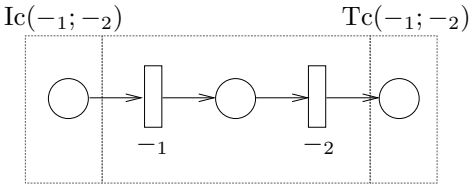
so $K[N_2]{:}\sigma_0 \Downarrow \sigma'$. The proof for the reverse implication is symmetric. □

6.3 Refinement for granularity

We can think of the definitions of net contexts and substitutions as giving a semantic way of interpreting the subprocesses of a net: a net N is a *subprocess* of the net N_0 if there exists a context K for which $N_0 \cong K[N]$, where $\cong$ represents net isomorphism. For example, the definition of the while loop construction `while` b `do` t_0 `od` in Section 3.3 is the same (up to net isomorphism) as the result of substituting $\mathcal{N}[\![t_0]\!]$ into the hole in the following context. (We draw the net without the state conditions and only draw one event for the tests of b and $\neg b$.)

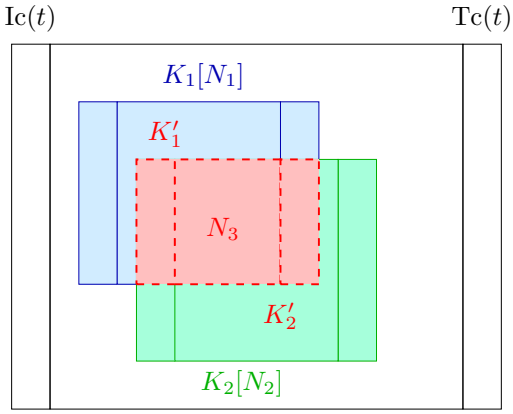


Similarly, following a slight generalization to allow contexts to have multiple holes, the net $\mathcal{N}[\![t_1; t_2]\!]$ can be obtained, up to isomorphism, by substitution of $\mathcal{N}[\![t_1]\!]$ into the hole labelled 1 and $\mathcal{N}[\![t_2]\!]$ into the hole labelled 2 in the following context:

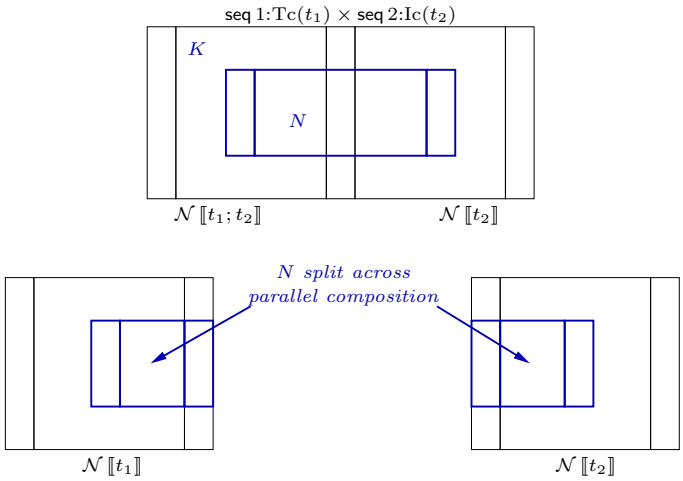


We have seen how the net semantics allows us to define the refinement operation for changing the granularity of actions providing they form non-interfering substitutions. It places us in a position where an attempt may be made to prove Reynold's' conjecture by showing proved processes are not susceptible to changes in granularity, in a way which we now outline.

Suppose that we have the net semantics $\mathcal{N}[\![t]\!]$ and wish to change the granularity of the subprocess represented by N within $\mathcal{N}[\![t]\!]$ to N'. There exists a context K such that $\mathcal{N}[\![t]\!] \cong K[N]$. To be able to apply the refinement operation described above, and in particular Theorem 6.1, we must show that the net $K[N]$ forms a non-interfering substitution. If N is a just a heap action, this follows from Theorem 5.5. In tackling the more general case, though, it is here that we encounter some difficulty in developing a workable understanding of how contexts span across the terms of the language. A general account may be developed by showing that, for any nets $K_1[N_1]$ and $K_2[N_2]$ such that $K_1[N_1] \cong K_2[N_2]$ there exists a net N_3 and contexts K_1' and K_2' such that $N_1 \cong K_1'[N_3]$ and $N_2 \cong K_2'[N_3]$. Furthermore, any condition that is both an N_1- and N_2-condition in $K_1[N_1] \cong K_2[N_2]$ is an N_3-condition in $K_1[K_1'[N_3]] \cong K_2[K_2'[N_3]]$. This may be drawn as:



For example, recalling that the sequential composition can be considered to be obtained by substitution as above, this will allow us to describe how substitutions spread across the sequential composition shown below:



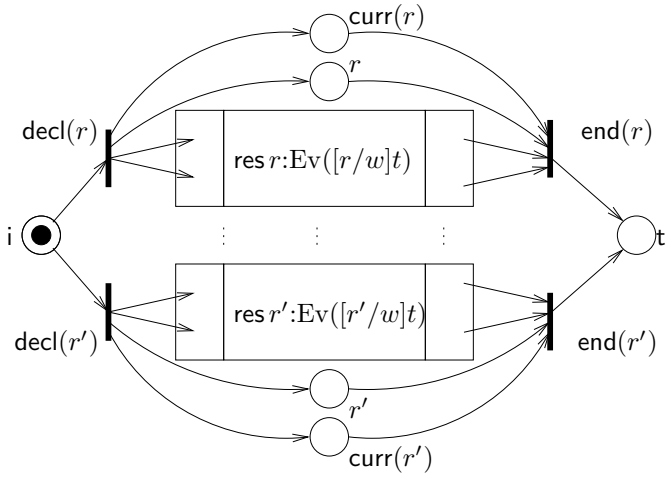
If we wish to prove that all subprocesses form non-interfering substitutions by induction on terms, this decomposition will be necessary.

Unfortunately, this result has proven difficult to obtain due to the rather complicated definition of substitution, in particular on the conditions of the net, so we must leave it for future work.

Chapter 7

Petri nets and unfolding

In the previous two chapters, we have shown how a Petri net semantics can be given to a simple programming language. It is worth considering how the net $\mathcal{N}[\![t]\!]$ representing a term t can be infinite. The first obvious way is that t might contain an action α which can act in infinitely many ways, so that the set $\mathcal{A}[\![\alpha]\!]$ is infinite. This can be overcome by working with slightly more abstract nets, where places can hold values — perhaps by using Jensen's coloured Petri nets [Jen96]. More subtly, recall that the semantics for the term `resource` w `do` t `od` involves making an arbitrary choice between the (assumedly infinitely many) non-current resources.



The distinction between whichever particular resource is chosen is of little significance: the behaviours of the subnets $\mathcal{N}[\![[r/w]t]\!]$ and $\mathcal{N}[\![[r'/w]t]\!]$ are exactly the same apart from one using r instead of r'. The particular choice made is irrelevant since we ensure that the resource chosen for w does not occur free in t. In essence, the behaviours of the subnets that can be chosen will be *symmetric*. Similarly, the infinite branching arising from the allocation primitive might be mitigated by regarding (some of) the choices made as symmetric, though the situation there is somewhat more delicate due to considerations discussed in Section 5.6.

In this chapter and the next, we study a categorical framework for defining symmetry in the behaviour of Petri nets. As remarked in the introduction, there are, of course, many ways of adjoining symmetry to nets (for example, categories of net with symmetric structure are studied in [Sas98]). The method we use was motivated by the need to extend the expressive power of event structures and the maps between them [Win07a, Win07b]. Roughly, a symmetry on a Petri net is described as a relation between its runs

as causal nets, the relation specifying when one run is similar to another up to symmetry. Technically and generally, a symmetry is a span of open morphisms that form a pseudo-equivalence.

There are compelling reasons for applying the framework to place symmetry on Petri nets other than just to abstract away symmetry in the behaviour of the net semantics for terms described above.

The key reason for defining symmetry on Petri nets is to resolve the anomalous position of the category of general Petri nets, nets whose events have pre- and post-condition *multisets* and whose initial markings are multisets. This lies outside the standard framework relating categories of models for concurrency through adjunctions. This means that constructions such as parallel composition must be defined for the different forms of net as must forms of bisimulation. As we shall see in this chapter, the reason for this is that multiplicities break the uniqueness property required to obtain an adjunction between general forms of net and *occurrence nets*, special forms of net that are useful in establishing relationships with other models for concurrency. By representing the natural symmetry in the nets arising from multiplicities, we obtain uniqueness *up to symmetry* and therefore an adjunction up to symmetry.

Another reason is that we wish to test the general framework for defining symmetry away from its original setting in defining symmetry on event structures. In fact, this has led us to drop the constraint that the span of morphisms must be jointly monic, in which case the symmetry would be an equivalence rather than a pseudo-equivalence.

Finally, symmetry is present and plays a role, at least informally, in a wide range of areas related to the analysis of concurrent systems in addition to the specific areas in our semantics described above. It is present in security protocols due to the repetition of essentially similar sessions [DGT07, FHG98, CW01a], can be exploited to increase efficiency in model checking [Sis04], and is present whenever abstract names are involved [GP01].

Specifically, in this chapter we shall study the *unfolding* operation on general Petri nets. We shall show how a general Petri net can be unfolded to obtain an occurrence net. The process of unfolding a general net to form an occurrence net is analogous to the process of forming a synchronization tree from a transition system. Occurrence nets reveal how their events causally depend on each other and how they conflict with each other through the holding of conditions, and whether the occurrence of events is concurrent. They provide a useful bridge between nets and other models for concurrency.

The unfolding operation for general Petri nets is, in fact, a relatively straightforward extension of that for safe nets in [NPW81] — see [vG05, MMS96]. However, to apply the framework for defining symmetry, we are forced to study more general forms of net that can have more than one initial marking. This necessitates us reviewing the existing construction in this new setting. Once we have done so, characterizing the unfolding in Theorem 7.12, we show how, in general, unfolding does not form a basis for a coreflection between occurrence nets and general nets.

7.1 Varieties of Petri nets

In this section, we shall give an account of the forms of Petri net that we shall use, moving beyond the basic Petri nets introduced earlier. In doing so, we use some notation on multisets which is introduced fully in Appendix A, but we summarize the notation here:

Let A be any set. We write $X \subseteq_\mu A$ if X is a multiset over A. The number of occurrences of a in X, for any $a \in A$, is written $X[a]$. A multiset can contain only finitely

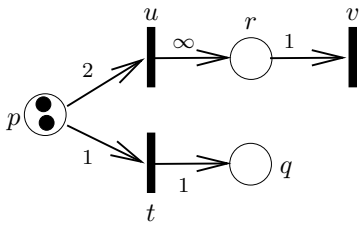
many occurrences of any particular element of A; otherwise it is an ∞-multiset, written $X \subseteq_{\mu_\infty} A$. A multiset is finite if it contains only finitely many elements. A multirelation $R \subseteq_\mu A \times B$ between sets A and B associates a natural number $X[a, b]$ to every pair $(a, b) \in A \times B$. Likewise, an infinity multirelation $R \subseteq_{\mu_\infty} A \times B$ associates either a natural number or ∞ to every pair in $A \times B$. The result of applying a multirelation $R \subseteq_\mu A \times B$ to a multiset $X \subseteq_\mu A$ is a multiset over B written $R \cdot X$. This notation is also applied to singletons, so we write $R \cdot x$ for $R \cdot \{x\}$, the multiset obtained by applying the multirelation R to the multiset with a single element, x. We write $+$ for the operation of multiset union and $-$ for the partial operation of multiset subtraction.

General nets

Just like the basic nets introduced in Chapter 2, a general Petri net consists of sets of conditions and events, though we now sometimes call these *places* and *transitions*, respectively. They are drawn in the same way as safe nets, with places drawn as circles and transitions drawn as rectangles, but with two important differences.

The first difference is that in a general net, there can be a (possibly infinite) *number* of tokens in any given place, so a marking of a general net is an ∞-multiset of places.

The second difference is that arcs from a place into a transition are now labelled with a natural number indicating the finite number of tokens in the place that are consumed by the occurrence of the transition. Arcs labelled with a natural number or ∞ may be drawn from a transition to a place, indicating the number of tokens to be deposited in the place by the occurrence of a transition. For example, the net



has an initial marking with two tokens in place p, and within the net the transition u consumes two tokens from place p and deposits infinitely many in place r.

More formally:

Definition 7.1.1 (General Petri net)**.** *A general Petri net is a 5-tuple,*

$$G = (P, T, Pre, Post, \mathbb{M}),$$

of which

- P *is the set of* places*;*
- T *is the set of* transitions*, disjoint from* P*;*
- Pre *is the* precondition *multirelation,* $Pre \subseteq_\mu T \times P$
- $Post$ *is the* postcondition ∞*-multirelation,* $Post \subseteq_{\mu_\infty} T \times P$*; and*
- $\mathbb{M}$ *is a set of* ∞*-multisets of* P*, forming the set of* initial markings *of* G*.*

Every transition must consume at least one token:

$$\forall t \in T \exists p \in P.\ Pre[t, p] > 0.$$

This is a mild generalization of the standard definition of Petri net in that we allow there to be a *set* of initial markings rather than just one initial marking, and will prove important later. We shall call a net *singly-marked* if it follows the standard definition, having only one initial marking. We shall sometimes, when necessary to explicitly disambiguate the the old singly-marked nets from the nets with sets of initial markings introduced here, call the new nets *multiply-marked*. Be aware, however, that the set of markings of a multiply-marked net could be a singleton set or even empty. The reason for extending the definition to allow multiple initial markings is that, without them, simple symmetries on nets would be inexpressible. A fuller explanation shall be given on page 168, once the framework for defining symmetry in models has been introduced. A guiding intuition in forming the definitions of multiply-marked nets, particularly in the coming extended definition of occurrence net, shall be that each initial marking can be thought of as being given rise to by some special, hidden event that is in conflict with all the other events giving rise to the other initial markings.

The marking of the net changes through the occurrence of transitions according to what is commonly called the *collective token game* for nets. For any place p, the occurrence of a transition t in marking M consumes $Pre[t,p]$ tokens from p and deposits $Post[t,p]$ tokens in p. The transition can only occur if $M[p]$, the number of tokens in p, is greater than or equal to $Pre[t,p]$. The token game extends to finite multisets of transitions, giving rise to a relation between markings labelled by a finite multiset of transitions $M \stackrel{X}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ defined[1] as

$$M \stackrel{X}{\longrightarrow\!\!\!\!\!\rightarrow} M' \text{ iff } Pre \cdot X \leq M \text{and } M' = M - Pre \cdot X + Post \cdot X.$$

Note that this is different from the token game for basic nets introduced in Chapter 2 since contact does not inhibit transitions: the presence of a token in a post-place does not prevent the occurrence of a transition; the token just remains in the place with the newly-added tokens. We shall use the same $\longrightarrow\!\!\!\!\!\rightarrow$ notation with the understanding that from here on it only refers to this token game, not the token game for basic nets.

The transition relation yields a notion of *reachable marking*, saying that a marking M' is reachable if there is some initial marking $M \in \mathbb{M}$ from which, following some finite sequence of transitions, the marking M' is obtained. We shall use the old notation $G{:}M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ to mean that the finite sequence of events π can occur from marking M to yield marking M', and shall often drop the $G:$ prefix where G is obvious from the context.

To give an example of the token game, in the net above the transition u can occur in the initial marking drawn:

$$\{p \mapsto 2\} \stackrel{u}{\longrightarrow\!\!\!\!\!\rightarrow} \{r \mapsto \infty\}$$

This yields a marking with infinitely many tokens residing in the place r. For any finite multiset V only containing occurrences of v, the following step can be obtained:

$$\{r \mapsto \infty\} \stackrel{V}{\longrightarrow\!\!\!\!\!\rightarrow} \{r \mapsto \infty\}$$

As such, the transition v can occur concurrently with itself. This is normally called *auto-concurrency*.

Before moving on, it is worth noting that we have placed a very slight restriction on general nets that $Pre \cdot t$ is a non-null multiset, *i.e.* every transition has at least one pre-place. Amongst other things, this makes defining the occurrence net unfolding of a general net much less technically arduous.

[1] Even though multiset subtraction may be undefined due to subtraction of infinity, $M - Pre \cdot X$ is always defined if $Pre \cdot X \leq M$ since X is finite and Pre is a multirelation rather than an ∞-multirelation.

Net morphisms

We saw earlier that morphisms between Petri nets, as introduced in [Win84], reveal how the structure of one net embeds into that of another in a way that preserves the behaviour of the original net by preserving the token game for nets. The definition was extended to general Petri nets in [Win87].

Definition 7.1.2 (General net morphisms)**.** *Let* $G = (P, T, Pre, Post, \mathbb{M})$ *and* $G' = (P', T', Pre', Post', \mathbb{M}')$ *be general Petri nets. A morphism* $(\eta, \beta){:}G \to G'$ *is a pair consisting of a partial function* $\eta{:}T \to_* T'$ *and an* ∞*-multirelation* $\beta \subseteq_{\mu_\infty} P \times P'$ *which jointly satisfy:*

- *for all* $M \in \mathbb{M}$*:* $\beta \cdot M \in \mathbb{M}'$
- *for all* $t \in T$*:* $\beta \cdot (Pre \cdot t) = Pre' \cdot \eta(t)$ *and* $\beta \cdot (Post \cdot t) = Post' \cdot \eta(t)$

The multirelation β *is required to be* countably injective[2]*:*

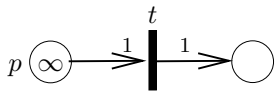
- *for all* $p \in P'$ *the set* $\{p \mid \beta[p, p'] > 0\}$ *is countable.*

We write $\eta(t) = *$ if $\eta(t)$ is undefined and in the above requirement regard $*$ as the empty multiset, so that if $\eta(t) = *$ then $\beta \cdot (Pre \cdot t)$ and $\beta \cdot (Post \cdot t)$ are both empty. For any multirelation R, we define $R[*, x]$ to be zero.

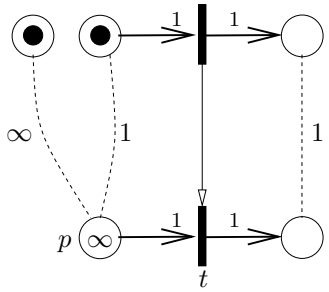
Any morphism $(\eta, \beta){:}G \to G'$ respects the collective token game for nets in the sense that for any t in G such that $\eta(t) \neq *$

$$\text{if } M \xrightarrow{t} M' \text{ in } G \text{ then } \beta \cdot M \xrightarrow{\eta(t)} \beta \cdot M' \text{ in } G'.$$

In fact, general nets are presented here in slightly more generality than in [Win87]: We allow the initial marking to be an ∞-multiset, whereas in [Win87], the initial marking has to be a multiset. Similarly, we allow an ∞-multiset of postconditions and do not require it to be a multiset. In this sense, the definition of morphism here is also slightly more general than that presented in [Win87] in that we allow a morphism to be an ∞-multirelation on conditions. The extra generality in the definition of general nets here allows, for example, the following net:



The ∞ symbol in place p represents there being infinitely many tokens in p. This example demonstrates how the added generality can allow finite presentations of infinite processes. The definition of morphism is generalized so that we can have, for example, a morphism



[2] The restriction to countably injective morphisms did not occur in [Win87]. We only require it here to obtain a coreflection between P/T nets and general nets, not to obtain the coreflection between occurrence nets and general nets. The restriction can be lifted by working with larger cardinals.

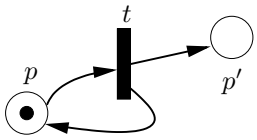


Figure 7.1: Example P/T net N

showing how a net with finite behaviour embeds into the general net.

From this definition of morphism, it is easy to see that we obtain a category $\mathbf{Gen}^\sharp$ of general nets. In particular, for any general net G the identity morphism $\mathsf{id}_G{:}G \to G$ is the identity function on events and the unit multirelation on places. Morphisms of general nets compose in the obvious way as partial functions on events and ∞-multirelations on places. We shall write $\mathbf{Gen}$ for the full subcategory of $\mathbf{Gen}^\sharp$ of singly-marked general nets.

We shall say that a net morphism (η, β) is *synchronous* if η is a total function on events. We add the subscript $_{\mathrm{s}}$ to denote categories with only synchronous morphisms, for example writing $\mathbf{Gen}^\sharp_{\mathrm{s}}$ for the category of multiply-marked general nets with synchronous morphisms between them. A morphism (η, β) is a *folding* morphism if it is synchronous and the relation β is also a total function. We add the subscript $_{\mathrm{f}}$ to denote categories with only folding morphisms, for example writing $\mathbf{Gen}^\sharp_{\mathrm{f}}$ for the category of multiply-marked general nets with folding morphisms between them.

P/T nets and safe nets

A particular form of general net is what we shall call the *P/T net*. A P/T net is a general net of which the pre- and post-condition (∞)-multirelations can be represented as relations and the initial markings can all be represented as sets. When drawing a P/T net, we can therefore omit the numbers on the arcs connecting places and transitions, and can only draw one token in any place of any initial marking. For technical reasons that shall become apparent later, we also require a P/T net to have no *isolated* places; we say that a place is isolated if it occurs in no initial marking and it does not occur in the pre- or post-conditions of any event. Following this account, P/T nets can be formally specified as follows.

Definition 7.1.3. *A P/T net is a 4-tuple*

$$(P, T, F, \mathbb{M})$$

where

- P *is the set of conditions or places,*
- T *is the set of events or transitions, disjoint from* P*,*
- $F \subseteq (P \times T) \cup (T \times P)$ *is the flow relation, and*
- $\mathbb{M} \subseteq \mathcal{P}ow(P)$ *is the set of initial markings, each of which is a set.*

The net must contain no isolated places and, for each event $t \in T$*, there must exist* $p \in P$ *such that* $p \, F \, t$*.*

The token game for P/T nets is exactly the same as that for general nets. For instance, the token game for the net N in Figure 7.1 allows the transition t to occur in the initial marking drawn, removing a token from place p and placing tokens in places p and p' to yield a marking in which both places contain exactly one token. A marking that can be obtained through a sequence of transitions from some initial marking is said to be *reachable* from that initial marking. Observe that although each *initial* marking of a P/T net is a set, it need not in general be the case that every *reachable* marking is itself a set. For instance, the net in Figure 7.1 has the following sequence of transition occurrences:

$$\left\{ \begin{array}{c} p \mapsto 1 \\ p' \mapsto 0 \end{array} \right\} \stackrel{t}{\longrightarrow\!\!\!\!\!\rightarrow} \left\{ \begin{array}{c} p \mapsto 1 \\ p' \mapsto 1 \end{array} \right\} \stackrel{t}{\longrightarrow\!\!\!\!\!\rightarrow} \left\{ \begin{array}{c} p \mapsto 1 \\ p' \mapsto 2 \end{array} \right\}$$

After two occurrences of the transition t, there are two tokens in the place p'.

For any transition t and place p, we re-adopt the notations from basic nets:

$$\begin{array}{rclcrcl} {}^\bullet t & = & \{p \mid p\ F\ t\} & \qquad & {}^\bullet p & = & \{t \mid t\ F\ p\} \\ t^\bullet & = & \{p \mid t\ F\ p\} & & p^\bullet & = & \{t \mid p\ F\ t\} \end{array}$$

The set ${}^\bullet t$ forms the *preconditions* of t and elements of the set $t^\bullet$ are *postconditions* of t. We shall avoid using these notations when referring to general nets that are not P/T nets, so these shall always be sets.

Apart from now having a *set* of initial markings, a P/T net has precisely the same structure as the basic nets introduced earlier. The only difference between the two forms of net is their token game: The token game for basic nets causes the occurrence of a transition to be inhibited should contact occur, whereas the token game for P/T nets allows such a transition to occur, to yield a marking with more than one token in some place.

We say that a P/T net is *safe* if all its reachable markings are sets. (In the literature, this is often referred to as 1-safety.) Note that this definition of safe net corresponds to the earlier definition of a safe net as a basic net that is contact-free.

Proposition 7.1. *All reachable markings of a P/T net N are sets iff the net N regarded as a basic net is contact-free from all initial markings.*

Proof. Let $N = (P, T, F, \mathbb{M})$.

($\Longrightarrow$): Suppose that there is a reachable marking of N that is not a set but, for contradiction, that N is contact-free from all initial markings. There exists least $n \in \mathbb{N}$ such that n is the length of a path from an initial marking $M_0 \in \mathbb{M}$ to a marking M that is not a set. Since every initial marking is a set, $n > 0$. Let the path be $\pi \cdot e$. It is easy to see there is contact in the marking M' obtained by $M_0 \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ since M' must be a set due to the minimality of n.

($\Longleftarrow$): Let M be reachable from some initial marking $M_0 \in \mathbb{M}$, so there exists a path π such that $M_0 \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M$. A simple induction on the length of π shows that M is a set. □

Morphisms

Morphisms between P/T nets are the net morphisms presented above. We obtain the category $\mathbf{PT}^\sharp$ of P/T nets and its full subcategory $\mathbf{PT}$ of singly-marked P/T nets.

The morphisms described in this section on safe nets might appear to be different from the morphisms of basic nets introduced in Chapter 2. The difference is, however, only superficial:

Proposition 7.2. *Let $N = (P, T, F, \mathbb{M})$ and $N' = (P', T', F', \mathbb{M}')$ be P/T nets, and consider a partial function $\eta{:}E \to_* E'$ and an ∞-multirelation $\beta \subseteq_{\mu_\infty} P \times P'$. The following two characterizations for the pair $(\eta, \beta){:}N \to N'$ being a morphism are equivalent:*

1. *For all $M \in \mathbb{M}$ and $t \in T$:*

$$\beta \cdot M \in \mathbb{M} \quad \& \quad \beta \cdot {}^\bullet t = {}^\bullet \eta(t) \quad \& \quad \beta \cdot t^\bullet = \eta(t)^\bullet$$

2. *β is a relation such that for all $M \in \mathbb{M}$ and $t \in T$:*

   - *there exists $M' \in \mathbb{M}'$ such that $\beta M \subseteq M'$ and $\forall p' \in M'.\exists! p \in M.\beta(p, p')$,*
   - *$\beta\ {}^\bullet e \subseteq {}^\bullet \eta(e)$ and $\forall p' \in {}^\bullet \eta(e).\exists! p \in {}^\bullet e.\beta(p, p')$, and*
   - *$\beta\ e^\bullet \subseteq \eta(e)^\bullet$ and $\forall p' \in \eta(e)^\bullet.\exists! p \in e^\bullet.\beta(p, p')$.*

Proof. It is obvious that (2) implies (1). Characterization (1) implies (2) as a straightforward consequence of P/T nets having no isolated conditions. □

Occurrence nets

Occurrence nets were introduced in [NPW81] as a class of net suited to giving the semantics of more general kinds of net in a way that directly represents the causal dependencies of elements of the net, for example that a particular event must have occurred at some earlier stage for a particular condition to become marked, and how the occurrence of elements of the net might conflict with each other. Technically, they can be thought-of as safe nets with acyclic flow relations such that every condition occurs as a postcondition of at most one event, for every condition there is a reachable marking containing that condition, and for every event there is a reachable marking in which the event can occur. We extend their original definition to account for the generalization to having a set of initial markings.

Definition 7.1.4. *An* occurrence net *$O = (B, E, F, \mathbb{M})$ is a P/T net satisfying the following restrictions:*

1. $\forall M \in \mathbb{M} : \forall b \in M : ({}^\bullet b = \emptyset)$
2. $\forall b' \in B : \exists M \in \mathbb{M} : \exists b \in M : (b\ F^*\ b')$
3. $\forall b \in B : (|{}^\bullet b| \leq 1)$
4. *F^+ is irreflexive and, for all $e \in E$, the set $\{e' \mid e'\ F^*\ e\}$ is finite*
5. *$\#$ is irreflexive, where*

$$\begin{aligned} e \#_{\mathrm{m}} e' &\iff^{\triangle} e \in E\ \&\ e' \in E\ \&\ e \neq e'\ \&\ {}^\bullet e \cap {}^\bullet e' \neq \emptyset \\ b \#_{\mathrm{m}} b' &\iff^{\triangle} \exists M, M' \in \mathbb{M} : (M \neq M'\ \&\ b \in M\ \&\ b' \in M') \\ x \# x' &\iff^{\triangle} \exists y, y' \in E \cup B : y \#_{\mathrm{m}} y'\ \&\ y\ F^*\ x\ \&\ y'\ F^*\ x' \end{aligned}$$

It is hopefully clear that any occurrence net is safe. From this definition, we obtain the category $\mathbf{Occ}^\sharp$ of occurrence nets and its full subcategory **Occ** of singly-marked occurrence nets. Morphisms in these categories are the net morphisms introduced above.

The flow relation F of an occurrence net O indicates how occurrences of events and conditions causally depend on each other and the relation $\#$ indicates how they conflict with each other, with $\#_{\mathrm{m}}$ representing immediate conflict. Two events are in immediate

conflict if they share a common precondition, so that the occurrence of one would mean that the other could not occur in any subsequent marking. Two conditions are in immediate conflict if they occur in different initial markings, so if one occurs in a reachable marking there is no subsequent reachable marking in which the other occurs. This corresponds to the intuition at the beginning of this section, that the hidden events giving rise to each initial marking should be in conflict with each other.

The *concurrency* relation $\mathrm{co}_O \subseteq (B \cup E) \times (B \cup E)$ of an occurrence net O may be defined as follows:

$$x \ \mathrm{co}_O \ y \stackrel{\triangle}{\iff} \neg(x \# y \text{ or } x \ F^+ \ y \text{ or } y \ F^+ \ x)$$

It shall shortly be shown that this relation has the intended meaning, relating two events if they can occur concurrently and relating two conditions if they both hold in some reachable marking. We often drop the subscript '$_O$' when we write co_O if the net O is obvious from the context. The concurrency relation is extended to sets of conditions A in the following manner:

$$\mathrm{co}\ A \stackrel{\triangle}{\iff} (\forall b, b' \in A : \ b \ \mathrm{co} \ b') \text{ and } \{e \in E \mid \exists b \in A . e \ F^* \ b\} \text{ is finite}$$

To help understand the conflict relation, notice that the events and conditions of an occurrence net $(B, E, F, \mathbb{M})$ can only occur from a unique initial marking. For $x \in B \cup E$ and $M \in \mathbb{M}$, write $M \ F^* \ x$ if there exists $b \in M$ such that $b \ F^* \ x$. It is easy to see that M is unique: for any $M, M' \in \mathbb{M}$, if $M \ F^* \ x$ and $M' \ F^* \ x$ then $M = M'$.

Before we can show that the relations of concurrency, causal dependency and conflict have the intended meanings, we must observe that the elements of an occurrence net O occur at a unique *depth*. For a condition b and an event e of an occurrence net $O = (B, E, F, \mathbb{M})$, depth is defined as:

$$\begin{array}{lcll} \mathrm{depth}(b) & = & 0 & \text{if } \exists M \in \mathbb{M} : (b \in M) \\ \mathrm{depth}(b) & = & \mathrm{depth}(e) & \text{if } e \ F \ b \end{array}$$

$$\mathrm{depth}(e) = 1 + \max\{\mathrm{depth}(b) \mid b \in B \ \& \ b \ F \ e\}$$

We denote by $O\lceil n$ the occurrence net obtained by restricting O to elements at depth less than or equal to n.

We are now able to prove the required property:

Proposition 7.3. *Let $O = (B, E, F, \mathbb{M})$ be an occurrence net. For any $b \in B$ and $M \in \mathbb{M}$, we have $M \ F^* \ b$ iff there exists M' such that $b \in M'$ and M' is reachable from M. For any $e \in E$ and $M \in \mathbb{M}$, we have $M \ F^* \ e$ iff there exists M' such that e has concession in M' and M' is reachable from M.*

The relations $\#_{\mathrm{m}} \subseteq B^2 \cup E^2$ and $\# \subseteq (B \cup E)^2$ are binary, symmetric, irreflexive relations. The relation of conflict $x \# x'$ holds iff either there exist distinct $M, M' \in \mathbb{M}$ such that $M \ F^* \ x$ and $M' \ F^* \ x'$ or there exists a reachable marking M and two events e, e' that have concession in M such that $e \#_m e'$ and $e \ F^* \ x$ and $e' \ F^* \ x'$.

The relation co *is a binary, symmetric, reflexive relation between conditions and events of N. On conditions, we have $b \ \mathrm{co} \ b'$ iff there is a reachable marking in which both b and b' hold. We have $e \ \mathrm{co} \ e'$ iff there is a reachable marking at which e and e' can occur concurrently.*

Any subset $A \subseteq B$ satisfies co A *iff there exists a reachable marking M of O such that $A \subseteq M$.*

Proof. All but the first part is a straightforward adaptation of Proposition 3.3.3 in [Win86].

The first part is shown in two parts. First, for any $M \in \mathbb{M}$, we show by induction on n that if $M\ F^n\ x$ then there is a path π of length $\leq n$ such that the marking M' obtained by $M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ satisfies $M' \stackrel{x}{\longrightarrow\!\!\!\!\!\rightarrow} M''$ if x is an event and $x \in M'$ if x is a condition.

Secondly, it can be shown by induction on the length of π that for any initial marking $M \in \mathbb{M}$ and path π such that $M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$, that if $M' \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow}$ then $M\ F^*\ e$ and if $b \in M'$ then $M\ F^*\ b$. □

Morphisms of occurrence nets

It will be useful later to point out now that any morphism $(\eta, \beta){:}O_1 \to O_2$ between occurrence nets preserves initial markings giving rise to elements of the occurrence net, it reflects conflict and it reflects the F relation in the following sense:

Proposition 7.4. *Let $O_1 = (B_1, E_1, F_1, \mathbb{M}_1)$ and $O_2 = (B_2, E_2, F_2, \mathbb{M}_2)$ be occurrence nets. For events $e_1, e_1' \in E_1$, write $e_1 \asymp_1 e_1'$ iff either $e_1 = e_1'$ or $e_1 \# e_1'$. Define $\asymp_2$ similarly for events in E_2. For any morphism $(\eta, \beta){:}O_1 \to O_2$ in $\mathbf{Occ}^\sharp$:*

- *for any $b_1 \in B_1$ and $M \in \mathbb{M}_1$, if $M\ {F_1}^*\ b_1$ and $\beta(b_1, b_2)$ then $\beta \cdot M\ {F_2}^*\ b_2$*
- *for any $e_1 \in E_1$, if $\eta(e_1)$ defined and $M\ {F_1}^*\ e_1$ then $\beta \cdot M\ {F_2}^*\ \eta(e_1)$*
- *for any $e_1, e_1' \in E_1$ and $e_2, e_2' \in E_2$:*

$$\eta(e_1) = e_2\ \&\ \eta(e_1') = e_2'\ \&\ e_2 \asymp_2 e_2'\ \Longrightarrow\ e_1 \asymp_1 e_1'$$

- *for any $b_1, b_1' \in B_1$ and $b_2, b_2' \in B_2$:*

$$\beta(b_1, b_2)\ \&\ \beta(b_1', b_2')\ \&\ b_2 \asymp_2 b_2'\ \Longrightarrow\ b_1 \asymp_1 b_1'$$

- *for any $e_2 \in E_2$, $b_1 \in B_1$ and $b_2 \in B_2$:*

$$e_2\ F_2\ b_2\ \&\ \beta(b_1, b_2)\ \Longrightarrow\ \exists! e_1 \in E_1 :\ e_1\ F_1\ b_1\ \&\ \eta(e_1) = e_2$$

- *for any $e_1 \in E_1$, $e_2 \in E_2$ and $b_2 \in B_2$:*

$$\eta(e_1) = e_2\ \&\ b_2\ F_2\ e_2\ \Longrightarrow\ \exists! b_1 \in B_1 :\ b_1\ F_1\ e_1\ \&\ \beta(b_1, b_2)$$

Proof. The first two items follow straightforwardly from Proposition 7.3 and the fact the the token game is preserved by net morphisms. The remaining four items follow from the nets O_1 and O_2 being occurrence nets. □

It follows that morphisms in the category $\mathbf{Occ}^\sharp$ also preserve the concurrency relation on both events and conditions.

7.2 Marking decompositions and coproducts

A multiply-marked occurrence net can be thought-of as a *family* of singly-marked occurrence nets placed side-by-side. A family is just an indexed collection of objects.

Definition 7.2.1 (Family)**.** *Let $\mathcal{C}$ be any category. A* family *of $\mathcal{C}$, written $(X_i)_{i \in I}$ is an indexing set I and a function associating each element of $i \in I$ with an element X_i of $\mathcal{C}$. A morphism between families $f{:}(X_i)_{i \in I} \to (Y_j)_{j \in J}$ is a function $\hat{f}{:}I \to J$ and, for each $i \in I$, a morphism $f_i{:}X_i \to Y_{\hat{f}(i)}$ in $\mathcal{C}$.*

We write $\mathcal{F}am(\mathcal{C})$ for the category of families of $\mathcal{C}$, with the obvious identities and composition of morphisms. A more complete account of categories of families can be found in Appendix C.

An occurrence net O gives rise to a family of singly-marked occurrence nets obtained by splitting the net O at each initial marking.

Definition 7.2.2 (Marking decomposition)**.** *Let $O = (B, E, F, \mathbb{M})$ be an occurrence net. The* marking decomposition *of O is a family of singly-marked occurrence nets $(O_M)_{M \in \mathbb{M}}$ in which the net O_M has conditions B_M and events E_M defined as*

$$B_M = \{b \in B \mid M\ F^*\ b\} \qquad E_M = \{e \in E \mid M\ F^*\ e\},$$

each net O_M inherits the flow relation of O and has initial marking M.

We write $\mathsf{decomp}(O)$ for the marking decomposition of an occurrence net O.

We shall soon show that the occurrence net O can be recovered, up to isomorphism, by placing the elements of its marking decomposition side-by-side, each element with its own initial marking. This will amount to taking the coproduct of the nets in its marking decomposition.

First, though, we shall show that the operation extends to a functor,

$$\mathsf{decomp}{:}\mathbf{Occ}^\sharp \to \mathcal{F}am(\mathbf{Occ}).$$

Let $O = (B, E, F, \mathbb{M})$ and $O' = (B', E', F', \mathbb{M}')$ be occurrence nets with marking decompositions $(O_M)_{M \in \mathbb{M}}$ and $(O'_{M'})_{M' \in \mathbb{M}'}$ respectively. A morphism $(\eta, \beta){:}O \to O'$ is sent to a morphism $f = \mathsf{decomp}(\eta, \beta)$ defined as

$$\hat{f}(M) = \beta \cdot M$$

for any $M \in \mathbb{M}$, and $f_M = (\eta_M, \beta_M){:}O_M \to O'_{\beta \cdot M}$ is defined to be

$$\eta_M(e) = \eta(e) \qquad \beta_M(b, b') \iff \beta(b, b')$$

for any e an event in O_M and b and b' conditions of O_M and $O'_{\beta \cdot M}$ respectively. It is clear from the first two items of Proposition 7.4 that f_M is a morphism in **Occ**.

We now consider how to form a multiply-marked occurrence net from a family of singly-marked occurrence nets. Coproducts in the categories of singly-marked safe nets and singly-marked occurrence nets were studied in [Win84]. There, the construction of $N_1 + N_2$ essentially involves 'gluing' the nets N_1 and N_2 together at their initial markings. The generalization to allow multiple initial markings allows a somewhat simpler construction in the categories $\mathbf{Occ}^\sharp$ and $\mathbf{PT}^\sharp$, where the nets are forced to operate on disjoint sets of conditions.

Proposition 7.5. *Let $(N_i)_{i \in I}$ be a family of P/T nets where $N_i = (P_i, T_i, F_i, \mathbb{M}_i)$ for each $i \in I$. The net $\sum_{i \in I} N_i = (P, T, F, \mathbb{M})$ defined as*

$$\begin{array}{rcl} P & = & \{\mathsf{in}_i p \mid i \in I\ \&\ p \in P_i\} \\ T & = & \{\mathsf{in}_i t \mid i \in I\ \&\ t \in T_i\} \\ (\mathsf{in}_i x)\ F\ (\mathsf{in}_j y) & \iff & i = j\ \&\ x\ F_i\ y \\ \mathbb{M} & = & \{\{\mathsf{in}_i p \mid p \in M\} \mid i \in I\ \&\ M \in \mathbb{M}_i\} \end{array}$$

is a coproduct in the category $\mathbf{PT}^\sharp$ with coproduct injections $\mathsf{in}_i{:}N_i \to \sum_{j \in I} N_j$.

Furthermore, the construction gives coproducts in the categories $\mathbf{Occ}^\sharp$ and $\mathbf{Safe}^\sharp$ and the categories with synchronous morphisms.

Proof. To see that $\sum_{i\in I} N_i$ is a coproduct in $\mathbf{PT}^\sharp$, suppose that for each $i \in I$ there is a morphism $f_i = (\eta_i, \beta_i){:}N_i \to N$ for some P/T net N. We must show that there is a unique morphism, which we shall denote $\sum_{i\in I} f_i{:}\sum_{i\in I} N_i \to N$, such that $f_i = (\sum_{i\in I} f_i) \circ \mathsf{in}_i$ for every $i \in I$. It is easy to see that $(\eta, \beta){:}\sum_{i\in I} N_i \to N$ defined as

$$\eta(\mathsf{in}_i e) = \eta_i(e) \qquad \beta(\mathsf{in}_i b, b') \iff \beta_i(b, b')$$

is such a morphism, and hence $\sum_{i\in I} N_i$ is a coproduct in $\mathbf{PT}^\sharp$. This net is safe if the nets N_i are safe and is an occurrence net if the nets N_i are occurrence nets, and so the construction also yields a coproduct in the categories of safe nets and occurrence nets. □

An important result is that the category $\mathbf{Occ}^\sharp$ is equivalent to $\mathcal{F}am(\mathbf{Occ})$. The functor $\mathsf{decomp}{:}\mathbf{Occ}^\sharp \to \mathcal{F}am(\mathbf{Occ})$ is part of this equivalence. The functor in the opposite direction, which we shall call $\mathsf{join}{:}\mathcal{F}am(\mathbf{Occ}) \to \mathbf{Occ}^\sharp$, is defined on objects as

$$\mathsf{join}(O_i)_{i\in I} = \sum_{i\in I} O_i$$

for the coproduct in $\mathbf{Occ}^\sharp$ described above. On morphisms, the the functor takes a morphism of families $f{:}(O_i)_{i\in I} \to (O'_{i'})_{i'\in I'}$ to

$$\mathsf{join}(f) = \sum_{i\in I} f_i$$

for f_i defined in Definition 7.2.1 and $\sum_{i\in I} f_i$ as defined in the proof of Proposition 7.5.

Proposition 7.6. *There categories* $\mathbf{Occ}^\sharp$ *and* $\mathcal{F}am(\mathbf{Occ})$ *are equivalent through the functors* decomp *and* join.

Proof. This is equivalent to saying that there are isomorphisms

$$\varphi_O{:}O \cong \mathsf{join}(\mathsf{decomp}(O)) \quad \text{and} \quad \psi_{(O_i)_{i\in I}}{:}(O_i)_{i\in I} \cong \mathsf{decomp}(\mathsf{join}((O_i)_{i\in I})),$$

natural in O and the family $(O_i)_{i\in I}$ respectively.

Let $O = (B, E, F, \mathbb{M})$. The isomorphism φ_O takes an element x of O to $\mathsf{in}_M x$ for the unique marking $M \in \mathbb{M}$ such that $M\ F^*\ x$. The isomorphism $\psi_{(O_i)_{i\in I}}$ is a consequence of the universal characterization of the coproduct, acting as the identity on the indexing sets and sending x in O_i to $\mathsf{in}_i x$ in $\mathsf{decomp}(\mathsf{join}((O_i)_{i\in I}))_i$.

Naturality of the two isomorphisms is straightforwardly shown. □

As an immediate consequence, we see that a multiply-marked occurrence net can be reformed by taking the coproduct of the family formed by taking its marking decomposition.

Corollary 7.7. *Let* O *be an occurrence net and* $(O_M)_{M\in\mathbb{M}}$ *be its marking decomposition. Then* $O \cong \sum_{M\in\mathbb{M}} O_M$ *through an isomorphism natural in* O.

We conclude this section by noting that the categories $\mathbf{PT}^\sharp$ and $\mathcal{F}am(\mathbf{PT})$ are not equivalent. The construction join lifts to P/T nets in a straightforward manner. However, for a condition or event x in a P/T net $(P, T, F, \mathbb{M})$, there need not be a unique $M \in \mathbb{M}$ such that $M\ F^*\ x$. As a consequence, we cannot define a functor $\mathsf{decomp}{:}\mathbf{PT}^\sharp \to \mathcal{F}am(\mathbf{PT})$ that yields an isomorphism $N \cong \mathsf{decomp}(\mathsf{join}(N))$ natural in N.

7.3 Inductive definition of nets

The process of forming an occurrence net from a general net is called *unfolding*. The construction takes place in stages, first defining the unfolding to form an occurrence net of depth 0, then an occurrence net of depth 1, and so on. In showing that the net obtained is indeed an occurrence net, it is useful to have some understanding of the *subnet order* on occurrence nets and how the limit of an (ordered) chain of occurrence nets is itself an occurrence net.

The subnet order on nets was first introduced in [Win84]. We generalize the order to account for multiple initial markings (though we shall content ourselves with considering just P/T nets rather than nets with multiplicities on arcs and in the initial markings).

Definition 7.3.1. *Let $N = (B, E, F, \mathbb{M})$ and $N' = (B', E', F', \mathbb{M}')$ be P/T nets. Then N is a subnet of N', written $N \leq N'$, iff*

- $B \subseteq B' \qquad E \subseteq E' \qquad \mathbb{M} \subseteq \mathbb{M}'$,
- $\forall e \in E : \forall b \in B' : b\ F'\ e \iff b \in B\ \&\ b\ F\ e$, *and*
- $\forall e \in E : \forall b \in B' : e\ F'\ b \iff b \in B\ \&\ e\ F\ b$.

As such, a net N is a subnet of N' if, and only if, there is a morphism $(\eta, \beta){:}N \to N'$ where both η and β are inclusions. The subnet relation is clearly a partial order.

With this subnet order, nets form a domain (ccpo) in the sense that there is a least net and that there are limits of ω-chains. The limit is defined as follows.

Definition 7.3.2. *Let $N_0, N_1, \ldots$ be an $\mathbb{N}$-indexed set of P/T nets with $N_i \leq N_{i+1}$ for all $i \in \mathbb{N}$ (that is, an ω-chain). Define the net*

$$\bigsqcup_{i\in\mathbb{N}} N_i = (\bigcup_{i\in\mathbb{N}} B_i, \bigcup_{i\in\mathbb{N}} E_i, \bigcup_{i\in\mathbb{N}} F_i, \bigcup_{i\in\mathbb{N}} \mathbb{M}_i).$$

We now show that this is a limit and that there is a least net according to the subnet order.

Proposition 7.8. *Let the P/T net $\bot = (\emptyset, \emptyset, \emptyset, \emptyset)$. For any P/T net N, we have*

$$\bot \leq N.$$

For any ω-chain of P/T nets $N_0 \leq N_1 \leq N_2 \leq \ldots$, the net $\bigsqcup_{i\in\mathbb{N}} N_i$ is a P/T net and is the limit of the chain.

Proof. Let $\bigsqcup_{i\in\mathbb{N}} N_i = (B, E, F, \mathbb{M})$. The only non-trivial part of the proof is to show that, for any $e \in E_i$:

$$\{b \in B \mid b\ F\ e\} = \{b \in B_i \mid b\ F_i\ e\} \quad \{b \in B \mid e\ F\ b\} = \{b \in B_i \mid e\ F_i\ b\}$$

Both follow straightforwardly from the definition. □

The limit of a chain of safe nets is a safe net.

Proposition 7.9. *Let $\bigsqcup_{i\in\mathbb{N}} N_i$ be the limit of an ω-chain of safe nets. Then $\bigsqcup_{i\in\mathbb{N}} N_i$ is a safe net.*

Proof. As in the preceding proof, for each i, let $N_i = (B_i, E_i, F_i, \mathbb{M}_i)$ and let $N = (B, E, F, \mathbb{M}) = \bigsqcup_{i\in\mathbb{N}} N_i$. Suppose, for contradiction, that N is not safe; there exists a path π and a marking M such that $N{:}M' \xrightarrow{\pi} M$ for some $M' \in \mathbb{M}$ and there exists an event $e \in E$ such that ${}^\bullet e \subseteq M$ but $M \setminus {}^\bullet e \cap e^\bullet \neq \emptyset$ for ${}^\bullet e$ and $e^\bullet$ the pre- and postconditions of e in N. Let E' be the set of events of the path π along with e. The set E' is finite. By induction on the size of E', we can show that there exists an i such that $E' \subseteq E_i$, $M' \in \mathbb{M}_i$ and the following two equalities hold for every $e' \in E'$:

$$\{b \in B \mid b\ F\ e'\} = \{b \in B_i \mid b\ F_i\ e'\} \qquad \{b \in B \mid e'\ F\ b\} = \{b \in B_i \mid e'\ F_i\ b\}.$$

It follows that the marking M is reachable from M' in N_i and that the event e causes contact in this marking in N_i. We therefore arrive at a contradiction, as required, because the net N_i is not safe. □

More importantly for us, the limit of a chain of occurrence nets is an occurrence net.

Proposition 7.10. *Let $O_0 \leq O_1 \leq O_2 \leq \ldots$ be an ω-chain of occurrence nets. For the net $O_i = (B_i, E_i, F_i, \mathbb{M}_i)$, let its concurrency relation be co_i and its conflict relation be $\#_i$. The net $\bigsqcup_{i\in\mathbb{N}} O_i = (B, E, F, \mathbb{M})$ is an occurrence net satisfying, for any $x, y \in B_i \cup E_i$, any $n \in \mathbb{N}$ and any $A \subseteq B_i$:*

$$\begin{array}{ccc} x\ F^n\ y & \iff & x\ {F_i}^n\ y \\ x \# y & \iff & x \#_i y \\ x\ \mathrm{co}\ y & \iff & x\ \mathrm{co}_i\ y \\ \mathrm{co}\ A & \iff & \mathrm{co}_i\ A \end{array}$$

Proof. We begin the proof by showing that if $x\ F\ y$ and $y \in B_i \cup E_i$ then $x \in B_i \cup E_i$. There exists j such that $x, y \in B_j \cup E_j$ and $x\ F_j\ y$. If $j \leq i$ then $x \in B_i \cup E_i$ as required. If $i \leq j$ and y is an event, so $y \in E_i$, then from requirement (4) on the order $\leq$ we have $x \in B_i$. Suppose instead, for contradiction, that $i \leq j$ and $y \in B_i$ and $x\ F_j\ y$ but $x \notin E_i$. The net N_i is an occurrence net, so either $y \in M$ for some $M \in \mathbb{M}_i$ or there exists an event $e \in E_i$ such that $e\ F_i\ y$. In the first case, we would have $M \in \mathbb{M}_j$ and, in the second case, we would have $e\ F_j\ y$, both contradicting the net N_j being an occurrence net.

Another important property is that for any $M \in \mathbb{M}$, if $M \neq \emptyset$ and $M \subseteq B_i$ then $M \in \mathbb{M}_i$. This is straightforwardly seen from the net N_i being an occurrence net; there would otherwise exist a different marking M' in $\mathbb{M}_i$ containing a condition b for $b \in M$, which would give rise to b being in conflict with itself in the net N_j for some j such that $M \in \mathbb{M}_j$. As a consequence of this property, recalling that we write $M\ F^*\ x$ iff there exists $b \in M$ such that $b\ F^*\ x$, we see that $M\ F^*\ x$ iff $M\ {F_i}^*\ x$ for any $M \in \mathbb{M}$ and $x \in B_i \cup E_i$.

It follows straightforwardly from these properties that the net $\bigsqcup_{i\in\mathbb{N}} N_i$ is an occurrence net that satisfies the requirements. □

Just as we can define a P/T as the limit of a chain of P/T nets, so a morphism of nets from the limit can be defined. It will be useful later to have the morphism from the limit to a general Petri net.

Proposition 7.11. *Consider a general net G and an ω-chain of P/T nets $N_0 \leq N_1 \leq \ldots$, where $N_i = (B_i, E_i, F_i, \mathbb{M}_i)$. A set of morphisms $(\eta_i, \beta_i){:}N_i \to G$ in $\mathbf{Gen}^\sharp$ for each $i \in \mathbb{N}$ that are coherent in the sense that*

- *$\eta_i(e) = \eta_j(e)$ for all i, j such that $e \in E_i \cap E_j$, and*
- *$\beta_i(b, p) = \beta_j(b, p)$ for all i, j such that $b \in B_i \cap B_j$ and all places p in G*

gives rise to a morphism $(\eta, \beta)\colon \bigsqcup_{i\in\mathbb{N}} N_i \to G$ *defined as*

$$\begin{array}{rcll} \eta(e) & = & \eta_i(e) & \text{if } e \in E_i \\ \beta(b,p) & = & \beta_i(b,p) & \text{if } b \in B_i. \end{array}$$

Proof. Let $\bigsqcup_{i\in\mathbb{N}} N_i = (B, E, F, \mathbb{M})$. The proof is a straightforward consequence of the coherence of the morphisms and the earlier observation that

$$\begin{array}{rcl} \{b \in B \mid b\ F\ e'\} & = & \{b \in B_i \mid b\ F_i\ e'\} \\ \{b \in B \mid e'\ F\ b\} & = & \{b \in B_i \mid e'\ F_i\ b\}. \end{array}$$

□

7.4 Causal nets and paths of nets

There are many ways that paths of Petri nets can be described. We have already seen that the collective token game for nets lends itself to paths that are sequences of events between markings represented as multisets. More generally, we might take a path to be a so-called *elementary event structure*, a partially-ordered set of event occurrences, also known as a *pomset* [Pra86].

Here, we shall consider *causal nets*[3]. These are a well-understood, net based representations of paths of Petri nets.

Definition 7.4.1. *A causal net is an occurrence net with an empty conflict relation and at most one initial marking.*

We shall use the symbol C to range over causal nets and write **Caus** for the category of causal nets with net morphisms between them.

Say that a causal net is *finite* if its set of events is finite — though note that its set of conditions might still be infinite.

Any finite causal C net has a unique marking following the occurrence of all its events; denote this marking $\mathrm{mkg}(C)$. This is easily seen to be equal to the set of conditions that do not occur as a precondition to any event.

A folding morphism $\iota\colon C \to G$ from a finite causal net is a direct characterization of a path of G according to the *individual token game* for nets.

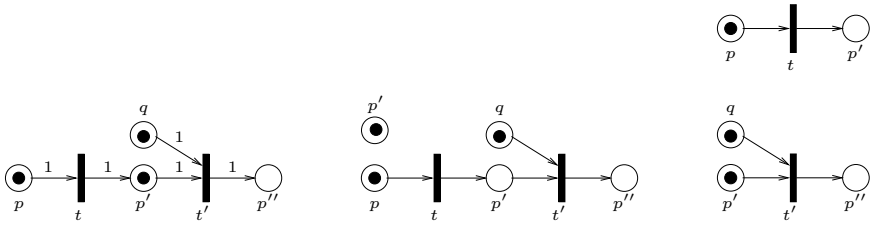
We shall not go into much detail on the different forms of token game here — see [vG05] for a comprehensive account. Intuitively, the individual token game is different from the collective token game in that, in the individual token game, tokens record the occurrence of the transition that gave rise to them. Consider the general net G in Figure 7.2. According to the collective token game for nets, following the occurrence of t we obtain the marking

$$\{p' \mapsto 2, q \mapsto 1\}.$$

In this marking, the transition t' can occur. The notion of concurrency arising the collective token game is subtle, but we would normally say that the two transitions occurred concurrently since the the transition t' can occur without the earlier occurrence of t.

The individual token game removes all this ambiguity. The marking following the occurrence of t has two tokens in place p', but we are able to distinguish the token that was initially in p' from the token that is placed in p' by t. As such, when the transition t' occurs and consumes one of these tokens, it is known whether the occurrence of t' depends

[3] We generalize them to account for nets now having sets of initial markings. A causal net according to this definition might have no initial marking. An alternative would be to require them to have precisely one initial marking, giving rise to a coarser form of open map bisimulation.



(a) General net G

(b) Path with causal dependency

(c) Path without causal dependency

Figure 7.2: Causal net paths of a general net

on the earlier occurrence of t (since it uses the token that was recorded as such) or whether t' does not depend on the earlier occurrence of t (since it uses the token that was initially in the place p'). This is manifested by there being two causal net paths that embed into G by folding morphisms, presented in Figure 7.2; recall that a folding morphism comprises a total function on events and a total function on conditions. We have drawn the nets with conditions and events labelled with their images under their embeddings into G. The causal net in the middle represents the run in which t' depends on t through the transition t depositing a token in place p' that t' consumes. The net on the right represents the run in which t' does not not depend on t, so the token in place p' consumed by t' is not the one placed there by t.

Before showing a few elementary facts about causal nets as paths, we introduce a little convenient notation. These facts will be used in Appendix D, where open maps with respect to the category of causal nets as paths are characterized.

Definition 7.4.2. *Let G be a general net. Write $P(G)$ for its set of conditions, $T(G)$ for its set of events, $Pre(G)$ for its precondition multirelation, $Post(G)$ for its postcondition ∞-multirelation and $\mathbb{M}(G)$ for its set of initial markings. If G is a P/T net, we write $F(G)$ for its flow relation.*

For any morphism, we add the subscript $_{\mathsf{e}}$ to represent its component on events and $_{\mathsf{c}}$ to represent its component on conditions. For example, for a morphism $f = (\eta, \beta)$ we have $f_{\mathsf{e}} = \eta$ and $f_{\mathsf{e}} = \beta$. If f is a folding morphism, we may drop the subscripts.

We begin by showing how finite causal net paths can be extended. Suppose that there is a finite causal net C and a folding morphism $\iota{:}C \to G$. Notice that $\iota \cdot \mathrm{mkg}(C)$ is the marking of G obtained by following the path C (for $\mathrm{mkg}(C)$ as introduced following Definition 7.4.1). If an event t can occur in the marking $\iota \cdot \mathrm{mkg}(C)$, there must exist a subset $A \subseteq \mathrm{mkg}(C)$ such that $\iota \cdot A = Pre(G) \cdot t$. We now define a causal net $C +_A t$ and folding morphism $\iota +_A t{:}C \to G$ that represents the extension of C by an event with image t under the new morphism that has preconditions A.

Formally, the events of the net $C +_A t$ are the disjoint union of the events of C and the event t. The set of conditions of the net $C +_A t$ is formed by taking the disjoint union of the conditions of C and the set

$$X = \{(p, i) \mid p \in P(G) \ \&\ 0 \leq i < Post(G)[t, p]\}.$$

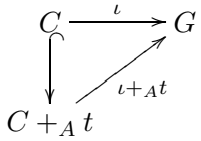
In $C +_A t$, the preconditions of t will be A and its post-conditions will be X. Since $\mathrm{mkg}(C)$ is the set of conditions of C that do not occur as a precondition to any event, it is easy to see that $C +_A t$ is a causal net.

The new folding morphism $\iota +_A t$ is precisely the same as ι apart from being extended by sending

$$t \mapsto t \qquad (p, i) \mapsto p.$$

It is very easy to see that the following lemma holds:

Lemma 7.4.1. *The net $C +_A t$ is a causal net and the folding $\iota +_A t{:}C +_A t \to G$ is a morphism in $\mathbf{Gen}^\sharp$ for which the following diagram commutes:*



It follows that any marking M reachable in a general net according to the collective token game is reachable according to individual token game.

Lemma 7.4.2. *Let π be a finite sequence of events of G such that $G{:}M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M'$ for some initial marking M of G. There exist a finite causal net C and a folding morphism $\iota{:}C \to G$ for which there is a path $C{:}M_0 \stackrel{\pi_0}{\longrightarrow\!\!\!\!\!\rightarrow} \mathrm{mkg}(C)$ from the (unique) initial marking M_0 of C such that $\iota(\pi_0) = \pi$ and $\iota \cdot M_0 = M$ and $\iota \cdot \mathrm{mkg}(C) = M'$.*

Proof. The proof is by induction on the length of π. The base case, where π is the empty sequence, is easy; just take C to be the causal net with no events and an initial marking defined as

$$\{(p, i) \mid p \in P(G) \;\&\; 0 \leq i < M[p]\},$$

and define the morphism ι as $(p, i) \mapsto p$. The inductive case is shown straightforwardly using Lemma 7.4.1. □

The converse property, that any marking reachable according to the individual token game with the history of tokens stripped away is reachable in the collective token game, follows immediately from the observation that $\mathrm{mkg}(C)$ is reachable in the net C from its initial marking and hence $\iota \cdot \mathrm{mkg}(C)$ is reachable from the initial marking of the general net according to the definition of net morphisms.

It is a very basic fact of nets that morphisms preserve paths derived according to the collective token game. For any morphism $(\eta, \beta){:}G \to G'$,

$$\text{if } G{:}M \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} M' \text{ then } G'{:}\beta \cdot M \stackrel{\eta(\pi)}{\longrightarrow\!\!\!\!\!\rightarrow} \beta \cdot M'.$$

The individual token game is also preserved by morphisms in the sense that from a causal net C, a folding map $\iota{:}C \to G$ and a morphism $f{:}G \to G'$ in $\mathbf{Gen}^\sharp$, the image of the causal net C under $f \circ \iota$ represents a path of G' according to the individual token game. That is, the image, which we shall denote by $\hat{f}C$, embeds into G' through a folding morphism $\hat{\iota}{:}\hat{f}C \to G'$.

The net $\hat{f}C$ is defined as follows:

$$\begin{array}{rcl}
P(\hat{f}C) & = & \{(b, i, p) \mid b \in P(C) \;\&\; p \in P(G') \;\&\; 0 \leq i < f_{\mathsf{c}}[\iota(b), p]\} \\
T(\hat{f}C) & = & \{(e, t) \mid e \in T(C) \;\&\; t \in T(G') \;\&\; t = f_{\mathsf{e}}(\iota(e)) \\
 & & \quad F(\hat{f}C)((b, i, p), (e, t)) \iff F(C)(b, e) \\
 & & \quad F(\hat{f}C)((e, t), (b, i, p)) \iff F(C)(e, b) \\
\mathbb{M}(\hat{f}C) & = & \{\{(b, i, p) \mid (b, i, p) \in P(\hat{f}C) \;\&\; b \in M\} \mid M \in \mathbb{M}(C)\}
\end{array}$$

The folding morphism $\hat{\iota}$ is defined as

$$\hat{\iota}(b,i,p) = p \qquad \hat{\iota}(e,t) = t.$$

Accompanying the causal net is a (relational) morphism $\hat{f}{:}C \to \hat{f}C$ defined as

$$\hat{f}_{\mathsf{c}}(b,(b',i,p)) \iff b = b' \qquad \hat{f}_{\mathsf{e}}(e) = f(\iota(e)).$$

Lemma 7.4.3. *The net $\hat{f}C$ is a causal net, ι is a folding morphism and $\hat{f}$ is a relational morphism such that following diagram commutes:*

$$\begin{array}{ccc} C & \xrightarrow{\iota} & G \\ {\scriptstyle \hat{f}}\downarrow & & \downarrow{\scriptstyle f} \\ \hat{f}C & \xrightarrow[\hat{\iota}]{} & G' \end{array}$$

Furthermore, $\mathrm{mkg}(\hat{f}C) = \hat{f} \cdot \mathrm{mkg}(C)$ *and if f is a folding morphism then so is $\hat{f}$.*

Proof. Every part of the proof is straightforward apart from showing that $\hat{\iota}$ is a morphism. We shall show only that $\hat{\iota}{\cdot}{}^{\bullet}(e,t) = Pre(G'){\cdot}t$ for any $(e,t) \in T(\hat{f}C)$; the other requirements are similar.

Suppose that $(e,t) \in T(\hat{f}C)$. Then $e \in T(C)$ and $t \in T(G')$ and $t = f(\iota(e))$. Recalling that $Pre(G')[t,p]$ is the number attached to the arc from p to t in G', we must show that, for any $p \in P(G')$, there is a bijection

$$\begin{array}{rl} & \{(b',i',p') \in P(\hat{f}C) \mid (b',i',p') \in {}^{\bullet}(e,t) \ \&\ \hat{\iota}(b',i',p') = p\} \\ \cong & \{i \mid 0 \le i < Pre(G')[t,p]\}. \end{array}$$

Since f is a morphism and $t = f(\iota(e))$, there is a bijection

$$\begin{array}{rl} & \{(p_0,i,j) \mid p_0 \in P(G) \ \&\ 0 \le i < Pre(G)[\iota(e),p_0] \ \&\ 0 \le j < f_{\mathsf{c}}[p_0,p]\}\} \\ \cong & \{i \mid 0 \le i < Pre(G')[t,p]\}. \end{array}$$

This relies on f_{c} being a countably injective multirelation (Appendix A). Since ι is a folding morphism there is a bijection for any $p_0 \in P(G)$

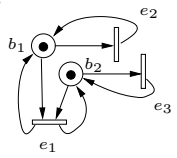
$$\{b \in P(C) \mid b \in {}^{\bullet}e \ \&\ \iota(b) = p_0\} \cong \{i \mid 0 \le i < Pre(G)[\iota(e),p_0]\}.$$

This relies on ι being a countably injective multirelation on conditions. The result follows by combining the two bijections. □

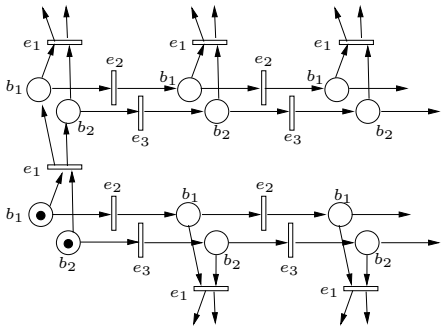
7.5 Unfolding

As discussed above, occurrence nets can be used to give the semantics of more general forms of net. The process of forming the occurrence net semantics of a net is called *unfolding*, first defined for safe nets in [NPW81], and is analogous to the process of unfolding a transition system to obtain a synchronization tree. The result of unfolding a net N is an occurrence net $\mathcal{U}(N)$ accompanied by a morphism $\varepsilon_N{:}\mathcal{U}(N) \to N$ relating the unfolding back to the original net. An example unfolding of a safe net is presented below, with the conditions and events of the unfolding labelled by their image under ε_N:

N:



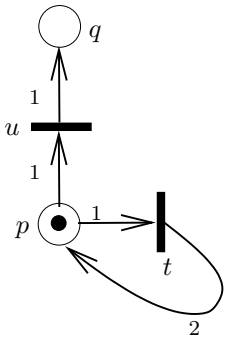
$\mathcal{U}(N)$:



We now show how this definition extends to general nets. In fact, the techniques for unfolding safe nets are almost directly applicable to general nets. The construction is rather technical, requiring an inductive definition, but is neatly characterized in Theorem 7.12.

Overview

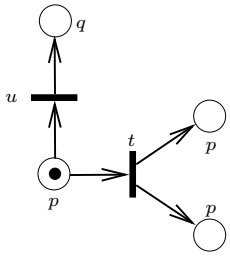
To help motivate the formal definitions to follow, we present a simple example of how a general Petri net is unfolded. Let G be the following general Petri net.



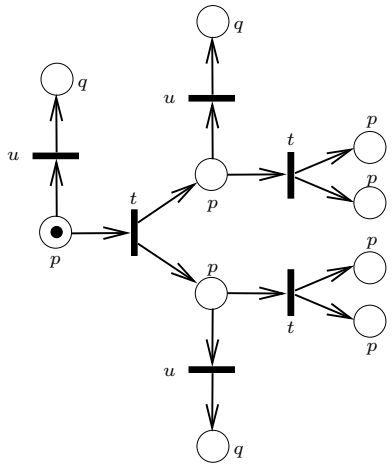
The unfolding of G will be formed from conditions and events to represent *occurrences* of conditions and events in the reachable markings of G. Since the place p occurs holding one token in the initial marking of G, it occurs once in the first stage of the unfolding, $\mathcal{U}_0(G)$. We draw the net $\mathcal{U}_0(G)$ with labels to indicate to which element of G its conditions and events correspond.

$\mathcal{U}_0(G)$: 

Since the place p can become marked, the events t and u can occur. Either occurrence will consume the token from p, so the two occurrences conflict with each other. The occurrence of t will yield two occurrences of tokens in p and the occurrence of u will yield one occurrence of a token in q. This yields the unfolding $\mathcal{U}_1(G)$.



Note that there are correspondingly two postconditions of the event labelled t. Each of the two occurrences of the place p in the postconditions of the occurrence of t allow either the event t or the event u to occur. We add these occurrences to the unfolding $\mathcal{U}_2(G)$.



The unfolding $\mathcal{U}_2(G)$ represents that the occurrences of u causally depend on the prior occurrence of t and that they can occur concurrently in the marking obtained following t.

Notice how the unfolding represents the individual token game discussed in Section 7.4: The two occurrences of p in the postconditions of t in $\mathcal{U}_1(G)$ each give rise to a distinct occurrence of the event u in $\mathcal{U}_2(G)$.

The net $\mathcal{U}(G)$ is formed by continuing with this process of adding possible event occurrences, so taking the limit described in Proposition 7.10 of the chain of occurrence nets

$$\mathcal{U}_0(G) \leq \mathcal{U}_1(G) \leq \mathcal{U}_2(G) \leq \cdots$$

to yield an infinite occurrence net. We can see that $\mathcal{U}_n(G)$ is the occurrence net unfolding of G restricted to elements at depth n.

Inductive characterization

The inductive definition of the unfolding involves constructing the unfolding of a general net G to a particular depth. This will consist of the unfolding $\mathcal{U}_n(G)$ and a folding morphism $(\eta_n, \beta_n){:}\mathcal{U}_n(G) \to G$. Recall that since this is a folding morphism, β_n will be a function on places.

Definition 7.5.1. *Let $G = (P, T, Pre, Post, \mathbb{M})$ be a general Petri net. On a tuple (X, p, i) define the projection $\beta(X, p, i) \triangleq p$ and on a tuple (A, t) define the projection $\eta(A, t) \triangleq t$. The* unfolding *of the net G to depth $n \in \mathbb{N}$ is the tuple $\mathcal{U}_n(G) = (B_n, E_n, F_n, \mathbb{M}_0)$ defined by induction on n as*

$$\begin{array}{rcl}
E_0 & \triangleq & \emptyset \\
B_0 & \triangleq & \{(M, p, i) \mid M \in \mathbb{M} \;\&\; 0 \leq i < M[p]\} \\
\mathbb{M}_0 & \triangleq & \{\{(M, p, i) \mid (M, p, i) \in B_0\} \mid M \in \mathbb{M}\} \\
E_{n+1} & \triangleq & E_n \cup \{(A, t) \mid A \subseteq B_n \;\&\; \mathrm{co}_n\, A \;\&\; t \in T \;\&\; \beta \cdot A = Pre \cdot t\} \\
B_{n+1} & \triangleq & B_n \cup \{(\{e\}, p, i) \mid e \in E_{n+1} \;\&\; p \in P \;\&\; 0 \leq i < Post[\eta(e), p]\}
\end{array}$$

with flow relation $F_n \subseteq (B_n \times E_n) \cup (E_n \times B_n)$ *defined as*

$$b\ F_n\ e \quad \stackrel{\triangle}{\iff} \quad \exists A, t:\ e = (A, t)\ \textit{and}\ b \in A$$
$$e\ F_n\ b \quad \stackrel{\triangle}{\iff} \quad \exists p, i:\ b = (\{e\}, p, i).$$

The concurrency relation on the net $\mathcal{U}_n(G)$ *is written* co_n *and its conflict relation is written* $\#_n$.

Define η_n *to be the restriction of* η *to* E_n *and* β_n *to be the restriction of* β *to* B_n.

We now begin to consider the structure of the net $\mathcal{U}_n(G)$, first by looking at its structure as a P/T net.

Lemma 7.5.1. *The net* $\mathcal{U}_n(G)$ *is a P/T net such that* $\mathcal{U}_m(G) \leq \mathcal{U}_n(G)$ *for any* $m \leq n$. *Furthermore,*

1. *if* $x, y \in B_m \cup E_m$ *then:*

$$x\ F_n\ y \iff x\ F_m\ y \qquad x \#_n y \iff x \#_m y \qquad x\ \mathrm{co}_n\ y \iff x\ \mathrm{co}_m\ y$$

2. *for any* $i \in \mathbb{N}$, *if* $x, y \in B_m \cup E_m$ *then* $x {F_n}^i y$ *iff* $x {F_m}^i y$.

Proof. It is relatively straightforward to show by induction on n that $\mathcal{U}_n(G)$ is a P/T net. Another easy induction, this time on $n - m$, shows that $\mathcal{U}_m(G) \leq \mathcal{U}_n(G)$ for any $m \leq n$. By considering the ω-chain $\mathcal{U}_m(G) \leq \mathcal{U}_n(G) \leq \mathcal{U}_n(G) \leq \cdots$, it follows from Proposition 7.10 that the remaining properties hold. □

We can now show that $(\eta_n, \beta_n) : \mathcal{U}_n(G) \to G$ is a net morphism and that the chain of morphisms generated is consistent.

Lemma 7.5.2. *The pair* (η_n, β_n) *is a net morphism and the morphisms* (η_m, β_m) *and* (η_n, β_n) *are consistent for all* m *and* n.

Proof. The morphisms (η_m, β_m) and (η_n, β_n) are consistent immediately from their definition.

Let the net $G = (P, T, Pre, Post, \mathbb{M})$. We shall show that (η_n, β_n) is a morphism in $\mathbf{Gen}^\sharp$ by induction on n. The base case is easy, being a simple verification that $\beta_0 \cdot M \in \mathbb{M}$ for all $M \in \mathbb{M}_0$. For the inductive case, since $\mathbb{M}_n = \mathbb{M}_0$, all that we must show is that for any event $e \in E_n$:

$$\beta_n \cdot {}^\bullet e = Pre \cdot \eta_n(e) \qquad \beta_n \cdot e^\bullet = Post \cdot \eta_n(e).$$

If $e \in E_{n-1}$, these both hold as a direct consequence of the induction hypothesis and the fact that $\mathcal{U}_{n-1}(G) \leq \mathcal{U}_n(G)$ according to Lemma 7.5.1. Otherwise, if $e \in E_n \setminus E_{n-1}$, we have $e = (A, t)$ for some $A \subseteq B_{n-1}$ such that $\beta \cdot A = Pre \cdot t$. It follows immediately from the observation that ${}^\bullet e = A$ and the definitions of β_n and η_n that $\beta_n \cdot {}^\bullet e = Pre \cdot \eta_n(e)$.

From the definition F_n and B_n, we can see that

$$\begin{aligned} e^\bullet &= \{(\{e\}, p, i) \mid (\{e\}, p, i) \in B_{n+1}\} \\ &= \{(\{e\}, p, i) \mid p \in P\ \&\ 0 \leq i < Post[t, p])\}. \end{aligned}$$

It follows from the definition of β_n that $\beta \cdot e^\bullet = Post \cdot t$. □

We shall now show that $\mathcal{U}_n(G)$ is an occurrence net.

Lemma 7.5.3. *The net $\mathcal{U}_n(G)$ is an occurrence net.*

Proof. We begin by observing two important properties from the inductive definition of $\mathcal{U}_n(G)$. For any $n > 0$, if $e \in E_n \setminus E_{n-1}$ then:

(a) if $b\ F_n\ e$ then $b \in B_{n-1}$, and

(b) if $e\ F_n\ b$ then $b \in B_n \setminus B_{n-1}$.

We now prove that $\mathcal{U}_n(G)$ is an occurrence net by induction on n. We must show that:

1. $\forall M \in \mathbb{M}_0 : \forall b \in M : (\{e \mid e \in E_n\ \&\ e\ F_n\ b\} = \emptyset)$
2. $\forall b' \in B_n : \exists M \in \mathbb{M}_0 : \exists b \in M : (b {F_n}^* b')$
3. $\forall b \in B_n : (|\{e \mid e \in E_n\ \&\ e\ F_n\ b\}| \leq 1)$
4. F_n^+ is irreflexive, and for all $e \in E_n$ the set $\{e' \mid e' {F_n}^* e\}$ is finite
5. $\#_n$ is irreflexive

It is clear that these properties are satisfied for $n = 0$. We shall show that they hold at $n + 1$ given that they hold at n.

1. A simple consequence of (b) above with the fact that if $b \in M \in \mathbb{M}_0$ then $b \in B_0$.
2. Suppose that $b \in B_{n+1}$. If $b \in B_n$, the result follows from Lemma 7.5.1(2) and the induction hypothesis. If $b \in B_{n+1} \setminus B_n$ then $b = (\{e\}, p, i)$ for some $p \in P$ and $i \in \mathbb{N}$ and $e \in E_{n+1}$, and furthermore $e\ F_{n+1}\ b$. We have $e \notin E_n$ since otherwise $b \in B_n$. It follows from the requirement that $Pre \cdot t$ is non-null for any transition t in G that there exists a condition $b' \in B_{n+1}$ such that $b'\ F_{n+1}\ e$. Furthermore, $b' \in B_n$ by (a). From the induction hypothesis, there exists $M \in \mathbb{M}_0$ and $b_0 \in M$ such that $b_0 {F_n}^* b'$. Therefore, from Lemma 7.5.1(2) we obtain $b_0 {F_{n+1}}^* b'$, as required.
3. If $b \in B_n$ then the result follows from (b) and the induction hypothesis. If $b \in B_{n+1} \setminus B_n$, it must be the case from the definition of B_{n+1} that $b = (\{e\}, p, i)$ for some $e \in E_{n+1} \setminus E_n$, and so there is a unique e', namely e, such that $e'\ F_{n+1}\ b$.
4. If follows from (a) and (b) that F_{n+1}^+ is irreflexive. We now show that $\{e' \mid e'\ F_{n+1}^*\ e\}$ is finite.

   Suppose first that $e \in E_n$. For any $e' \in E_{n+1}$, if $e'\ F_{n+1}^*\ e$ then $e' \in E_n$ and $e'\ F_n^*\ e$ according to a simple inductive argument based on (a), (b) and Lemma 7.5.1. Hence, by induction, $\{e' \in E_{n+1} \mid e'\ F_{n+1}^*\ e\}$ is finite. Now suppose that $e \in E_{n+1} \setminus E_n$, in which case $e = (A, t)$ for some $t \in T$ and $A \subseteq B_n$ such that $\mathrm{co}_n\ A$. From the definition of $\mathrm{co}_n\ A$, the set $X = \{e' \mid e' \in E_n\ \&\ \exists b \in A : e'\ F_n^*\ b\}$ is finite. A simple inductive argument using (a) and (b) shows that

   $$\forall e_0, b_0 : b_0 \in B_n\ \&\ e_0\ F_{n+1}^*\ b_0 \implies e_0\ F_n^*\ b_0,$$

   and as a consequence $X = \{e' \mid e' \in E_{n+1}\ \&\ \exists b \in A : e'\ F_{n+1}^*\ b\}$ from which we obtain the required result since $A = \{b \mid b \in B_{n+1}\ \&\ b\ F_{n+1}\ e\}$.

5. Suppose for contradiction that $\#_{n+1}$ is not irreflexive, first because there exists $e \in E_{n+1}$ such that $e\#_{n+1}e$. We cannot have $e \in E_n$ since, by Lemma 7.5.1(1), we would have $e\#_n e$ which contradicts the induction hypothesis. Hence $e \in E_{n+1} \setminus E_n$. From the definition of $(\#_{\mathrm{m}})_{n+1}$ we cannot have $e(\#_{\mathrm{m}})_{n+1}e$, so there must exist $b, b' \in B_{n+1}$ such that $b\#_{n+1}b'$ and $b\ F_{n+1}\ e$ and $b'\ F_{n+1}\ e$. However, from the definition of $E_{n+1} \setminus E_n$, we have $b\ \mathrm{co}_n\ b'$ and hence $\neg(b\#_{n+1}b')$ by Lemma 7.5.1(1). This gives us the required contradiction.

   Now suppose that there exists $b \in B_{n+1}$ such that $b\#_{n+1}b$. A similar argument to that above shows that $b \notin B_n$ since otherwise the induction hypothesis would be contradicted. The definition of $(\#_{\mathrm{m}})_{n+1}$ informs that $\neg(b(\#_{\mathrm{m}})_{n+1}b)$, so there must exist $e, e' \in E_{n+1}$ such that $e\ F_{n+1}\ b$ and $e'\ F_{n+1}\ b$ and $e\#_{n+1}e'$. From the above argument, we have $e \neq e'$, but this contradicts point (3) above, that each condition has at most one pre-event. □

We have now shown that for any general net G we have an ω-chain of occurrence nets

$$\mathcal{U}_0(G) \leq \mathcal{U}_1(G) \leq \mathcal{U}_2(G) \leq \cdots .$$

The unfolding $\mathcal{U}(G)$ can therefore be defined as the limit of this ω-chain specified in Definition 7.3.2,

$$\mathcal{U}(G) = \bigsqcup_{n\in\mathbb{N}} \mathcal{U}_n(G).$$

This is an occurrence net by Proposition 7.10.

We are now able to summarize all these technical matters by giving a characterization of the unfolding.

Theorem 7.12. *The occurrence net unfolding $\mathcal{U}(G) = (B, E, F, \mathbb{M}_0)$ of a general net $G = (P, T, Pre, Post, \mathbb{M})$ is the unique occurrence net to satisfy*

$$\begin{aligned}
B &= \{(M,p,i) \mid M \in \mathbb{M}\ \&\ p \in P\ \&\ 0 \leq i < M[p]\} \\
&\quad \cup\ \ \{(\{e\},p,i) \mid e \in E\ \&\ p \in P\ \&\ 0 \leq i < Post[\eta(e),p]\} \\
E &= \{(A,t) \mid A \subseteq B\ \&\ t \in T\ \&\ \mathrm{co}\ A\ \&\ \beta\cdot A = Pre\cdot t\} \\
&\quad b\ F\ (A,t) \iff b \in A \\
&\quad (A,t)\ F\ b \iff \exists p,i : (b = (\{(A,t)\},p,i)) \\
\mathbb{M}_0 &= \{\{(M,p,i) \mid (M,p,i) \in B\} \mid M \in \mathbb{M}\},
\end{aligned}$$

where co *and* $\#$ *are the concurrency and conflict relations arising from F on B and E. Furthermore, $\eta{:}E \to P$ and $\beta{:}B \to P$ defined as*

$$\eta(A,t) = t \qquad \beta(X,p,i) = p$$

form a morphism $\varepsilon_G = (\eta,\beta){:}\mathcal{U}(G) \to G$ in $\mathbf{Gen}^\sharp$.

Proof. It follows from Proposition 7.11 that $(\eta,\beta){:}\mathcal{U}(G) \to G$ is a morphism in $\mathbf{Gen}^\sharp$.

To see that the net $\mathcal{U}(G)$ satisfies the requirements above, we must show that the following two equations hold:

$$\begin{aligned}
\bigcup_{n\in\mathbb{N}} B_n &= \{(M,p,i) \mid M \in \mathbb{M}\ \&\ p \in P\ \&\ 0 \leq i < M[p]\} \\
&\quad \cup\{(\{e\},p,i) \mid e \in \bigcup_{n\in\mathbb{N}} E_n\ \&\ p \in P\ \&\ 0 \leq i < Post[\eta(e),p]\} \\
\bigcup_{n\in\mathbb{N}} E_n &= \{(A,t) \mid A \subseteq \bigcup_{n\in\mathbb{N}} B_n\ \&\ t \in T\ \&\ \mathrm{co}\ A\ \&\ \beta\cdot A = Pre\cdot t\}
\end{aligned}$$

The first is reasonably straightforward, so we shall show only the second.

($\subseteq$): Suppose that $e \in \bigcup_{n\in\mathbb{N}} E_n$. Since $\mathcal{U}_m(G) \leq \mathcal{U}_{m+1}(G)$ for all $m \in \mathbb{N}$ by Lemma 7.5.1, there is a least $n \in \mathbb{N}$ such that $e \in E_n$. It is easy to see that $n > 0$. Considering the definition of the net $\mathcal{U}_n(N)$, there exist $A \subseteq B_{n-1}$ and $t \in T$ such that $\mathrm{co}_{n-1}\ A$ and $\beta \cdot A = Pre \cdot t$. By Proposition 7.10, we have co A, and thus the required inclusion is shown.

($\supseteq$): Suppose that the pair (A, t) comprises $A \subseteq \bigcup_{m\in\mathbb{N}} B_m$ and $t \in T$, and satisfies co A and $\beta \cdot A = Pre \cdot t$. Since we have co A, the set $\{e \mid \exists b \in A :\ e \in E\ \&\ e\ F^*\ b\}$ is finite; let this set be called X. By Lemma 7.5.1, we have $\mathcal{U}_m(G) \leq \mathcal{U}_{m+1}(G)$ for all $m \in \mathbb{N}$, so there is a least $n \in \mathbb{N}$ such that $X \subseteq E_n$. We shall show that $A \subseteq B_n$. It will then follow from the definition of E_{n+1} that $(A, t) \in E_{n+1}$, completing the case.

Suppose that $b \in A$. Then $b \in B_k$ for some least $k \in \mathbb{N}$ since $A \subseteq \bigcup_{m\in\mathbb{N}} B_m$. If $k = 0$, we clearly have $b \in B_n$. Otherwise, $b = (\{e\}, p, i)$ for some $e \in E_k$ and $p \in P$ and i satisfying $0 \leq i < Post \cdot \eta(e)$. However, $e\ F_k\ b$ so $e \in X$ and hence $e \in E_n$. Therefore, by point (b) made at the start of the proof of Lemma 7.5.3, we have $b \in B_n$.

We now show that $\mathcal{U}(G)$ is the unique such occurrence net. Suppose that $O' = (B', E', F', \mathbb{M}')$ is any occurrence net satisfying the above constraints. Recall that $O'\lceil n$ is the occurrence net obtained by restricting O' to elements at depth less than or equal to n. It is easy to see that $O' = \bigsqcup_{n\in\mathbb{N}} O'\lceil n$. To show uniqueness, it is sufficient to show that

$$O'\lceil n = \mathcal{U}_n(G),$$

which we shall prove by induction on n.

Let $O'\lceil n = (B'\lceil n, E'\lceil n, F'\ \lceil n, \mathbb{M}'\lceil n)$ and recall that the unfolding $\mathcal{U}_n(G) = (B_n, E_n, F_n, \mathbb{M}_0)$. The base case of the induction is straightforward. For the inductive case, we shall show that

$$E'\lceil n = E_n;$$

the similar property for conditions follows from this straightforwardly, and the other parts follow immediately from the fact that O' satisfies the constraints above.

$\underline{E'\lceil n \subseteq E_n}$: Let $e \in E'$ with $\mathrm{depth}'(e) \leq n$. As O' satisfies the constraints above, we have $e = (A, t)$ for some $A \subseteq B'$ and $t \in T'$ such that $\mathrm{co}'\ A$ and $\beta \cdot A = Pre \cdot t$. Furthermore, $A = {}^\bullet e$ so, as O' is an occurrence net, we must have $\mathrm{depth}(b) \leq n - 1$ for all $b \in A$. From the induction hypothesis we have $b \in B_{n-1}$ for all $b \in A$, so we have $A \subseteq B_{n-1}$. From Proposition 7.10, the set of conditions A must be concurrent in $O'\lceil n-1$ and hence $\mathrm{co}_{n-1}\ A$. It follows immediately from the definition of E_n that $e \in E_n$.

$\underline{E'\lceil n \supseteq E_n}$: Suppose that $e \in E_n$. It is sufficient to consider the case where $e \notin E_{n-1}$ since otherwise the result follows straightforwardly by induction. Since $e \in E_n \setminus E_{n-1}$, from the definition there exist A and t such that $e = (A, t)$ and $A \subseteq B_{n-1}$ and $t \in T$, satisfying $\beta \cdot A = {}^\bullet\cdot t$ and $\mathrm{co}_{n-1}\ A$. Since $\mathcal{U}_{n-1}(G) = O'\lceil n - 1$ by induction, we have $A \subseteq B'\lceil n - 1$ and hence $A \subseteq B'$. We also have $\mathrm{co}'\ A$ from the fact that A is a concurrent set of conditions in $O'\lceil n-1$ and Proposition 7.10. Since the net O' satisfies the constraints above, we have $e \in E'$ and clearly $\mathrm{depth}(e) \leq n$, as required. □

We can see from the final part of this proof that, as one might expect, each occurrence net $\mathcal{U}_n(G)$ represents the unfolding $\mathcal{U}(G)$ restricted to elements at depth less than or equal to n.

Corollary 7.13. *Let $\mathcal{U}(G) = (B, E, F, \mathbb{M}_0)$. For any $x \in B \cup E$ and $n \in \mathbb{N}$:*

$$\text{depth}(x) \leq n \iff x \in B_n \cup E_n.$$

Openness

Earlier, we saw that the unfolding $\mathcal{U}(G)$ represents the behaviour of G according to the individual token game for nets. Paths according to the individual token game were described earlier, in Section 7.4, as causal nets. We would therefore expect the unfolding $\mathcal{U}(G)$ to be related to G through a bisimulation that respects the individual token game. We shall show that this is indeed the case. In particular, we shall show the morphism $\varepsilon_G{:}\mathcal{U}(G) \to G$ described in Theorem 7.12 above to be **Caus**-open[4].

In Appendix D, open morphisms from occurrence nets into general nets are characterized in the following way:

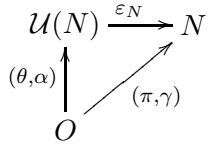
Proposition 7.14 (Theorem D.1 in Appendix D)**.** *Let O be an occurrence net and G be a general net. A morphism $f{:}O \to G$ is* **Caus***-open in* $\mathbf{Gen}^\sharp$ *if, and only if, it is a folding morphism and reflects any initial marking of G to an initial marking of O and satisfies the following 'transition lifting' property:*

> *for any subset A of conditions of O such that* co A *for which there exists a transition t of G such that $f \cdot A = Pre_G \cdot t$, there exists an event e of O such that $A = {}^\bullet e$ and $f(e) = t$.*

The morphism $\varepsilon_G{:}\mathcal{U}(G) \to G$ of Theorem 7.12 is readily seen to be a folding morphism satisfying this property. This will prove important later on in Proposition 8.3, where it allows us to form a symmetry on the unfolding.

Cofreeness

For a safe net N, we are able to say that the occurrence net $\mathcal{U}(N)$ and morphism $\varepsilon_N{:}\mathcal{U}(N) \to N$ are *cofree* [Mac71]. That is, for any occurrence net O and morphism $(\pi, \gamma){:}O \to N$, there is a *unique* morphism $(\theta, \alpha){:}O \to \mathcal{U}(N)$ such that the following triangle commutes:



The result, first shown in [Win86] (for singly-marked nets), ensures that $\mathbf{Occ}^\sharp$ is a coreflective subcategory of $\mathbf{Safe}^\sharp$ with the operation of unfolding giving rise to a functor that is right-adjoint to the inclusion functor $\mathbf{Occ}^\sharp \hookrightarrow \mathbf{Safe}^\sharp$. In fact, as we shall see, the result generalizes straightforwardly to multiply-marked nets and also applies to give a coreflection between the category of P/T nets $\mathbf{PT}^\sharp$ and the category of occurrence nets $\mathbf{Occ}^\sharp$. More generally still, it generalizes to give a coreflection between semi-weighted nets (nets

[4]It would be interesting to establish a connection between openness with respect to causal nets and the bisimulation in [vG05], perhaps corresponding to openness with respect to *extensional* causal nets.

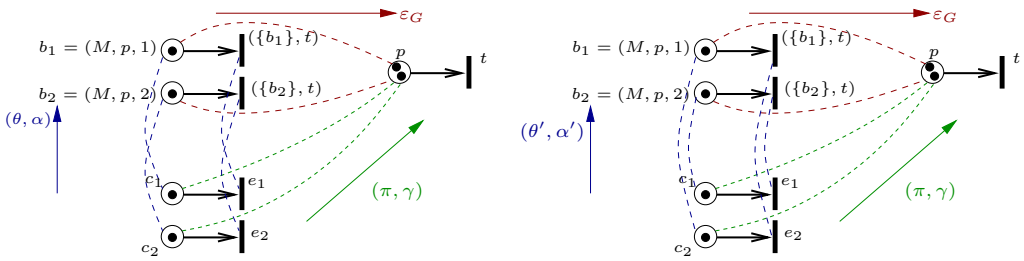


Figure 7.3: Non-uniqueness of mediating morphism (all multiplicities 1)

with single multiplicity in the post-places of each transition and that have at most one token in each place in the initial marking) and occurrence nets, as shown in [MMS96].

A coreflection is not, however, obtained when we consider the unfoldings of arbitrary general nets to occurrence nets (either singly- or multiply-marked). As we have seen, the problem does not lie in defining the unfolding of general nets; the unfolding operation is extended straightforwardly to general nets. Instead, the reason why we do not obtain a coreflection between the categories $\mathbf{Occ}^\sharp$ and $\mathbf{Gen}^\sharp$ (or $\mathbf{Occ}$ and $\mathbf{Gen}$) is that the uniqueness property required for cofreeness fails. That is, the morphism (θ, α) need not be the *unique* such morphism making the diagram above commute. In Figure 7.3, we present a general net G, its unfolding $\mathcal{U}(G)$ with the morphism ε_G and an occurrence net O, which happens to be isomorphic to $\mathcal{U}(G)$, with a morphism $(\pi, \gamma){:}O \to G$ alongside two distinct morphisms $(\theta, \alpha), (\theta', \alpha'){:}O \to \mathcal{U}(G)$ making the diagram commute.

In the net $\mathcal{U}(G)$ in Figure 7.3, the two conditions b_1 and b_2 are symmetric: they arise from there being two indistinguishable tokens in the initial marking of G in the place p. The events $(\{b_1\}, t)$ and $(\{b_2\}, t)$ are also symmetric since they are only distinguished by their symmetric pre-conditions; they have common image under ε_G. Our goal shall be to show that there is a unique mediating morphism *up to symmetry*, *i.e.* any two morphisms from O to $\mathcal{U}(G)$ making the diagram commute are only distinguished through their choice of symmetric elements of the unfolding. We first summarize the part of the cofreeness property that does hold.

Theorem 7.15. *Let G be a general Petri net, O be an occurrence net and $(\pi, \gamma){:}O \to G$ be a morphism in $\mathbf{Gen}^\sharp$. There is a morphism $(\theta, \alpha){:}O \to \mathcal{U}(G)$ in $\mathbf{Gen}^\sharp$ such that the following diagram commutes:*

$$\begin{array}{ccc} \mathcal{U}(G) & \xrightarrow{(\eta,\beta)=\varepsilon_G} & G \\ {\scriptstyle(\theta,\alpha)}\uparrow & \nearrow_{(\pi,\gamma)} & \\ O & & \end{array}$$

Furthermore, if the net G is a P/T net then (θ, α) is the unique such morphism.

Proof. Recall that $O\lceil n$ is the occurrence net O restricted to its elements at depth n. From the morphism $(\pi, \gamma){:}O \to G$, we obtain a morphism which we shall write $(\pi_n, \gamma_n){:}O\lceil n \to G$. For $n \in \mathbb{N}$, we shall begin by constructing a morphism $(\theta_n, \alpha_n){:}O\lceil n \to \mathcal{U}(G)$ such that the following diagram commutes:

$$\begin{array}{ccc} \mathcal{U}_n(G) & \xrightarrow{(\eta_n,\beta_n)} & G \\ {\scriptstyle(\theta_n,\alpha_n)}\uparrow & \nearrow_{(\pi_n,\gamma_n)} & \\ O\lceil n & & \end{array}$$

We shall also show that this is the unique such morphism if G is a P/T net.

We construct the morphism (θ_n, α_n), which will comprise a partial function on events and a *relation* on conditions, by induction on n. Let the nets

$$\begin{aligned} G &= (P, T, Pre, Post, \mathbb{M}) \\ O\lceil n &= (B, E, F, \mathbb{M}_0) \\ \mathcal{U}(G) &= (B_{\mathcal{U}}, E_{\mathcal{U}}, F_{\mathcal{U}}, \mathbb{M}_{\mathcal{U}}). \end{aligned}$$

The base case has $n = 0$. Any element of O at depth zero is a condition in an initial marking, so we define θ_0 to be the empty function. For any $M \in \mathbb{M}_0$, we have $\gamma \cdot M \in \mathbb{M}$ since (π, γ) is a morphism. There is therefore a bijection (unique if G is a P/T net since $(\gamma \cdot M)[p] \leq 1$ in this case) for every $M \in \mathbb{M}_0$ and $p \in P$

$$\varphi_{M,p}{:}\{(b, i) \mid b \in M \ \&\ 0 \leq i < \gamma[b, p]\} \cong \{j \mid 0 \leq j < (\gamma \cdot M)[p]\}.$$

This relies on the definition in Appendix A of application of a multirelation to a multiset. Any condition b at depth zero in an occurrence net is in a unique initial marking M. Define, for any $b' \in \mathcal{U}_n(G)$:

$$\alpha_0(b, b') \iff \exists p, i : 0 \leq i < \gamma[b, p] \ \&\ b' = (\gamma \cdot M, p, \varphi_{M,p}(b, i))$$

It is easy to see that this is a morphism $(\theta_0, \alpha_0){:}O\lceil 0 \to \mathcal{U}G$ such that the triangle above commutes and is the unique such morphism if G is a P/T net.

We now consider the case where $n > 0$. For any event e or condition b at depth less than n in $O\lceil n$ and any condition $b' \in B_{\mathcal{U}}$, define

$$\theta_n(e) = \theta_{n-1}(e) \qquad \alpha_n(b, b') \iff \alpha_{n-1}(b, b').$$

Now let e be an event at depth precisely n in $O\lceil n$. We extend θ_{n-1} by defining

$$\theta_n(e) = \begin{cases} * & \text{if } \pi(e) = * \\ (A, t) & \text{if } t = \pi(e) \ \&\ A = \theta_{n-1}{}^{\bullet}e \end{cases}$$

If G is a P/T net, since $(\theta_{n-1}, \alpha_{n-1})$ is the unique morphism up to depth $n-1$, it can be seen that θ_n has to be defined in this way.

Now let b be any condition at depth precisely n in $O\lceil n$. There exists a unique event e such that $b \in e^{\bullet}$ since O is an occurrence net. Since (π, γ) is a morphism, for every place $p \in P$ and event $e \in E$ there is a bijection

$$\varphi_{e,p}{:}\{(b, i) \mid b \in e^{\bullet} \ \&\ 0 \leq i < \gamma[b, p]\} \cong \{j \mid 0 \leq j < Post[\eta(e), p]\}.$$

This bijection is unique if G is a P/T net because $Post[\eta(e), p] \leq 1$ in this case. We define

$$\alpha_n(b, b') \iff \exists p, i : 0 \leq i < \gamma[b, p] \ \&\ b' = (\{e\}, p, \varphi_{e,p}(b, i)).$$

A straightforward analysis shows that this generates a morphism $(\theta_n, \alpha_n){:}O \to \mathcal{U}(G)$ such that the triangle above commutes and that this is the unique such morphism if G is a P/T net.

We now show how the set of morphisms $(\theta_n, \alpha_n){:}O\lceil n \to \mathcal{U}(G)$ generates a morphism $(\theta, \alpha){:}O \to \mathcal{U}(G)$ such that $(\eta, \beta) \circ (\theta, \alpha) = (\pi, \gamma)$, and that this is the unique such morphism if G is a P/T net.

We begin with the observation that

$$O\lceil 0 \leq O\lceil 1 \leq O\lceil 2 \leq \cdots$$

is an ω-chain and that $\bigsqcup_{n\in\mathbb{N}} O\lceil n = O$. Furthermore, we have a coherent set of morphisms $\{(\theta_n, \alpha_n) \mid n \in \mathbb{N}\}$. It follows from Proposition 7.11 that (θ, α) defined as

$$\begin{array}{rcll} \theta(e) & = & \theta_n(e) & \text{if depth}(e) \leq n \\ \alpha(b, b') & \iff & \alpha_n(b, b') & \text{if depth}(b) \leq n \end{array}$$

is a morphism $(\theta, \alpha){:}O \to \mathcal{U}(G)$ in $\mathbf{Gen}^\sharp$. It follows from commutation at depth n,

$$(\pi_n, \gamma_n) = (\eta, \beta) \circ (\theta_n, \alpha_n),$$

and the fact that any element of the occurrence net occurs at finite depth that $(\eta, \beta) \circ (\theta, \alpha) = (\pi, \gamma)$.

Suppose that G is a P/T net. By induction on n, using the remarks on uniqueness made above, we can show that $(\theta_n, \alpha_n){:}O\lceil n \to \mathcal{U}(G)$ is the unique morphism in $\mathbf{Gen}^\sharp$ such that $(\pi_n, \gamma_n) = (\eta, \beta) \circ (\theta_n, \alpha_n)$. It is easy to show from this and the fact that any element of the occurrence net O occurs at finite depth that (θ, α) is the unique morphism such that $(\pi, \gamma) = (\eta, \beta) \circ (\theta, \alpha)$. □

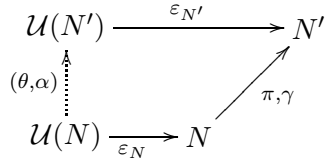
It will be of use later to note that if the multirelation γ above is a function then so is α, so the above result also applies if we substitute the category $\mathbf{Gen}^\sharp_\mathrm{f}$ of general nets with folding morphisms (morphisms that are total functions on conditions) for $\mathbf{Gen}^\sharp$.

From this result, we obtain an adjunction between the categories of occurrence nets and P/T nets.

$$\mathbf{Occ}^\sharp \underset{\mathcal{U}}{\overset{\perp}{\rightleftarrows}} \mathbf{PT}^\sharp$$

Since $\mathbf{Occ}^\sharp$ is a full subcategory of $\mathbf{PT}^\sharp$, the adjunction is a coreflection. The result is also seen to yield adjunctions between categories of singly-marked nets and, due to the observation above, categories of folding morphisms. However, due to the mediating morphism (θ, α) not necessarily being unique when we consider general Petri nets, we do not obtain an adjunction between $\mathbf{Occ}^\sharp$ and $\mathbf{Gen}^\sharp$. As discussed earlier, we must describe symmetry in the unfolding which will allow us to obtain a *pseudo*-adjunction.

As is the general case for an adjunction [Mac71], the cofreeness result specifies how the operation of the right adjoint on objects extends to a morphisms, yielding a functor $\mathcal{U}{:}\mathbf{PT}^\sharp \to \mathbf{Occ}^\sharp$. Suppose that we have a morphism between P/T nets $(\pi, \gamma){:}N \to N'$. The morphism $\mathcal{U}(\pi, \gamma)$ is defined (uniquely) to be the mediating morphism $(\theta, \alpha){:}\mathcal{U}(N) \to \mathcal{U}(N')$ arising from the cofreeness property in the following diagram:



Chapter 8

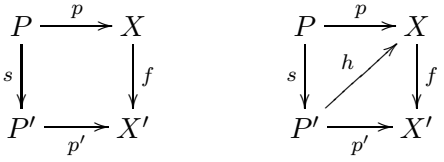
Symmetry and nets

We have seen in the previous chapter the unfolding operation on general Petri nets. In this chapter, we apply an abstract framework for defining symmetry in the behaviour of models for concurrency to enrich Petri nets with symmetry. This will be used to give a coreflection, up to symmetry, relating general Petri nets and occurrence nets. The key result is Theorem 8.6, where the key cofreeness property is shown to hold up to symmetry on nets. Following this result, we show that symmetry allows a coreflection between the category of P/T nets with symmetry and general nets with symmetry.

8.1 Categories with symmetry

It is shown in [Win07a, Win07b] how *symmetry* can be defined between the paths of event structures, and more generally on any category of models satisfying certain properties. As we shall see, we must generalize the framework to obtain an account of symmetry in Petri nets.

The definition of symmetry makes use of *open* morphisms [JNW95]. Let $\mathcal{C}_0$ be a category (typically a category of models such as Petri nets) with a distinguished subcategory $\mathcal{P}$ of path objects (such as causal nets: see Section 7.4) to describe the shape of computation paths, and morphisms specifying how a path extends to another. A morphism $f{:}X \to X'$ in $\mathcal{C}_0$ is $\mathcal{P}$-open if whenever there exists a morphism $s{:}P \to P'$ in $\mathcal{P}$ and morphisms $p{:}P \to X$ and $p'{:}P' \to X'$ in $\mathcal{C}_0$ such that the diagram on the left below commutes, there exists a morphism $h{:}P' \to X$ in $\mathcal{C}_0$ such that the two triangles in the diagram on the right commute:



We now describe the categories required for adding symmetry. Assume categories

$$\mathcal{P} \subseteq \mathcal{C}_0 \subseteq \mathcal{C}$$

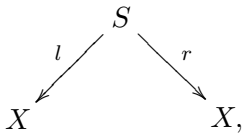
where $\mathcal{P}$ is a distinguished subcategory of path objects and path morphisms, $\mathcal{C}_0$ has pullbacks and shares the same objects as the (possibly larger) category $\mathcal{C}$, with the restriction that the inclusion functor $\mathcal{C}_0 \hookrightarrow \mathcal{C}$ preserves weak pullbacks[1]. Then, we will be able to add

[1] A weak pullback of the morphism $f_1{:}X_1 \to Y$ against $f_2{:}X_2 \to Y$ is an object W with morphisms $w_1{:}W \to X_1$ and $w_2{:}W \to X_2$ such that $f_1 \circ w_1 = f_2 \circ w_2$, and whenever the outer square in the diagram

symmetry to $\mathcal{C}$, and at the same time maintain constructions dependent on pullbacks of open morphisms, such as Lemma 8.1.1 below, which will be central to constructing symmetries on unfoldings.[2] (The earlier method for introducing symmetry used in [Win07a] corresponds to the situation where $\mathcal{C}_0$ and $\mathcal{C}$ coincide.)

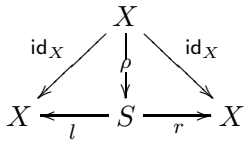
The role of $\mathcal{P} \subseteq \mathcal{C}_0$ is to determine open morphisms; the role of the subcategory $\mathcal{P}$ is to specify the form of path objects and extension, while the generally larger category $\mathcal{C}_0$ fixes the form of paths $p{:}P \to X$ from a path object P in an object X of $\mathcal{C}_0$.

Now we show how $\mathcal{C}$ can be extended with symmetry to yield a category $\mathcal{SC}$. (We sometimes write $\mathcal{S}_{\mathcal{P}\subseteq\mathcal{C}_0}C$ when we wish to highlight the particular path category with respect to which morphisms are open.) The objects of $\mathcal{SC}$ are spans



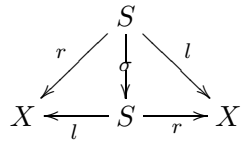
which we sometimes write (X, S, l, r), consisting of an object X of $\mathcal{C}$ and two $\mathcal{P}$-open morphisms $l, r{:}S \to X$ in $\mathcal{C}_0$ which make l, r a *pseudo-equivalence* [CV98] in the category $\mathcal{C}$. A span is a pseudo-equivalence if it satisfies the following axioms of reflexivity, symmetry and transitivity.

Reflexivity there is a morphism $\rho{:}X \to S$ in $\mathcal{C}$ such that



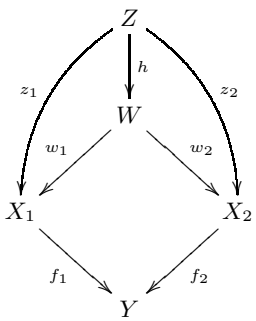
commutes;

Symmetry there is a morphism $\sigma{:}S \to S$ in $\mathcal{C}$ such that



---

below commutes, *i.e.* $f_1 \circ z_1 = f_2 \circ z_2$, there is a morphism $h{:}Z \to W$ such that the upper two triangles commute, *i.e.* $w_1 \circ h = z_1$ and $w_2 \circ h = z_2$.

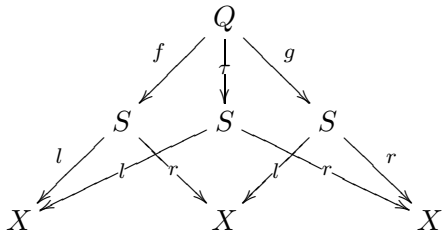


Note that this is the same as the universal characterization of a pullback apart from the mediating morphism h not necessarily being unique.

[2]General conditions have been chosen that work for our purposes here. It might become useful to replace the role of $\mathcal{P} \subseteq \mathcal{C}_0$ by an axiomatization of a subcategory of open morphisms in $\mathcal{C}$ and in this way broaden the class of situations in which we can adjoin symmetry.

commutes; and

Transitivity there is a weak pullback Q, f, g of r, l and a morphism $\tau{:}Q \to S$ in $\mathcal{C}$ such that



commutes.

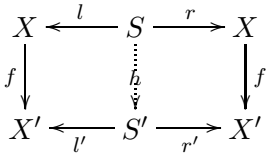
The requirements on l and r are slightly weaker than those in [Win07a] in that we do not require that the morphisms l and r are jointly monic — see Section 8.5 for an example of a symmetry on a safe net that cannot be expressed with the jointly-monic condition. They are also slightly weaker in the axiom for transitivity, which involves a weak pullback rather than a pullback. If the maps l and r are jointly monic and the axiom for transitivity is satisfied through taking a pullback, then they form an *equivalence* [Joh02].

It will be useful later to have a little notation to abbreviate the tuples forming an object with symmetry.

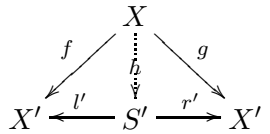
Notation 8.1.1. *We use the notation* $\mathbf{X}$ *to range over objects with symmetry. We write* X *for the object in which the symmetry is,* S_X *for the object representing the symmetry, and* $l_X, r_X{:}S_X \to X$ *for the symmetry morphisms. It follows that*

$$\mathbf{X} = (X, S_X, l_X, r_X).$$

The morphisms of $\mathcal{SC}$ are morphisms of $\mathcal{C}$ that *preserve symmetry*. Let $f{:}X \to X'$ be a morphism in $\mathcal{C}$ and (X, S, l, r) and (X', S', l', r') be objects of $\mathcal{SC}$. The morphism $f{:}X \to X'$ preserves symmetry if there is a morphism $h{:}S \to S'$ such that the following diagram commutes:



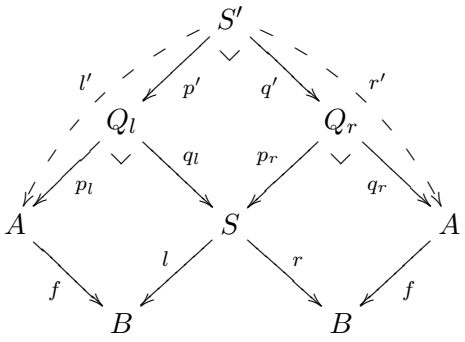
With the definition of symmetry on objects, we can define the equivalence relation $\sim$ expressing when morphisms are *equal up to symmetry*:
Let $f, g{:}(X, S, l, r) \to (X', S', l', r')$ be morphisms in $\mathcal{SC}$. Define $f \sim g$ iff there is a morphism $h{:}X' \to X'$ in $\mathcal{C}$ such that following diagram commutes in $\mathcal{C}$:



Composition of morphisms in $\mathcal{SC}$ coincides with composition in $\mathcal{C}$ and the two categories share the same identity morphisms. The category $\mathcal{SC}$ is more fully described as a category enriched in equivalence relations.

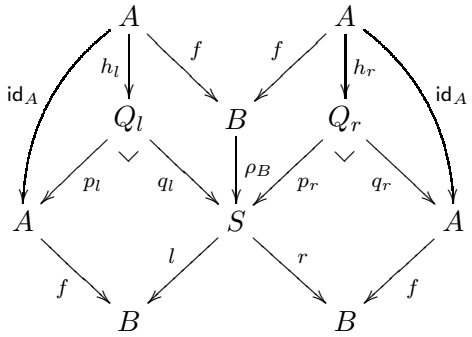
Later we make significant use of the following construction, the inverse image of a symmetry along an open morphism.

Lemma 8.1.1. *Given a symmetry l, r on B and a $\mathcal{P}$-*open *morphism $f : A \to B$ in $\mathcal{C}_0$ we obtain a symmetry l', r' on A as its inverse image along f, obtained via the following three pullbacks in $\mathcal{C}_0$:*



Proof. We must show that the maps l' and r' are open, which follows from pullbacks of open maps being open [JNW95], and that they form a pseudo-equivalence.

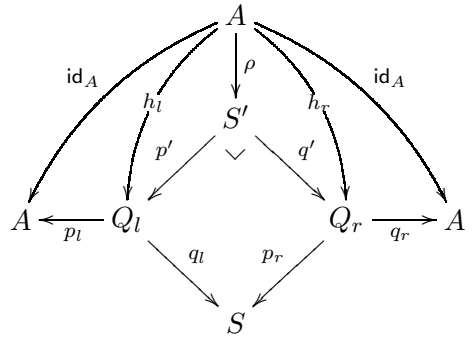
Reflexivity Let ρ_B be the morphism arising from the reflexivity of S on B. From Q_l and Q_r being pullbacks in $\mathcal{C}_0$ and hence weak pullbacks in $\mathcal{C}$, there are morphisms h_l and h_r such that the following diagram commutes:



It follows from S' being a weak pullback and, from commutation above, that

$$\rho_B \circ f = q_l \circ h_l = p_r \circ h_r,$$

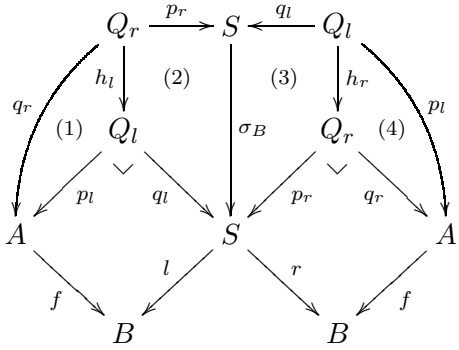
that there is a morphism $\rho{:}A \to S'$ such that the following diagram commutes:



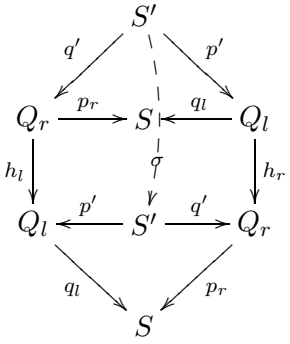
It is easy to see that this is a morphism satisfying the reflexivity axiom for (A, S', l', r').

Symmetry Let σ_B be the morphism arising from the symmetry of S on B. It follows from the definitions of σ_B, Q_l and Q_r that the following diagram without h_l and h_r

commutes. Since Q_l and Q_r are weak pullbacks in $\mathcal{C}$, there are morphisms h_l and h_r such that the whole of the following diagram commutes, and in particular squares (1)–(4) commute:



Recall that S' is the pullback of $q_l{:}Q_l \to S$ against $p_r{:}Q_r \to S$ in $\mathcal{C}_0$ and therefore a weak pullback in $\mathcal{C}$, with pullback morphisms $p'{:}S' \to Q_l$ and $q'{:}S' \to Q_r$. Consider the following diagram:



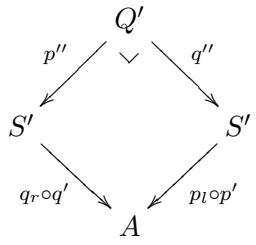
The outer hexagon commutes:

$$\begin{aligned} q_l \circ h_l \circ q' &= \sigma_B \circ p_r \circ q' && \text{by (2)} \\ &= \sigma_B \circ q_l \circ p' && \text{by def. } S' \text{ as pullback} \\ &= p_r \circ h_r \circ p' && \text{by (3)} \end{aligned}$$

It follows from S' being a pullback that there is a morphism $\sigma{:}S' \to S'$ such that $p' \circ \sigma = h_l \circ q'$ and $q' \circ \sigma = h_r \circ p'$. Using (1) and (4), a straightforward calculation shows that σ is the morphism required to show that (S', A', l', r') satisfies the symmetry requirement.

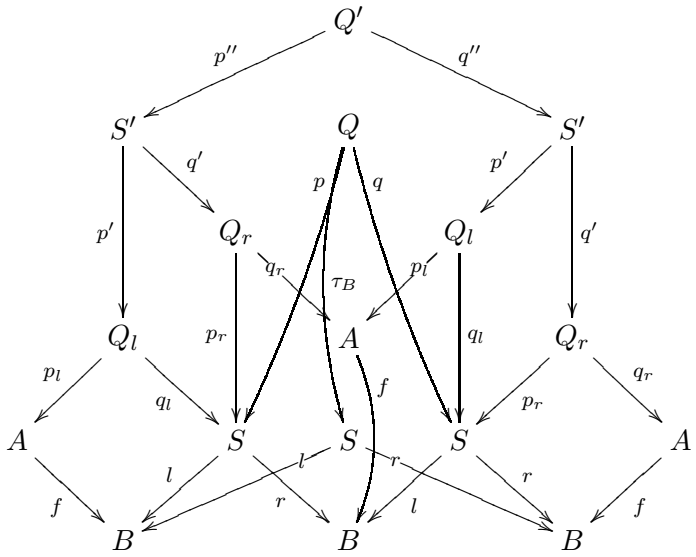
Transitivity The symmetry S on B is transitive, so there exists a weak pullback Q, p, q in $\mathcal{C}$ of r against l and a morphism $\tau_B{:}Q \to S$ such that $l \circ \tau_B = l \circ p$ and $r \circ \tau_B = r \circ q$.

Recalling the construction of the symmetry S' on A above, let Q', p'', q'' be the pullback of $r' = q_r \circ q'$ against $l' = p_l \circ p'$ in the category $\mathcal{C}_0$; this is a weak pullback in the category $\mathcal{C}$ since the inclusion of $\mathcal{C}_0$ in $\mathcal{C}$ preserves weak pullbacks.



We shall construct a morphism $\tau{:}Q' \to S'$ to show that the symmetry S', r', l' on A is transitive.

The pullback diagram above can be expanded by adding in the rest of the inverse image construction (twice) and drawing the morphism τ_B:



Let $y = p_r \circ q'{:}S' \to S$. We have $y = q_l \circ p'$ due to the definition of S' in the inverse image as a pullback.

From the commutation of the pullback diagram defining Q'' drawn above, we have

$$q_r \circ q' \circ p'' = p_l \circ p' \circ q'' \qquad (1).$$

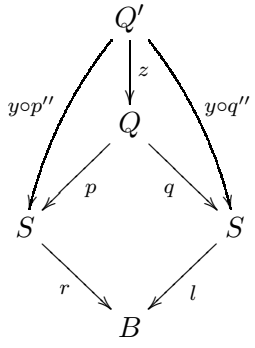
From commutation of the pullback diagrams defining Q_l and Q_r, respectively, we have

$$f \circ p_l = l \circ q_l \qquad (2) \qquad\qquad r \circ p_r = f \circ q_r \qquad (3).$$

Hence

$$\begin{array}{rcll} r \circ y \circ p'' & = & r \circ p_r \circ q' \circ p'' & \text{by def. } y \\ & = & f \circ q_r \circ q' \circ p'' & \text{by (3)} \\ & = & f \circ p_l \circ p' \circ q'' & \text{by (1)} \\ & = & l \circ q_l \circ p' \circ q'' & \text{by (2)} \\ & = & l \circ y \circ q'' & \text{by def. } y. \end{array}$$

It follows from Q being a weak pullback that there exists a morphism $z{:}Q \to Q'$ such that the following diagram commutes:



In particular,

$$p \circ z = y \circ p'' \qquad (4) \qquad\qquad q \circ z = y \circ q'' \qquad (5).$$

It follows that

$$l \circ y \circ p'' = l \circ p \circ z = l \circ \tau_B \circ z \qquad (6);$$

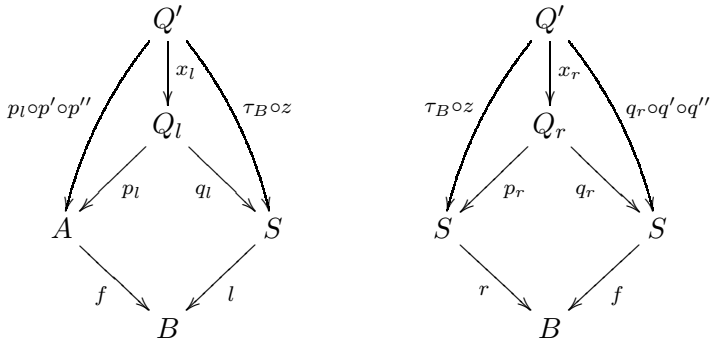
the first equality follows from (4) and the second from the definition of τ_B being a morphism demonstrating transitivity. Similarly, we have

$$r \circ y \circ q'' = r \circ q \circ z = r \circ \tau_B \circ z \qquad (7).$$

Now,

$$\begin{array}{rcll} f \circ p_l \circ p' \circ p'' & = & l \circ q_l \circ p' \circ p'' & \text{by (2)} \\ & = & l \circ y \circ p'' & \text{by def. } y \\ & = & l \circ \tau_B \circ z & \text{by (6).} \end{array}$$

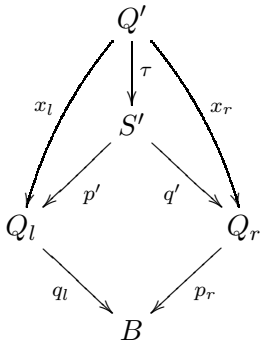
Similarly, we can show that $f \circ q_r \circ q' \circ q'' = r \circ \tau_B \circ z$. Since Q_l and Q_r are (weak) pullbacks, there exist morphisms x_l and x_r such that the following diagrams commute:



In particular, we have

$$p_l \circ x_l = p_l \circ p' \circ p'' \qquad (8) \qquad\qquad q_r \circ x_r = q_r \circ q' \circ q'' \qquad (9).$$

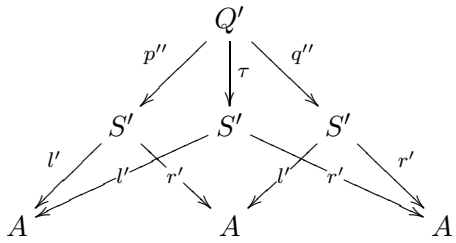
We also have $\tau_B \circ z = q_l \circ x_l = p_r \circ x_r$, so it follows from the definition of S' as a pullback that there is a morphism τ such that the following diagram commutes:



That is,

$$x_l = p' \circ \tau \qquad (10) \qquad\qquad x_r = q' \circ \tau \qquad (11).$$

To check that the morphism τ is indeed the morphism demonstrating transitivity of the symmetry S' on A, we must show that the following diagram commutes:



Recalling that $r' = q_r \circ q'$ and $l' = p_l \circ p'$, the inner square commutes according to the definition of p'' and q'' as pullback morphisms.

We have

$$\begin{aligned} l' \circ p'' &= p_l \circ p' \circ p'' \\ &= p_l \circ x_l && \text{by (8)} \\ &= p_l \circ p' \circ \tau && \text{by (10)} \\ &= l' \circ \tau, \end{aligned}$$

showing commutation of the left triangle. It can be shown similarly that $r' \circ q'' = r' \circ \tau$, as required to complete the proof. □

Lemma 8.1.1 depends crucially on the existence of pullbacks in $\mathcal{C}_0$ and the property that pullbacks of open morphisms are open (here weak pullbacks do not suffice) — without this we would not know that l' and r' were open.

Symmetry, functors and adjunctions

Let $\mathcal{C}$ and $\mathcal{D}$ be categories upon which symmetry can be placed, *i.e.* categories with subcategories $\mathcal{P} \subseteq \mathcal{C}_0 \subseteq \mathcal{C}$ and $\mathcal{Q} \subseteq \mathcal{D}_0 \subseteq \mathcal{D}$ such that $\mathcal{C}_0$ and $\mathcal{D}_0$ have pullbacks that are preserved as weak pullbacks in $\mathcal{C}$ and $\mathcal{D}$, respectively, and $\mathcal{P}$ and $\mathcal{Q}$ are path categories from which open maps can be drawn. We obtain the categories with symmetry $\mathcal{S}_{\mathcal{P} \subseteq \mathcal{C}_0}\mathcal{C}$ and $\mathcal{S}_{\mathcal{P} \subseteq \mathcal{C}_0}\mathcal{D}$, which from now on we shall abbreviate to $\mathcal{SC}$ and $\mathcal{SD}$.

Say that a functor $F{:}\mathcal{C} \to \mathcal{D}$ *preserves open maps in* $\mathcal{C}_0$ if, for any morphism $f{:}X \to X'$ in $\mathcal{C}_0$ that is $\mathcal{P}$-open in $\mathcal{C}_0$, the morphism $F(f){:}F(X) \to F(X')$ is in $\mathcal{D}_0$ and, moreover, is $\mathcal{Q}$-open in $\mathcal{D}_0$. Say that F preserves weak pullbacks of $\mathcal{P}$-open morphisms in $\mathcal{C}_0$ if, for any $f{:}X \to Y$ and $f'{:}X' \to Y$ that are $\mathcal{P}$-open in $\mathcal{C}_0$ and have a weak pullback P in $\mathcal{C}$ with pullback morphisms $p{:}P \to X$ and $p'{:}P \to X'$, then the object $F(P)$ with pullback morphisms $F(p)$ and $F(p')$ is a weak pullback of $F(f)$ and $F(f')$ in $\mathcal{D}$.

Proposition 8.1. *A functor $F{:}\mathcal{C} \to \mathcal{D}$ between categories described above yields a functor $\mathcal{S}F{:}\mathcal{SC} \to \mathcal{SD}$ defined on objects (X, S, l, r) of $\mathcal{SC}$ as*

$$\mathcal{S}F(X, S, l, r) = (FX, FS, Fl, Fr)$$

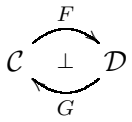
and on morphisms $f{:}(X, S, l, r) \to (X', S', l', r')$ as

$$\mathcal{S}F(f) = F(f)$$

if F preserves open maps in $\mathcal{C}_0$ and preserves weak pullbacks of $\mathcal{P}$-open maps in $\mathcal{C}_0$.

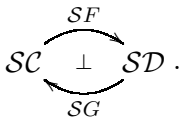
Proof. It is easy to see, given that F preserves weak pullbacks of $\mathcal{P}$-open morphisms in $\mathcal{C}_0$, that (FX, FS, Fl, Fr) satisfies the requirements to be an element of $\mathcal{SD}$. It is also easy to show that $\mathcal{S}F(f)$ is a map preserving symmetry as a consequence of f being a map preserving symmetry. □

Any adjunction



in which the functors F and G satisfy the constraints above of preserving open maps and preserving weak pullbacks of open maps (noting that the functor G automatically preserves all weak pullbacks as a consequence of it being a right adjoint: this is a specialization of the well-known property that right adjoints preserve limits) gives rise to an adjunction between the categories enriched with symmetry.

Proposition 8.2. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories on which symmetry can be placed. Suppose that the functor $F{:}\mathcal{C} \to \mathcal{D}$ preserves $\mathcal{P}$-open maps in $\mathcal{C}_0$ and preserves weak pullbacks of $\mathcal{P}$-open morphisms in $\mathcal{C}_0$, and suppose that the functor $G{:}\mathcal{C} \to \mathcal{D}$ preserves $\mathcal{Q}$-open maps in $\mathcal{D}_0$. If $F \dashv G$, i.e. F is left adjoint to G, then the functor $\mathcal{S}F{:}\mathcal{SC} \to \mathcal{SD}$ defined in Proposition 8.1 is left adjoint to the functor $\mathcal{S}G{:}\mathcal{SD} \to \mathcal{SC}$, i.e.*



Proof. Immediately from the assumptions, the functor $F{:}\mathcal{C} \to \mathcal{D}$ extends to a functor $\mathcal{S}F{:}\mathcal{SC} \to \mathcal{SD}$ according to Proposition 8.1. Since G is a right adjoint, it preserves weak pullbacks and therefore preserves weak pullbacks of $\mathcal{Q}$-open morphisms in $\mathcal{D}_0$. Therefore the functor $G{:}\mathcal{D} \to \mathcal{C}$ extends to a functor $\mathcal{S}G{:}\mathcal{SD} \to \mathcal{SC}$ by Proposition 8.1.

From the adjunction $F \dashv G$, there is an isomorphism of hom sets

$$\varphi_{X,Y}{:}\mathcal{C}(X, GY) \cong \mathcal{D}(FX, Y){:}\varphi^{-1}_{X,Y},$$

natural in X and Y. We must show that this yields an isomorphism

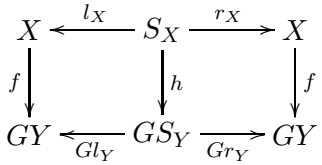
$$\psi_{\mathbf{X},\mathbf{Y}}{:}\mathcal{SC}(\mathbf{X}, \mathcal{S}G\mathbf{Y}) \cong \mathcal{SD}(\mathcal{S}F\mathbf{X}, \mathbf{Y}){:}\psi^{-1}_{\mathbf{X},\mathbf{Y}}$$

natural in $\mathbf{X} = (X, S_X, l_X, r_X)$ and $\mathbf{Y} = (Y, S_Y, l_Y, r_Y)$. In particular, we shall show that defining

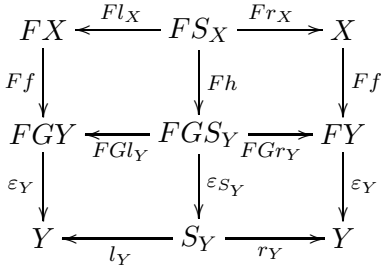
$$\psi_{\mathbf{X},\mathbf{Y}}(f) = \varphi_{X,Y}(f) \qquad \text{and} \qquad \psi^{-1}_{\mathbf{X},\mathbf{Y}}(g) = \varphi^{-1}_{X,Y}(g)$$

for any morphism $f{:}\mathbf{X} \to \mathcal{S}G\mathbf{Y}$ in $\mathcal{SC}$ and any morphism $g{:}\mathcal{S}F\mathbf{X} \to \mathbf{Y}$ in $\mathcal{SD}$ is such an isomorphism.

First, suppose that there is a morphism $f{:}\mathbf{X} \to \mathcal{S}G\mathbf{Y}$ in $\mathcal{SC}$. We shall show that $\varphi_{X,Y}(f){:}\mathcal{S}F\mathbf{X} \to \mathbf{Y}$ is a morphism in $\mathcal{SD}$. There is a morphism $h{:}S_X \to S_Y$ such that the following diagram commutes because f is a map preserving symmetry.



We now apply the functor F and add morphisms to represent the counit $\varepsilon{:}FG \Rightarrow \mathsf{id}_{\mathcal{C}}$ of the adjunction $F \dashv G$. The lower squares commute by the naturality of ε.



Due to the relationship between φ and ε from the adjunction $F \dashv G$, we have

$$\varepsilon_Y \circ Ff = \varphi_{X,Y}(f) \qquad \text{and} \qquad \varepsilon_{S_Y} \circ Fh = \varphi_{S_X,S_Y}(h).$$

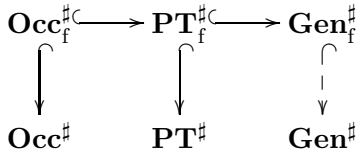
If follows that if $f{:}\mathbf{X} \to \mathcal{S}G\mathbf{Y}$ is a morphism in $\mathcal{SC}$, then $\varphi_{X,Y}(f){:}\mathcal{S}F\mathbf{X} \to \mathbf{Y}$ is a morphism in $\mathcal{SD}$, as required.

We can show, dually, that if $g{:}\mathcal{S}F\mathbf{X} \to G$ is a morphism in $\mathcal{SD}$ then $\varphi^{-1}_{X,Y}(g){:}\mathbf{X} \to \mathcal{S}G\mathbf{Y}$ is a morphism in $\mathcal{SD}$. It follows immediately from $\varphi_{X,Y}$ being an isomorphism that $\psi_{\mathbf{X},\mathbf{Y}}$ is an isomorphism. Naturality of $\psi_{\mathbf{X},\mathbf{Y}}$ in $\mathbf{X}$ and $\mathbf{Y}$ is a straightforward consequence of the naturality of $\varphi_{X,Y}$ in X and Y. □

8.2 Nets with symmetry

Applying symmetry to nets requires two key ingredients: the ability to take pullbacks as described above and open maps of nets with respect to some path category. In Appendix B, we consider pullbacks within categories of nets. We show that the categories $\mathbf{Occ}^\sharp$ and $\mathbf{PT}^\sharp$ have pullbacks, as do their subcategories $\mathbf{Occ}^\sharp_{\mathrm{f}}$ and $\mathbf{PT}^\sharp_{\mathrm{f}}$ — the categories restricted to folding morphisms. We show in Theorem B.2 that although the category $\mathbf{Gen}^\sharp$ does not have pullbacks, its subcategory $\mathbf{Gen}^\sharp_{\mathrm{f}}$ does. The following key lemma informs that pullbacks in $\mathbf{Gen}^\sharp_{\mathrm{f}}$ are *weak* pullbacks in $\mathbf{Gen}^\sharp$, which led to the adaptation of the abstract definition of symmetry above from that presented in [Win07a] to accommodate the failure of $\mathbf{Gen}^\sharp$ to have pullbacks. (The definition of weak pullbacks is repeated in the footnote on page 157)

Lemma 8.2.1 (Lemma B.2.8 in Appendix B)**.** *The solid inclusion functors in the following diagram preserve pullbacks and the dashed inclusion functor preserves weak pullbacks.*



With the understanding of pullbacks of nets and open maps with respect to causal nets as objects, we can apply the framework for defining symmetry described in Section 8.1 to obtain categories of nets enriched with symmetry.

Recall that a model equipped with symmetry is represented by three categories,

$$\mathcal{P} \subseteq \mathcal{C}_0 \subseteq \mathcal{C},$$

where $\mathcal{P}$ is the path category, the category $\mathcal{C}_0$ has pullbacks that are preserved as weak pullbacks in $\mathcal{C}$. As stated above, the category of general nets and all morphisms does not

have pullbacks. We therefore use the category of general nets with folding morphisms between them, $\mathbf{Gen}^\sharp_\mathrm{f}$, for the category $\mathcal{C}_0$. For general nets, a reasonable choice for the paths $\mathcal{P}$ would be $\mathbf{Caus}_\mathrm{f}$, taking path objects to be causal nets and expressing path extensions by foldings between them. The categories

$$\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Gen}^\sharp_\mathrm{f} \subseteq \mathbf{Gen}^\sharp$$

meet the requirements needed to construct $\mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Gen}^\sharp_\mathrm{f}}\mathbf{Gen}^\sharp$, so adjoining symmetry to general nets. The particular requirements that $\mathbf{Gen}^\sharp_\mathrm{f}$ should have pullbacks and that they are preserved as weak pullbacks in $\mathbf{Gen}^\sharp$ are met according to Theorem B.2 and Lemma B.2.8, respectively.

The requirements are also met by

$$\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Occ}^\sharp_\mathrm{f} \subseteq \mathbf{Occ}^\sharp$$

yielding $\mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Occ}^\sharp_\mathrm{f}}\mathbf{Occ}^\sharp$. In particular, the category $\mathbf{Occ}^\sharp_\mathrm{f}$ has pullbacks due to Theorem B.2 and Lemma B.2.6.

However, since the category $\mathbf{Occ}^\sharp$ has pullbacks of *all* morphisms, an 'alternative' category of occurrence nets with symmetry can be defined by relaxing the requirement that all symmetries should be formed from folding maps. This yields the category $\mathcal{S}_{\mathbf{Caus} \subseteq \mathbf{Occ}^\sharp}\mathbf{Occ}^\sharp$ obtained through the inclusions

$$\mathbf{Caus} \subseteq \mathbf{Occ}^\sharp \subseteq \mathbf{Occ}^\sharp.$$

In fact, these turn out to be exactly the same categories.

Lemma 8.2.2.

$$\mathcal{S}_{\mathbf{Caus} \subseteq \mathbf{Occ}^\sharp}\mathbf{Occ}^\sharp = \mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Occ}^\sharp_\mathrm{f}}\mathbf{Occ}^\sharp$$

Proof. The only non-trivial parts of the proof are to show that any **Caus**-open morphism $f{:}O \to O'$ in $\mathbf{Occ}^\sharp$ is a folding morphism and that any folding morphism is **Caus**-open in $\mathbf{Occ}^\sharp$ iff it is $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Occ}^\sharp_\mathrm{f}$.

Let $f{:}O \to O'$ be any **Caus**-open morphism in $\mathbf{Occ}^\sharp$. It is **Caus**-open in $\mathbf{Gen}^\sharp$ according to Lemma D.0.1. Therefore, according to Lemma D.0.2, the morphism f is a folding map. We shall now show that the folding morphism f is **Caus**-open in $\mathbf{Occ}^\sharp$ iff it is $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Occ}^\sharp_\mathrm{f}$.

($\Rightarrow$): Assume that f is **Caus**-open in $\mathbf{Occ}^\sharp$. By Lemma D.0.1 it is **Caus**-open in $\mathbf{Gen}^\sharp$. Since it is a folding morphism, by Lemma D.0.3 it is $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Gen}^\sharp_\mathrm{f}$. Just like in Lemma D.0.1, because $\mathbf{Occ}^\sharp_\mathrm{f}$ is a full subcategory of $\mathbf{Gen}^\sharp_\mathrm{f}$, a folding morphism $f{:}O \to O'$ in $\mathbf{Occ}^\sharp_\mathrm{f}$ is $\mathbf{Caus}_\mathrm{f}$-open iff it is $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Gen}^\sharp_\mathrm{f}$. It follows that f is $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Occ}^\sharp_\mathrm{f}$, as required.

($\Leftarrow$): Symmetric. □

We therefore just write $\mathcal{S}\mathbf{Occ}^\sharp$ for the category of occurrence nets with symmetry.

The same convenient property does not hold for P/T nets: we obtain the distinct categories with symmetry $\mathcal{S}_{\mathbf{Caus} \subseteq \mathbf{PT}^\sharp}\mathbf{PT}^\sharp$ and $\mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{PT}^\sharp_\mathrm{f}}\mathbf{PT}^\sharp$ arising from the respective inclusions

$$\mathbf{Caus} \subseteq \mathbf{PT}^\sharp \subseteq \mathbf{PT}^\sharp \qquad \text{and} \qquad \mathbf{Caus}_\mathrm{f} \subseteq \mathbf{PT}^\sharp_\mathrm{f} \subseteq \mathbf{PT}^\sharp.$$

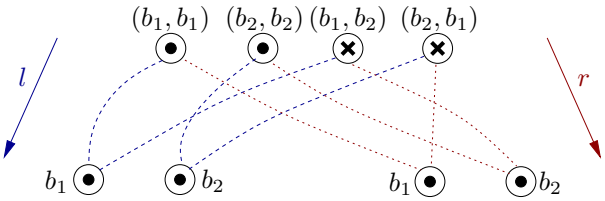


Figure 8.1: Symmetry in a net with two places

This is due to there being open maps between P/T nets that are not foldings — though as we see in Lemma D.0.2, the only way in which a **Caus**-open morphism in $\mathbf{PT}^\sharp$ can fail to be a folding morphism is through it being non-functional on a condition that never becomes marked.

There are other possibilities for defining symmetry on general nets, for example restricting paths to be the causal nets associated with with finite elementary event structures. They would, however, lead to less refined equivalences up to symmetry, leading to a weaker cofreeness-up-to-symmetry characterization of the unfolding.

8.3 Symmetry in unfolding

In Section 7.5, we showed how a general Petri net may be unfolded to form an occurrence net. This was shown not to yield a coreflection due to the mediating morphism not necessarily being unique, though it was observed that that uniqueness might be obtained by regarding the net up to the evident symmetry between paths in the unfolding. This led us to define a category of general nets with symmetry. To give an example of the forms of symmetry that can be expressed, consider the simple net with two places, b_1 and b_2, both initially marked once. Suppose that we wish to express that the two places are symmetric; for instance, the net might be thought of as the unfolding of the general net with a single place initially marked twice. The span to express that symmetry is presented in Figure 8.1. Without the extension of the definition of net to allow multiple initial markings, this simple symmetry would be inexpressible. This accompanies the fact that the category of singly-marked general nets (even when restricted to folding morphisms) does not have pullbacks.

In general, the symmetry in an unfolding is obtained by unfolding the *kernel* of the morphism $\varepsilon_G{:}\mathcal{U}(G) \to G$, which is the pullback of ε_G against itself in $\mathbf{Gen}^\sharp_{\mathrm{f}}$:

$$\begin{array}{ccccc} \mathcal{U}(S) & \xrightarrow{\varepsilon_S} & S & \xrightarrow{r} & \mathcal{U}(G) \\ & & {\scriptstyle l}\downarrow & \lrcorner & \downarrow{\scriptstyle \varepsilon_G} \\ & & \mathcal{U}(G) & \xrightarrow[\varepsilon_G]{} & G \end{array}$$

To see that $(\mathcal{U}(G), \mathcal{U}(S), l \circ \varepsilon_S, r \circ \varepsilon_S)$ is a symmetry in $\mathcal{S}\mathbf{Occ}^\sharp$, we must show that the morphisms $l \circ \varepsilon_S$ and $r \circ \varepsilon_S$ are **Caus**-open and form a pseudo equivalence.

Proposition 8.3. *The tuple $(\mathcal{U}(G), \mathcal{U}(S), l \circ \varepsilon_S, r \circ \varepsilon_S)$ is an occurrence net with symmetry, i.e. an object in $\mathcal{S}\mathbf{Occ}^\sharp$.*

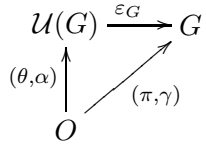
Proof. (Here we take $\mathcal{S}\mathbf{Occ}^\sharp$ to be $\mathcal{S}_{\mathbf{Caus}_{\mathrm{f}} \subseteq \mathbf{Occ}^\sharp_{\mathrm{f}}}\mathbf{Occ}^\sharp$.) For any general net G', the morphism $\varepsilon_{G'}{:}\mathcal{U}(G) \to G$ is readily seen using Theorem D.1 to be a $\mathbf{Caus}_{\mathrm{f}}$-open morphism in

$\mathbf{Gen}^\sharp_{\mathrm{f}}$. The pullback of open morphisms is open [JNW95], so the morphisms l and r are both $\mathbf{Caus}_{\mathrm{f}}$-open in $\mathbf{Gen}^\sharp_{\mathrm{f}}$. Open morphisms compose to form open morphisms [JNW95], so the maps $\varepsilon_S \circ l$ and $\varepsilon_S \circ r$ are also both $\mathbf{Caus}_{\mathrm{f}}$-open in $\mathbf{Gen}^\sharp_{\mathrm{f}}$. Hence, since $\mathbf{Occ}^\sharp_{\mathrm{f}}$ is a full subcategory of $\mathbf{Gen}^\sharp_{\mathrm{f}}$, by an argument similar to that in the proof of Lemma D.0.1 they are $\mathbf{Caus}_{\mathrm{f}}$-open in $\mathbf{Occ}^\sharp_{\mathrm{f}}$.

A standard diagrammatic argument, generalized in Proposition 8.5 shows that that the morphisms $l, r{:}\mathcal{U}(S) \to \mathcal{U}(G)$ form a pseudo-equivalence. □

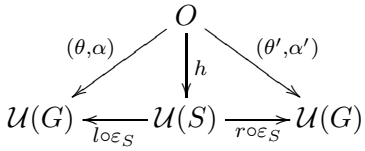
With the symmetry on $\mathcal{U}(G)$ at our disposal, we obtain the equivalence relation $\sim$ on morphisms from any occurrence net to $\mathcal{U}(G)$. This is used to extend Theorem 7.15 to obtain cofreeness 'up to symmetry'.

Theorem 8.4. *Let G be a general Petri net and O be an occurrence net. For any morphism $(\pi, \gamma){:}O \to G$ in $\mathbf{Gen}^\sharp$, there is a morphism $(\theta, \alpha){:}O \to \mathcal{U}(G)$ in $\mathbf{Gen}^\sharp$ such that*



commutes, i.e. $\varepsilon_G \circ (\theta, \alpha) = (\pi, \gamma)$. Furthermore, any morphism $(\theta', \alpha'){:}O \to \mathcal{U}(G)$ in $\mathbf{Gen}^\sharp$ such that $\varepsilon_G \circ (\theta', \alpha') = (\pi, \gamma)$ satisfies $(\theta, \alpha) \sim (\theta', \alpha')$ with respect to the symmetry $(\mathcal{U}(S), l \circ \varepsilon_S, r \circ \varepsilon_S)$ on $\mathcal{U}(G)$ defined above (and the identity symmetry on O).

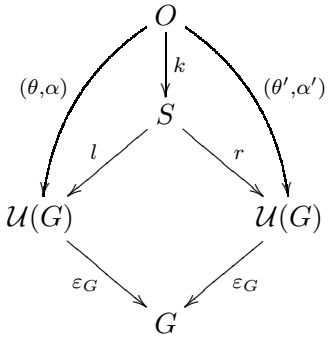
Proof. The morphism (θ, α) exists according to Theorem 7.15. Suppose that (θ', α') is a morphism as described. We wish to show that $(\theta, \alpha) \sim (\theta', \alpha')$ with respect to the given symmetry. This amounts to showing that there exists a morphism $h{:}O \to \mathcal{U}(S)$ such that the following diagram commutes:



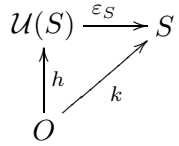
Recall that S, l, r is a pullback in $\mathbf{Gen}^\sharp_{\mathrm{f}}$ of ε_G against itself. It is therefore a weak pullback in $\mathbf{Gen}^\sharp$ since the inclusion $\mathbf{Gen}^\sharp_{\mathrm{f}} \hookrightarrow \mathbf{Gen}^\sharp$ preserves pullbacks as weak pullbacks. Since

$$\varepsilon_G \circ (\theta, \alpha) = (\pi, \gamma) = \varepsilon_G \circ (\theta', \alpha')$$

by assumption, there exists a morphism $k{:}O \to S$ in $\mathbf{Gen}^\sharp$ such that the following diagram commutes:



By Theorem 7.15, there exists a morphism $h{:}O \to \mathcal{U}(S)$ such that the following diagram commutes, *i.e.* $k = \varepsilon_S \circ h$:



We therefore have

$$\begin{array}{rrcccl} & (\theta, \alpha) & = & l \circ k & = & l \circ \varepsilon_S \circ h \\ \text{and} & (\theta', \alpha') & = & r \circ k & = & r \circ \varepsilon_S \circ h, \end{array}$$

as required. □

8.4 A coreflection up to symmetry

We now show how the results of the last section are part of a more general coreflection from occurrence nets *with symmetry* to general nets *with symmetry*, culminating in the key result, Theorem 8.6. In the last section, we showed how to unfold a general net to an occurrence net with symmetry. For the coreflection, we need to extend this construction to unfold general nets themselves with symmetry.

To show that the 'inclusion' $I{:}\mathcal{S}\mathbf{Occ}^\sharp \to \mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Gen}^\sharp_\mathrm{f}}\mathbf{Gen}^\sharp$ taking an occurrence net with symmetry (O, S, l, r) to a general net with symmetry is a functor, it is necessary to show that the transitivity property holds of the symmetry in $\mathcal{S}\mathbf{Gen}^\sharp$. For this it is important that pullbacks are not disturbed in moving from $\mathbf{Occ}^\sharp_\mathrm{f}$ to the larger category $\mathbf{Gen}^\sharp_\mathrm{f}$, as is assured by Lemma B.2.8.

Lemma 8.4.1. *The inclusion* $I{:}\mathcal{S}\mathbf{Occ}^\sharp \to \mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Gen}^\sharp_\mathrm{f}}\mathbf{Gen}^\sharp$ *is a functor.*

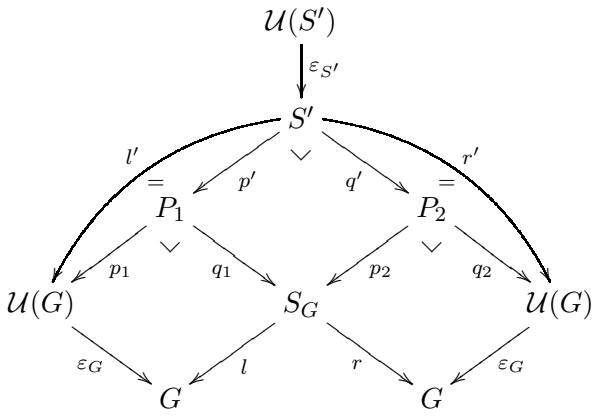
Proof. Let (O, S, l, r) be an object in $\mathcal{S}\mathbf{Occ}^\sharp$. By Lemma 8.2.2, the morphisms $l, r{:}S \to O$ are $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Occ}^\sharp_\mathrm{f}$. Since $\mathbf{Occ}^\sharp_\mathrm{f}$ is a full subcategory of $\mathbf{Gen}^\sharp_\mathrm{f}$, it can be shown in the same way as Lemma D.0.1 that l and r are $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Gen}^\sharp_\mathrm{f}$.

Reflexivity and symmetry of the span (O, S, l, r) in $\mathbf{Gen}^\sharp$ is an immediate consequence of its reflexivity and symmetry in $\mathbf{Occ}^\sharp$. Transitivity in $\mathbf{Gen}^\sharp$ relies on the fact that the inclusion $\mathbf{Occ}^\sharp_\mathrm{f} \hookrightarrow \mathbf{Gen}^\sharp_\mathrm{f} \hookrightarrow \mathbf{Gen}^\sharp$ preserves weak pullbacks as a consequence of $\mathbf{Occ}^\sharp_\mathrm{f} \hookrightarrow \mathbf{Gen}^\sharp_\mathrm{f}$ preserving pullbacks and $\mathbf{Gen}^\sharp_\mathrm{f} \hookrightarrow \mathbf{Gen}^\sharp$ preserving weak pullbacks.

Any morphism in $\mathbf{Occ}^\sharp$ is clearly a morphism in $\mathbf{Gen}^\sharp$, and it follows immediately from the definition that any map preserving symmetry in $\mathbf{Occ}^\sharp$ is a map preserving symmetry in $\mathbf{Gen}^\sharp$. □

We now have a functor $I{:}\mathcal{S}\mathbf{Occ}^\sharp \to \mathcal{S}\mathbf{Gen}^\sharp$, respecting $\sim$, regarding an occurrence net with symmetry (O, S, l, r) itself directly as a general net with symmetry.

It remains for us to define the unfolding operation on objects of the category of general nets with symmetry. Its extension to a *pseudo*-functor will follow from the biadjunction. Let (G, S_G, l, r) be a general net with symmetry. Let $\varepsilon_G{:}\mathcal{U}(G) \to G$ be the folding morphism given earlier in Theorem 7.12. It is open by Theorem D.1. The general net (G, S_G, l, r) is 'unfolded' to the occurrence net with symmetry $\mathcal{U}(G, S_G, l, r) = (\mathcal{U}(G), S_0, l_0, r_0)$; its symmetry, $S_0 \triangleq \mathcal{U}(S')$, $l_0 \triangleq l' \circ \varepsilon_{S'}$ and $r_0 \triangleq r' \circ \varepsilon_{S'}$, is given by unfolding the inverse image of the symmetry in G along the open morphism $\varepsilon_G{:}\mathcal{U}(G) \to G$:



The pullbacks are in $\mathbf{Gen}^\sharp_\mathrm{f}$. This diagram makes clear that ε_G is a morphism preserving symmetry.

Proposition 8.5. *Let* $\mathbf{G}$ *be an object in* $\mathcal{S}_{\mathbf{Caus}_\mathrm{f}\subseteq\mathbf{Gen}^\sharp_\mathrm{f}}\mathbf{Gen}^\sharp$*, i.e. a general net with symmetry. Then* $\mathcal{U}(\mathbf{G})$ *as defined above is an object in* $\mathcal{S}\mathbf{Occ}^\sharp$*, i.e. an occurrence net with symmetry.*

Proof. We shall show that $l' \circ \varepsilon_S$ and $r' \circ \varepsilon_S$ are a pair of $\mathbf{Caus}_\mathrm{f}$-open morphisms in $\mathbf{Occ}^\sharp_\mathrm{f}$ that form a pseudo-equivalence in $\mathbf{Occ}^\sharp$.

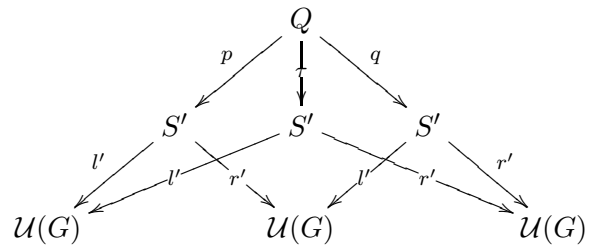
The morphism ε_G is $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Gen}^\sharp_\mathrm{f}$, as is the morphism $\varepsilon_{S'}$. Pullbacks of open morphisms are open, so the morphisms l' and r' are both $\mathbf{Caus}_\mathrm{f}$-open. Open morphisms compose to form open morphisms, so $l' \circ \varepsilon_S$ and $r' \circ \varepsilon_S$ are both $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Gen}^\sharp_\mathrm{f}$. Since $\mathbf{Occ}^\sharp_\mathrm{f}$ is a full subcategory of $\mathbf{Gen}^\sharp_\mathrm{f}$, they are also $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Occ}^\sharp_\mathrm{f}$ by an analogue of Lemma D.0.1.

In Lemma 8.1.1, it is shown that there are morphisms

$$\rho{:}\mathcal{U}(G) \to S' \qquad \sigma{:}S' \to S' \qquad \tau{:}Q \to S'$$

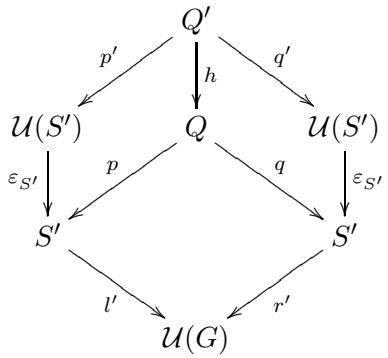
respectively demonstrating the reflexivity, symmetry and transitivity in $\mathbf{Gen}^\sharp$ of the symmetry (S', l', r') on $\mathcal{U}(G)$, the inverse image of the symmetry on G under the morphism ε_G. The first two give rise to morphisms demonstrating reflexivity and symmetry of the symmetry $\mathcal{U}(S)$, $\varepsilon_S \circ l'$, $\varepsilon_S \circ r'$ by a straightforward argument using the cofreeness property of the unfolding $\mathcal{U}(S)$ in Theorem 7.15. We now consider transitivity.

The morphism $\tau{:}Q \to S'$, for some weak pullback Q, p, q in $\mathbf{Gen}^\sharp$ of r' against l', makes the following diagram commute:

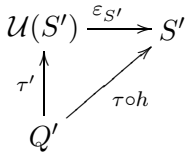


Let Q', p', q' be the pullback in $\mathbf{Occ}^\sharp_\mathrm{f}$ of $l' \circ \varepsilon_{S'}{:}\mathcal{U}(S') \to \mathcal{U}(G)$ against $r' \circ \varepsilon_{S'}{:}\mathcal{U}(S') \to \mathcal{U}(G)$. The inclusion $\mathbf{Occ}^\sharp_\mathrm{f} \hookrightarrow \mathbf{Occ}^\sharp$ preserves (weak) pullbacks by Lemma B.2.8, so Q' is a weak

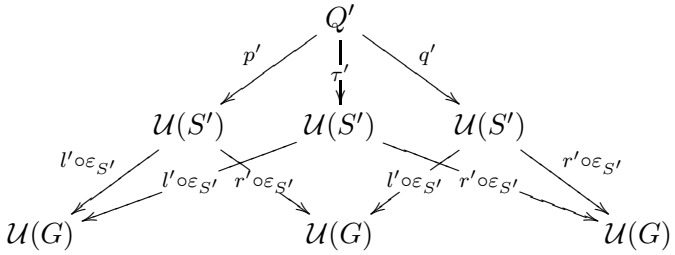
pullback in $\mathbf{Occ}^\sharp$ of $l' \circ \varepsilon_{S'}$ against $r' \circ \varepsilon_{S'}$. Using the assumption that Q is a weak pullback in $\mathbf{Gen}^\sharp$, there is a morphism $h{:}Q' \to Q$ such that the following diagram commutes:



Applying the cofreeness property as stated in Theorem 7.15 for the unfolding of S', there is a morphism $\tau'{:}Q' \to \mathcal{U}(S')$ in $\mathbf{Gen}^\sharp$ such that the following diagram commutes:



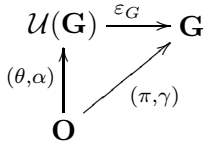
Since $\mathbf{Occ}^\sharp$ is a full subcategory of $\mathbf{Gen}^\sharp$, the morphism τ' is a morphism in $\mathbf{Occ}^\sharp$. It is now straightforward to show that the following diagram commutes:



Hence we have demonstrated that the symmetry induced on $\mathcal{U}(G)$ by the symmetry on G is transitive. □

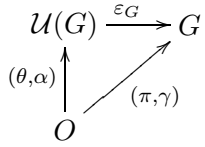
Now that we have the inclusion $I{:}\mathcal{S}\mathbf{Gen}^\sharp \to \mathcal{S}\mathbf{Occ}^\sharp$ and the operation of unfolding a general net with symmetry, we are able to improve Theorem 7.15 to give a 'cofreeness up to symmetry' result. This key result shows how regarding the unfolding of general nets up to their natural symmetry allows a form of coreflection to be obtained.

Theorem 8.6. *Let $\mathbf{G} = (G, S_G, l_G, r_G)$ be a general net with symmetry and $\mathbf{O} = (O, S_O, l_O, r_O)$ be an occurrence net with symmetry. For any $(\pi, \gamma){:}\mathbf{O} \to \mathbf{G}$ in $\mathcal{S}\mathbf{Gen}^\sharp$, there is a morphism $(\theta, \alpha){:}\mathbf{O} \to \mathcal{U}(\mathbf{G})$ in $\mathcal{S}\mathbf{Gen}^\sharp$ such that the following diagram commutes:*

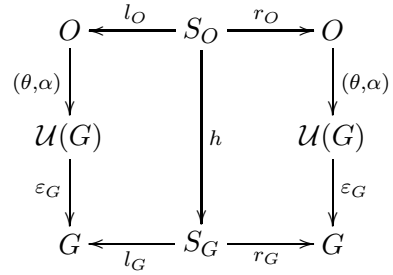


Furthermore, (θ, α) is unique up to symmetry: any $(\theta', \alpha'){:}\mathbf{O} \to \mathcal{U}(\mathbf{G})$ such that $\varepsilon_G \circ (\theta', \alpha') \sim (\pi, \gamma)$ satisfies $(\theta, \alpha) \sim (\theta', \alpha')$.

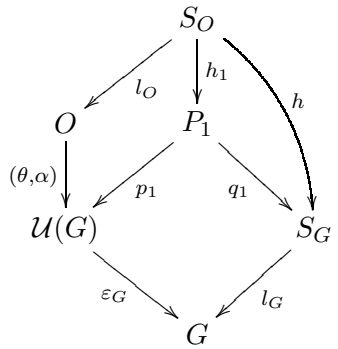
Proof. By Theorem 7.15, there is a morphism $(\theta, \alpha){:}O \to G$ such that the following diagram commutes:



We begin by showing that the morphism (θ, α) is in $\mathcal{S}\mathbf{Gen}^\sharp$, *i.e.* that (θ, α) is a map preserving symmetry. We know that $(\pi, \gamma){:}O \to G$ is a map preserving symmetry, so there is a morphism $h{:}S_O \to S_G$ such that the following diagram commutes:



Recall from the definition of the pullbacks in $\mathbf{Gen}^\sharp_{\mathrm{f}}$ used to define the symmetry on $(\mathcal{U}(G), \mathcal{U}(S'), l' \circ \varepsilon_{S'}, r' \circ \varepsilon_{S'})$ on page 171 that P_1 is a pullback in $\mathbf{Gen}^\sharp_{\mathrm{f}}$ of ε_G against l_G. It is a weak pullback in $\mathbf{Gen}^\sharp$. It follows that there exists a morphism $h_1{:}S_O \to P_1$ in $\mathbf{Gen}^\sharp$ such that the following diagram commutes:



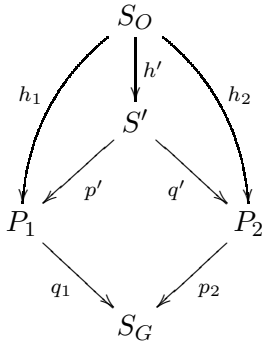
Similarly, there exists a morphism $h_2{:}S_O \to P_2$ in $\mathbf{Gen}^\sharp$ such that

$$(\theta, \alpha) \circ r_O = q_2 \circ h_2 \qquad \text{and} \qquad h = p_2 \circ h_2.$$

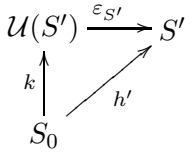
Note that

$$q_1 \circ h_1 = h = p_2 \circ h_2.$$

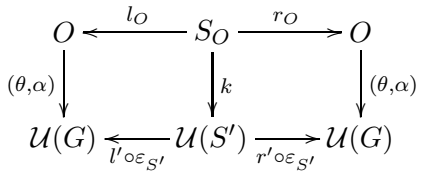
Since S' is a weak pullback in $\mathbf{Gen}^\sharp$, there exists a morphism $h'{:}S_O \to S'$ such that the following diagram commutes:



From the cofreeness property as presented in Theorem 7.15 for S', there is a morphism $k{:}S_0 \to \mathcal{U}(S')$ such that the following diagram commutes:

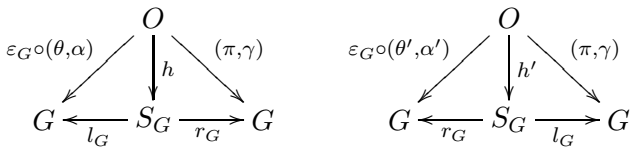


It is now a straightforward calculation to show that k is a map making the two squares in the following diagram commute, and therefore (θ, α) is a map preserving symmetry.

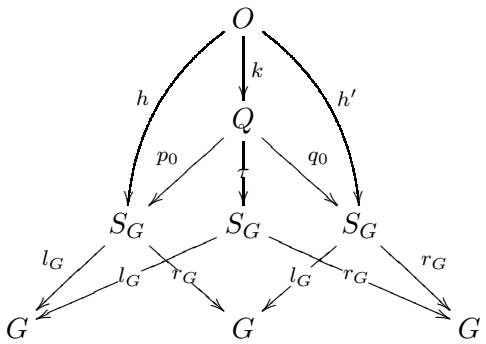


We now show that any map preserving symmetry $(\theta', \alpha'){:}\mathbf{O} \to \mathcal{U}(\mathbf{G})$ such that $(\pi, \gamma) \sim \varepsilon_G \circ (\theta', \alpha')$ satisfies $(\theta, \alpha) \sim (\theta', \alpha')$.

Since we have $(\pi, \gamma) = \varepsilon_G \circ (\theta, \alpha)$ we must have $(\pi, \gamma) \sim \varepsilon_G \circ (\theta, \alpha)$ since $\sim$ is reflexive. By assumption, we also have $(\pi, \gamma) \sim \varepsilon_G \circ (\theta', \alpha')$, so there exist morphisms $h, h'{:}O \to S_G$ such that the following two diagrams commute:

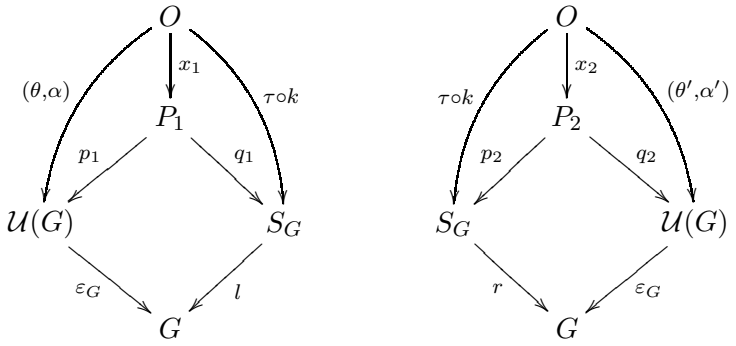


The symmetry (G, S_G, l, r) is required to be transitive, so there exists a morphism $k{:}O \to Q$ and morphism $\tau{:}Q \to S_G$ such that the following diagram commutes, recalling that Q, p_0, q_0 is a weak pullback in $\mathbf{Gen}^\sharp$ of r against l:

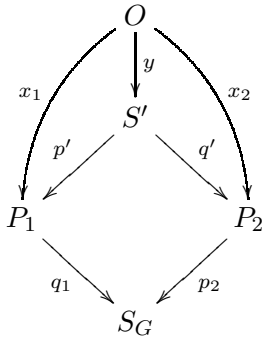


Let P_1 and P_2 be the pullbacks in $\mathbf{Gen}^\sharp_{\mathrm{f}}$ drawn in the definition of the symmetry $\mathcal{U}(S')$ on page 171. They are weak pullbacks in $\mathbf{Gen}^\sharp$, so there exist morphisms $x_1{:}O \to P_1$ and

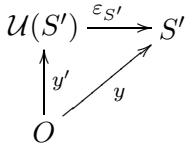
$x_2{:}O \to P_2$ such that the following diagrams commute:



Since S', p', q' is a pullback in $\mathbf{Gen}^\sharp_\mathrm{f}$ of q_1 against p_2 and therefore a weak pullback in $\mathbf{Gen}^\sharp$, there is a morphism $y{:}O \to S'$ such that the following diagram commutes:



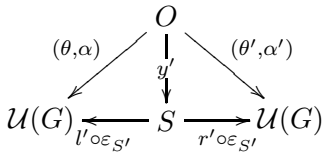
From the 'cofreeness' result, Theorem 7.15, there is a map $y'{:}O \to \mathcal{U}(S')$ such that the following diagram commutes:



From commutation of the diagrams above, we see that

$$\begin{array}{lllllllll} l' \circ \varepsilon_{S'} \circ y' & = & l' \circ y & = & p_1 \circ p' \circ y & = & p_1 \circ x_1 & = & (\theta, \alpha) \\ r' \circ \varepsilon_{S'} \circ y' & = & r' \circ y & = & q_2 \circ q' \circ y & = & q_2 \circ x_2 & = & (\theta', \alpha'), \end{array}$$

or, diagrammatically, that the following diagram commutes:



It follows immediately that $(\theta, \alpha) \sim (\theta', \alpha')$, as required. □

Technically, we have a biadjunction from the category of occurrence nets with symmetry $\mathcal{S}\mathbf{Occ}^\sharp$ to the category of general nets with symmetry $\mathcal{S}\mathbf{Gen}^\sharp$ with I left biadjoint to $\mathcal{U}$ (which extends to a pseudo-functor). Its counit is ε and its unit is a natural isomorphism $\mathbf{O} \cong \mathcal{U}(\mathbf{O})$. In this sense, we have established a coreflection from $\mathcal{S}\mathbf{Occ}^\sharp$ to $\mathcal{S}\mathbf{Gen}^\sharp$ up to symmetry [Pow98].

8.5 Symmetry and P/T nets

Now that we have shown the main result, giving a 'cofreeness up to symmetry' result to characterize the occurrence net unfolding of general nets, we begin to complete the picture by giving adjunctions between other categories of nets with symmetry. The key result is Theorem 8.9, in which a coreflection up to symmetry between P/T nets and general nets is shown.

Earlier, two categories of P/T nets with symmetry were highlighted:

$$\mathcal{S}_{\mathbf{Caus}\subseteq\mathbf{PT}^\sharp}\mathbf{PT}^\sharp \qquad \text{and} \qquad \mathcal{S}_{\mathbf{Caus}_f\subseteq\mathbf{PT}^\sharp_f}\mathbf{PT}^\sharp$$

The category on the left is simpler and allows more symmetries to be expressed, since a symmetry need not be a span of *folding* morphisms (that are open and form a pseudo equivalence) as they must be to give a symmetry in the category on the right. The extent of this distinction is, however, limited due to the observation in Lemma D.0.2 that all **Caus**-open morphisms in $\mathbf{PT}^\sharp$ are foldings apart from on places that can never become marked or on events that can never occur.

Coreflections with occurrence nets

We first show how the coreflection

$$\mathbf{Occ}^\sharp \underset{\mathcal{U}}{\overset{}{\perp}} \mathbf{PT}^\sharp$$

between occurrence nets and P/T nets extends to give coreflections between the categories enriched with symmetry

$$\mathcal{S}\mathbf{Occ}^\sharp \underset{\mathcal{SU}}{\overset{}{\perp}} \mathcal{S}\mathbf{PT}^\sharp\,,$$

where $\mathcal{S}\mathbf{PT}^\sharp$ is either one of the categories of P/T nets with symmetry.

Recall that

$$\mathcal{S}_{\mathbf{Caus}_f\subseteq\mathbf{Occ}^\sharp_f}\mathbf{Occ}^\sharp = \mathcal{S}_{\mathbf{Caus}\subseteq\mathbf{Occ}^\sharp}\mathbf{Occ}^\sharp = \mathcal{S}\mathbf{Occ}^\sharp.$$

To demonstrate the coreflection where $\mathcal{S}\mathbf{PT}^\sharp = \mathcal{S}_{\mathbf{Caus}\subseteq\mathbf{PT}^\sharp}\mathbf{PT}^\sharp$, we note that the inclusion $\mathbf{Occ}^\sharp \hookrightarrow \mathbf{PT}^\sharp$ and the functor $\mathcal{U}{:}\mathbf{PT}^\sharp \to \mathbf{Occ}^\sharp$ both preserve **Caus**-openness. This follows from the earlier coreflection and a general result about open maps being preserved through coreflections [JNW95, Lemma 6]. We also note that the functor $\mathcal{U}{:}\mathbf{PT}^\sharp \to \mathbf{Occ}^\sharp$ preserves weak pullbacks of all morphisms as a consequence of it being a right adjoint. All that remains before Proposition 8.2 can be applied to obtain the required coreflection between the categories with symmetry is to show that the inclusion $\mathbf{Occ}^\sharp \hookrightarrow \mathbf{PT}^\sharp$ preserves weak pullbacks of morphisms that are **Caus**-open in in $\mathbf{Occ}^\sharp$. This is a consequence of all **Caus**-open morphisms in $\mathbf{Occ}^\sharp$ being folding morphisms (Lemmas D.0.1 and D.0.2) and the preservation of pullbacks through the inclusion $\mathbf{Occ}^\sharp_f \hookrightarrow \mathbf{PT}^\sharp$ shown in Lemma B.2.8. Note that this is sufficient to show that the inclusion preserves weak pullbacks of all **Caus**-open morphisms in $\mathbf{Occ}^\sharp$ according to a general result, noted at the start of the proof of Lemma B.2.7, that a functor from a category with pullbacks preserves weak pullbacks iff it preserves pullbacks as weak pullbacks.

Theorem 8.7. *The unfolding functor*

$$\mathcal{SU}{:}\mathcal{S}_{\mathbf{Caus}\subseteq\mathbf{PT}^\sharp}\mathbf{PT}^\sharp \to \mathcal{S}_{\mathbf{Caus}\subseteq\mathbf{Occ}^\sharp}\mathbf{Occ}^\sharp$$

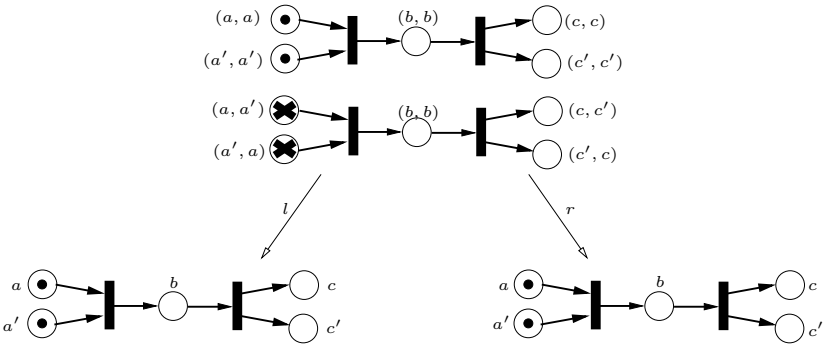


Figure 8.2: A symmetry (N, S, l, r) with (folding) morphisms $l(x, y) = x$ and $r(x, y) = y$.

is right adjoint to the inclusion

$$\mathcal{S}_{\mathbf{Caus}\subseteq\mathbf{Occ}^\sharp}\mathbf{Occ}^\sharp \hookrightarrow \mathcal{S}_{\mathbf{Caus}\subseteq\mathbf{PT}^\sharp}\mathbf{PT}^\sharp.$$

Furthermore, the adjunction is a coreflection. □

We now consider the coreflection where symmetries are restricted to being spans of folding maps, *i.e.* where $\mathcal{S}\mathbf{PT}^\sharp = \mathcal{S}_{\mathbf{Caus}_\mathrm{f}\subseteq\mathbf{PT}^\sharp_\mathrm{f}}\mathbf{PT}^\sharp$. As noted following Theorem 7.15, the functor $\mathcal{U}$ restricts to yield a coreflection

$$\mathbf{Occ}^\sharp_\mathrm{f} \underset{\mathcal{U}}{\overset{\curvearrowright}{\underset{\curvearrowleft}{\perp}}} \mathbf{PT}^\sharp_\mathrm{f}.$$

As before, it follows that both the inclusion $\mathbf{Occ}^\sharp \hookrightarrow \mathbf{PT}^\sharp$ preserves $\mathbf{Caus}_\mathrm{f}$-open maps in $\mathbf{Occ}^\sharp_\mathrm{f}$ and the functor $\mathcal{U}{:}\mathbf{PT}^\sharp \to \mathbf{Occ}^\sharp$ preserves $\mathbf{Caus}_\mathrm{f}$-open maps in $\mathbf{PT}^\sharp_\mathrm{f}$ according to Lemma 6 in [JNW95]. Since $\mathcal{U}$ is a right adjoint, it preserves pullbacks (and hence weak pullbacks) of all morphisms in $\mathbf{PT}^\sharp_\mathrm{f}$. Any pullback in $\mathbf{Occ}^\sharp_\mathrm{f}$ is a pullback in $\mathbf{PT}^\sharp_\mathrm{f}$ according to Lemma B.2.8, so the inclusion $\mathbf{Occ}^\sharp \hookrightarrow \mathbf{PT}^\sharp$ preserves weak pullbacks in $\mathbf{Occ}^\sharp_\mathrm{f}$ in the way required to apply Proposition 8.2. Applying this proposition, we obtain the desired coreflection:

Theorem 8.8. *The unfolding functor*

$$\mathcal{S}\mathcal{U}{:}\mathcal{S}_{\mathbf{Caus}_\mathrm{f}\subseteq\mathbf{PT}^\sharp_\mathrm{f}}\mathbf{PT}^\sharp \to \mathcal{S}_{\mathbf{Caus}_\mathrm{f}\subseteq\mathbf{Occ}^\sharp_\mathrm{f}}\mathbf{Occ}^\sharp$$

is right adjoint to the inclusion

$$\mathcal{S}_{\mathbf{Caus}_\mathrm{f}\subseteq\mathbf{Occ}^\sharp_\mathrm{f}}\mathbf{Occ}^\sharp \hookrightarrow \mathcal{S}_{\mathbf{Caus}_\mathrm{f}\subseteq\mathbf{PT}^\sharp_\mathrm{f}}\mathbf{PT}^\sharp.$$

Furthermore, the adjunction is a coreflection. □

Joint monicity

The definition of symmetry in Section 8.1 used to define the categories with symmetry here is different from that in [Win07a, Win07b] since it requires a symmetry to be a span that is a *pseudo* equivalence rather than an equivalence. As such, the maps l and r of an object with symmetry need not be jointly monic according to the definition here. In Figure 8.2,

we give a symmetry that happens to be jointly monic in $\mathbf{Occ}^\sharp$. When the morphisms l and r are considered in the category $\mathbf{PT}^\sharp$ (or in $\mathbf{Safe}^\sharp$), however, the morphisms are not jointly monic. With the restriction to jointly monic maps, a symmetry of occurrence nets would not be a symmetry of P/T nets by virtue of the fact that any occurrence net is a P/T net. Let the symmetry in Figure 8.2 be denoted (N, S, l, r). In fact, it can be seen that there is no corresponding jointly monic symmetry in the category $\mathbf{PT}^\sharp$ since the image of the net S in $N \times N$, taking the product in $\mathbf{PT}^\sharp$, has more behaviour than S. Consequently, we would fail to obtain the above coreflections if we were to restrict attention to jointly monic maps.

Unfolding general nets to P/T nets

We now show how the biadjunction between occurrence nets with symmetry and general nets with symmetry factors through the category of P/T nets with symmetry.

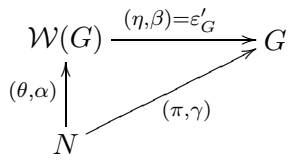
Let $G = (P_G, T_G, Pre_G, Post_G, \mathbb{M}_G)$ be a general net. The *P/T net unfolding* of a general net $\mathcal{W}(G) = (P, T, F, \mathbb{M})$ can be formed that has copies of places in P_G to account for multiplicities. Extending the definition given earlier for P/T nets, we can say that a place $p \in P_G$ is *isolated* if $M[p] = 0$ for all $M \in \mathbb{M}_G$ and $Pre[t,p] = Post[t,p] = 0$ for all transitions $t \in T_G$. The P/T net $\mathcal{W}(G)$ with a folding morphism $(\eta, \beta){:}\mathcal{W}(G) \to G$ is defined as follows:

$$\begin{array}{rcl} P & = & \{(p,i) \mid p \in P \;\&\; p \text{ not isolated in } G \;\&\; i \in \mathbb{N}\} \\ T & = & \{(A,t,B) \mid A, B \subseteq P \;\&\; t \in T \\ & & \qquad \&\; \beta \cdot A = Pre \cdot t \;\&\; \beta \cdot B = Post \cdot t\} \\ & & \quad (p,i)\; F\; (A,t,B) \iff (p,i) \in A \\ & & \quad (A,t,B)\; F\; (p,i) \iff (p,i) \in B \\ \mathbb{M} & = & \{M \mid M \subseteq P \;\&\; \beta \cdot M \in \mathbb{M}_G\}, \end{array} \qquad \begin{array}{rcl} \text{where} & & \\ \beta(p,i) & = & p \\ \eta(A,t,B) & = & t \end{array}$$

It follows from this definition that $\mathcal{W}(G)$ contains no isolated conditions and is therefore a P/T net.

We shall write ε'_G for the morphism $(\eta, \beta){:}\mathcal{W}(G) \to G$ to distinguish it from the counit of the adjunction [Mac71] arising from the occurrence net unfolding $\varepsilon_G{:}\mathcal{U}(G) \to G$.

Theorem 8.9. *Let G and $\mathcal{W}(G)$ be as defined above. For any any P/T net N and any morphism $(\pi, \gamma){:}N \to G$ in $\mathbf{Gen}^\sharp$, there is a morphism $(\theta, \alpha){:}N \to \mathcal{W}(G)$ such that the following diagram commutes:*



Proof. Let $N = (P_N, T_N, F_N, \mathbb{M}_N)$. For any $p' \in P_G$, there exists a set $A_{p'} \subseteq \mathbb{N}$ and a bijection

$$\theta_{p'}{:}A_{p'} \cong \{(p,i) \mid p \in P_N \;\&\; 0 \le i < \gamma[p,p']\}.$$

This relies on γ being countably injective. The morphism (θ, α), which will be a relation on conditions rather than a multirelation, is defined as follows:

$$\alpha(p,(p',j)) \iff j \in A_{p'} \;\&\; \exists i : \theta_{p'}(j) = (p,i)$$

$$\theta(t) = \begin{cases} * & \text{if } \pi(t) = * \\ (\alpha \cdot {}^\bullet t, \pi(t), \alpha \cdot t^\bullet) & \text{otherwise} \end{cases}$$

It is straightforward to check that this is indeed a morphism and that the diagram commutes. □

From Proposition D.2 in Appendix D, we can check that the morphism ε'_G is $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Gen}^\sharp_\mathrm{f}$ by checking that the morphism $\varepsilon_{\mathcal{W}(G)} \circ \varepsilon'_G{:}\mathcal{U}(\mathcal{W}(G)) \to G$ is $\mathbf{Caus}_\mathrm{f}$-open using the 'transition lifting' property identified in Theorem D.1.

Lemma 8.5.1. *The morphism $\varepsilon'_G{:}\mathcal{W}(G) \to G$ is $\mathbf{Caus}_\mathrm{f}$-open in $\mathbf{Gen}^\sharp_\mathrm{f}$.*

Proof. We check that the morphism $\varepsilon_{\mathcal{W}(G)} \circ \varepsilon'_G{:}\mathcal{U}(\mathcal{W}(G)) \to G$ is $\mathbf{Caus}_\mathrm{f}$-open using the 'transition lifting' property identified in Theorem D.1.

Let $G = (P, T, Pre, Post, \mathbb{M})$. Let $\varepsilon_{\mathcal{W}(G)} = (\eta, \beta)$ and let $\varepsilon'_G = (\eta', \beta')$. It is easy to verify that for any marking $M \in \mathbb{M}$ there is an initial marking M' of $\mathcal{U}(\mathcal{W}(G))$ such that $\beta' \cdot \beta \cdot M' = M$.

Let A be a subset of conditions of $\mathcal{U}(\mathcal{W}(G))$ such that co A and $\beta'{\cdot}\beta{\cdot}A = Pre{\cdot}t$ for some transition t. Let $B = \{(p, i) \mid p \in P \,\&\, 0 \leq i < Post[t, p]\}$. We have $\beta'{\cdot}B = Post{\cdot}t$, so from the definition of $\mathcal{W}(G)$ it is the case that $(\beta{\cdot}A, t, B)$ is a transition of $\mathcal{W}(G)$. From Theorem 7.12, the event $(A, (\beta{\cdot}A, t, B))$ is an event in $\mathcal{U}(\mathcal{W}(G))$, and clearly $\eta'(\eta(A, (\beta{\cdot}A, t, B))) = t$. The transition lifting property is therefore satisfied. □

As we did for the unfolding to occurrence nets, we can form a symmetry on $\mathcal{W}(G)$ by taking the kernel of the morphism ε'_G in the category $\mathbf{Gen}^\sharp_\mathrm{f}$:

$$\begin{array}{ccc} S & \xrightarrow{\;r\;} & \mathcal{W}(G) \\ {\scriptstyle l}\downarrow & \lrcorner & \downarrow{\scriptstyle \varepsilon'_G} \\ \mathcal{W}(G) & \xrightarrow[\varepsilon'_G]{} & G \end{array}$$

Note that the net S has to be a P/T net since l (or r) is a folding morphism into a P/T net, so the construction of the symmetry is slightly simpler than that for occurrence nets since there is no need to apply the functor $\mathcal{W}(S)$ to obtain an P/T net.

With the observation that ε'_G is open, in the same manner as in Proposition 8.3 we can show that $(\mathcal{W}(G), S, l, r)$ is a P/T net with symmetry, or more specifically an object in $\mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{PT}^\sharp_\mathrm{f}}\mathbf{PT}^\sharp$. With this symmetry, we may conclude in an analogous way to Theorem 8.4 that the morphism $(\theta, \alpha){:}N \to \mathcal{W}(G)$ described in Theorem 8.9 is the unique morphism *up to symmetry* to make the diagram there commute. That is, any morphism $(\theta', \alpha'){:}N \to \mathcal{W}(G)$ such that $\varepsilon'_G \circ (\theta', \alpha') = (\pi, \gamma)$ also satisfies $(\theta, \alpha) \sim (\theta', \alpha')$ with respect to the symmetry S, l, r on the P/T net unfolding $\mathcal{W}(G)$ described above.

A coreflection up to symmetry

Just as we did for occurrence nets, we can construct a coreflection between P/T nets *with symmetry* and general nets *with symmetry*.

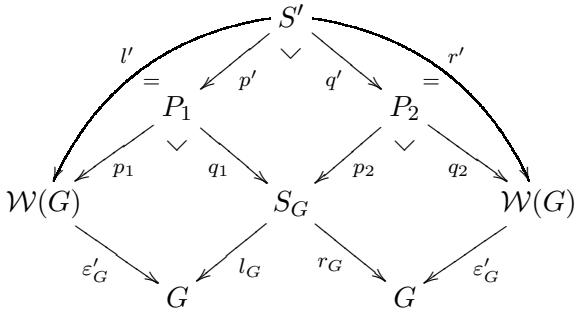
$$\mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{PT}^\sharp_\mathrm{f}}\mathbf{PT}^\sharp \underset{\mathcal{SW}}{\overset{I}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Gen}^\sharp_\mathrm{f}}\mathbf{Gen}^\sharp \,,$$

In a directly analogous way to Lemma 8.4.1, we can show that the 'inclusion'

$$I{:}\mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{PT}^\sharp_\mathrm{f}}\mathbf{PT}^\sharp \hookrightarrow \mathcal{S}_{\mathbf{Caus}_\mathrm{f} \subseteq \mathbf{Gen}^\sharp_\mathrm{f}}\mathbf{Gen}^\sharp$$

is a functor.

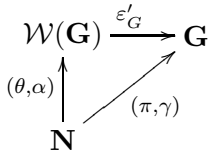
It remains for us to define the unfolding operation $\mathcal{SW}$ on objects of the category of general nets with symmetry. Let $\mathbf{G} = (G, S_G, l_G, r_G)$ be a general net with symmetry, with $\mathcal{W}(G)$ and $\varepsilon'_G{:}\mathcal{W}(G) \to G$ as defined above. The general net with symmetry $\mathbf{G}$ is unfolded to the P/T net with symmetry $\mathcal{W}(G) = (\mathcal{W}(G), S', l', r')$. Its symmetry is given by the inverse image of the symmetry in G along the open morphism ε'_G.



The pullbacks are in $\mathbf{Gen}^\sharp_{\mathrm{f}}$ and the diagram makes it clear that ε'_G is a map preserving symmetry. In a directly analogous way to Proposition 8.5, we can show that $\mathcal{U}(\mathbf{G})$ is a P/T net with symmetry. The proof is slightly simpler since we do not need to apply the operation $\mathcal{W}$ to S' to obtain a P/T net as was the case for occurrence nets.

We can now give a 'cofreeness up to symmetry' result on the P/T net unfolding $\mathcal{W}(G)$ of a general net G.

Theorem 8.10. *Let $\mathbf{G} = (G, S_G, l_G, r_G)$ be a general net with symmetry and $\mathbf{N} = (N, S_N, l_N, r_N)$ be a P/T net with symmetry. For any $(\pi, \gamma){:}\mathbf{N} \to \mathbf{G}$ in $\mathcal{S}_{\mathbf{Caus}_{\mathrm{f}} \subseteq \mathbf{Gen}^\sharp_{\mathrm{f}}}\mathbf{Gen}^\sharp$, there is a morphism $(\theta, \alpha){:}\mathbf{N} \to \mathcal{W}(\mathbf{G})$ in $\mathcal{S}_{\mathbf{Caus}_{\mathrm{f}} \subseteq \mathbf{Gen}^\sharp_{\mathrm{f}}}\mathbf{PT}^\sharp$ such that the following diagram commutes:*



Furthermore, (θ, α) is unique up to symmetry: any $(\theta', \alpha'){:}\mathbf{N} \to \mathcal{W}(\mathbf{G})$ such that $\varepsilon'_G \circ (\theta', \alpha') \sim (\pi, \gamma)$ satisfies $(\theta, \alpha) \sim (\theta', \alpha')$.

Proof. Similar to the proof of Theorem 8.6. □

Technically, we have a biadjunction from the category of P/T nets with symmetry $\mathcal{S}_{\mathbf{Caus}_{\mathrm{f}} \subseteq \mathbf{PT}^\sharp_{\mathrm{f}}}\mathbf{PT}^\sharp$ to the category of general nets with symmetry $\mathcal{S}_{\mathbf{Caus}_{\mathrm{f}} \subseteq \mathbf{Gen}^\sharp_{\mathrm{f}}}\mathbf{Gen}^\sharp$ with I left biadjoint to $\mathcal{W}$ (which extends to a pseudo functor). The counit of the biadjunction is ε. The inclusion I is easily seen to be full (and faithful). In this sense, we have established a coreflection from $\mathcal{S}\mathbf{PT}^\sharp$ to $\mathcal{S}\mathbf{Gen}^\sharp$ up to symmetry.

8.6 Multiply-marked nets and event structures

In defining symmetry on Petri nets, we have seen that it is necessary to introduce nets with multiple initial markings. We begin to place these nets in the context of other models for concurrency in this section. In particular, we shall establish a coreflection between a category of multiply-marked occurrence nets and a category of event structures. This shall

arise from the existing coreflection between event structures and singly-marked occurrence nets. There are, however, difficulties in showing that this yields a coreflection between event structures with symmetry and occurrence nets with symmetry, as shall be described on page 189, due to the functor from event structures to occurrence nets nor preserving pullbacks of open maps. Though this is disappointing, the results do connect nets with multiple initial markings to event structures, and hence to other models for concurrency.

Categories of event structures

Event structures [NPW81, Win86] represent a computational process as a set of event occurrences, recording how these event occurrences *causally depend* on each other. An event structure also records how the occurrence of an event indicates that the process has taken a particular branch. For the variant of event structure that we shall consider, called *prime* event structures, this amounts to recording how event occurrences *conflict* with each other.

Definition 8.6.1. *A (prime) event structure is a 3-tuple*

$$ES = (E, \leq \#),$$

where

- E *is the set of* events *(more precisely, event occurrences),*
- $\leq \subseteq E \times E$ *is the partial order of* causal dependency*, and*
- $\# \subseteq E \times E$ *is the irreflexive, symmetric binary relation of* conflict*.*

An event structure must satisfy the following axioms:

1. *each event causally depends on only finitely many other events, i.e.* $\{e' \mid e' \leq e\}$ *is finite for all* $e \in E$*, and*
2. *if* $e_1 \# e_2$ *and* $e_1 \leq e_1'$ *then* $e_1' \# e_2$*.*

The intuition is that if we have $e \leq e'$ for two events e and e', then the event e must have occurred prior to any occurrence of e'. If we have $e \# e'$, then the occurrence of e precludes the occurrence of event e' at any later stage. An event structure is said to be *elementary* if the conflict relation is empty. The first axiom ensures that an event structure only consists of event occurrences that can eventually take place, not relying on an infinite number of prior event occurrences. The second axiom asserts that if the occurrence of an event e_2 precludes the occurrence of an event e_1 upon which the event e_1' causally depends then the event e_2 precludes the occurrence of the event e_1'. We say that two events e_1 and e_2 are *concurrent*, written e_1 co e_2, if there is no causal dependency between them and they do not conflict, *i.e.* e_1 co $e_2 \iff \neg(e_1 \# e_2$ or $e_1 \leq e_2$ or $e_2 \leq e_1)$. We write $e < e'$ if $e \leq e'$ but $e \neq e'$.

The computational states of an event structure, called its *configurations*, are represented by the sets of events that have occurred. Every configuration must be consistent with the relations of conflict and causal dependency. Formally, $x \subseteq E$ is a configuration of an event structure $(E, \leq, \#)$ if it satisfies the following two properties:

- Conflict-freedom: $\forall e, e' \in x : \ \neg(e \# e')$
- Downwards-closure: $\forall e, e' \in E : \ e \leq e' \ \& \ e' \in x \implies e \in x$.

We write $\mathcal{D}(ES)$ for the set of configurations of ES and write $\mathcal{D}^0(ES)$ for the set of finite configurations of ES. We write $\lceil e \rceil$ for $\{e' \mid e' \leq e\}$, the least configuration containing the event e.

We now introduce morphisms of event structures. A morphism $\eta{:}ES \to ES'$ is a function from the events of ES to the events of ES' that expresses how the behaviour of ES embeds into ES' in the sense that the function preserves the configurations of the event structure and also preserves the atomicity of events.

Definition 8.6.2. *Let $ES = (E, \leq, \#)$ and $ES' = (E', \leq', \#')$ be event structures. A morphism $\eta{:}ES \to ES'$ consists of a partial function $\eta{:}E \to_* E'$ such that for all $x \in \mathcal{D}(ES)$:*

$$\begin{array}{ll} & \eta x \in \mathcal{D}(ES) \\ \& & \forall e, e' \in x : \ (\eta(e), \eta(e') \text{ defined } \ \& \ \eta(e) = \eta(e')) \implies e = e' \end{array}$$

A morphism is said to be synchronous *if it is a total function on events.*

In fact, it is only necessary to consider finite configurations $x \in \mathcal{D}^0(ES)$ in the requirement on morphisms above. It is easy to see that if $x \xrightarrow{e} x'$ then $\eta x \xrightarrow{\eta(e)} \eta x'$.

We obtain a category **ES** of (prime) event structures with event structures as objects and morphisms as described above. The identity morphism on an event structure is the identity function on its underlying set of events, and composition of morphisms occurs as composition of functions. We also obtain a category $\mathbf{ES}_{\mathrm{s}}$ of event structures with synchronous morphisms between then. We write **Elem** for the category of elementary event structures (event structures with no conflict) and $\mathbf{Elem}_{\mathrm{s}}$ for the category of elementary event structures with synchronous morphisms between them. Elementary event structures can be thought of as paths of event structures and nets, and these categories will later be used to define open maps of event structures and nets.

Before moving on to consider their relationship with Petri nets, we note that the categories **ES** and $\mathbf{ES}_{\mathrm{s}}$ have coproducts obtained by forming the disjoint union of their events using injections in_i, placing two events in conflict if they occur in different components of the coproduct.

Proposition 8.11. *Let $(ES)_{i \in I}$ be a family of event structures indexed by I, where $ES_i = (E_i, \leq_i, \#_i)$. A coproduct of these event structures in the category* **ES** *and also in the category* $\mathbf{ES}_{\mathrm{s}}$ *is the event structure $\sum_{i \in I} ES_i = (E, \leq, \#)$ with events $E = \{\mathsf{in}_i e \mid i \in I \ \& \ e \in E_i\}$ and relations*

$$\begin{array}{rcl} \mathsf{in}_i e \ \leq \ \mathsf{in}_j e' & \iff & i = j \ \& \ e \leq_i e' \\ \mathsf{in}_i e \ \# \ \mathsf{in}_j e' & \iff & i \neq j \text{ or } (i = j \ \& \ e \#_i e') \end{array}$$

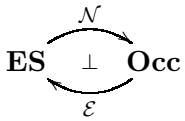
For each $j \in I$, the function $\mathsf{in}_j{:}ES_j \to \sum_{i \in I} ES_i$ defined as $\mathsf{in}_i(e) = \mathsf{in}_i e$ is the associated injection into the coproduct.

Proof. The proof of the existence of binary coproducts in Theorem 2.2.9 of [Win86] extends straightforwardly. □

The category of event structures also has pullbacks. Their construction is given in Appendix C of [Win07a], and is hard, being most easily seen in the category of stable families, so we shall not present it here.

Singly-marked occurrence nets: a coreflection

There is a coreflection that embeds the category of event structures into the category of singly-marked occurrence nets.



The functor $\mathcal{N}$ constructs an occurrence net from an event structure, saturating the events of the event structure with as many conditions as possible that are consistent with the the relations of causal dependency and conflict in the original event structure. The functor $\mathcal{E}$ strips away the conditions from the occurrence net to reveal the underlying causal dependency and conflict relations on events. Since we shall use the constructions in forming a coreflection between event structures and multiply-marked occurrence nets, we now give the constructions $\mathcal{E}$ and $\mathcal{N}$ concretely. The operation $\mathcal{N}$ was first defined in [NPW81] and was shown to yield a coreflection in [Win86]. A coreflection can also be obtained via the category of asynchronous transition systems as in [WN95].

The functor $\mathcal{E}$:**Occ** $\to$ **ES**

The functor $\mathcal{E}$ takes an occurrence net to an event structure by interpreting causal dependency on the events of the occurrence net as the transitive closure of the flow relation and obtaining the conflict relation as in Definition 7.1.4.

Definition 8.6.3. *Let $O = (B, E, F, \mathbb{M})$ be an occurrence net. The event structure $\mathcal{E}(O) = (E, \leq, \#)$ has the same events as O, inherits conflict from O as in Definition 7.1.4 and has $e \leq e'$ iff they are related through the transitive closure of the flow relation, $e\ F^*\ e'$.*

It is an immediate consequence of the definitions that $\mathcal{E}(O)$ is an event structure for any occurrence net O. Recalling that a morphism between occurrence nets O and O' is a pair (η, β) of which $\eta{:}E \to_* E'$ is a partial function on their underlying sets of events, we obtain the operation of the functor on morphisms.

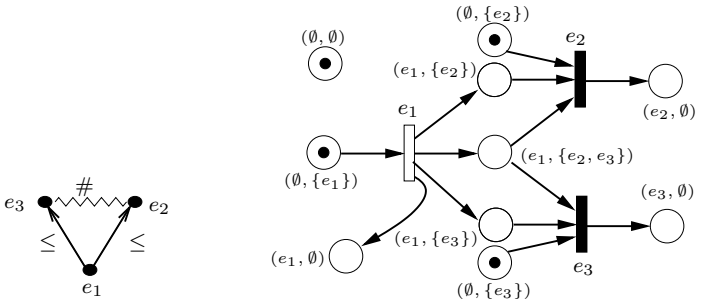
Proposition 8.12. *Let $(\eta, \beta){:}O \to O'$ be a morphism in* **Occ**. *Then $\eta{:}\mathcal{E}(O) \to \mathcal{E}(O')$ is a morphism in* **ES**.

Proof. Lemma 3.4.2 in [Win86] □

It is straightforward to see that defining $\mathcal{E}(\eta, \beta) = \eta$ yields an operation that preserves identities and composition, so $\mathcal{E}$:**Occ** $\to$ **ES** is a functor. This is easily seen to restrict to categories with synchronous morphisms, so also $\mathcal{E}$:$\mathbf{Occ}_{\mathrm{s}} \to \mathbf{ES}_{\mathrm{s}}$.

The functor $\mathcal{N}$:**ES** $\to$ **Occ**

We now consider how to form an occurrence net from an event structure. As stated earlier, the essential idea is to form an occurrence net with the same events as the original event structure, adding as many conditions as possible that are consistent with the causal dependency and conflict relations of the original event structure. We extend the notation $e \asymp e'$ as introduced in Proposition 7.4 to event structures to mean that either $e = e'$ or $e \# e'$.



(a) Event structure ES

(b) Occurrence net $\mathcal{N}(ES)$

Figure 8.3: An event structure with its associated occurrence net

Definition 8.6.4. *Let $ES = (E, \leq, \#)$ be a event structure. The net $\mathcal{N}(ES)$ is defined as $(B, E, F, \{M\})$, where*

$$\begin{array}{rcl} M & = & \{(\emptyset, A) \mid A \subseteq E \;\&\; (\forall a, a' \in A : a \asymp a')\} \\ B & = & M \cup \{(e, A) \mid \; e \in E \;\&\; A \subseteq E \;\&\; (\forall a, a' \in A : a \asymp a') \\ & & \qquad \&\; (\forall a \in A : e < a)\} \end{array}$$
$$e \; F \; (x, A) \iff x = e \qquad (x, A) \; F \; e \iff e \in A$$

The net is formed with conditions (e, A) indicating that all the events in A are in conflict with each other and all causally depend on e. There are conditions $(\emptyset, A)$ to indicate just that the events in A are in conflict with each other but might not causally depend on some other event. The net formed is *condition-extensional* in the sense that any two conditions with precisely the same beginning- and end-events are identified. The occurrence net of an example event structure is presented in Figure 8.3.

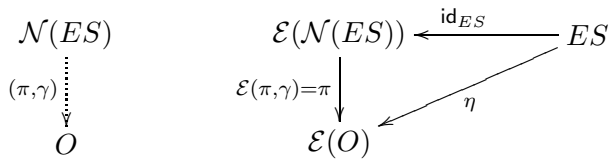
Proposition 8.13. *The net $\mathcal{N}(ES)$ is an occurrence net. Furthermore, for any event structure ES we have $\mathcal{E}(\mathcal{N}(ES)) = ES$.*

Proof. The first part of Theorem 3.4.11 in [Win86]. □

Freeness and morphisms

In order to obtain a coreflection, this time it is easier to show a *freeness* result. This is sufficient, also, to show how the operation $\mathcal{N}$ extends to a functor [Mac71].

Proposition 8.14. *For any event structure in* **ES**, *the net $\mathcal{N}(ES)$ and morphism id_{ES}:$ES \to \mathcal{E}(\mathcal{N}(ES))$ is free over ES with respect to $\mathcal{E}$. That is, for any occurrence net O and morphism η:$ES \to \mathcal{E}(O)$ in* **ES** *there is a unique morphism (π, γ):$\mathcal{N}(ES) \to O$ in* **Occ** *such that the triangle in the following diagram commutes:*



Proof. The second part of Theorem 3.4.11 in [Win86]. □

Hence the functor $\mathcal{N}$:**ES** $\to$ **Occ** is left-adjoint to the functor $\mathcal{E}$:**Occ** $\to$ **ES**. Since the unit of the adjunction is a natural isomorphism (in this case, the identity), the adjunction is a coreflection.

Proposition 8.14 also applies using the categories $\mathbf{ES}_\mathrm{s}$ and $\mathbf{Occ}_\mathrm{s}$ in place of **ES** and **Occ**, so a coreflection is obtained for the categories with synchronous morphisms. We shall to use the same symbols to represent the functors $\mathcal{N}$:$\mathbf{ES}_\mathrm{s} \to \mathbf{Occ}_\mathrm{s}$ and $\mathcal{E}$:$\mathbf{Occ}_\mathrm{s} \to \mathbf{ES}_\mathrm{s}$.

Multiply-marked occurrence nets: a coreflection

To obtain an adjunction when we allow multiple initial markings, it is necessary to restrict attention to categories of occurrence nets and event structures with synchronous morphisms (morphisms that are total on events). We shall briefly mention how partiality could be recovered at the end of this section.

We first define how a multiply-marked occurrence net forms an event structure, giving rise to a functor $\mathcal{E}_\mathrm{s}^\sharp$:$\mathbf{Occ}_\mathrm{s}^\sharp \to \mathbf{ES}_\mathrm{s}$.

Let O be any occurrence net. The events of $\mathcal{E}_\mathrm{s}^\sharp(O)$ are simply the events of O; causal dependency of events is obtained from the flow relation F; and the conflict relation on events is obtained from the conflict relation of O. Recall that the conflict relation on the multiply-marked occurrence net places two events in conflict if they are given rise to by different initial markings.

Definition 8.6.5. *Let $O = (B, E, F, \mathbb{M})$ be an occurrence net. The event structure $\mathcal{E}_\mathrm{s}^\sharp(N)$ is $(E, \leq, \#)$ where $e \leq e'$ iff $e\ F^*\ e'$ and $\#$ is the conflict relation on the occurrence net O in Definition 7.1.4.*

The operation $\mathcal{E}_\mathrm{s}^\sharp$ extends to a functor $\mathcal{E}_\mathrm{s}^\sharp$:$\mathbf{Occ}_\mathrm{s}^\sharp \to \mathbf{ES}_\mathrm{s}$ by taking a morphism of occurrence nets (η, β):$O \to O'$ to

$$\mathcal{E}_\mathrm{s}^\sharp(\eta, \beta) = \eta.$$

It can be shown straightforwardly that η:$\mathcal{E}_\mathrm{s}^\sharp(O) \to \mathcal{E}_\mathrm{s}^\sharp(O')$ is a morphism of event structures and that $\mathcal{E}_\mathrm{s}^\sharp$ satisfies the requirements for being a functor.

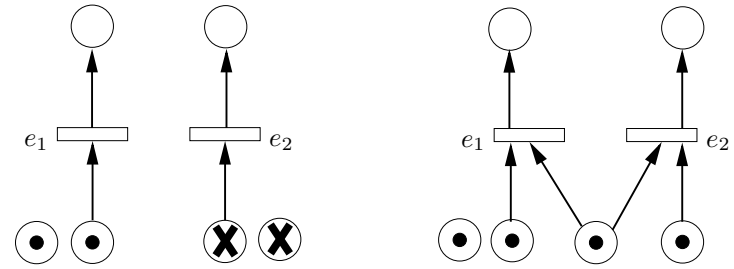
The specification of a functor from event structures to occurrence nets with multiple initial markings is a little trickier. The generalization of occurrence nets as introduced on page 136 to allow them to possess more than one initial marking gives rise to two distinct ways in which their events may be in conflict.

'Early' conflict Any event in an occurrence net can occur in a marking reachable from precisely one initial marking. The events may conflict if they arise from distinct initial markings.

'Late' conflict As with singly-marked occurrence nets, two events e_1 and e_2 might be in conflict because they either share a precondition or there might exist events e_1' and e_2' that share a common precondition for which e_1 causally depends on e_1' and e_2 causally depends on e_2'.

Quite clearly, all conflict in singly-marked occurrence nets is late conflict. The old functor $\mathcal{N}$:$\mathbf{ES}_\mathrm{s} \to \mathbf{Occ}_\mathrm{s}$ from event structures to singly-marked occurrence nets therefore uses late conflict to represent conflict in the event structure.

In the category $\mathbf{Occ}_\mathrm{s}^\sharp$, early conflict embeds into late conflict. Consider, for instance, the nets in Figure 8.4. There is a morphism preserving events from N to N'. Late conflict, however, does not embed into early conflict; there is no morphism preserving events from N' to N.



(a) Net N with early conflict

(b) Net N' with late conflict

Figure 8.4: Nets with early conflict and late conflict

The old functor $\mathcal{N}$ can be seen, as a result, not to be left adjoint to the new functor $\mathcal{E}^\sharp_{\rm s}$. If it were, there would have to be a morphism preserving events of the form $\mathcal{N}(\mathcal{E}^\sharp_{\rm s}(N)) \rightarrow N$. It is easy to see that the event structures $\mathcal{E}^\sharp_{\rm s}(N)$ and $\mathcal{E}^\sharp_{\rm s}(N')$ are equal, comprising two events e_1 and e_2 that are in conflict. The net $\mathcal{N}(\mathcal{E}^\sharp_{\rm s}(N))$ is isomorphic to N', so the required morphism does not exist. The problem is that, in constructing the net $\mathcal{N}(\mathcal{E}^\sharp_{\rm s}(N))$, early conflict is replaced by late conflict. We must therefore define a new functor $\mathcal{N}^\sharp_{\rm s}{:}\mathbf{ES}_{\rm s} \rightarrow \mathbf{Occ}^\sharp_{\rm s}$ that ensures that if two events of a net N are in early conflict then they remain in early conflict in the net $\mathcal{N}^\sharp_{\rm s}(\mathcal{E}^\sharp_{\rm s}(N))$.

It is best to define the new functor $\mathcal{N}^\sharp_{\rm s}{:}\mathbf{ES}_{\rm s} \rightarrow \mathbf{Occ}^\sharp_{\rm s}$ by considering *families* of event structures. Families were defined in Section 7.2. In particular, we shall establish the following chain of coreflections:

$$\mathbf{ES}_{\rm s} \underset{\sum}{\overset{\mathcal{F}}{\rightleftarrows}} \mathcal{F}am(\mathbf{ES}_{\rm s}) \underset{\mathcal{F}am(\mathcal{E}_{\rm s})}{\overset{\mathcal{F}am(\mathcal{N}_{\rm s})}{\rightleftarrows}} \mathcal{F}am(\mathbf{Occ}_{\rm s}) \simeq \underset{\text{decomp}}{\overset{\text{join}}{\rightleftarrows}} \mathbf{Occ}^\sharp_{\rm s}$$

Recall that in Section 7.2, it was shown that the category of multiply-marked occurrence nets is equivalent to the category of families of singly-marked occurrence nets. This accounts for the rightmost adjunction.

The pictorially central adjunction is the lifting of the previous coreflection between event structures and singly-marked occurrence nets to their categories of families. The functors $\mathcal{F}am(\mathcal{N}_{\rm s})$ and $\mathcal{F}am(\mathcal{E}_{\rm s})$ are obtained from Proposition C.1 in Appendix C, where it is shown how functors lift to families. The coreflection then follows from Proposition C.2.

We now turn to the leftmost adjunction in the picture, which is where the key result of this section, Theorem 8.15, is required.

The right adjoint $\sum{:}\mathcal{F}am(\mathbf{ES}_{\rm s}) \rightarrow \mathbf{ES}_{\rm s}$ takes a family of event structures $(ES_i)_{i\in I}$ to their coproduct $\sum_{i\in I} ES_i$ defined in Proposition 8.11. Let $\eta{:}(ES_i)_{i\in I} \rightarrow (ES'_j)_{j\in J}$ in $\mathcal{F}am(\mathbf{ES}_{\rm s})$ be a morphism in $\mathcal{F}am(\mathbf{ES}_{\rm s})$, where

$$\eta = (\hat{\eta}{:}I \rightarrow J, (\eta_i{:}ES_i \rightarrow ES'_{\hat{\eta}(i)})_{i\in I}).$$

As a functor, $\sum$ takes η to the morphism $\sum\eta$, which is the unique morphism, known to exist since $\sum_{i\in I} ES_i$ is a coproduct, such that the following diagram commutes for all

$i \in I$:

$$\begin{array}{ccc} ES_i & \xrightarrow{\mathsf{in}_i} & \sum_{i\in I} ES_i \\ {\scriptstyle \eta_i}\downarrow & & \downarrow{\scriptstyle \sum \eta} \\ ES'_{\hat{\eta}(i)} & \xrightarrow[\mathsf{in}'_{\hat{\eta}(i)}]{} & \sum_{j\in J} ES'_j \end{array}$$

The morphisms $\mathsf{in}_i{:}ES_i \to \sum_{i\in I} ES_i$ and $\mathsf{in}'_{\hat{\eta}(i)}{:}ES'_{\hat{\eta}(i)} \to \sum_{j\in J} ES'_j$ in the diagram above are the coproduct injections.

For the reasons discussed above, the left adjoint $\mathcal{F}$ will have to ensure that early conflict is preserved in the sense that if two events are in early conflict in the occurrence net O then they are in early conflict in the net obtained by successively applying the right and then left adjoints drawn above. This means that the functor $\mathcal{F}$ must ensure that if two events in $\sum_{i\in I} ES_i$ come from different components of the family $(ES_i)_{i\in I}$ then they are in different components of the family $\mathcal{F}\left(\sum_{i\in I} ES_i\right)$. To define such a functor, we must use the *compatibility* relation on events. Let $ES = (E, \leq, \#)$ be an event structure. The compatibility relation $\frown \subseteq E \times E$ is defined as:

$$e \frown e' \overset{\triangle}{\iff} \neg(e \# e')$$

Two events are compatible if there exists a configuration containing them both. The compatibility relation is symmetric and reflexive, so its transitive closure $\frown^+$ is an equivalence relation. The event structure ES can be partitioned into a family $(ES_c)_{c\in C}$ of non-empty $\frown^+$-equivalence classes. Each $\frown^+$-equivalence class ES_c is an event structure with conflict and causal dependency inherited from ES. Any event of ES_c is in conflict with every event of ES_d in the event structure ES if $c \neq d$.

Lemma 8.6.1. *Let ES be an event structure and let the $\frown^+$-equivalence classes contained in ES form the family $(ES_c)_{c\in C}$ for some indexing set C. Each equivalence class ES_c is an event structure and $ES \cong \sum_{c\in C} ES_c$ through an isomorphism natural in ES, taking the coproduct in the category $\mathbf{ES}_{\mathrm{s}}$ defined in Proposition 8.11.*

Proof. It is easy to see that ES_c is an event structure for every $c \in C$. Denote the isomorphism between ES and $\sum_{c\in C} ES_c$ by

$$\varphi_{ES}{:}\quad ES \overset{\cong}{\longleftrightarrow} \sum_{c\in C} ES_c \quad {:}\varphi_{ES}^{-1}.$$

For any event e in ES with $\frown^+$-equivalence class c, the morphisms φ_{ES} and φ_{ES}^{-1} are defined as:

$$\varphi_{ES}(e) = \mathsf{in}_c e \qquad \varphi_{ES}^{-1}(\mathsf{in}_c e) = e.$$

It is clear that this forms an isomorphism. Naturality of the isomorphism follows immediately from the definition above of $\sum \eta$ for any morphism $\eta{:}ES \to ES'$. □

Two events of the event structure $\sum_{c\in C} ES_c$ cannot arise from different components of the family $(ES_c)_{c\in C}$ if they are $\frown^+$-related.

Given an event structure ES, we are now able to define the action of the functor $\mathcal{F}$ on objects as

$$\mathcal{F}(ES) = (ES_c)_{c\in C},$$

where $(ES_c)_{c\in C}$ is the family of $\frown^+$-equivalence classes of ES.

An important observation when considering how $\mathcal{F}$ extends to morphisms is that synchronous morphisms of event structures preserve the relation $\frown^+$.

Lemma 8.6.2. *Let $ES_1 = (E_1, \leq_1, \#_1)$ and $ES_2 = (E_2, \leq_2, \#_2)$ be event structures with compatibility relations $\frown_1$ and $\frown_2$ respectively. Let $\eta{:}ES_1 \to ES_2$ be a morphism in $\mathbf{ES}_\mathrm{s}$. For any $e, e' \in E_1$, if $e \frown_1^+ e'$ then $\eta(e) \frown_2^+ \eta(e')$.*

Proof. We show that if $e \frown_1 e'$ then $\eta(e) \frown_2 \eta(e')$. The result then follows by a straightforward induction. If $e \frown_1 e'$, there must exist a configuration x of ES_1 such that $e, e' \in x$. The configuration ηx is a configuration of ES_2 since η is a morphism (Definition 8.6.2). Hence there exists a configuration ηx of ES_2 such that $\eta(e), \eta(e') \in \eta x$, so we must have $\eta(e) \frown_2 \eta(e')$. □

The previous lemma relies on the fact that morphisms are synchronous, *i.e.* total on events. It need not be the case that if $e_1 \frown_1^+ e_2$ and $\eta(e_1)$ and $\eta(e_2)$ are defined for some non-synchronous morphism η then $\eta(e_1) \frown_2^+ \eta(e_2)$.

Let the $\frown_1^+$-equivalence classes of ES_1 be the family $(ES_c)_{c \in C}$ and the $\frown_2^+$-equivalence classes of ES_2 be the family $(ES_d)_{d \in D}$, and suppose that there is a synchronous morphism $\eta{:}ES_1 \to ES_2$ in $\mathbf{ES}_\mathrm{s}$. As a consequence of the previous lemma, for all $c \in C$ and $d \in D$:

$$\begin{array}{rl} & \exists e \in E_c : \ E_d = \{e_2 \mid \eta(e) \frown_2^+ e_2\} \\ \iff & \forall e \in E_c : \ E_d = \{e_2 \mid \eta(e) \frown_2^+ e_2\}. \end{array}$$

We may therefore define a function $\hat{\eta}{:}C \to D$ as $\hat{\eta}(c) = d$ iff $\exists e \in E_c : \ E_d = \{e_2 \mid \eta(e) \frown_2^+ e_2\}$, which informs that the event structure ES_c within ES_1 is taken by η to $ES_{\hat{\eta}(c)}$ in ES_2. The morphism $\eta{:}ES_1 \to ES_2$ therefore restricts to a morphism $\eta_c{:}ES_c \to ES_{\hat{\eta}(c)}$. We therefore obtain the operation of the functor $\mathcal{F}$ on morphisms as

$$\mathcal{F}(\eta) = (\hat{\eta}, (\eta_c)_{c \in C}).$$

It is straightforward to show that $\mathcal{F}$ preserves identities and composition.

Using these definitions, it is possible to demonstrate the coreflection between event structures and families of event structures, the key coreflection in relating multiply-marked occurrence nets and event structures.

Theorem 8.15. *The functors $\mathcal{F}$ and $\sum$ form a coreflection with $\mathcal{F} \dashv \sum$. That is, there is an isomorphism of hom-sets*

$$\varphi_{ES,(ES'_j)_{j \in J}}{:}\mathbf{ES}_\mathrm{s}(ES, \sum_{j \in J} ES'_j) \cong \mathcal{F}am(\mathbf{ES}_\mathrm{s})(\mathcal{F}(ES), (ES'_j)_{j \in J}),$$

natural in ES and $(ES'_j)_{j \in J}$ and, furthermore, the functor $\mathcal{F}$ is full and faithful.

Proof. Suppose that the $\frown^+$-decomposition of the event structure ES is the family $(ES_c)_{c \in C}$. We have the following chain of isomorphisms, in which $\prod$ and $\coprod$ represent the indexed product and coproduct defined in Appendix C.

$$\begin{array}{rcll} \mathbf{ES}_\mathrm{s}(ES, \sum_{j \in J} ES'_j) & \cong & \mathbf{ES}_\mathrm{s}(\sum_{c \in C} ES_c, \sum_{j \in J} ES'_j) & (1) \\ & \cong & \prod_{c \in C} \mathbf{ES}_\mathrm{s}(ES_c, \sum_{j \in J} ES'_j) & (2) \\ & \cong & \prod_{c \in C} \coprod_{j \in J} \mathbf{ES}_\mathrm{s}(ES_c, ES_j) & (3) \\ & \cong & \mathcal{F}am(\mathbf{ES}_\mathrm{s})((ES_c)_{c \in C}, (ES'_j) j \in J) & (4) \\ & = & \mathcal{F}am(\mathbf{ES}_\mathrm{s})(\mathcal{F}(ES), (ES'_j) j \in J) & (5) \end{array}$$

Isomorphism (1) arises from the fact that $ES \cong \sum_{c \in C}(ES_c)$ shown in Lemma 8.6.1. Isomorphism (2) is from the universal characterization of the coproduct of event structures.

Isomorphism (4) is from Proposition C.3 in Appendix C, and (5) is from the definition of $\mathcal{F}(ES)$. Naturality of all of these is straightforward.

All that remains is to give isomorphism (3) and to show that it is natural. Let the isomorphism be denoted

$$\psi_{ES,(ES_j)_{j\in J}} : \prod_{c\in C} \mathbf{ES}_{\mathrm{s}}(ES_c, \sum_{j\in J} ES'_j) \cong \prod_{c\in C} \coprod_{j\in J} \mathbf{ES}_{\mathrm{s}}(ES_c, ES_j).$$

Suppose that for every $c \in C$ there is a morphism $\eta_c : ES_c \to \sum_{j\in J} ES'_j$ in $\mathbf{ES}_{\mathrm{s}}$. There exists $j \in J$ such that the image of ES_c under η_c is in ES'_j since ES_c is non-empty due to $(ES_c)_{c\in C}$ being the set of (non-empty) $\subset^+$-equivalence classes of ES. Furthermore, by Lemma 8.6.2, j is unique. Let $\hat{c}$ denote this j for each c. It is easy to see that sending the indexed family $(\eta_c)_{c\in C}$ to $(\mathsf{in}_{\hat{c}}\eta_c)_{c\in C}$ yields the required isomorphism.

For naturality in ES, suppose that there is a morphism $\eta : ES \to ES'$ in $\mathbf{ES}_{\mathrm{s}}$. Let the $\subset^+$-equivalence classes of ES and ES' be $(ES_c)_{c\in C}$ and $(ES'_d)_{d\in D}$ respectively. According to the discussion immediately preceding this theorem, we obtain a function $\hat{\eta} : C \to D$ and morphisms $\eta_c : ES_c \to ES'_{\hat{\eta}(c)}$ for every $c \in C$. It is straightforward to check that the following diagram commutes, thereby demonstrating naturality in ES:

$$\begin{array}{ccc}
\prod_{c\in C} \mathbf{ES}_{\mathrm{s}}(ES_c, \sum_{j\in J} ES''_j) & \xrightarrow[\cong]{\psi_{ES,(ES_j)_{j\in J}}} & \prod_{c\in C} \coprod_{j\in J} \mathbf{ES}_{\mathrm{s}}(ES_c, ES''_j) \\
\uparrow{\scriptstyle \prod_{c\in C}(-\circ\eta_c)} & & \uparrow{\scriptstyle \prod_{c\in C} \coprod_{d\in D}(-\circ\eta_c)} \\
\prod_{d\in D} \mathbf{ES}_{\mathrm{s}}(ES'_d, \sum_{j\in J} ES''_j) & \xrightarrow[\psi_{ES',(ES_j)_{j\in J}}]{\cong} & \prod_{d\in D} \coprod_{j\in J} \mathbf{ES}_{\mathrm{s}}(ES'_d, ES''_j)
\end{array}$$

For naturality in the family $(ES'_j)_{j\in J}$, suppose that there is a morphism $\eta : (ES'_j)_{j\in J} \to (ES''_i)_{i\in I}$. The morphism $\sum \eta : \sum_{j\in J} ES'_j \to \sum_{i\in I} ES''_i$ is the result of applying the functor $\sum$ to η. It is easy to see that the following diagram commutes, thereby demonstrating naturality in $(ES'_j)_{j\in J}$.

$$\begin{array}{ccc}
\prod_{c\in C} \mathbf{ES}_{\mathrm{s}}(ES_c, \sum_{j\in J} ES'_j) & \xrightarrow[\cong]{\psi_{ES,(ES_j)_{j\in J}}} & \prod_{c\in C} \coprod_{j\in J} \mathbf{ES}_{\mathrm{s}}(ES_c, ES'_j) \\
\downarrow{\scriptstyle \prod_{c\in C}(\sum \eta\circ-)} & & \downarrow{\scriptstyle \prod_{c\in C} \coprod_{j\in J}(\eta_j\circ-)} \\
\prod_{c\in C} \mathbf{ES}_{\mathrm{s}}(ES_c, \sum_{i\in I} ES''_i) & \xrightarrow[\psi_{ES,(ES_i)_{i\in I}}]{\cong} & \prod_{c\in C} \coprod_{i\in I} \mathbf{ES}_{\mathrm{s}}(ES_c, ES''_i)
\end{array}$$

Finally, the functor $\mathcal{F}$ is easily seen to be full and faithful, thus completing the proof of the coreflection. □

The account so far has been restricted to categories of nets and event structures with synchronous morphisms. To lift this restriction and still obtain an adjunction, event structures may presumably be extended to record information on early conflict, so that we obtain a category of 'event structures with early conflict' that is equivalent to to the category of families of event structures in the same way that the category of multiply-marked occurrence nets is equivalent to the category of families of singly-marked occurrence nets. In a sense, this would cut off the left part of the chain of coreflections drawn at the start of this section, which is where we had to require the morphisms to be synchronous.

Symmetry

We would now wish to show that the coreflection above between event structures and occurrence nets with synchronous morphisms extends to the categories with symmetry. Unfortunately, for reasons that we shall now explain, this is problematic.

To attach symmetry to these categories, we would have to choose a path category such that open maps are preserved by the adjunctions. The appropriate paths of event structures in this situation seem to be elementary event structures and the appropriate paths of occurrence nets are the images under the functor $\mathcal{N}_{\mathrm{s}}^{\sharp}$:$\mathbf{ES}_{\mathrm{s}} \to \mathbf{Occ}_{\mathrm{s}}^{\sharp}$ of elementary event structures. Open maps of safe nets, and hence occurrence nets, with elementary event structures as paths were studied in [NW96].

A general result about open maps presented in [JNW95] shows that the functors $\mathcal{E}_{\mathrm{s}}^{\sharp}$ and $\mathcal{N}_{\mathrm{s}}^{\sharp}$ preserve open maps as defined here. The adjunction between the categories with symmetry is, however, stymied by the fact that the functor $\mathcal{N}_{\mathrm{s}}^{\sharp}$ does not preserve pullbacks of $\mathbf{Elem}_{\mathrm{s}}$-open maps. This can be seen by considering the event structure ES, an event structure with two events e_1 and e_2 that are in conflict, and the event structure ES', an event structure with one event, e. The morphism η:$ES \to ES'$ defined as

$$\eta(e_1) = \eta(e_2) = e$$

is $\mathbf{Elem}_{\mathrm{s}}$-open. It can be shown that the pullback Q of the morphism η taken against itself is equal to the event structure with events

$$\{(e_1, e_1), (e_1, e_2), (e_2, e_1), (e_2, e_2)\},$$

all of which are in conflict with each other. This is not preserved as a pullback by the functor $\mathcal{N}_{\mathrm{s}}^{\sharp}$.

Chapter 9

Conclusion

Unifying this thesis have been three forms of Petri net semantics: techniques for defining the net semantics of programming languages, a semantics for concurrent separation logic based on nets, and the semantics of Petri nets themselves and how symmetry plays a vital role there.

Hopefully the first two of these components demonstrate how Petri nets are well-suited to the systematic and comprehensive study of programming language semantics. We have seen how a structural Petri-net semantics can be given to terms, resulting in a companion to Plotkin's structural operational semantics [Plo81] which is based on transition systems. The semantics identifies the particular structure of nets required in general for giving a net semantics, such as splitting the net into control and data parts and that the nets must have particular properties such as well-termination in order to have the expected semantics. It was then demonstrated how to obtain the expected results that the nets defined have the desired behaviour.

With the net semantics of a simple C-like programming language, we developed a semantics for concurrent separation logic and showed that this was sound. The basis of the semantics for the logic was to model ownership, providing an explicit account of the intuitions in [O'H07]. The process of forming a net to model interference and then synchronizing with the original semantics seems to be a direct and general way of interpreting the form of rely-guarantee reasoning employed by the logic.

By defining a net semantics rather than using an interleaving model, we retain information on the concurrency of events. We have seen that this allows us to capture directly within the semantics important properties, for example the race-freedom of programs proved in concurrent separation logic. Such properties are important when considering concurrent programs, and we anticipate shall become increasingly so with the move towards programming concurrency on modern processors. For example, Boudol has recently shown in [BP09] that race-freedom allows a more tractable understanding of the weak memory model on multi-core processors. We hope to establish a proper connection between that work and our net semantics for separation logic in the future. The separation result obtained is, in fact, stronger than just race freedom, for example showing that interaction between parallel processes may occur through allocation and deallocation. This is significant, as such interaction leads to examples of the incompleteness of concurrent separation logic.

Turning to defining symmetry on models for concurrency, we have seen how symmetry plays a central role in defining the semantics of general forms of Petri net. By enriching with symmetry, we can recover the cofreeness characterization of the unfolding 'up to symmetry'. One might wonder whether a tighter characterization of the unfolding might

be obtained. One possible way is to consider 'nominal' nets, nets that allow permutations of names [GP01]. This would equip markings of the net with a permutation structure on tokens, and might be used to capture the symmetry in conditions in the unfolding of general nets directly.

Of course, symmetry is important in its own right in concurrency, and can be used to reduce state spaces encountered when model checking [Sis04]. In the semantics of programming languages, it occurs wherever identical threads are spawned, when names are bound, when locations are allocated, and in many other places. The work here begins to form an abstract account of how symmetry on nets can be represented, leading us to study a more general framework than that in [Win07a], for example through dropping the jointly-monic constraint. In fact, this is also necessary in other work [Win], so this suggests that not all models fit the simple scheme appropriate for event structures and stable families described in [Win07a].

Related work

Net semantics and separation logic

The first component of this work provides an inductive definition of the semantics as a net of programs operating in a (shared) state. This is a relatively novel technique, but has in the past been applied to give the semantics of a language for investigating security protocols, SPL [CW01b], though our language involves a richer collection of constructs. Other independence models for terms include the Box calculus [BDH92] and the event structure and net semantics of CCS [Stu80, Win82, GM84, WN95], though these model interaction as synchronized communication rather than occurring through shared state. We hope that the novel Petri net semantics presented here and in [CW01b] can be the start of *systematic and comprehensive* methods to attribute structural Petri net semantics to a full variety of programming languages, resulting in a Petri net companion to Plotkin's structural operational semantics (SOS) based on transition systems [Plo81]. Paralleling the (inductive) definitions of data and transitions of SOS would be (inductive) definitions of conditions and events of Petri nets.

The proof of soundness of separation logic here is led by Brookes' earlier pioneering proof of soundness based on action traces [Bro07]. There are a few minor differences in the syntax of processes, including that we allow the dynamic binding of resource variables. Another minor difference between the programming language and logic considered here and that introduced by O'Hearn and proved sound by Brookes is that we do not distinguish *stack variables*. These may be seen as locations to which other locations may not point and are the only locations that terms can directly address. In Brookes' model, as in [O'H07], interference of parallel processes through stack variables is constrained by the use of a side condition on the rule rather than using the concept of ownership (the area of current research on 'permissions' [BCOP05, BCY05, Bro06] promises a uniform approach). In particular, the rule allows the concurrent reading of stack locations. Though we have chosen not to include stack variables in our model in order to highlight the concept of ownership, our model and proofs could be easily extended to deal with them. Concurrent reading of memory would be at the cost of a more sophisticated notion of independence that allowed independent events to access the same condition providing that neither affects the marking of that condition.

More notably, at the core of Brookes' work is a 'local enabling relation', which gives the semantics of programs over a restricted set of 'owned' locations. Our notion of validity

involves maintaining a record of ownership and using this to constrain the occurrence of events in the interference net augmented to the process. This allows the intuition of ownership in O'Hearn's introduction of concurrent separation logic [O'H07] to be seen directly as constraining interference. Though the intuitive relationship between our model and Brookes' is fairly obvious, we believe that our approach leads to a clearer parallel decomposition lemma, upon which the proof of soundness of the logic critically stands.

The most significant difference between our work and Brookes' is that the net model captures, as a primitive property, the independence of parallel processes enforced by the logic. We have used this property to define a straightforward, yet general, form of refinement suited to changing the atomicity of commands within the semantics of a term. This is in contrast to [Bro05], which gives a new form of trace semantics to race-free processes that abstracts entirely away from attaching any form of atomicity to the semantics of heap actions. It has proven difficult, however, to show that this corresponds to the semantics in [Bro07].

Another study of the semantics of concurrent separation logic is conducted by Calcagno, O'Hearn and Yang in [COY07]. There, the goal is to provide a semantics that abstracts away components of the particular structure on which processes operate. In particular, their semantics is not dependent on processes interacting on a shared memory formed of locations holding values: it instead assumes that the processes operate on an arbitrary structure providing it allows the effect of processes to be specified through their local effect on the abstract state. Perhaps this form of action semantics might be combined with that in [BCHK93], where an operational semantics is embellished with extra structure on location and is used to reveal concurrency of actions.

Symmetry and unfoldings

Occurrence nets were first introduced in [NPW81] together with the operation of unfolding singly-marked safe nets. The coreflection between occurrence nets and safe nets was first shown in [Win84]. A number of attempts have been made since then to characterize the unfoldings of more general forms of net.

Engelfriet defines the unfolding of (singly-marked) P/T nets in [Eng91]. Rather than giving a coreflection between the categories, the unfolding is characterized as the greatest element of a complete lattice of occurrence nets embedding into the P/T net.

A coreflection between a subcategory of (singly-marked) general nets and a category of embellished forms of transition system is given in [Muk92]. There, the restriction to particular kinds of net morphism is of critical importance; taking the more general morphisms of general Petri nets presented here would have resulted in the cofreeness property failing for an analogous reason to the failure of cofreeness of the unfolding of general nets to occurrence nets without symmetry.

A coreflection between a subcategory of general nets and generalized event structures is presented in [HKT96]. However, there auto-concurrency of events is ruled out, thereby side-stepping the issues of symmetry presented here.

An adjunction between a subcategory of singly-marked general nets and the category of occurrence nets is given in [MMS96]. The restriction imposed on the morphisms of general nets there, however, precludes in general there being a morphism from $\mathcal{U}(G)$ to G in their category of general nets if $\mathcal{U}(G)$, the occurrence net unfolding of G, is regarded directly as a general net. To obtain an adjunction, the functor from the category of occurrence nets into the category of general nets is not regarded as the direct inclusion, but instead occurs through a rather detailed construction and does not yield a coreflection apart from when restricted to the subcategory of semi-weighted nets as defined in [MMS96].

Future work

In addition to the future work described above on using nominal techniques to define symmetry and establishing a connection between our model and the work on weak memory models in [BP09], there are several topics worthy of further investigation.

One broad area of research might be to consider how we can relate our work on net semantics to other forms of Petri net. In particular, there has been extensive study of Petri nets with time and probability [Mar89]. Presumably the techniques for defining the net semantics described here can also be applied to obtain such nets, which might be suited to the analysis of the cost of memory access — an issue of concern in multi-core processors, where there are large differences in the time taken to access different forms of memory location [Myc07]. Also, independence models are often claimed to be well-suited to the study of fairness and liveness. It would be interesting to consider how the net semantics could be used to provide a semantics to recent work on proving various liveness properties established using an extension of separation logic [GCPV09].

Another area for further research, described at the end of Chapter 6, is to prove that the refinement operator there can be used to address the issue of granularity. Other more specific areas, not described so far, are:

Imprecision of invariants

The rules of separation logic presented in [O'H07] rely on the resource invariants to be precise, as we saw on page 67. An example due to Reynolds shows that if imprecise (not precise) invariants are allowed, the rules of concurrent separation logic become unsound with respect to any sensible semantics. However, the example relies on the presence of the rule representing Hoare's law of conjunction:

$$(\text{L-Conjunction}) : \frac{\Gamma \vdash \{\varphi_1\}t\{\psi_1\} \qquad \Gamma \vdash \{\varphi_2\}t\{\psi_2\}}{\Gamma \vdash \{\varphi_1 \wedge \varphi_2\}t\{\psi_1 \wedge \psi_2\}}$$

It is not hard to see how this rule is inconsistent with allowing imprecision. For example, consider the imprecise invariant $\ell \mapsto 0 \vee \mathsf{empty}$. Using the rules for nil process, logical equivalence and critical regions, we can derive the following two judgements:

$$r{:}\ell \mapsto 0 \vee \mathsf{empty} \vdash \{\ell \mapsto 0\}\texttt{with}\ r\ \texttt{do}\ \varepsilon\ \texttt{od}\{\ell \mapsto 0\}$$
$$r{:}\ell \mapsto 0 \vee \mathsf{empty} \vdash \{\ell \mapsto 0\}\texttt{with}\ r\ \texttt{do}\ \varepsilon\ \texttt{od}\{\mathsf{empty}\}$$

Hence, using the law of conjunction, we may derive the following judgement, which with a normal semantics would imply that the process never terminates:

$$r{:}\ell \mapsto 0 \vee \mathsf{empty} \vdash \{\ell \mapsto 0\}\texttt{with}\ r\ \texttt{do}\ \varepsilon\ \texttt{od}\{\bot\}$$

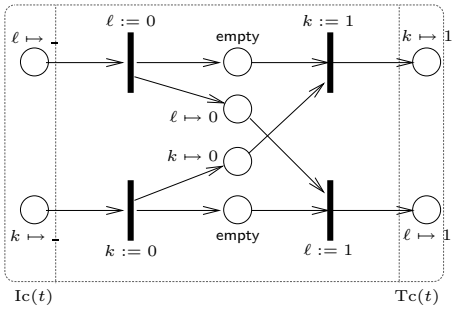
If we drop the law of conjunction from the logic, it seems as though a sensible semantics can be given that at ground level would yield the same form of validity, *i.e.* without affecting Corollary 5.3. This would allow us to drop the checks for precision when the conjunction rule is not used. The semantics for open terms is an interesting variant of the semantics for ownership presented earlier but now based on strategies for games.

More specifically, the semantics for judgements will involve a game played by a refuter and a verifier. For a judgement $\Gamma \vdash \{\varphi\}t\{\psi\}$, the game is played on the ownership net $\mathcal{W}[\![t]\!]_\Gamma$ from any initial state that satisfies φ in Γ. Play starts with the refuter who continues to choose enabled events, the marking of the net changing accordingly, until it

chooses an event that releases a resource with an invariant. It then becomes the verifier's turn, who must choose which part of the owned heap, satisfying the invariant, the process is to release ownership of. Play then resumes with the verifier who proceeds as before. The refuter wins if either a violating marking is encountered, if the verifier cannot find part of the heap to release ownership of, or if a terminal marking is encountered that does not satisfy ψ in Γ. The judgement is valid if the refuter does not have a winning strategy (*i.e.* cannot force play so that it certainly encounters a winning state within a finite number of turns).

Abstraction

Another rather more speculative area for further research seeks to use the net semantics to develop a more abstract account of the logic, where we abstract away the role of the heap conditions from the semantics. To replace them, we label the control conditions with a formula of the heap logic. Events will be labelled to characterize their effect as actions on the state. For example, we might have the following net:



There will obviously be rules that the formulae on conditions must obey, such that whenever an action occurs in a heap initially satisfying the multiplicative conjunction of the formulae labelling the preconditions of its event, the multiplicative conjunction of the formulae labelling the postconditions of the event holds of the heap. Conditions to represent resources will be labelled similarly with the associated resource invariant. One would then seek to prove that if $\Gamma \vdash \{\varphi\}t\{\psi\}$ then there exists a way of labelling the conditions of the net for t in such a way that their multiplicative conjunction is implied by φ, and dually to label the terminal conditions in such a way that their multiplicative conjunction implies ψ. This would provide an alternative way of proving soundness of the logic. In addition to providing a graphical way of presenting proofs of programs, this kind of structure with formulae labelling conditions is of interest since it might provide a connection with the net semantics of linear logic [EW94] or of bunched implications [POY04]. It would also be interesting to consider whether some sort of technique like abstract interpretation could be performed with this structure, and whether any benefits from locality would ensue.

Bibliography

[BCHK93] Gérard Boudol, Ilaria Castellani, Matthew Hennessy, and Astrid Kiehn. A theory of processes with localities. *Formal Aspects of Computing*, 6(2):165–200, 1993.

[BCOP05] Richard Bornat, Cristiano Calcagno, Peter W. O'Hearn, and Matthew Parkinson. Permission accounting in separation logic. In *Proc. Priniples of Programming Languages 2005 (POPL '05)*. ACM Press, 2005.

[BCY05] Richard Bornat, Cristiano Calcagno, and Hongseok Yang. Variables as resource in separation logic. In *Proc. Mathematical Foundations of Programming Semantics (MFPS XXI)*, ENTCS, 2005.

[BDH92] Eike Best, Raymond Devillers, and Jon G. Hall. The box calculus: A new causal algebra with multi-label communication. In *Advances in Petri Nets 1992*, volume 609 of *Lecture Notes in Computer Science*. Springer-Verlag, 1992.

[Bed88] Marek A. Bednarczyk. *Categories of Asynchronous Systems*. PhD thesis, Computer Science, University of Sussex, 1988. report number 1/88.

[BP09] Gérard Boudol and Gustavo Petri. Relaxed memory models: an operational approach. In *Proc. Principles of Programming Languages 2009 (POPL '09)*, 2009.

[Bri72] Per Brinch Hansen. Structured multiprogramming. *Communications of the ACM*, 15(7):574–578, 1972.

[Bro04] Stephen Brookes. A semantics for concurrent separation logic. In *Proc. International Conference on Concurrency Theory 2004 (CONCUR '04)*, volume 3170 of *LNCS*. Springer-Verlag, 2004.

[Bro05] Stephen Brookes. A grainless semantics for parallel programs with shared mutable data. In *Proc. Mathematical Foundations of Programming Semantics (MFPS XXI)*, ENTCS, 2005.

[Bro06] Stephen Brookes. Variables as resource for shared-memory programs: Semantics and soundness. In *Proc. Mathematical Foundations of Programming Semantics (MFPS XXII)*, ENTCS, 2006.

[Bro07] Stephen Brookes. A semantics for concurrent separation logic. *Theoretical Computer Science*, 375(1–3), 2007. Extended version of [Bro04].

[BRR87] Wilfried Brauer, Wolfgang Reisig, and Grzegorz Rozenberg, editors. *Advances in Petri Nets 1986: Proceedings of an Advanced Course*, volume 254–255 of *Lecture Notes in Computer Science*. Springer, 1987.

[CMP87] Luca Castellano, Giorgio De Michelis, and Lucia Pomello. Concurrency vs. interleaving: An instructive example. *Bulletin of the European Associatation for Theoretical Computer Science*, 31:12–14, 1987.

[COY07] Cristiano Calcagno, Peter W. O'Hearn, and Hongseok Yang. Local action and abstract separation logic. In *Proc. Logic in Computer Science 2007 (LICS '07)*, pages 366–378. IEEE Press, 2007.

[CV98] A. Carboni and E. M. Vitale. Regular and exact completions. *Journal of Pure and Applied Algebra*, 125(1–3):79–116, March 1998.

[CW01a] Federico Crazzolara and Glynn Winskel. Events in security protocols. In *ACM Conference on Computer and Communications Security*, 2001.

[CW01b] Federico Crazzolara and Glynn Winskel. Events in security protocols. In *Proc. ACM Conference on Computer and Communications Security 2001 (CCS '01)*, New York, 2001. ACM Press.

[DGT07] Shaddin F. Doghmi, Joshua D. Guttman, and F. Javier Thayer. Searching for shapes in cryptographic protocols. In *Proc. Tools and Algorithms for the Contruction and Analysis of Systems 2007 (TACAS'07)*, 2007.

[Dij68] Edsger Dijkstra. Cooperating sequential processes. In F. Genuys, editor, *Programming Languages*. Academic Press, 1968.

[Eng91] Joost Engelfriet. Branching processes of Petri nets. *Acta Informatica*, 28:575–591, 1991.

[EW94] Uffe Engberg and Glynn Winskel. Linear Logic on Petri nets. In *A Decade of Concurrency: Reflections and Perspectives. Proc. REX school/symposium*, volume 803 of *Lecture Notes in Computer Science*, pages 176–229. Springer-Verlag, 1994.

[Fab06] Eric Fabre. On the construction of pullbacks for safe Petri nets. In *Proc. International Conference on Application and Theory of Petri Nets 2006 (ICATPN '06)*, volume 4024 of *Lecture Notes in Computer Science*. Springer-Verlag, 2006.

[FHG98] F. Javier Thayer Fabrega, Jonathan C. Herzog, and Joshua D. Guttman. Strand spaces: Why is a security protocol correct. In *Proc. IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 1998.

[GCPV09] Alexey Gotsman, Byron Cook, Matthew Parkinson, and Viktor Vafeiadis. Proving that non-blocking algorithms don't block. In *Proc. Principles of Programming Languages 2009 (POPL '09)*, 2009.

[GM84] Ursula Goltz and Alan Mycroft. On the relationship of CCS and Petri nets. In *Proc. International Colloquium on Automata, Languages and Programming 1984 (ICALP '84)*, volume 172 of *Lecture Notes in Computer Science*. Springer-Verlag, 1984.

[GP01] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2001.

[HKT96] P.W. Hoogers, H. C. M Kleijn, and P. S. Thiagarajan. An event structure semantics for general Petri nets. *Theoretical Computer Science*, 153(1–2):129–170, 1996.

[Hoa69] C. A. R. Hoare. An axiomatic basis for computer programming. *Comm. ACM*, 12(10):576–580, 1969.

[Hoa72] C. A. R. Hoare. Towards a theory of parallel programming. In C. A. R. Hoare and R. H. Perrot, editors, *Operating Systems Techniques*. Academic Press, 1972.

[HW08a] Jonathan Hayman and Glynn Winskel. Independence and concurrent separation logic. *Logical Methods in Computer Science*, 4(1), 2008. To appear in a special issue for LICS '06.

[HW08b] Jonathan Hayman and Glynn Winskel. Symmetry in Petri nets. In Kamal Lodaya, Madhavan Mukund, and R. Ramanujam, editors, *Perpectives in Concurrency*. Universities Press, 2008.

[HW08c] Jonathan Hayman and Glynn Winskel. The unfolding of general Petri nets. In *Proc. Foundations of Software Technology and Theoretical Computer Science 2008 (FSTTCS '08)*, 2008.

[IO01] Samin S. Ishtiaq and Peter W. O'Hearn. BI as an assertion language for mutable data structures. In *Proc. Principles of Programming Languages 2001 (POPL '01)*. ACM Press, 2001.

[Jen96] Kurt Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*. Springer, second edition, 1996.

[JNW95] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. In *Proc. Logic in Computer Science 1993 (LICS '93)*, volume 127(2) of *Information and Computation*, 1995.

[Joh02] Peter T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium*, volume 1. Oxford University Press, 2002.

[Jon83] Cliff B. Jones. Specification and design of (parallel) programs. In R. E. A. Mason, editor, *Information Processing 83: Proc. IFIP Congress*, pages 321–332, 1983.

[Lam86] Leslie Lamport. On interprocess communication. *Distributed Computing*, 1(2):77–101, June 1986.

[Mac71] Saunders MacLane. *Categories for the Working Mathematician*. Springer, 1971.

[Mar89] M. Ajmone Marsan. Stochastic Petri nets: An elementary introduction. In *Advances in Petri Nets 1989*, volume 424 of *Lecture Notes in Computer Science*, pages 1–29. Springer-Verlag, 1989.

[Maz77] Antoni Mazurkiewicz. Concurrent program schemes and their interpretations. Technical Report DAIMI PB-78, University of Aarhus, 1977.

[Maz89] Antoni Mazurkiewicz. Basic notions of trace theory. In *Linear Time, Branching Time and Partial Order in Logics and Models for Concurrency*, volume 354 of *Lecture Notes in Computer Science*, pages 285–363, 1989.

[MMS96] José Meseguer, Ugo Montanari, and Vladminiro Sassone. On the semantics of Place/Transition Petri nets. *Mathematical Structures in Computer Science*, 7:359–397, 1996.

[Moo65] Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics Magazine*, 38(8), April 1965. Available from `http://download.intel.com/research/silicon/moorespaper.pdf`.

[Moo05] Gordon E. Moore. Excerpts from A Conversation with Gordon Moore. Available from `download.intel.com/museum/MooresLaw/`, 2005.

[Muk92] Madhavan Mukund. Petri nets and step transition systems. *International Journal of Foundations of Computer Science*, 3(4):443–478, 1992.

[Myc07] Alan Mycroft. Programming language design and analysis motivated by hardware evolution. In *Static Analysis Symposium 2007*, volume 4634 of *Lecture Notes in Computer Science*, pages 18–33. Springer-Verlag, 2007.

[NPW81] Mogens Nielsen, Gordon Plotkin, and Glynn Winskel. Petri nets, event structures and domains, Part 1. *Theoretical Computer Science*, 13:85–108, 1981.

[NW96] Mogens Nielsen and Glynn Winskel. Petri nets and bisimulation. *Theoretical Computer Science*, 1–2(153):211–244, Janurary 1996.

[OG76] Susan Owicki and David Gries. Verifying properties of parallel programs: an axiomatic approach. *Communications of the ACM*, 19(5):279–285, 1976.

[O'H04] Peter W. O'Hearn. Resources, concurrency and local reasoning. In *Proc. International Conference on Concurrency Theory 2004 (CONCUR '04)*, volume 3170. Springer-Verlag, 2004.

[O'H07] Peter W. O'Hearn. Resources, concurrency and local reasoning. *Theoretical Computer Science*, 375(1–3):271–307, 2007. Extended version of [O'H04].

[OP99] Peter W. O'Hearn and David J. Pym. The logic of bunched implications. *Bulletin of Symbolic Logic*, 5(2), 1999.

[Pet62] Carl Adam Petri. *Kommunikation mit Automaten*. PhD thesis, Institut für Instrumentelle Mathematik, Bonn, 1962.

[Pey08] Simon Peyton Jones. Harnessing the multicores: Nested data parallelism in Haskell. In *Proc. Foundations of Computer Science and Theoretical Computer Science 2008 (FSTTCS '08)*, Bangalore, 2008.

[Plo81] Gordon D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.

[Pow98] John Power. 2-categories. Technical Report NS-98-7, BRICS, August 1998.

[POY04] David J. Pym, Peter W. O'Hearn, and Hongseok Yang. Possible worlds a resources: the semantics of **BI**. *Theoretical Computer Science*, 315(1):257–305, 2004.

[Pra86] Vaughan Pratt. Modeling concurrency with partial orders. *International Journal of Parallel Programming*, 15(1), 1986.

[Pym02] David J. Pym. *The Semantics and Proof Theory of the Logic of the Logic of Bunched Implications*, volume 26 of *Applied Logic Series*. Kluwer Academic Publishers, 2002.

[Rei85] Wolfgang Reisig. *Petri Nets*. EATCS Monographs on Theoretical Computer Science. Springer-Verlag, 1985.

[Rey00] John C. Reynolds. Intuitionistic reasoning about shared mutable data structure. In *Millennial Perspectives in Computer Science*, 2000.

[Rey02] John C. Reynolds. Separation logic: A logic for shared mutable data structures. In *Proc. Logic in Computer Science 2002 (LICS '02)*. IEEE Computer Society, 2002.

[Rey04] John C. Reynolds. Towards a grainless semantics for shared variable concurrency. In *Proc. Foundations of Software Technology and Theoretical Computer Science 2004 (FSTTCS '04)*, volume 3328 of *LNCS*. Springer-Verlag, 2004.

[RT88] Alexander Moshe Rabinovich and Boris A. Trakhtenbrot. Behaviour structures and nets. *Fundamenta Informaticae*, 11(4), 1988.

[Sas98] Vladimiro Sassone. An axiomatization of the category of petri net computations. *Mathematical Structures in Computer Science*, 8(2):117–151, 1998.

[Shi85] Mike W. Shields. Concurrent machines. *The Computer Journal*, 28:449–465, 1985.

[Sis04] A. Prasad Sistla. Employing symmetry reductions in model checking. *Computer Languages, Systems and Structures*, 30(3–4):99–137, 2004.

[Stu80] Students. Projects for the course "Models for Concurrency" given by G. Winskel and M. Nielsen. Dept. of Computer Science, University of Aarhus, Denmark, 1980.

[vG05] Robert J. van Glabbeek. The individual and collective token interpretations of Petri nets. In *Proc. International Conference on Concurrency Theory 2005 (CONCUR '05)*, volume 3653 of *Lecture Notes in Computer Science*. Springer-Verlag, 2005.

[vGG89] Rob J. van Glabbeek and Ursula Goltz. Equivalence notions for concurrent systems and refinement of actions. In *Proc. International Symposium on Mathematical Foundations of Computer Science 1989 (MFCS '89)*, volume 379 of *LNCS*. Springer-Verlag, 1989.

[vGV87] Robert van Glabbeek and Fritz Vaandrager. Petri net models for algebraic theories of concurrency. In *Proc. Parallel Architectures and Languages Europe 1987 (PARLE '87)*, volume 259 of *Lecture Notes in Computer Science*. Springer-Verlag, 1987.

[VP07] Viktor Vafeiadis and Matthew Parkinson. A marriage of Rely/Guarantee and separation logic. In *Proc. International Conference on Concurrency Theory 2007 (CONCUR '07)*, volume 4703 of *Lecture Notes in Computer Science*, pages 256–271. Springer-Verlag, 2007.

[Win] Glynn Winskel. The symmetry of stability. Forthcoming.

[Win82] Glynn Winskel. Event structure semantics for CCS and related languages. In *Proc. International Colloquium on Automata, Languages and Programming 1982 (ICALP '82)*, volume 140 of *Lecture Notes in Computer Science*. Springer-Verlag, 1982.

[Win84] Glynn Winskel. A new definition of morphism on Petri nets. In *Proc. Symposium on Theoretical Aspects of Computer Science 1984 (STACS '84)*, volume 166 of *Lecture Notes in Computer Science*, 1984.

[Win86] Glynn Winskel. Event structures. In *Advances in Petri Nets, Part II*, volume 255 of *Lecture Notes in Computer Science*. Springer, 1986.

[Win87] Glynn Winskel. Petri nets, algebras, morphisms and compositionality. *Information and Computation*, 72(3):197–238, 1987.

[Win07a] Glynn Winskel. Event structures with symmetry. *Electronic Notes in Theoretical Computer Science*, 172, 2007.

[Win07b] Glynn Winskel. Symmetry and concurrency. In *Proc. Conference on Algebra and Coalgebra in Computer Science 2007 (CALCO '07)*, volume 4624 of *Lecture Notes in Computer Science*. Springer-Verlag, August 2007. Invited talk.

[WN95] Glynn Winskel and Mogens Nielsen. Models for concurrency. In *Handbook of Logic and the Foundations of Computer Science*, volume 4, pages 1–148. Oxford University Press, 1995.

Appendix A

Multisets

Let the set of natural numbers $\{0, 1, 2, \ldots\}$ be denoted $\mathbb{N}$. A multiset over a set A is a vector of elements of $\mathbb{N}$ indexed by elements of A, or equivalently a function from A to $\mathbb{N}$. For instance, let $A = \{a_0, a_1\}$ and suppose that the multiset X contains two occurrences of a_0 and one of a_1; the corresponding multiset is:

$$\begin{array}{c} a_0 \\ a_1 \end{array} \left[\begin{array}{c} 2 \\ 1 \end{array}\right]$$

Let the set of multisets over a set A be denoted $\mu(A)$ and write $X \subseteq_\mu A$ if X is a multiset over A. Let $X[a]$ denote the value of the vector at a. Write $\vec{0}_A$ for the empty multiset with basis A. Denote multiplication of the multiset X by a scalar $n \in \mathbb{N}$ by $n.X$. A multiset X with basis A is said to be *finite* if $\sum_{a \in A} X[a]$ is finite.

Define the set $\mathbb{N}_\infty = \mathbb{N} \cup \{\infty\}$. An ∞-multiset over the set A is a vector of elements of $\mathbb{N}_\infty$ indexed by elements of A. The set of all ∞-multisets over A is denoted $\mu_\infty(A)$, and we write $X \subseteq_{\mu_\infty} A$ if X is an ∞-multiset over A. Addition and multiplication on integers is extended to the element ∞ by defining

$$\begin{array}{rcrcll} \infty + n & = & n + \infty & = & \infty & (\forall n \in \mathbb{N}_\infty) \\ \infty \cdot n & = & n \cdot \infty & = & \infty & (\forall n \in \mathbb{N}_\infty \setminus \{0\}) \\ \infty \cdot 0 & = & 0 \cdot \infty & = & 0 & \end{array}$$

Subtraction $m - n$ of two elements $m, n \in \mathbb{N}_\infty$ is a partial operation, defined iff $n \leq m$ and $n \neq \infty$. As such, the value of $\infty - \infty$ is left undefined.

Addition and subtraction of multisets are defined in the usual way as addition and subtraction of vectors. On ∞-multisets, vector addition and subtraction is defined with respect to the arithmetic above. Note that subtraction on the natural numbers is a partial operation, and therefore so is subtraction of (∞)-multisets.

A multirelation R between sets A and B is a matrix of elements of $\mathbb{N}$, and similarly an ∞-multirelation is a matrix of elements in $\mathbb{N}_\infty$. The number of times that the element a is related to b is given by $R[a, b]$, which is the natural number (or ∞) occurring in the a-indexed row and b-indexed column of R. We write $R \subseteq_\mu A \times B$ if R is a multirelation between A and B and $R \subseteq_{\mu_\infty} A \times B$ if R is an ∞-multirelation between A and B, noting the equivalent formulation of a multirelation as a multiset over the basis $A \times B$.

Application of a multirelation to a multiset is dealt with at greater generality. We regard ∞ as the first infinite cardinal number. A *cardinal multiset* X over a set A associates a cardinal number $X[a]$ to any element of A. A *cardinal multirelation* between sets A and B is a cardinal multiset over $A \times B$, the set product of X and Y. Application of a cardinal

multirelation R to a cardinal multiset X is obtained as their inner product $R \cdot X$. In particular, $R \cdot X$ is a cardinal multiset over B and for any $b \in B$

$$(R \cdot X)[b] = \sum_{a \in A} R[a, b] \cdot X[a].$$

Care has to be taken since application of a multirelation $R \subseteq_\mu A \times B$ to a multiset X may fail to yield a multiset if the above sum is greater than or equal to ∞ at any $b \in B$. It can also fail to be an ∞-multiset. For example, let $A = \{a\}$ and $B = \mathbb{R}$, the set of real numbers. Let R be defined as $R[x, a] = 1$ for any $x \in \mathbb{R}$. Let $\hat{a}$ denote the multiset with a single element a, so $\hat{a}[a] = 1$. We have $R \cdot \hat{a} > \infty$, so $R \cdot \hat{a}$ is not an ∞-multiset.

Say that an ∞-multirelation R over $A \times B$ is *countably injective* if for every $b \in B$ the set $\{a \in A \mid R[a, b] > 0\}$ is countable. Application of a countably injective ∞-multirelation to a ∞-multiset always yields an ∞-multiset.

Sets, relations and (partial) functions

We say that a multiset $X \subseteq_\mu A$ is a *set* if $X[a] \leq 1$ for all $a \in A$. All the usual notation for sets is adopted in this situation, for example $a \in X$ for $X[a] = 1$.

A *relation* R on sets A and B, written $R \subseteq A \times B$ is identified with a multirelation $R \subseteq_\mu A \times B$ such that $R(a, b) \leq 1$ for all $a \in A$ and $b \in B$. We now write $R(a, b)$ or aRb if, as a multirelation, $R[a, b] = 1$. We write R^* for the reflexive, transitive closure of a relation R, and write R^+ for the transitive closure of R.

If f is a *partial* function from set X to set Y, written $f{:}X \to_* Y$, that is undefined on $x \in X$, we write $f(x) = *$. As with relations, we identify partial functions with certain multirelations. We write $f{:}X \to Y$ if f is a function from X to Y.

Appendix B

Pullbacks of nets

In this appendix, we study pullbacks in categories of Petri nets. These are used to show how nets support the abstract framework for their extension with symmetry and for composition of open maps.

B.1 Pullbacks of P/T nets

We shall begin by describing pullbacks of P/T nets.

It is shown in [NW96] that the category of labelled safe Petri nets has pullbacks, and a similar result for unlabelled nets (thereby allowing morphisms to be partial functions on events) is studied in detail in [Fab06].

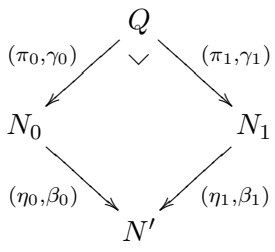
We begin this section by generalizing the construction so that it can be applied to give pullbacks in the category $\mathbf{PT}^\sharp$ of multiply-marked P/T nets, and so that it can later be used to give pullbacks of particular kinds of morphism in $\mathbf{Gen}^\sharp$.

It is necessary to impose the restriction that no P/T net has any isolated conditions in order to obtain pullbacks, where a condition is said to be isolated if it is neither a pre- nor a postcondition to any event nor in some initial marking. Without this additional constraint (also seen in the constructions in [NW96, Fab06]), simple counterexamples can be given to show that the category of P/T nets would not have pullbacks, analogous to those establishing the fact that the category of sets with relations between them does not have pullbacks.

Let

$$\begin{aligned} N_0 &= (P_0, T_0, F_0, \mathbb{M}_0) \\ N_1 &= (P_1, T_1, F_1, \mathbb{M}_1) \\ N' &= (P', T', F', \mathbb{M}') \end{aligned}$$

be P/T nets with morphisms $(\eta_0, \beta_0){:}N_0 \to N'$ and $(\eta_1, \beta_1){:}N_1 \to N'$. We shall show how to construct the pullback of (η_0, β_0) against (η_1, β_1).



As we do so, we shall assume that the sets P_0 and P_1 are disjoint and that the sets T_0 and T_1 are disjoint. This makes no difference to the generality of the construction since we can always form the pullback of nets isomorphic to N_0 and N_1 satisfying these properties which has no effect since pullbacks are only defined up to isomorphism.

Candidate pullback

The construction of the pullback involves equivalence classes of conditions. Define a relation $\sim_0 \subseteq P_0 \times P_1$ as

$$p_0 \sim_0 p_1 \iff \exists p' \in P' : \beta_0(p_0, p') \;\&\; \beta_1(p_1, p').$$

For sets $X_0 \subseteq P_0$ and $X_1 \subseteq P_1$, the relation $\sim_{X_0}^{X_1}$ is the equivalence relation generated by $\sim_0$ on X_0 and X_1. That is, $\sim_{X_0}^{X_1}$ is the least reflexive, transitive and symmetric relation on $X_0 \cup X_1$ satisfying

$$p_0 \in X_0 \;\&\; p_1 \in X_1 \;\&\; p_0 \sim_0 p_1 \implies p_0 \sim_{X_0}^{X_1} p_1.$$

For any $p \in X_0 \cup X_1$, we write $[p]_{X_0}^{X_1}$ for its $\sim_{X_0}^{X_1}$-equivalence class.

Note that if β_0 is locally injective on the set X_0, for example if X_0 is an initial marking or the set of pre- or post-conditions of some event, then $[p]_{\emptyset}^{X_0} = \{p\}$.

The conditions of the pullback, forming the set P_Q, are formed by taking the equivalence classes c such that

$$c = [p]_{X_0}^{X_1}$$

for some $p \in X_0 \cup X_1$ where $X_0 \subseteq P_0$ and $X_1 \subseteq P_1$ are sets such that $\beta_0 \cdot X_0 = \beta_1 \cdot X_1$. We also require, to ensure that there are no isolated conditions in the pullback, that either:

- $X_0 \in \mathbb{M}_0$ and $X_1 \in \mathbb{M}_1$,
- either $X_0 = {}^\bullet e_0$ and $X_1 = {}^\bullet e_1$ or $X_0 = e_0{}^\bullet$ and $X_1 = e_1{}^\bullet$ for some $e_0 \in E_0$ and $e_1 \in E_1$ such that $\eta_0(e_0) = \eta_1(e_1)$,
- either $X_0 = {}^\bullet e_0$ or $X_0 = e_0{}^\bullet$ and $X_1 = \emptyset$ for some $e_0 \in E_0$ such that $\eta_0(e_0) = *$, or
- $X_0 = \emptyset$ and either $X_1 = {}^\bullet e_1$ or $X_1 = e_1{}^\bullet$ for some $e_1 \in E_1$ such that $\eta_1(e_1) = *$.

Note that the first case, where we write $\eta_0(e_0) = \eta_1(e_1)$, includes the possibility that both $\eta_0(e_0) = *$ and $\eta_1(e_1) = *$.

The events of the pullback are defined by taking the pullback of the partial functions η_0 and η_1:

$$\begin{array}{rcll} & \{\ (t_0, t_1) & \mid & t_0 \in T_0 \;\&\; t_1 \in T_1 \;\&\; \eta_0(t_0) = \eta_1(t_1) \ \} \\ T_Q = \cup & \{\ (t_0, *) & \mid & t_0 \in T_0 \;\&\; \eta_0(t_0) = * \ \} \\ \cup & \{\ (*, t_1) & \mid & t_1 \in T_1 \;\&\; \eta_1(t_1) = * \ \} \end{array}$$

The pullback morphism $(\pi_0, \gamma_0){:}Q \to N_0$ is defined as

$$\begin{array}{ll} \pi_0(t_0, t_1) = & t_0 \\ \pi_0(t_0, *) = & t_0 \\ \pi_0(*, t_1) = & * \end{array} \qquad \gamma([p]_{X_0}^{X_1}, p') \iff p' \in [p]_{X_0}^{X_1}$$

with (π_1, γ_1):$Q \to N_1$ defined similarly. Using this definition, the flow relation on the pullback is defined as:

$$\begin{aligned} {}^\bullet e &= \{[p]^{\bullet\pi_1(e)}_{\bullet\pi_0(e)} \mid p \in {}^\bullet\pi_0(e) \cup {}^\bullet\pi_1(e)\} \\ e^\bullet &= \{[p]^{\pi_1(e)^\bullet}_{\pi_0(e)^\bullet} \mid p \in \pi_0(e)^\bullet \cup \pi_1(e)^\bullet\} \end{aligned}$$

The initial markings of the pullback are obtained as:

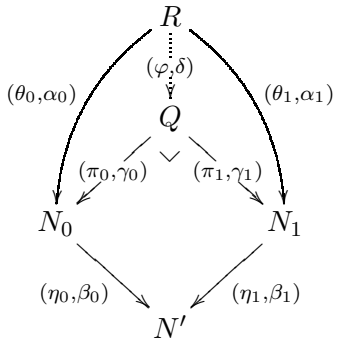
$$\mathbb{M} = \{\{[p]^{\mathbb{M}_1}_{\mathbb{M}_0} \mid p \in \mathbb{M}_0 \cup \mathbb{M}_1\} \mid M_0 \in \mathbb{M}_0 \;\&\; M_\in \mathbb{M}_1 \;\&\; \beta_0 \cdot M_0 = \beta_1 \cdot M_1\}$$

It is straightforward from these definitions to show that the candidate pullback is a P/T net (all that must be shown is that there are no isolated conditions and that every event has at least one precondition) and that (π_0, γ_0) and (π_1, γ_1) are morphisms such that

$$(\eta_0, \beta_0) \circ (\pi_0, \gamma_0) = (\eta_1, \beta_1) \circ (\pi_1, \gamma_1).$$

Universal property

It remains to show that the P/T net Q constructed above satisfies the universal property of being a pullback in the category $\mathbf{PT}^\sharp$. That is, we must show that for any pair of morphisms (θ_0, α_0):$R \to N_0$ and (θ_1, α_1):$R \to N_1$ in $\mathbf{PT}^\sharp$ such that the outer square in the diagram commutes there is a unique morphism (φ, δ):$R \to Q$ making the two triangles beneath R commute.



Let the net $R = (P_R, T_R, F_R, \mathbb{M}_R)$. The morphism (φ, δ):$R \to Q$ is defined on events as follows:

$$\varphi(t) = \begin{cases} * & \text{if } \theta_0(t) = * \;\&\; \theta_1(t) = * \\ (\theta_0(t), \theta_1(t)) & \text{otherwise} \end{cases}$$

The relation δ is defined as being the least relation such that $\delta(b, c)$ if either:

- $c = [p]^{\alpha_1 \cdot M}_{\alpha_0 \cdot M}$ for some $M \in \mathbb{M}_R$ and $p \in \alpha_0 \cdot M \cup \alpha_1 \cdot M$ such that $b \in M$ and either $\alpha_0(b, p)$ or $\alpha_1(b, p)$,
- $c = [p]^{\alpha_1 \cdot {}^\bullet t}_{\alpha_0 \cdot {}^\bullet t}$ for some $t \in T_R$ and $p \in \alpha_0 \cdot {}^\bullet t \cup \alpha_1 \cdot {}^\bullet t$ such that $b \in {}^\bullet t$ and either $\alpha_0(b, p)$ or $\alpha_1(b, p)$, or
- $c = [p]^{\alpha_1 \cdot t^\bullet}_{\alpha_0 \cdot t^\bullet}$ for some $t \in T_R$ and $p \in \alpha_0 \cdot t^\bullet \cup \alpha_1 \cdot t^\bullet$ such that $b \in t^\bullet$ and either $\alpha_0(b, p)$ or $\alpha_1(b, p)$.

The key to showing that (φ, δ) satisfies the constraints for being a morphism is the following lemma.

Lemma B.1.1. *Let $X, Y \subseteq P_R$, and let α_0, β_0, α_1 and β_1 be relations as in the diagram above. If*

- *β_0 is locally injective on the sets $\alpha_0 \cdot X$ and $\alpha_0 \cdot Y$,*
- *β_1 is locally injective on the sets $\alpha_1 \cdot X$ and $\alpha_1 \cdot Y$, and*
- *$p \in \alpha_0 \cdot (X \cap Y)$ or $p \in \alpha_1 \cdot (X \cap Y)$*

then

$$[p]_{\alpha_0 \cdot X}^{\alpha_1 \cdot X} = [p]_{\alpha_0 \cdot Y}^{\alpha_1 \cdot Y}.$$

Proof. Recall that the relation $\sim_{\alpha_0 \cdot X}^{\alpha_1 \cdot X}$ is defined inductively using the relation $\sim_0$. We shall prove that for any $p \in \alpha_0 \cdot (X \cap Y)$, if $p \sim_0 p'$ and $p' \in \alpha_1 \cdot X$ then $p' \in \alpha_1 \cdot (X \cap Y)$. The result then follows (with the symmetric property) by a straightforward induction on the length of the chain demonstrating $\sim_{\alpha_0 \cdot X}^{\alpha_1 \cdot X}$.

Suppose, then, that $p \sim_0 p'$ and $p \in \alpha_0 \cdot (X \cap Y)$ and $p' \in \alpha_1 \cdot X$. There exists $b \in X \cap Y$ such that $\alpha_0(b, p)$. Since $p \sim_0 p'$, there exists b' such that $\beta_0(p, b')$, and hence $b' \in \beta_0 \cdot \alpha_0 \cdot (X \cap Y)$, and also $\beta_1(p', b')$. We have $\beta_0 \circ \alpha_0 = \beta_1 \circ \alpha_1$ so there must exist p'' such that $\alpha_1(b, p'')$ and $\beta_1(p'', b')$. Hence $p'' \in \alpha_1 \cdot (X \cap Y)$. The relation β_0 is locally injective on the set $\alpha_1 \cdot X$, so we must therefore have $p' = p''$, completing the proof. □

We now show that $(\varphi, \delta) : R \to Q$ satisfies the constraints for being a net morphism.

Lemma B.1.2. *The pair (φ, δ) is a morphism of nets.*

Proof. Let $Q = (P_Q, T_Q, F_Q, \mathbb{M}_Q)$ and $R = (P_R, T_R, F_R, \mathbb{M}_R)$. We shall show only one part of the proof that (φ, δ) is a morphism, that if $t \in T_R$ then $\delta \cdot {}^\bullet t \subseteq {}^\bullet \varphi(t)$ and $\forall c \in {}^\bullet \varphi(t) : \exists! b \in {}^\bullet t : \delta(b, c)$. The other parts are similar.

$\underline{\delta \cdot {}^\bullet t \subseteq {}^\bullet \varphi(t)}$: Suppose that $b \in {}^\bullet t$ and $\delta(b, c)$. From the definition of the definition of δ, we have $c = [p]_{\alpha_0 \cdot X}^{\alpha_1 \cdot X}$ for some $X \subseteq P_R$ and $p \in \alpha_0 \cdot X \cup \alpha_1 \cdot X$ such that either $\alpha_0(b, p)$ or $\alpha_1(b, p)$. Without loss of generality, assume that $p \in P_0$, so we have $\alpha_0(b, p)$. Notice that therefore $\theta_0(t) \neq *$ so $\varphi(t) = (\theta_0(t), \theta_1(t)) \neq *$. It is easy to see from (η_0, β_0) and (η_1, β_1) being morphisms and the forms that X may take according to the definition of δ that β_0 is locally injective on $\alpha_0 \cdot X$ and β_1 is locally injective on $\alpha_1 \cdot X$. We also have β_0 locally injective on ${}^\bullet \theta_0(t)$ and β_1 locally injective on ${}^\bullet \theta_1(t)$. It follows from the previous lemma that

$$[p]_{\alpha_0 \cdot X}^{\alpha_1 \cdot X} = [p]_{\alpha_0 \cdot {}^\bullet t}^{\alpha_1 \cdot {}^\bullet t}.$$

Since $\alpha_0(b, p)$, we have $p \in {}^\bullet \theta_0(t)$ and therefore $[p]_{\alpha_0 \cdot {}^\bullet t}^{\alpha_1 \cdot {}^\bullet t} \in {}^\bullet(\theta_0(t), \theta_1(t))$, as required.

$\underline{\forall c \in {}^\bullet \varphi(t) : \exists! b \in {}^\bullet t : \delta(b, c)}$: Now suppose that $c \in {}^\bullet \varphi(t)$. We have $\varphi(t) \neq *$, so $\varphi(t) = (\theta_0(t), \theta_1(t))$. From the definition of ${}^\bullet \varphi(t)$, there exists $p \in {}^\bullet \theta_0(t) \cup {}^\bullet \theta_1(t)$ such that $c = [p]_{{}^\bullet \theta_0(t)}^{{}^\bullet \theta_1(t)}$. Without loss of generality, suppose that $p \in {}^\bullet \theta_0(t)$. Since (θ_0, α_0) is a morphism, there exists a (unique) $b \in {}^\bullet t$ such that $\alpha_0(b, p)$ and hence $\delta(b, c)$.

We now consider uniqueness of b. Suppose that there exists $b' \in {}^\bullet t$ such that $\delta(b', c)$. From the definition of δ, there exists $p' \in {}^\bullet \theta_0(t) \cup {}^\bullet \theta_1(t)$ such that $[p]_{{}^\bullet \theta_0(t)}^{{}^\bullet \theta_1(t)} = [p']_{{}^\bullet \theta_0(t)}^{{}^\bullet \theta_1(t)}$ and either $\alpha_0(b', p')$ or $\alpha_1(b', p')$. We shall show that, for any p_0 and p_1:

$$b \in {}^\bullet t \;\&\; p_0 \in {}^\bullet \theta_0(t) \;\&\; p_1 \in {}^\bullet \theta_1(t) \;\&\; \alpha_0(b, p_0) \;\&\; p_0 \sim_0 p_1 \implies \alpha_1(b, p_1).$$

A straightforward induction will then show that the same property holds for $\sim_{^\bullet\theta_0(t)}^{^\bullet\theta_1(t)}$, from which it immediately follows that $b = b'$.

Suppose that there exist $p_0 \in {}^\bullet\theta_0(t)$ and $p_1 \in {}^\bullet\theta_1(t)$ such that $\alpha_0(b, p_0)$ for some $b \in {}^\bullet t$ and $p_0 \sim_0 p_1$. Since $p_0 \sim_0 p_1$, there exists $b' \in B'$ such that $\beta_0(p_0, b')$ and $\beta_1(p_1, b')$. We have $\beta_0 \circ \alpha_0 = \beta_1 \circ \alpha_1$, so there exists $p_1' \in P_1$ such that $\alpha_1(b, p_1')$ and $\beta_1(p_1', b')$. Since α_1 is a morphism, we have $p_1' \in {}^\bullet\theta_1(t)$. From the local injectivity of β_1 on ${}^\bullet\theta_1(t)$ arising from (η_1, β_1) being a morphism, we have $p_1 = p_1'$ and hence $\alpha_1(b, p_1)$ as required. □

To complete the proof that Q as defined above is a pullback in the category $\mathbf{PT}^\sharp$, it remains to show that the two triangles in the diagram above commute and that the morphism (φ, δ) is the unique such morphism.

Proposition B.1. *For the morphisms described above:*

$$(\pi_0, \gamma_0) \circ (\varphi, \delta) = (\theta_0, \alpha_0) \qquad (\pi_1, \gamma_1) \circ (\varphi, \delta) = (\theta_1, \alpha_1).$$

Furthermore, for any morphism $(\varphi', \delta'){:}R \to Q$ such that

$$(\pi_0, \gamma_0) \circ (\varphi', \delta') = (\theta_0, \alpha_0) \qquad (\pi_1, \gamma_1) \circ (\varphi', \delta') = (\theta_1, \alpha_1),$$

it is the case that $\varphi = \varphi'$ and $\delta = \delta'$.

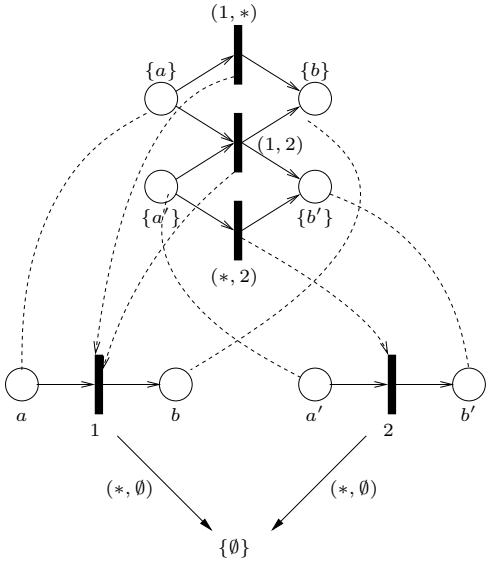
Proof. Commutation is proved by a straightforward calculation and it is easy to see that $\varphi = \varphi'$.

We first show that $\delta \subseteq \delta'$. Suppose that $\delta(b, c)$ for some $b \in P_R$ and $c \in P_Q$. Without loss of generality, suppose that $b \in {}^\bullet e$ for some $e \in T_R$; the other cases arising from the condition b not being isolated are similar. As argued in the previous lemma, we have $c = [p]_{^\bullet\theta_0(e)}^{^\bullet\theta_1(e)}$ for some p such that either $p \in {}^\bullet\theta_0(e)$ and $\alpha_0(b, p)$ or $p \in {}^\bullet\theta_1(e)$ and $\alpha_1(b, p)$. Without loss of generality, suppose that $p \in {}^\bullet\theta_0(e)$ and $\alpha_0(b, p)$. We therefore have $\gamma_0(c, p)$. Since $\gamma_0 \cdot \delta' = \alpha_0$, there must exist $c' \in P_Q$ such that $\delta'(b, c')$ and $\gamma_0(c', p)$. Since $b \in {}^\bullet e$ and $\delta'(b, c')$, we must have $c' \in {}^\bullet\varphi(e)$ and similarly $c \in {}^\bullet\varphi(e)$. It follows from (π_0, γ_0) being a morphism that $c = c'$, as required.

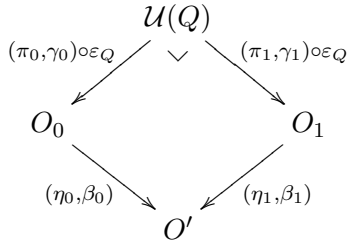
We now show that $\delta' \subseteq \delta$. Suppose that $\delta'(b, c)$. As above, without loss of generality we shall assume that there exists $e \in T_R$ such that $b \in {}^\bullet e$. Since $\varphi = \varphi'$ and (φ', δ') is a morphism, we have $c \in {}^\bullet\varphi(e)$. Recall that $\varphi(e) = (\theta_0(e), \theta_1(e))$. Again, as shown in the previous lemma when considering the preconditions of $\varphi(e)$, we have $c = [p]_{^\bullet\theta_0(e)}^{^\bullet\theta_1(e)}$ for some p such that either $p \in {}^\bullet\theta_0(e)$ and $\alpha_0(b, p)$ or $p \in {}^\bullet\theta_1(e)$ and $\alpha_0(b, p)$. In either case, it follows immediately from the definition of δ that $\delta(b, [p]_{^\bullet\theta_0(e)}^{^\bullet\theta_1(e)})$, as required. □

It is easy to see that the pullback of safe nets is also a safe net: If there were a reachable marking M of Q in which a condition c were marked twice, the markings $\gamma_0 \cdot M$ and $\gamma_1 \cdot M$ would be reachable in N_0 and N_1 respectively. Since there exists p such that either $\gamma_0(c, p)$ or $\gamma_1(c, p)$, it would follow that the condition p occurred twice in either $\gamma_0 \cdot M$ or $\gamma_1 \cdot M$, contradicting either N_0 or N_1 being a safe net.

The same cannot be said for occurrence nets. Taking the pullback in $\mathbf{PT}^\sharp$ of morphisms $(\eta_0, \beta_0){:}O_0 \to O'$ and $(\eta_1, \beta_1){:}O_1 \to O'$ for occurrence nets O_0, O_1 and O' does not necessarily yield an occurrence net. As a particular example, denote by $\{\emptyset\}$ the net with no conditions, no events and one initial marking that is empty. For any occurrence net O, denote by $(*, \emptyset){:}O \to \{\emptyset\}$ the morphism that is undefined on events and the empty relation on conditions The pullback drawn in the following diagram is not an occurrence net.



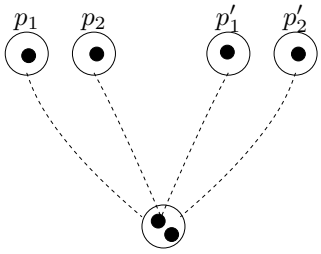
This is not to say that the category $\mathbf{Occ}^\sharp$ of occurrence nets does not have pullbacks. They arise from the coreflection between the categories $\mathbf{Occ}^\sharp$ and $\mathbf{PT}^\sharp$ shown in Theorem 7.15. Let Q, (π_0, γ_0), (π_1, γ_1) be the pullback in $\mathbf{PT}^\sharp$ of the morphisms $(\eta_0, \beta_0){:}O \to O'$ and $(\eta_1, \beta_1){:}O_1 \to O'$. Since right adjoints preserve all limits, and in particular pullbacks, the net $\mathcal{U}(Q)$ is a pullback in $\mathbf{Occ}^\sharp$ of (η_0, β_0) against (η_1, β_1) with pullback morphisms $(\pi_0, \gamma_0) \circ \varepsilon_Q$ and $(\pi_1, \gamma_1) \circ \varepsilon_Q$.



B.2 Pullbacks of general nets and folding morphisms

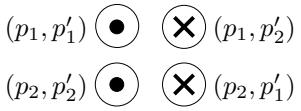
The construction of pullbacks in the previous section does not extend to giving pullbacks in the category $\mathbf{Gen}^\sharp$. The reason for this is that morphisms of general nets, even when they are relations rather than multirelations, need not be locally injective on initial markings or on the pre- or post-condition sets of events.

To see this, consider candidate pullbacks of the following morphisms:



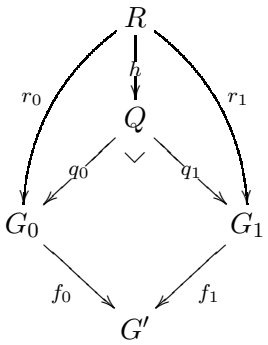
Let the morphism on the left be denoted $f_0{:}G_0 \to G'$ and the morphism on the right be denoted $f_1{:}G_1 \to G'$. Naïvely applying the construction for P/T nets would give a 'pullback' with just one condition which would be initially marked. This is straightforwardly seen not to be a pullback in the category $\mathbf{Gen}^\sharp$.

Let the net Q be

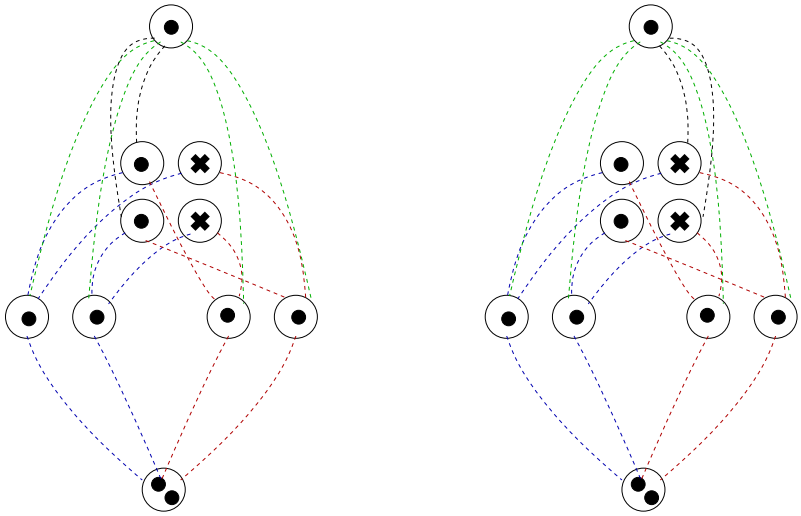


Notice that the net has two initial markings, namely $\{(p_1, p_1'), (p_2, p_2')\}$ and $\{(p_1, p_2'), (p_2, p_1')\}$. Define morphisms $q_0{:}Q \to G_0$ and $q_1{:}Q \to G_1$ as the obvious projections, so for example q_0 relates the condition (p_1, p_1') only to p_1 and q_1 relates (p_1, p_1') only to p_1'.

It can be argued straightforwardly that Q with q_0 and q_1 is a *weak* pullback of f_0 against f_1. That is, the square in the diagram below commutes, so $f_0 \circ q_0 = f_1 \circ q_1$, and for any general net R and morphisms $r_0{:}R \to G_0$ and $r_1{:}R \to G_1$ such that $f_0 \circ r_0 = f_1 \circ r_1$, there must exist a morphism $h{:}R \to Q$ such that $r_0 = q_0 \circ h$ and $r_1 = q_0 \circ h$.



The morphism h is *not* required to be unique; if it were, we would have shown that Q was a pullback. In fact, it is easy to see that Q is not a pullback of f_0 against f_1 since the morphism h need not be unique. This is easily seen in the following example:



We now show that there is *no* pullback of the two morphisms f_0 and f_1. For contradiction, suppose that there is such a pullback; call it Q' with pullback morphisms q_0' and q_1'. Since Q' is a pullback, there must exist a unique morphism $h{:}Q \to Q'$ such that $q_0 = q_0' \circ h$ and $q_1 = q_1' \circ h$. Considering Q, it must be the case that h is a function on conditions and injective; it is therefore monic in the category $\mathbf{Gen}^\sharp$. Taken with Q being a weak pullback, we may conclude that Q is also a pullback of f_0 against f_1. This contradicts the point made earlier, so there is no pullback of f_0 against f_1.

Folding morphisms

The absence of pullbacks in the category $\mathbf{Gen}^\sharp$ is problematic. Indeed, this led to the generalization of the framework for applying symmetry described earlier in this chapter. The situation may be resolved by observing that pullbacks may be obtained if we restrict attention to *folding* morphisms of Petri nets. Recall that a morphism $(\eta, \beta){:}G \to G'$ is a folding morphism if η and β are both total functions. As such, we shall sometimes denote folding morphisms using just one symbol rather than as pairs. The category of general nets with folding morphisms between them is written $\mathbf{Gen}^\sharp_{\mathrm{f}}$.

We now show that the category $\mathbf{Gen}^\sharp_{\mathrm{f}}$ has pullbacks. Let the following be general nets

$$\begin{aligned} G_0 &= (P_0, T_0, Pre_0, Post_0, \mathbb{M}_0) \\ G_1 &= (P_1, T_1, Pre_1, Post_1, \mathbb{M}_1) \\ G' &= (P', T', Pre', Post', \mathbb{M}') \end{aligned}$$

and suppose that there are folding morphisms $(\eta_0, \beta_0){:}G_0 \to G'$ and $(\eta_1, \beta_1){:}G_1 \to G$.

Candidate pullback

We now proceed to define the pullback object Q and morphisms $(\pi_0, \gamma_0){:}Q \to G_0$ and $(\pi_1, \gamma_1){:}Q \to G_1$. They will be obtained by stripping away all the isolated conditions from the net $Q_0 = (P_Q, T_Q, Pre_Q, Post_Q, \mathbb{M}_Q)$, defined in the following way:

Conditions The set of conditions P_Q is defined as:

$$P_Q \triangleq \{(p_0, p_1) \mid p_0 \in P_0 \ \&\ p_1 \in P_1 \ \&\ \beta_0(p_0) = \beta_1(p_1)\}$$

The pullback morphisms act on places in the following way:

$$\gamma_0(p_0, p_1) = p_0 \qquad \gamma_1(p_0, p_1) = p_1$$

Events The set of events of Q is defined to be:

$$\begin{aligned} T_Q \triangleq \{(C, t_0, t_1, D) \mid\ & C \subseteq_\mu P_Q \ \&\ D \subseteq_{\mu_\infty} P_Q \ \&\ \eta_0(t_0) = \eta_1(t_1) \\ & \&\ \gamma_0 \cdot C = Pre_0 \cdot t_0 \ \&\ \gamma_1 \cdot C = Pre_1 \cdot t_1 \\ & \&\ \gamma_0 \cdot D = Post_0 \cdot t_0 \ \&\ \gamma_1 \cdot D = Post_1 \cdot t_1 \ \} \end{aligned}$$

The folding morphisms act on transitions in the following way:

$$\pi_0(C, t_0, t_1, D) = t_0 \qquad \pi_1(C, t_0, t_1, D) = t_1$$

Preconditions

$$Pre_Q \cdot (C, t_0, t_1, D) = C$$

Postconditions

$$Post_Q \cdot (C, t_0, t_1, D) = D$$

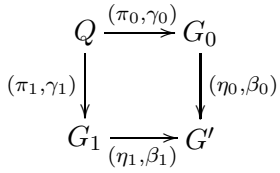
Initial markings

$$\mathbb{M}_Q = \{M \mid M \subseteq_{\mu_\infty} P_Q \ \&\ \gamma_0 \cdot M \in \mathbb{M}_0 \ \&\ \gamma_1 \cdot M \in \mathbb{M}_1\}$$

Universal property

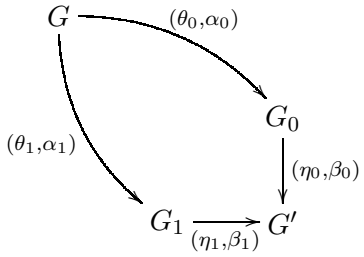
The first part of showing that Q, (π_0, γ_0) and (π_1, γ_1) form a pullback in $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$ is to show that they lie within the $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$ and that the morphisms commute with (η_0, β_0) and (η_1, β_1).

Lemma B.2.1. *The net Q is a general Petri net and, for both $i \in \{0, 1\}$, the morphism (π_i, γ_i):$Q \to G_i$ is a folding morphism. Furthermore, the following diagram commutes:*

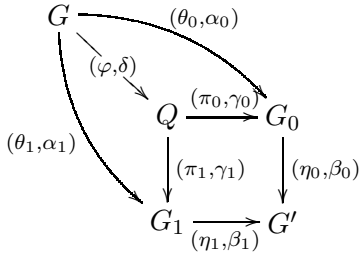


Proof. Easily checked to be immediate consequences of the definitions. □

We now progress to show the universality of the pullback in the category $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$. Suppose that there are morphisms (θ_0, α_0):$G \to G_0$ and (θ_1, α_1):$G \to G_1$ in $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$ making the following diagram commute:



We shall show that (φ, δ) as defined below is the unique morphism in $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$ that makes the two triangles in the following diagram commute:



Let the net $G = (P, T, F, \mathbb{M})$. For a condition $p \in P$, define

$$\delta(p) = (\theta_0(p), \theta_1(p)).$$

It is easy to see that this is a condition of Q from the commutation of the diagram above. In particular, the condition $(\theta_0(p), \theta_1(q))$ is non-isolated because the condition p is.

For an event $t \in T$, define

$$\varphi(t) = (C, \theta_0(t), \theta_1(t), D)$$

for multisets $C \subseteq_{\mu} P_Q$ and $D \subseteq_{\mu_\infty} P_Q$ defined as

$$\begin{aligned} C[(p_0, p_1)] &= \sum_{\{p \in P \mid \alpha_0(p) = p_0 \ \&\ \alpha_1(p) = p_1\}} Pre \cdot t[p] \\ D[(p_0, p_1)] &= \sum_{\{p \in P \mid \alpha_0(p) = p_0 \ \&\ \alpha_1(p) = p_1\}} Post \cdot t[p] \end{aligned}$$

Lemma B.2.2. *The functions $\varphi{:}T \to T_Q$ and $\delta{:}P \to P_Q$ are well-defined.*

Proof. As mentioned, it is clear that δ is a function and that $\delta(p) \in P_Q$ for any $p \in P$.

Let t be a transition of G, so $t \in T$. Let $\varphi(t) = (C, t_0, t_1, D)$. To see that $\varphi(t) \in T_Q$, we shall show the following:

1. C is a multiset rather than an ∞-multiset
2. $\eta_0(t_0) = \eta_1(t_1)$
3. $\gamma_0 \cdot C = Pre_0 \cdot t_0$

The remaining requirements, that for example $\gamma_1 \cdot D = Post_1 \cdot t_1$, are all similar to (3).

To see that C is a multiset, for contradiction suppose that there exists $(p_0, p_1) \in P_Q$ such that $C[(p_0, p_1)] = \infty$. Since G is a general net, $Pre \cdot t[p]$ is finite for all $p \in P$, so there must be infinitely many $p \in P$ such that $\alpha_0(p) = p_0$ and $Pre \cdot t[p] > 0$. However, then, since (θ_0, α_0) is a morphism, we would have $Pre_0 \cdot t_0[p_0] = \infty$, contradicting the requirements for G_0 to be a general net.

It is clear from the definition that $\eta_0(t_0) = \eta_1(t_1)$ since $t_0 = \theta_0(t)$ and $t_1 = \theta_1(t)$, and, from commutation, $\eta_0 \circ \theta_0 = \eta_1 \circ \theta_1$.

To show that $\gamma_0 \cdot C = Pre_0 \cdot t_0$, we shall show that $\gamma_0 \cdot C[p_0] = Pre_0 \cdot t_0[p_0]$ for all $p_0 \in P_0$. We have the following:

$$
\begin{array}{lll}
 & (\gamma_0 \cdot C)[p_0] & \\
= & \sum_{\{p_1 \in P_1 \mid \beta_0(p_0) = \beta_1(p_1)\}} C[(p_0, p_1)] & (1) \\
= & \sum_{\{p_1 \in P_1 \mid \beta_0(p_0) = \beta_1(p_1)\}} \left(\sum_{\{p \in P \mid \alpha_0(p) = p_0 \ \&\ \alpha_1(p) = p_1\}} (Pre \cdot t)[p] \right) & (2) \\
= & \sum_{\{p \in P \mid \exists p_1 \in P_1 (\beta_0(p_0) = \beta_1(p_1) \ \&\ \alpha_0(p) = p_0 \ \&\ \alpha_1(p) = p_1)\}} (Pre \cdot t)[p] & (3) \\
= & \sum_{\{p \in P \mid \alpha_0(p) = p_0\}} (Pre \cdot t)[p] & (4) \\
= & (\alpha_0 \cdot Pre \cdot t)[p_0] & (5) \\
= & (Pre_0 \cdot t_0)[p_0] & (6)
\end{array}
$$

Every equation apart from (4) is straightforward from the definitions. Equation (4) is an immediate consequence of the fact that $\beta_0 \circ \alpha_0 = \beta_1 \circ \alpha_1$. □

We now give the key lemma in showing that Q as defined above satisfies the universal property of being a pullback.

Lemma B.2.3. *The pair (φ, δ) is a folding morphism of general nets and, furthermore, is the unique morphism in $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$ such that $(\theta_0, \alpha_0) = (\pi_0, \gamma_0) \circ (\varphi, \delta)$ and $(\theta_1, \alpha_1) = (\pi_1, \gamma_1) \circ (\varphi, \delta)$.*

Proof. It is clear from the definitions that $(\theta_0, \alpha_0) = (\pi_0, \gamma_0) \circ (\varphi, \delta)$ and $(\theta_1, \alpha_1) = (\pi_1, \gamma_1) \circ (\varphi, \delta)$. Using this, it is easy to show that (φ, δ) is a morphism in the category $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$.

Let $(\varphi', \delta'){:}G \to Q$ be a morphism in the category $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$ such that

$$
\begin{array}{lll}
(\theta_0, \alpha_0) & = & (\pi_0, \gamma_0) \circ (\varphi', \delta') \quad (1) \\
(\theta_1, \alpha_1) & = & (\pi_1, \gamma_1) \circ (\varphi', \delta') \quad (2).
\end{array}
$$

We shall show that $(\varphi, \delta) = (\varphi', \delta')$.

It is easy to see that we must have $\delta = \delta'$. Let t be a transition in T and suppose that $\varphi(t) = (C, t_0, t_1, D)$. From the definition of $\varphi(t)$, we must have $\theta_0(t) = t_0$ and $\theta_1(t) = t_1$. From (1) and (2) above, we must have $\varphi'(t) = (C', t_0, t_1, D')$ for some C' and D'. We shall show that $C = C'$; the proof that $D = D'$ will be similar.

We already know that (φ, δ) is a morphism, so

$$\delta \cdot Pre \cdot t = Pre \cdot \varphi(t) = C.$$

The latter equality is from the definition of the preconditions of $\varphi(t)$. Since (φ', δ') is also a morphism, we have

$$\delta' \cdot Pre \cdot t = Pre \cdot \varphi'(t) = C'.$$

We saw earlier that $\delta = \delta'$, so it immediately follows that $C = C'$, thus completing the proof. □

It follows immediately that Q is the pullback.

Theorem B.2. *The net Q with morphisms $(\theta_0, \alpha_0){:}Q \to G_0$ and $(\theta_1, \alpha_1){:}Q \to G_1$ is a pullback in the category $\mathbf{Gen}^\sharp_f$ of the morphisms $(\eta_0, \beta_0){:}G_0 \to G'$ and $(\eta_1, \beta_1){:}G_1 \to G'$.* □

Pullbacks in other categories

To help understand the pullback of morphisms between P/T nets in the category $\mathbf{Gen}^\sharp_f$, we have the following lemma. This shows that the multisets C and D in an event (C, e_0, e_1, D) in the pullback of P/T nets are, in fact, sets, and are uniquely determined by the events.

Lemma B.2.4. *Let the pullback of folding morphisms $(\eta_0, \beta_0){:}G_0 \to G'$ and $(\eta_1, \beta_1){:}G_1 \to G'$ be as described above.*

$$\begin{array}{ccc} Q & \xrightarrow{(\pi_0,\gamma_0)} & G_0 \\ {\scriptstyle (\pi_1,\gamma_1)}\downarrow & \lrcorner & \downarrow{\scriptstyle (\eta_0,\beta_0)} \\ G_1 & \xrightarrow[(\eta_1,\beta_1)]{} & G' \end{array}$$

For any sets $X_0 \subseteq P_0$ and $X_1 \subseteq P_1$ such that $\beta_0 \cdot X_0 = \beta_1 \cdot X_1$, if β_0 is locally injective on X_0 and β_1 is locally injective on X_1 then there is a unique set $X \subseteq P_Q$ such that $\gamma_0 \cdot X = X_1$ and $\gamma_1 \cdot X = X_0$.

Proof. Let

$$X = \{(x_0, x_1) \mid \exists x \in \beta_0 \cdot X : \beta_0(x_0) = x \ \&\ \beta_1(x_1) = x\}.$$

It is straightforward to see that this satisfies the above constraints and, by local injectivity, is the unique such set. □

From this, it is easy to see that the net Q defined above is a P/T net if the nets G_0 and G_1 are P/T nets, so Q is also a pullback in the category $\mathbf{PT}^\sharp_f$. This result will be useful in showing that pullbacks are preserved through the inclusion $\mathbf{PT}^\sharp_f \hookrightarrow \mathbf{PT}^\sharp$.

If the nets G_0 and G_1 are safe nets then the net Q is also a safe net. To see this, suppose for contradiction that the net Q is not a safe net. We know that it is a P/T net, so there must exist a reachable marking M that is not a multiset. Since (θ_0, α_0) and (θ_1, α_1) are morphisms, the markings $\alpha_0 \cdot M$ and $\alpha_1 \cdot M$ are reachable in G_0 and G_1, respectively, according to Lemma 2.3.1 (which extends straightforwardly to morphisms in the category $\mathbf{Gen}^\sharp$). The maps are folding morphisms, so α_0 and α_1 are total on conditions, so neither $\alpha_0 \cdot M$ nor $\alpha_1 \cdot M$ are sets (they are multisets) and therefore neither G_0 nor G_1 are safe, contradicting the assumption. It follows that the construction also yields pullbacks in the category $\mathbf{Gen}^\sharp_f$.

We shall now show that the pullback Q is an occurrence net if G_0, G_1 and G' are occurrence nets, demonstrating that the construction also yields a pullback in the category $\mathbf{Occ}^\sharp_{\mathrm{f}}$. First, we show how conflict in the pullback gives rise to conflict in the nets G_0 and G_1. From this, it will follow immediately that the conflict relation on the pullback is irreflexive.

Lemma B.2.5. *Suppose that G_0, G_1 and G' are occurrence nets and that Q, (π_0, γ_0) and (π_1, γ_1) is the pullback of folding morphisms described above. For any $b, b' \in P_Q$ and $e, e' \in T_Q$, if $b \#_Q b'$ then either $\gamma_0(b) \#_0 \gamma_0(b')$ or $\gamma_1(b) \#_1 \gamma_1(b')$. If $e \#_Q e'$ then either $\pi_0(e) \# \pi_0(e')$ or $\pi_1(e) \# \pi_1(e')$.*

Proof. Let x and x be either a pair of conditions or a pair of events in Q such that $x \#_Q x'$. Let $q_0 = (\pi_0, \gamma_0) : Q \to G_0$ and $q_1 = (\pi_1, \gamma_1) : Q \to G_1$ be the folding morphisms from the pullback in $\mathbf{Gen}^\sharp$. We shall show, by induction on

$$\mathrm{depth}_0(q_0(x)) + \mathrm{depth}_1(q_1(x)) + \mathrm{depth}_0(q_0(x')) + \mathrm{depth}_1(q_1(x')),$$

that either $q_0(x) \#_0 q_0(x')$ or $q_1(x) \#_1 q_1(x')$.

The base case, where the sum is zero, is relatively simple: It is easy to see from the fact that G_0 and G_1 are occurrence nets that any two such elements of the net Q must be conditions $b = (b_0, b_1)$ and $b' = (b'_0, b'_1)$ that are in immediate conflict, *i.e.* there exist distinct initial markings M and M' of Q such that $b \in M$ and $b' \in M'$. From the definition of the initial markings of Q and Lemma B.2.4, it follows that either $\gamma_0 \cdot M \neq \gamma_0 \cdot M'$ or $\gamma_1 \cdot M \neq \gamma_1 \cdot M'$. In the first case (the second is similar), it follows that $b_0 \in \gamma_0 \cdot M$ and $b'_0 \in \gamma_0 \cdot M'$, and hence b_0 and b'_0 are in immediate conflict in G_0, as required.

The inductive case begins by considering two events that are in conflict in Q. Let the events be $e = (C, e_0, e_1, D)$ and $e' = (C', e'_0, e'_1, D')$. There are two ways in which e and e' might be in conflict in Q. First, the event e might have a precondition $b = (b_0, b_1)$ and e' might have a precondition $b' = (b'_0, b'_1)$ such that b is in conflict with b'. From the induction hypothesis, b_0 is in conflict with b'_0 (or, symmetrically, b_1 is in conflict with b'_1). Since b_0 is a precondition of e_0 and b'_0 is a precondition of e'_0, the result follows. The second way in which e and e' might be in conflict is for e and e' to be distinct, and hence from the definition of the events of Q and Lemma B.2.4 either $e_0 \neq e'_0$ or $e_1 \neq e'_1$, and for them to share a precondition. If $e_0 \neq e'_0$, it follows straightforwardly that e_0 and e_0 share a common precondition, and hence they are in conflict, and if $e_1 \neq e'_1$ the case is symmetric.

The second part of the inductive case involves considering two conditions, $b = (b_0, b_1)$ and $b' = (b'_0, b'_1)$, that are in conflict. Without loss of generality, suppose that $\mathrm{depth}_0(q_0(b)) > 0$. There must exist an event e_0 such that $b_0 \in e_0{}^\bullet$. Since G_0, G_1 and G' are occurrence nets and $\beta_0(b_0) = \beta_1(b_1)$, we can see that there exists an event e_1 such that $\eta_1(e_1) = \eta_0(e_0)$ and $b_1 \in e_1{}^\bullet$. From Lemma B.2.4, there exist unique C and D such that $(C, e_0, e_1, D) \in T_Q$. Additionally, according to the lemma, we have $b \in D = (C, e_0, e_1, D)^\bullet$ and since G_0 and G_1 are occurrence nets this is the unique event of which b is a postcondition and b is not in any initial marking. It follows that b and b' are not in immediate conflict. There must therefore exist a condition $b'' \in {}^\bullet(C, e_0, e_1, D)$ such that either $b'' \#_Q b'$ or there exists an event $(C', e'_0, e'_1, D') \in T_Q$ not equal to (C, e_0, e_1, D) such that $b' \in (C', e'_0, e'_1, D')^\bullet$ and $b'' \in {}^\bullet(C', e'_0, e'_1, D')$. Let $b'' = (b''_0, b''_1)$. In the first case, we have $b''_0 \#_0 b'_0$ by induction (or, symmetrically, $b''_1 \#_1 b'_1$) and hence $b_0 \# b''_0$ since $b''_0 \in {}^\bullet e_0$ and $b_0 \in e_0{}^\bullet$. Turning to the second case, since (C, e_0, e_1, D) and (C', e'_0, e'_1, D') are distinct events in T_Q, according to the definition of the events of Q and Lemma B.2.4, we must have either $e_0 \neq e'_0$ or $e_1 \neq e'_1$. Suppose that $e_0 \neq e'_0$; the argument is similar if $e_1 \neq e'_1$. We have $b''_0 \in {}^\bullet e_0 \cap {}^\bullet e'_0$, so $e_0 (\#_m)_0 e'_0$. It follows immediately that $b_0 \# b'_0$, as required. ∎

We now proceed to show that the pullback of occurrence nets in $\mathbf{Gen}_{\mathrm{f}}^{\sharp}$ is an occurrence net.

Lemma B.2.6. *If the nets G_0, G_1 and G' are occurrence nets then the pullback of folding morphisms Q described above is also an occurrence net.*

The net Q is a safe net since the nets G_0 and G_1 are safe. We must show the following:

Proof. $\forall M \in \mathbb{M}_Q : \forall b \in M : ({}^\bullet b = \emptyset)$: Suppose, for contradiction, that there exists $e \in T_Q$ such that $b \in e^\bullet$. It follows from (π_0, γ_0) being a morphism that $\gamma_0(b) \in \gamma_0 \cdot M$ and $\gamma_0(b) \in \pi_0(e)^\bullet$. Recalling that $\gamma_0 \cdot M \in \mathbb{M}_0$ since (π_0, γ_0) is a morphism, we arrive at the desired contradiction since then G_0 would not be an occurrence net.

$\forall b \in P_Q : \exists M \in \mathbb{M}_Q : \exists p \in M : (p\ F^*\ b)$: Let $b = (b_0, b_1)$, so $\beta_0(b_0) = \beta_1(b_1)$. We shall show that b is reachable from a condition in some initial marking by induction on the depth of b_0.

The base case has $b_0 \in M_0$ for some $M_0 \in \mathbb{M}_0$. We have $b_1 \in M_1$ for some $M_1 \in \mathbb{M}_1$; otherwise, there would have to exist an event e_1 such that $b_1 \in {e_1}^\bullet$ since G_1 is an occurrence net, but then $\beta_1(b_1) \in \eta_1(e_1)^\bullet$ and $\beta_1(b_1) \in \gamma_0 \cdot M_0$, contradicting G' being an occurrence net. Let $b' \in P'$ be the condition such that $\beta_0(b_0) = b'$. Recall that, since $(b_0, b_1) \in P_Q$, we have $\beta_1(b_0) = b'$. Since $b' \in \beta_0 \cdot M_0$ and $b' \in \beta_1 \cdot M_1$ because (η_0, β_0) and (η_1, β_1) are morphisms and therefore preserve initial markings, we must have $\beta_0 \cdot M_0 = \beta_1 \cdot M_1$ since no condition of the occurrence net G' can occur in more than one initial marking. It is now straightforward to see, from the fact that the relations β_0 and β_1 are locally injective on the initial markings M_0 and M_1 that there is a marking of the pullback M such that $(b_0, b_1) \in M$.

We now consider the inductive case. There exists a unique event $e_0 \in T_0$ such that $b_0 \in {e_0}^\bullet$, and similarly there exists a unique event $e_1 \in T_1$ such that $b_1 \in {e_1}^\bullet$. Since we have $\beta_0(b_0) = b' = \beta_1(b_1)$ and G' is an occurrence net, we must have $\eta_0(e_0) = \eta_1(e_1)$. The net G_0 is an occurrence net, so e_0 must have a precondition; call this b'_0. Since (η_1, β_1) is a morphism, there exists a condition $b'_1 \in {}^\bullet e_1$ such that $\beta_0(b'_0) = \beta_1(b'_1)$. From the local injectivity of the β_0 and β_1 on the pre- and postconditions of e_0 and e_1, it is straightforward to show that there exist (unique) sets $C \subseteq P_Q$ and $D \subseteq P_Q$ such that $(C, e_0, e_1, D) \in T_Q$ and $(b'_0, b'_1) \in C$ and $(b_0, b_1) \in D$. From the induction hypothesis, since b'_0 is at lower depth in G_0 than b_0, there exists a condition $p \in P_Q$ and initial marking $M \in \mathbb{M}_Q$ such that $p\ {F_Q}^*\ (b'_0, b'_1)$. Hence $p\ {F_Q}^*\ (b'_0, b'_1)$, as required.

$\forall b \in P_Q : (|{}^\bullet b| \leq 1)$: Suppose, for contradiction, that there exists $b \in P_Q$ and two distinct events $(C, e_0, e_1, D), (C', e'_0, e'_1, D') \in T_Q$ with $b \in D \cap D'$. According to the definition of T_Q and Lemma B.2.4, since the events are distinct we must have either $e_0 \neq e'_0$ or $e_1 \neq e'_1$. Suppose that $e_0 \neq e'_0$; the other case is similar. Let $b = (b_0, b_1)$. It follows from (π_0, γ_0) being a morphism that $b_0 \in {e_0}^\bullet \cap {e'_0}^\bullet$, thus contradicting the net G_0 being an occurrence net.

F^+ is irreflexive and for all $e \in E$, the set $\{e' \mid e'\ F^*\ e\}$ is finite: Let the morphism (π_0, γ_0) be written q_0, a function from the conditions and events of Q to G_0. A straightforward induction on n shows that, for any $x, y \in P_0 \cup T_0$, if $x\ F^n\ y$ then $q_0(x)\ {F_0}^n\ q_o(y)$. It follows immediately from the net G_0 being an occurrence net that the flow relation in Q is irreflexive.

Suppose that $e'\ F^+\ e$, and therefore $q_0(e')\ F^+\ q_0(e)$. It follows from the flow relation F_0 in G_0 being irreflexive that q_0 is injective on the set $\{e' \mid e'\ F^*\ e\}$. Consequently, since q_0 is total on events, if the set $\{e' \mid e'\ F^*\ e\}$ were infinite, the set $\{e'_0 \mid e'_0\ {F_0}^*\ q_0(e)\}$ would be infinite, contradicting G_0 being an occurrence net.

is irreflexive: The result follows immediately from Lemma B.2.5. □

Weak pullbacks

We have seen that the category $\mathbf{Gen}^\sharp$ does not have pullbacks. We were forced to restrict to the category $\mathbf{Gen}^\sharp_{\mathrm{f}}$ of general nets with folding morphisms to remedy this. The example above shows that the inclusion $\mathbf{Gen}^\sharp_{\mathrm{f}} \hookrightarrow \mathbf{Gen}^\sharp$ does not preserve pullbacks, and that the category of general nets does not have pullbacks of folding morphisms.

The inclusion $\mathbf{Gen}^\sharp_{\mathrm{f}} \hookrightarrow \mathbf{Gen}^\sharp$ does, however, embed any pullback in the category $\mathbf{Gen}^\sharp_{\mathrm{f}} \hookrightarrow \mathbf{Gen}^\sharp$ as a *weak* pullback in the category $\mathbf{Gen}^\sharp$. Weak pullbacks were described on Page 157. This is equivalent to stating that the inclusion preserves weak pullbacks.

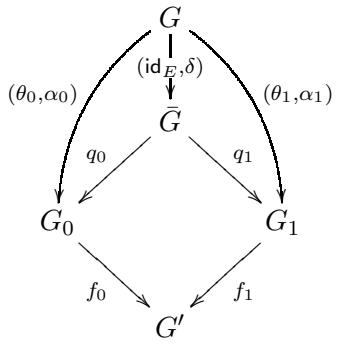
Lemma B.2.7. *The inclusion* $\mathbf{Gen}^\sharp_{\mathrm{f}} \hookrightarrow \mathbf{Gen}^\sharp$ *preserves weak pullbacks.*

Proof. We first recall a general fact from category theory. Let $\mathcal{C}$ be a category with pullbacks that is a subcategory of $\mathcal{D}$. The following two statements are equivalent:

- Any weak pullback in $\mathcal{C}$ is a weak pullback in $\mathcal{D}$.
- Any pullback in $\mathcal{C}$ is a weak pullback in $\mathcal{D}$.

We shall show that the pullback Q, $(\pi_0, \gamma_0){:}Q \to G_0$ and $(\pi_1, \gamma_1){:}Q \to G_1$ in the category $\mathbf{Gen}^\sharp_{\mathrm{f}}$ of the folding morphisms $f_0 = (\eta_0, \beta_0){:}G_0 \to G'$ and $f_1 = (\eta_1, \beta_1){:}G_1 \to G'$ is a weak pullback in the category $\mathbf{Gen}^\sharp$.

Let $G = (B, E, F, \mathbb{M})$ be a general net and $(\theta_0, \alpha_0){:}G \to G_0$ and $(\theta_1, \alpha_1){:}G \to G_1$ be any morphisms in $\mathbf{Gen}^\sharp$ such that $f_0 \circ (\theta_0, \alpha_0) = f_1 \circ (\theta_1, \alpha_1)$. To show that the pullback of folding morphisms is a weak pullback in the category $\mathbf{Gen}^\sharp$, we shall define a net $\bar{G} = (\bar{B}, E, \bar{F}, \bar{\mathbb{M}})$, a (relational) morphism of general nets $(\mathsf{id}_E, \delta){:}G \to \bar{G}$ and folding morphisms $q_0{:}\bar{G} \to G_0$ and $q_1{:}\bar{G} \to G_1$ such that the two triangles and the square in the following diagram commute:



It then follows immediately from Q, (π_0, γ_0) and (π_1, γ_1) being a pullback of folding morphisms that it is a weak pullback in the category of general nets.

Recall that $G_0 = (P_0, T_0, F_0, \mathbb{M}_0)$, $G_1 = (P_1, T_1, F_1, \mathbb{M}_1)$ and $G = (B, E, F, \mathbb{M})$. The net $\bar{G}$ has conditions

$$\bar{B} \quad = \quad \{(b, i_0, p_0) \mid b \in B\ \&\ p_0 \in P_0\ \&\ 0 \leq i_0 < \alpha_0[b, p_0]\},$$

the same events as G, and flow relation

$$\bar{F}[e,(b,i_0,p_0)] = F[e,b] \qquad \bar{F}[(b,i_0,p_0),e] = F[b,e].$$

We define $\bar{\mathbb{M}}$ to be the least set containing the marking $\bar{M}$ for any marking $M \in \mathbb{M}$, where $\bar{M}$ is defined as

$$\bar{M}[(b,i_0,p_0)] = M[b].$$

We define the morphism (id_E, δ):$G \to \bar{R}$ as

$$\delta(b,(b',i_0,p_0)) \iff b = b'.$$

The morphism q_0:$\bar{G} \to G_0$ is straightforwardly defined as

$$q_0(e) = \theta_0(e) \qquad q_0(b,i_0,p_0) = p_0.$$

It is easy to see that this is a morphism such that the upper-left triangle commutes. Since, by assumption, the outer square commutes, *i.e.* $f_0 \circ (\theta_0, \alpha_0) = f_1 \circ (\theta_1, \alpha_1)$, for each $b \in B$ and $p \in P$, where P is the set of conditions of G', there is a bijection

$$\begin{array}{rcl} \theta_{b,p} & : & \{(p_0,i_0) \mid f_0(p_0) = p \;\&\; 0 \le i_0 < \alpha_0[b,p_0]\} \\ & \cong & \{(p_1,i_1) \mid f_1(p_1) = p \;\&\; 0 \le i_1 < \alpha_1[b,p_1]\} \end{array}$$

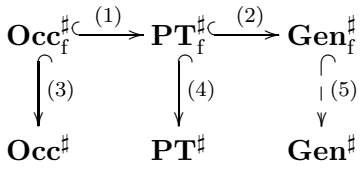
From this bijection, it is easy to see that q_1:$\bar{R} \to G_1$ defined as

$$q_1(e) = \theta_1(e) \qquad q_1(b,i_0,p_0) = p_1 \text{ for } (p_1,i_1) = \theta_{b,f_0(p_0)}(p_0,i_0)$$

make the upper-right triangle and the square commute, and from this that q_1 is a morphism. □

We conclude this section by summarizing how the inclusions of the various categories of net preserve pullbacks.

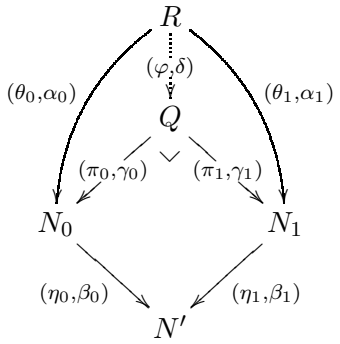
Lemma B.2.8. *The solid inclusion functors in the following diagram preserve pullbacks and the dashed inclusion functor preserves weak pullbacks.*



Proof. We have just shown that inclusion (5) preserves weak pullbacks. We remarked following Lemma B.2.4 that (2) preserves pullbacks. It follows immediately from Lemma B.2.6 that the inclusion $\mathbf{Occ}_f^\sharp \hookrightarrow \mathbf{Gen}_f^\sharp$ preserves pullbacks, and hence that (1) preserves pullbacks.

It is now sufficient to show that (4) preserves pullbacks. It follows from (1) and (4) preserving pullbacks that the pullback in $\mathbf{PT}^\sharp$ of folding morphisms between occurrence nets is an occurrence net, and therefore that (3) preserves pullbacks.

To see that (4) preserves pullbacks, we return to considering the pullback Q in the category $\mathbf{PT}^\sharp$ defined on page 205.



Recalling the definition of the conditions of Q on page 206, any condition in Q is a $\sim_{X_0}^{X_1}$-equivalence class

$$c = [p]_{X_0}^{X_1}$$

for a condition p of R and sets X_0 and X_1 that are either initial markings of N_0 and N_1 with the same image under β_0 and β_1, respectively, or the pre- or post-sets of events with the same image under η_0 and η_1, respectively. It is easy to see, using Lemma B.2.4, that if (η_0, β_0) and (η_1, β_1) are folding morphisms then each class contains precisely two conditions, one from N_0 and one from N_1. Hence the relations γ_0 and γ_1 are, in fact, total functions. Similarly, π_0 and π_1 are total on events and hence (π_0, γ_0) and (π_1, γ_1) are folding morphisms.

If the net R drawn above is a P/T net and (θ_0, α_0) and (θ_1, α_1) drawn above are folding morphisms, it is easy to see from the definition on page 207 that (φ, δ) is also a folding morphism.

From the morphisms (π_0, γ_0), (π_1, γ_1) and (φ, δ) being folding morphisms, the pullback of folding morphisms taken in $\mathbf{PT}^\sharp$ is also a pullback in $\mathbf{PT}^\sharp_{\mathrm{f}}$. Pullbacks in any category are uniquely defined up to isomorphism, so any pullback in $\mathbf{PT}^\sharp_{\mathrm{f}}$ is also a pullback in $\mathbf{PT}^\sharp$, as required to complete the proof. □

Note that the inclusion functor $\mathbf{Occ}^\sharp \hookrightarrow \mathbf{PT}^\sharp$ does not preserve pullbacks, as we showed on page 209

Appendix C

Categories of families

Definition C.0.1 (Family)**.** *Let $\mathcal{C}$ be any category. A* family *of $\mathcal{C}$, written $(X_i)_{i\in I}$ is an indexing set I and a function associating each element of $i \in I$ with an element X_i of $\mathcal{C}$. A morphism between families $f{:}(X_i)_{i\in I} \to (Y_j)_{j\in J}$ is a function $\hat{f}{:}I \to J$ and, for each $i \in I$, a morphism $f_i{:}X_i \to Y_{\hat{f}(i)}$ in $\mathcal{C}$.*

We write $\mathcal{F}am(\mathcal{C})$ for the category of families of $\mathcal{C}$, with the obvious identities and composition of morphisms.

Proposition C.1. *Let $F{:}\mathcal{C} \to \mathcal{D}$ be a functor. Then $\mathcal{F}am(F)$, defined as follows, is a functor, $\mathcal{F}am(F){:}\mathcal{F}am(\mathcal{C}) \to \mathcal{F}am(\mathcal{D})$.*

- *For any family $(X_i)_{i\in I}$ in $\mathcal{F}am(\mathcal{C})$*

$$\mathcal{F}am(F)(X_i)_{i\in I} = (F(X_i))_{i\in I}.$$

- *Let $f{:}(X_i)_{i\in I} \to (Y_j)_{j\in J}$ be a morphism in $\mathcal{F}am(\mathcal{C})$. Recall that*

$$f = (\hat{f}{:}I \to J, (f_i{:}X_i \to Y_{\hat{f}(i)})_{i\in I}).$$

  Define

$$\mathcal{F}am(F)(f) = (\hat{f}, (F(f_i))_{i\in I}).$$

□

It is now easy to see that $\mathcal{F}am$ is an endofunctor on the category of categories since it preserves composition of functors and preserves identities.

Coreflections between categories lift to their categories of families.

Proposition C.2. *Let $\mathcal{C}$ and $\mathcal{D}$ be any categories related through an adjunction $F \dashv G$, so $F{:}\mathcal{C} \to \mathcal{D}$ is left adjoint to $G{:}\mathcal{D} \to \mathcal{C}$. The functor $\mathcal{F}am(F){:}\mathcal{F}am(\mathcal{C}) \to \mathcal{F}am(\mathcal{D})$ is left adjoint to the functor $\mathcal{F}am(G){:}\mathcal{F}am(\mathcal{D}) \to \mathcal{F}am(\mathcal{C})$. Furthermore, if $F \dashv G$ is a coreflection then so is $\mathcal{F}am(F) \dashv \mathcal{F}am(G)$.*

Proof. We are given an isomorphism of hom-sets

$$\varphi_{X,Y}{:}\mathcal{C}(X, GY) \cong \mathcal{D}(FX, Y),$$

natural in X and Y. For any family $(X_i)_{i\in I}$ in $\mathcal{F}am(\mathcal{C})$ and any family $(Y_j)_{j\in J}$, we wish to construct an isomorphism of hom-sets

$$\psi_{(X_i)_{i\in I},(Y_j)_{j\in J}}{:}\mathcal{F}am(\mathcal{C})((X_i)_{i\in I}, (GY_j)_{j\in J}) \cong \mathcal{F}am(\mathcal{D})((FX_i)_{i\in I}, (Y_j)_{j\in J}),$$

natural in the families $(X_i)_{i\in I}$ and $(Y_j)_{j\in J}$.

Let $f = (\hat{f}, (f_i)_{i\in I}):(X_i)_{i\in I} \to (GY_j)_{j\in J}$ be a morphism in $\mathcal{F}am(\mathcal{C})$. Define

$$\psi_{(X_i)_{i\in I},(Y_j)_{j\in J}}(f) = (\hat{f}, (\varphi_{X_i,Y_{\hat{f}(i)}}(f_i))_{i\in I}).$$

Let $g = (\hat{g}, (g_j)_{j\in J}):(FX_i)_{i\in I} \to (Y_j)_{j\in J}$ in $\mathcal{F}am(\mathcal{D})$. Define

$$\psi^{-1}_{(X_i)_{i\in I},(Y_j)_{j\in J}}(g) = (\hat{g}, (\varphi^{-1}_{X_{\hat{g}(j)},Y_j}(g_j))_{j\in J}).$$

It follows immediately from this that $\psi_{(X_i)_{i\in I},(Y_j)_{j\in J}}$ is a bijection. To see that this bijection is natural in $(X_i)_{i\in I}$ (naturality in $(Y_j)_{j\in J}$ will be similar), for any morphism $f:(X_i)_{i\in I} \to (X'_i)_{i\in I'}$ in $\mathcal{F}am(\mathcal{C})$ we must show that the following diagram commutes:

$$\begin{array}{ccc}
\mathcal{F}am(\mathcal{C})((X_i)_{i\in I}, (GY_j)_{j\in J}) & \xrightarrow{\psi_{(X_i)_{i\in I},(Y_j)_{j\in J}}} & \mathcal{F}am(\mathcal{D})((FX_i)_{i\in I}, (Y_j)_{j\in J}) \\
\uparrow{\scriptstyle -\circ f} & & {\scriptstyle -\circ \mathcal{F}am(F)(f)}\uparrow \\
\mathcal{F}am(\mathcal{C})((X'_i)_{i\in I'}, (GY_j)_{j\in J}) & \xrightarrow[\psi_{(X'_i)_{i\in I'},(Y_j)_{j\in J}}]{} & \mathcal{F}am(\mathcal{D})((FX'_i)_{i\in I'}, (Y_j)_{j\in J})
\end{array}$$

Let $h:(X'_i)_{i\in I'} \to (Y_j)_{j\in J}$ be any morphism in $\mathcal{F}am(\mathcal{C})((X_i)_{i\in I}, (GY_j)_{j\in J})$. Since f and h are morphisms of families,

$$\begin{aligned} h &= (\hat{h}{:}I' \to J, (h_i{:}X'_i \to Y_{\hat{h}(i)})_{i\in I'}) \\ f &= (\hat{f}{:}I \to I', (f_i{:}X_i \to X_{\hat{f}(i)})_{i\in I}). \end{aligned}$$

From the naturality of φ, the following diagram commutes for any $i \in I$:

$$\begin{array}{ccc}
X_i & \mathcal{C}(X_i, GY_{\hat{h}\hat{f}(i)}) & \xrightarrow{\varphi_{X_i,Y_{\hat{h}\hat{f}(i)}}} \mathcal{D}(FX_i, Y_{\hat{h}\hat{f}(i)}) \\
\downarrow{\scriptstyle f_i} & \uparrow{\scriptstyle -\circ f_i} & {\scriptstyle Ff_i}\uparrow \\
X'_{\hat{f}(i)} & \mathcal{C}(X'_{\hat{f}(i)}, GY_{\hat{h}\hat{f}(i)}) & \xrightarrow[\varphi_{X'_{\hat{f}(i)},Y_{\hat{h}\hat{f}(i)}}]{} \mathcal{D}(FX'_{\hat{f}(i)}, Y_{\hat{h}\hat{f}(i)})
\end{array}$$

We therefore have the following equations, as required to show commutation of the naturality diagram for ψ above:

$$\begin{aligned}
& \psi_{(X'_i)_{i\in I'},(Y_j)_{j\in J}}(h) \circ \mathcal{F}am(F)(f) \\
\overset{(1)}{=}\ & (\hat{h}, (\varphi_{X'_i,Y_{\hat{h}(i)}}(h_i))_{i\in I'}) \circ \mathcal{F}am(F)(f) \\
\overset{(2)}{=}\ & (\hat{h}\circ\hat{f}, (\varphi_{X'_{\hat{f}(i)},Y_{\hat{h}\hat{f}(i)}}(h_{\hat{f}(i)}) \circ F(f_i))_{i\in I}) \\
\overset{(3)}{=}\ & (\hat{h}\circ\hat{f}, (\varphi_{X_i,Y_{\hat{h}\hat{f}(i)}}(h_{\hat{f}(i)} \circ f_i))_{i\in I}) \\
\overset{(4)}{=}\ & \psi_{(X_i)_{i\in I},(Y_j)_{j\in J}}(h \circ f)
\end{aligned}$$

Equations (1) and (4) follow from the definition of ψ. Equation (2) follows from the definition of $\mathcal{F}am(F)$ and the definition of composition of morphisms of families. Equation (3) follows from the naturality diagram for φ drawn above.

To complete the proof, it can be shown straightforwardly that if F is full and faithful then $\mathcal{F}am(F)$ is full and faithful, so $\mathcal{F}am$ preserves coreflections. □

C.1 Indexed products and coproducts of sets

Let $(X_i)_{i\in I}$ be an I-indexed family of sets. We denote by $\prod_{i\in I} X_i$ the indexed product of the family $(X_i)_{i\in I}$, so an element of $\prod_{i\in I} X_i$ associates to each element i of I an element of the set X_i. It is therefore a family $(x_i)_{i\in I}$ such that $x_i \in X_i$ for all $i \in I$.

Let $(X_i)_{i\in I}$ and $(Y_j)_{j\in J}$ be indexed families of sets. Given a function $\hat{f}{:}I \to J$ and a family of functions $(f_i{:}Y_{\hat{f}(i)} \to X_i)_{i\in I}$, we obtain a morphism

$$\prod_{i\in I} f_i{:}\prod_{i\in I} X_i \to \prod_{j\in J} Y_j.$$

by taking a J-indexed set $(y_j)_{j\in J}$ to the I-indexed set $(f_i(y_{\hat{f}(i)}))_{i\in I}$.

The indexed coproduct of an I-indexed family of sets $(X_i)_{i\in I}$ is denoted $\coprod_{i\in I} X_i$. It is a set, and its elements are of the form $\mathsf{in}_i x$ for some $i \in I$ and $x \in X_i$. As for the product, let $(X_i)_{i\in I}$ and $(Y_j)_{j\in J}$ be indexed families of sets. Given a function $\hat{f}{:}I \to J$ and a family of functions $(f_i{:}X_i \to Y_{\hat{f}(i)})_{i\in I}$, for the coproduct we obtain a morphism

$$\coprod_{i\in I} f_i{:}\coprod_{i\in I} X_i \to \coprod_{j\in J} Y_j$$

which takes $\mathsf{in}_i x$ to $\mathsf{in}_{\hat{f}(i)} f_i(x)$.

Proposition C.3. *For any (locally small) category $\mathcal{C}$ and families $(X_i)_{i\in I}$ and $(Y_j)_{j\in J}$ of objects in $\mathcal{C}$ there is an isomorphism*

$$\prod_{i\in I}\coprod_{j\in J} \mathcal{C}(X_i, Y_j) \cong \mathcal{F}am(\mathcal{C})((X_i)_{i\in I}, (Y_j)_{j\in J})$$

natural in the families $(X_i)_{i\in I}$ and $(Y_j)_{j\in J}$.

Proof. A standard calculation. □

Appendix D

Open maps of Petri nets

In Section 7.4, we saw how causal nets represent paths of general nets according to the individual token game. We shall now describe the form of open map bisimulation ensuing from taking paths of general nets to be causal nets.

We shall begin by showing that restricting attention to folding maps does not affect openness, and we shall show that all **Caus**-open maps are foldings apart from on conditions and events that cannot occur finitely in any reachable marking. We shall then give a characterization of when a morphism from an occurrence net to a general net is **Caus**-open. This key result will allow us to show that the morphism from the unfolding of a general net back to the original general net is open and therefore forms a basis for defining symmetry on the unfolding.

As a preliminary result, note that the inclusion of occurrence nets into general nets preserves **Caus**-openness.

Lemma D.0.1. *Let $(\eta, \beta){:}O \to O'$ be a morphism in $\mathbf{Occ}^\sharp$. The morphism (η, β) is* **Caus***-open in $\mathbf{Gen}^\sharp$ iff it is* **Caus***-open in $\mathbf{Occ}^\sharp$.*

Proof. Follows directly from $\mathbf{Occ}^\sharp$ being a *full* subcategory of $\mathbf{Gen}^\sharp$ — *cf.* Proposition 5 of [WN95]. □

We now show that any **Caus**-open map in the category $\mathbf{Gen}^\sharp$ is a folding map, apart from on conditions that either can never be marked or that can only ever be infinitely marked. In this appendix, we continue to use the notation described in Definition 7.4.2, writing $P(G)$ for the places of a general net G, and so on.

Lemma D.0.2. *Let $(\eta, \beta){:}G \to G'$ be a* **Caus***-open morphism in $\mathbf{Gen}^\sharp$. If e is an event that can occur in some reachable marking of G then $\eta(e) \neq *$. For any place $p \in P(G)$:*

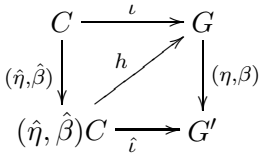
- *if there exists a reachable marking M such that $M[p] > 0$ then there exists a unique place $p' \in P(G')$ such that $\beta[p, p'] > 0$, and*
- *if there exists a reachable marking M such that $0 < M[p] < \infty$ then there exists $p' \in P(G')$ such that $\beta[p, p'] = 1$.*

Hence any morphism from an occurrence net $(\eta, \beta){:}O \to G'$ in $\mathbf{Gen}^\sharp$ that is **Caus***-open is a folding morphism.*

Proof. Since $Pre \cdot t$ is a non-empty multiset according to the definition of general net, it is sufficient just to consider the requirement on places.

Let p be a condition of G for which there is a reachable marking M such that $M[p] > 0$. By Lemma 7.4.2, there is a finite causal net C and folding morphism $\iota{:}C \to G$ such that

$\iota \cdot \mathrm{mkg}(C) = M$. From Lemma 7.4.3 and openness, there exist morphisms making the following diagram commute:



Furthermore, $\hat{\iota}$ is a folding morphism.

For the first part, suppose for contradiction that $\beta[p, p'] = 0$ for all $p' \in P(G')$. Since $\iota \cdot \mathrm{mkg}(C) = M$, there exists $b \in \mathrm{mkg}(C)$ such that $\iota(b) = p$. However, according to the definition of $(\hat{\eta}, \hat{\beta})$, there is no b' such that $\hat{\beta}(b, b')$, contradicting the commutation of the upper triangle.

Now suppose that there exist distinct $p'_1, p'_2 \in P(G')$ such that $\beta[p, p'_1] > 0$ and $\beta[p, p'_2] > 0$. Again, since $\iota \cdot \mathrm{mkg}(C) = M$, there exists $b \in \mathrm{mkg}(C)$ such that $\iota(b) = p$. From commutation of the upper triangle, there exists $b' \in P((\hat{\eta}, \hat{\beta})C)$ such that $\hat{\beta}(b, b')$ and $h(b', p)$. Since ι is a folding morphism, there is no $p' \neq p$ such that $h(b', p')$. It is therefore impossible for the lower triangle to commute since $\hat{\iota}$ is a folding morphism.

For the second part, assume further that $M[p] < \infty$. For contradiction, suppose that $\sum_{p' \in P(G')} \beta[p, p'] > 1$. Let X be the set $\{b \in \mathrm{mkg}(C) \mid \iota(b) = p\}$. We have $|X| = M[p]$ because $\iota \cdot \mathrm{mkg}(C) = M$. It follows from the definition of $\hat{\beta}$ that $|\hat{\beta}X| > M[p]$. However, since ι and $\hat{\iota}$ are foldings, it can be seen that we must have $h_{\mathsf{c}}(b', p)$ for all $b' \in \hat{\beta}X$. It follows that neither the upper nor the lower triangles commute, giving us the required contradiction. □

From this, it follows immediately that any open morphism from an occurrence net must be a folding since every event can occur in some reachable marking of an occurrence net and since every condition of an occurrence net occurs in some reachable marking (ensuring that for any condition p there exists a reachable marking M such that $M[p] > 0$) and all occurrence nets are safe (ensuring that the marking M satisfies $M[p] < \infty$). It can also be seen that if a place in a P/T net can become marked, there also exists a marking in which it is finitely marked. Hence any open morphism from a P/T net is a folding apart from on conditions that never become marked or on events that can never occur.

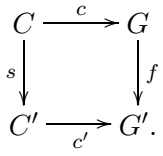
So far, we have considered **Caus**-open maps in the categories $\mathbf{Occ}^\sharp$ and $\mathbf{Gen}^\sharp$. The general framework for defining symmetry on general nets will involve us restricting to folding maps between general nets. As we have seen, many **Caus**-open maps between general nets are foldings, for example all maps from occurrence nets. The framework will, however, involve us considering $\mathbf{Caus}_{\mathrm{f}}$-openness in the category $\mathbf{Gen}^\sharp_{\mathrm{f}}$. We now show that this has no effect on the bisimulations obtained: Openness of folding morphisms in the category $\mathbf{Gen}^\sharp_{\mathrm{f}}$, which are the kind of morphism that we use to relate the unfolding of a net back to the original net, with respect to the path category $\mathbf{Caus}_{\mathrm{f}}$ coincides with openness of folding morphisms in the category $\mathbf{Gen}^\sharp$ with respect to the path category **Caus**. The same property holds for occurrence nets as a consequence of Lemma D.0.1.

Lemma D.0.3. *Let G and G' be general nets and $(\eta, \beta){:}G \to G'$ be a folding morphism. The morphism (η, β) is* **Caus***-open in* $\mathbf{Gen}^\sharp$ *if, and only if, it is* $\mathbf{Caus}_{\mathrm{f}}$*-open in* $\mathbf{Gen}^\sharp_{\mathrm{f}}$.

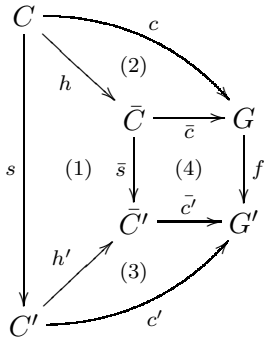
Proof. It is easy to show that a folding morphism is $\mathbf{Caus}_{\mathrm{f}}$-open in $\mathbf{Gen}^\sharp_{\mathrm{f}}$ if it is **Caus**-open in $\mathbf{Gen}^\sharp$.

Suppose that the folding morphism $f = (\eta, \beta){:}G \to G'$ is $\mathbf{Caus}_{\mathrm{f}}$-open in $\mathbf{Gen}^\sharp_{\mathrm{f}}$. To show that it is **Caus**-open in $\mathbf{Gen}^\sharp$, suppose that there are causal nets C and C' and

morphisms c, c' and s in $\mathbf{Gen}^\sharp$ such that the following diagram commutes:



We shall show that there are causal nets $\bar{C}$ and $\bar{C}'$, general net morphisms h and h' and folding morphisms $\bar{s}$, $\bar{c}$ and $\bar{c}'$ such that the squares (1) and (4) and triangles (2) and (3) in the diagram below commute. It will then follow from f being $\mathbf{Caus}_{\mathrm{f}}$-open in $\mathbf{Gen}^\sharp_{\mathrm{f}}$ that f is $\mathbf{Caus}$-open in $\mathbf{Gen}^\sharp$.



The net $\bar{C}$, defined as follows, will be a causal net and shall be defined with a flow relation $F(\bar{C})$:

$$\begin{array}{rcl} P(\bar{C}) & \triangleq & \{(b,i,p) \mid b \in P(C) \;\&\; p \in P(G) \;\&\; 0 \leq i < c_{\mathsf{c}}[b,p]\} \\ T(\bar{C}) & \triangleq & \{e \in T(C) \mid c_{\mathsf{e}}(e) \neq *\} \\ (b,i,p)\ F(\bar{C})\ e \iff b\ F(C)\ e & & e\ F(\bar{C})\ (b,i,p) \iff e\ F(C)\ b \\ \mathbb{M}(\bar{C}) & \triangleq & \{\{(b,i,p) \mid (b,i,p) \in P(C) \;\&\; b \in M\} \mid M \in \mathbb{M}(C)\} \end{array}$$

The net $\bar{C}'$ is defined similarly, using C' in place of C, c' in place of c and G' in place of G. The only slightly difficult part of showing that $\bar{C}$ is a causal net is to show that for all $x \in P(\bar{C}) \cup T(\bar{C})$ there exists $b \in M \in \mathbb{M}(\bar{C})$ such that $b\ F(C)^*\ x$. This follows from the following claim:

Claim. *For any $y \in P(C) \cup T(C)$:*

- *if $y \in P(C)$ and $(y,i,p) \in P(\bar{C})$ then there exists $b \in M \in \mathbb{M}(\bar{C})$ such that $b\ \ F\ (C)^*\ (y,i,p)$, and*
- *if $y \in T(C)$ and $y \in T(\bar{C})$ then there exists $b \in M \in \mathbb{M}(\bar{C})$ such that $b\ \ F\ (C)^*\ y$.*

Proof. Induction on the depth of y. □

On events, we define $\bar{c}_{\mathsf{e}}$ to be the restriction of c_{e} to $T(\bar{C})$, $\bar{c}'_{\mathsf{e}}$ to be the restriction of c'_{e} to $T(\bar{C}')$, $\bar{s}$ to be the restriction of s to $T(\bar{C})$, and h_{e} and h'_{e} to be the obvious inclusions. From these definitions, it is obvious that (1)–(4) above commute on events. On conditions, we define the functions $\bar{c}$ and $\bar{c}'$ as

$$\begin{array}{rcl} \bar{c}_{\mathsf{c}}(b,i,p) & \triangleq & p \\ \bar{c}'_{\mathsf{c}}(b',i,p') & \triangleq & p'. \end{array}$$

It is easy to show that $\bar{c}$ and $\bar{c}'$ are morphisms as a consequence of c and c' being morphisms. The morphism h need not be a folding morphism, but it shall be a relational morphism *i.e.* a relation on conditions.

$$h_{\mathsf{c}}(b,(b',i,p)) \iff b = b'.$$

Again, it is straightforward to prove that h is a morphism. The morphism $h'{:}C' \to \bar{C}'$ is defined similarly. The triangles (2) and (3) are now easily seen to commute, *i.e.* $\bar{c} \circ h = c$ and $\bar{c}' \circ h' = c'$.

All that remains is to define the morphism $\bar{s}$ and show that the squares (1) and (4) commute on conditions. Observe that the morphism s must be a relational morphism because it is between causal nets. From the commutation of the earlier square (*i.e.* $f \circ c = c' \circ s$), for every $b \in P(C)$ and $p' \in P(G')$, there exists a bijection

$$\begin{array}{rcl} \theta_{b,p'} & : & \{(p,i) \mid p \in P(G) \ \&\ f_{\mathsf{e}}(p) = p' \ \&\ 0 \le i < c_{\mathsf{c}}(b,p)\} \\ & \cong & \{(b',i) \mid b' \in P(C') \ \&\ s_{\mathsf{c}}(b,b') \ \&\ 0 \le i < c'_{\mathsf{c}}(b',p')\}. \end{array}$$

We define $\bar{s}_{\mathsf{c}}{:}P(\bar{C}) \to P(\bar{C}')$ as

$$\bar{s}_{\mathsf{c}}(b,i,p) = (b',i',f(p)) \quad \text{if } \theta_{b,f(p)}{:}(p,i) \mapsto (b',i').$$

A straightforward analysis shows that $\bar{s}$ is a (folding) morphism, and it is easy to see that the squares (1) and (4) commute. □

We conclude this section by characterizing when a morphism from an occurrence net to a general net is **Caus**-open.

Theorem D.1. *Let O be an occurrence net and G be a general net. A morphism $f{:}O \to G$ is* **Caus***-open in* $\mathbf{Gen}^{\sharp}$ *if, and only if, it is a folding morphism and reflects any initial marking of G to an initial marking of O and satisfies the following 'transition lifting' property:*

> *for any subset A of conditions of O such that* co A *for which there exists a transition t of G such that $f \cdot A = Pre_G \cdot t$, there exists an event e of O such that $A = {}^\bullet e$ and $f(e) = t$.*

Proof. 'Only if': It follows immediately from Lemma D.0.2 that f is a folding morphism. We first show that any **Caus**-open morphism reflects initial markings. Suppose that $M' \in \mathbb{M}(G)$. It follows from Lemma 7.4.2 that there is a causal net C and folding $\iota'{:}C \to G$ such that $\iota' \cdot \mathrm{mkg}(C) = M'$, and furthermore $\mathrm{mkg}(C)$ is the initial marking of C. Let 0 denote the causal net with no conditions, no events and no initial marking, $\iota{:}0 \to O$ be the inclusion of 0 into O, and s be the inclusion of 0 into C. The outer square of the following diagram commutes, so from openness of f there is a morphism h making the inner two triangles commute:

$$\begin{array}{ccc} 0 & \xrightarrow{\iota} & O \\ {\scriptstyle s}\downarrow & \overset{h}{\nearrow} & \downarrow{\scriptstyle f} \\ C & \xrightarrow[\iota']{} & G \end{array}$$

Since h is a morphism, $h \cdot \mathrm{mkg}(C) \in \mathbb{M}(O)$. By commutation of the lower triangle, $f \cdot h \cdot \mathrm{mkg}(C) = M'$, so, as required, f reflects initial markings if it is open.

We now show that for any $A \subseteq P(O)$, if co A and $f \cdot A = Pre \cdot t$ for some $t \in T(G)$ then there exists an event $e \in T(O)$ such that $A = {}^\bullet e$ and $f(e) = t$. Since we have

co A, according to Proposition 7.3 there exists a reachable marking M of O such that $A \subseteq M$. Hence, by Lemma 7.4.2, there is a causal net C and morphism $\iota{:}C \to O$ such that $\iota \cdot \mathrm{mkg}(C) = M$. It follows that there is a unique subset $A_0 \subseteq \mathrm{mkg}(C)$ such that $\iota \cdot A_0 = A$. According to Lemma 7.4.3, the following diagram commutes:

$$\begin{array}{ccc} C & \xrightarrow{\iota} & G \\ {\scriptstyle \hat{f}}\downarrow & & \downarrow{\scriptstyle f} \\ \hat{f}C & \xrightarrow[\hat{\iota}]{} & G' \end{array}$$

Recall that the path $\hat{f}C$ with embedding $\hat{\iota}{:}\hat{f}C \to G$ is the image under f of the path C with embedding $\iota{:}C \to O$. Since the square commutes, we have

$$Pre \cdot t = f \cdot A = f \cdot \iota \cdot A_0 = \hat{\iota} \cdot \hat{f} \cdot A_0.$$

Hence, according to Lemma 7.4.1, the following triangle commutes:

$$\begin{array}{ccc} \hat{f}C & \xrightarrow{\hat{\iota}} & G \\ \downarrow & \nearrow_{\iota +_{\hat{f}\cdot A_0} t} & \\ \hat{f}C +_{\hat{f}\cdot A_0} t & & \end{array}$$

Recall that $\hat{f}C +_{\hat{f}\cdot A_0} t$ is the net fC extended with a new event with preconditions $\hat{f} \cdot A_0$, which (to avoid confusion) we shall call t', and the morphism $\hat{\iota}$ is extended to form $\hat{\iota} +_{\hat{f}\cdot A_0} t$ which sends t' to t.

We can now apply the definition of f being an open map to the composition of these two diagrams to obtain a morphism h such that the two triangles in the following diagram commute:

$$\begin{array}{ccc} C & \xrightarrow{\iota} & O \\ {\scriptstyle \hat{f}}\downarrow & \nearrow_{h} & \downarrow{\scriptstyle f} \\ \hat{f}C +_{\hat{f}\cdot A_0} t & \xrightarrow[\hat{\iota} +_{\hat{f}\cdot A_0} t]{} & G \end{array}$$

We have $f(h(t')) = t$ by commutation of the lower square and the definition of $\hat{\iota} +_{\hat{f}\cdot A_0} t$. According to the definition of $\hat{f}C +_{\hat{f}\cdot A_0} t$, we also have ${}^\bullet t' = \hat{f} \cdot A_0$. It follows from commutation of the upper square and h being a morphism that ${}^\bullet h(t') = \iota \cdot A_0 = A$, as required.

'If': It is sufficient according to Lemma D.0.3 to show that if a folding morphism $f{:}O \to G$ reflects initial markings and transitions in the manner defined above then it is $\mathbf{Caus}_{\mathrm{f}}$-open in $\mathbf{Gen}^{\sharp}_{\mathrm{f}}$.

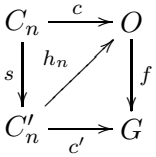
Suppose that the following diagram commutes, for causal nets C and C' and folding morphisms s, c and c':

$$\begin{array}{ccc} C & \xrightarrow{c} & O \\ {\scriptstyle s}\downarrow & & \downarrow{\scriptstyle f} \\ C' & \xrightarrow[c']{} & G \end{array}$$

For any $n \in \mathbb{N}$, let C_n denote the causal net obtained by restricting C to elements at depth less than or equal to n, and let C'_n denote C' restricted to elements at depth less than or

equal to n. It is clear that $C_0 \leq C_1 \leq \ldots$ and $C'_0 \leq C'_1 \leq \ldots$ are ω-chains of occurrence nets according to the subnet order defined in Section 7.3.

We shall give, by induction on n, a morphism $h_n{:}C'_n \to O$ such that the following diagram commutes:



The morphisms obtained shall be coherent in the sense that h_{n+1} and h_n shall coincide on elements of C'_n, and hence we shall obtain a morphism $h{:}C' \to O$ such that $c = h \circ s$ and $f \circ h = c'$ by applying Proposition 7.11.

Base case: We begin by constructing h_0. If C' has no initial marking, C'_0 has no events and no conditions and h_0 can be trivially defined to be the empty function.

If C' has an initial marking M' and C has an initial marking M, we must have $M' = s \cdot M$. The net C_0 consists precisely of the conditions in M and the net C'_0 consists precisely of the conditions in M'. For any condition b' in M', because s is a morphism, there is a unique condition b in M such that $s(b) = b'$. We define $h_0(b') = c(b)$. It is easy to see that this is a morphism that makes the two triangles commute.

Now suppose that C' has an initial marking M' but C has no initial marking. The net C has no conditions and no events. For any definition of h_0, the upper triangle will therefore trivially commute. Since c' is a morphism, $c' \cdot M' \in \mathbb{M}(G)$. By assumption, the morphism f reflects initial markings, so there exists $M \in \mathbb{M}(O)$ such that $f \cdot M = c' \cdot M'$. Hence, for any place $p \in P(G)$, there is a bijection

$$\theta_p{:}\{b' \in M' \mid c'(b) = p\} \cong \{b \in M \mid f(b) = p\}.$$

We define $h_0(b') = \theta_{c'(b)}(b)$. It is easy to see that this definition makes the lower triangle commute and that $h_0{:}C'_0 \to O$ is a morphism.

Inductive case: We now construct h_{n+1} from h_n. For any element x of C'_n at depth less than or equal to n, define $h_{n+1}(x) = h(x)$. It follows from the induction hypothesis that h_{n+1} satisfies the requirements for being a morphism on the initial marking of C'_{n+1} and any event of C'_{n+1} at depth less than or equal to n.

For any event e' of C' at depth precisely $n+1$, we now define $h_{n+1}(e)$ and define h_{n+1} on all postconditions of e. This leads to h_{n+1} being defined at all elements of C'_{n+1}.

If there exists an event e of C such that $s(e) = e'$, the event e must be the unique such event since otherwise the event e' could occur more than once in some run of the net C'. Define $h_{n+1}(e') = c(e)$. A simple induction shows that, for any folding morphism $f{:}C \to C'$ between causal nets, $\mathrm{depth}(x) = \mathrm{depth}(f(x))$ for any x in C. Hence $\mathrm{depth}(e) = \mathrm{depth}(e') = n+1$. For any condition $b' \in e'^{\bullet}$, there exists a unique condition $b \in e^{\bullet}$ such that $s(b) = b'$. Define $h_{n+1}(b') = c(b)$. It follows immediately from this definition that

$$h_{n+1} \cdot e'^{\bullet} \subseteq h_{n+1}(e')^{\bullet} \quad \& \quad \forall b \in h_{n+1}(e')^{\bullet} \,:\, \exists! b' \in e'^{\bullet} \,:\, h_{n+1}(b') = b.$$

From the induction hypothesis that the upper triangle commutes at depth n, *i.e.* $c_n = h_n \circ s_n$, it is easy to show that

$$h_{n+1} \cdot {}^{\bullet}e' \subseteq {}^{\bullet}h_{n+1}(e') \quad \& \quad \forall b \in {}^{\bullet}h_{n+1}(e') \,:\, \exists! b' \in {}^{\bullet}e' \,:\, h_{n+1}(b') = b,$$

and hence h_{n+1} satisfies the requirements for being a morphism on the event e'.

Now suppose that there is no event e of C such that $s(e) = e'$. From the induction hypothesis, the lower triangle commutes at depth n, so $f \circ h_n = c'_n$. Any precondition of e' occurs at depth less than or equal to n, so $c'_n \cdot {}^\bullet e' = c' \cdot {}^\bullet e'$ and hence $(f \circ h_n) \cdot {}^\bullet e' = c' \cdot {}^\bullet e' = Pre_G \cdot c'(e')$. Since we have $\mathrm{co}_n\ {}^\bullet e$ because C'_n is a causal net, we must also have co $(h_n \cdot {}^\bullet e')$. It follows from the 'event lifting' property above that there exists an event $e \in E(O)$ such that ${}^\bullet e = h_n \cdot {}^\bullet e'$ and $f(e) = c'(e')$. We define $h_{n+1}(e') = e$. We now consider postconditions of e'. Since f and c' are morphisms and $f(e) = c'(e')$, for every $p \in P(G)$ there is a bijection induced by f and c'

$$\theta_p{:}\{b' \in e'^\bullet \mid c'(b') = p\} \cong \{b \in e^\bullet \mid f(b) = p\}.$$

We define $h_{n+1}(b') = \theta_{c'(b')}(b')$ for any $b' \in e'^\bullet$, all of which will be at depth $n+1$. It is easy to see from these definitions that h_{n+1} satisfies the requirements for being a morphism on the event e'.

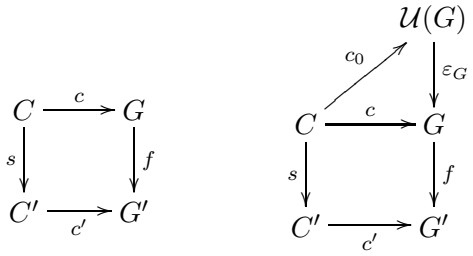
It follows that h_{n+1} is a morphism, clearly satisfying $c_{n+1} = h_{n+1} \circ s_{n+1}$ and $c'_{n+1} = f \circ h_{n+1}$. □

Using the cofreeness (apart from uniqueness of the mediating morphism) result in Theorem 7.15, the theorem above can be used to characterize when a morphism from *any* general net is open.

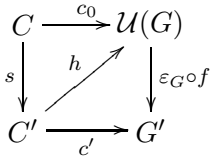
Proposition D.2. *A morphism $f{:}G \to G'$ is* **Caus***-open in* $\mathbf{Gen}^\sharp$ *if, and only if, the morphism $f \circ \varepsilon_G{:}\mathcal{U}(G) \to G'$ is* **Caus***-open.*

Proof. The "only if" direction is easy since, using the theorem above, the morphism $\varepsilon_G{:}\mathcal{U}(G) \to G$ is readily seen to be **Caus**-open, and the composition of open morphisms is an open morphism.

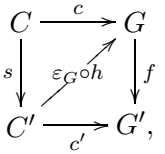
We now consider the "if" direction. If the diagram on the left commutes, by Theorem 7.15 there must exist a morphism $c_0{:}C \to \mathcal{U}(G)$ such that the outer square of the diagram on the right commutes because any causal net is an occurrence net:



The morphism $f \circ \varepsilon_G$ is assumed to be open, so there exists a morphism $h{:}C' \to \mathcal{U}(G)$ such that the following diagram commutes:



From this, we can see that the following diagram commutes



and hence $f{:}G \to G'$ is **Caus**-open. □

Appendix E

Correspondence

In this appendix, we show that the net semantics corresponds to the operational semantics. This shall involve equivalence on nets.

E.1 Net equivalence

As discussed in Section 3.11, the equivalences between terms t and t' that we shall consider are spans of **Pom**-open morphisms from a safe net N between $\mathcal{C}[\![t]\!]$ and $\mathcal{C}[\![t']\!]$. For our purposes, it is useful to require the net N at the apex of the span to have some additional structure, and it is convenient to restrict attention to spans of synchronous morphisms.

Definition E.1.1 (Spanning net)**.** *Let N be a safe net with initial marking written* $\mathrm{Ic}(N)$*. The net N is a* congruence spanning net *if it satisfies:*

1. *there exists a unique non-empty marking* $\mathrm{Tc}(N)$ *of N, reachable from all other reachable markings, such that there exists no e with concession in* $\mathrm{Tc}(N)$*,*
2. *for any event e of N, the sets ${}^\bullet e$ and $e^\bullet$ are nonempty, and*
3. *the net N is well-terminating.*

As such, the net N has all the structure assumed of an embedded net representing a term in, for example, the sequential path lemma (Lemma 3.6.2).

Lemma E.1.1. *For any closed term t, the net $\mathcal{C}[\![t]\!]$ is a congruence spanning net.*

Proof. Lemmas 3.9.1, 3.5.1 and 3.7.1. □

Definition E.1.2 (Congruence span)**.** *Two nets, $\mathcal{C}[\![t_1]\!]$ and $\mathcal{C}[\![t_2]\!]$ are related by a* congruence span *if there exists a congruence spanning net N and a pair of synchronous* **Pom***-open morphisms $f_1{:}N \to \mathcal{C}[\![t_1]\!]$ and $f_2{:}N \to \mathcal{C}[\![t_2]\!]$.*

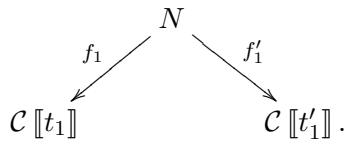
We write $\mathcal{C}[\![t]\!] \sim_{\mathsf{cc}} \mathcal{C}[\![t']\!]$, and sometimes just $t \sim_{\mathsf{cc}} t'$, if the nets $\mathcal{C}[\![t]\!]$ and $\mathcal{C}[\![t']\!]$ are related by a congruence span. We also write $(\mathcal{C}[\![t]\!], C) \sim_{\mathsf{cc}} (\mathcal{C}[\![t]\!], C')$ if the net $\mathcal{C}[\![t]\!]$ in initial marking C is related to $\mathcal{C}[\![t']\!]$ in initial marking C' by a congruence span.

An important property is that **Pom**-open morphisms preserve terminal markings, just as any morphism preserves the initial marking of a net.

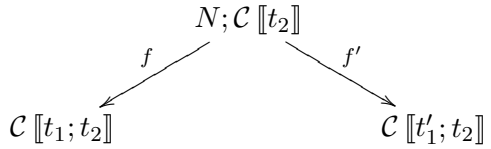
Lemma E.1.2. *Let $f = (\eta, \beta){:}N \to N'$ be a* **Pom***-open morphism between congruence spanning nets N and N'. Then* $\mathrm{Tc}(N') = \beta\mathrm{Tc}(N)$*.*

Proof. Suppose, for contradiction, that $\beta\mathrm{Tc}(N) \neq \mathrm{Tc}(N')$. Since the marking $\mathrm{Tc}(N)$ is reachable from $\mathrm{Ic}(N)$, the marking $\beta\mathrm{Tc}(N)$ is reachable from $\mathrm{Ic}(N')$. From the definition of $\mathrm{Tc}(N')$ as being the *unique* reachable marking of N' in which no event has concession, there exists an event e' such that $\beta\mathrm{Tc}(N) \stackrel{e'}{\longrightarrow\!\!\!\!\!\rightarrow}$. From openness of f, there must exist e such that $\eta(e) = e'$ and $\mathrm{Tc}(N) \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow}$. This contradicts terminality of $\mathrm{Tc}(N)$. □

It is reasonable to ask whether it is necessary to restrict attention to congruence spans rather than taking standard open map bisimulation on the control nets. We shall explain why it is required that the net N must be a congruence spanning net, first by drawing attention to sequential composition. Suppose that we have a span



We want a span relating $\mathcal{C}[\![t_1; t_2]\!]$ and $\mathcal{C}[\![t_1'; t_2]\!]$. The natural way of forming this is to form the sequential composition of N with $\mathcal{C}[\![t_2]\!]$, which we shall write $N; \mathcal{C}[\![t_2]\!]$, and show that the morphisms in the span



are open. Without restricting the net N to being a congruence spanning net, we lose a handle on the behaviour of the net $N; \mathcal{C}[\![t_2]\!]$ which we need in order to show that the morphisms f and f' are open.

Of course, this requires the operation of sequential composition of embedded nets to be defined; this just follows the earlier definition for sequential composition of nets representing terms.

Definition E.1.3. *Let N_1 and N_2 be congruence spanning nets where*

$$\begin{aligned} N_1 &= (C_1, E_1, {}^{\mathbf{C}}(-), (-)^{\mathbf{C}}, \mathrm{Ic}(N_1), |-|_1) \text{ and} \\ N_2 &= (C_2, E_2, {}^{\mathbf{C}}(-), (-)^{\mathbf{C}}, \mathrm{Ic}(N_2), |-|_2). \end{aligned}$$

The net

$$N_1; N_2 = (C, E, {}^{\mathbf{C}}(-), (-)^{\mathbf{C}}, \mathrm{Ic}(N_1; N_2), |-|)$$

is defined as:

$$\begin{aligned} C &= \mathsf{seq}\,1{:}(C_1 \setminus \mathrm{Tc}(t_1)) \cup \mathsf{seq}\,2{:}(C_2 \setminus \mathrm{Ic}(t_2)) \\ &\quad \cup (\mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)) \\ \mathrm{Ic}(N_1; N_2) &= P \triangleleft \mathsf{seq}\,1{:}\mathrm{Ic}(N_1) \\ E &= P \triangleleft \mathsf{seq}\,1{:}E_1 \cup P \triangleright \mathsf{seq}\,2{:}E_2 \end{aligned}$$

Let e be an event of N_1. Recall that it can be considered to be a tuple $({}^{\mathbf{C}}e, e^{\mathbf{C}}, \lambda)$, where λ indicates its effect on the state conditions, according to the discussion preceding Lemma 3.4.1. It gives rise to an event denoted $P \triangleleft \mathsf{seq}\,1{:}e$ which has preconditions $P \triangleleft \mathsf{seq}\,1{:}{}^{\mathbf{C}}e$, postconditions $P \triangleleft \mathsf{seq}\,1{:}e^{\mathbf{C}}$ and label λ.

Note that Lemma 3.6.2, which characterizes the runs of the net $\mathcal{C}[\![t_1; t_2]\!]$ can be extended to this setting to characterize the runs of $N_1; N_2$.

Lemma E.1.3. *Suppose that the nets N_1 and N_2 are congruence spanning nets. If $\mathrm{Ic}(N_1;N_2) \stackrel{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} C$ in $N_1;N_2$ then either:*

- *there exist C_1 and π_1 such that $C = P \triangleleft \mathsf{seq}\,1{:}C_1$ and $\pi = P \triangleleft \mathsf{seq}\,1{:}\pi_1$ and $\mathrm{Ic}(N_1) \stackrel{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ in N_1, or*
- *there exist C_2, π_1 and π_2 such that $C = P \triangleright \mathsf{seq}\,2{:}C_2$ and $\pi = (P \triangleleft \mathsf{seq}\,1{:}\pi_1) \cdot (P \triangleright \mathsf{seq}\,2{:}\pi_2)$ and $\mathrm{Ic}(N_1) \stackrel{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} \mathrm{Tc}(N_1)$ in N_1 and $\mathrm{Ic}(N_2) \stackrel{\pi_2}{\longrightarrow\!\!\!\!\!\rightarrow} C_2$ in N_2,*

where $P = \mathsf{seq}\,1{:}\mathrm{Tc}(N_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(N_2)$.

Proof. The proof is similar to that of Lemma 3.6.2. □

In fact, the net $N_1;N_2$ has all the structure needed to be a congruence spanning net.

Lemma E.1.4. *The net $N_1;N_2$ is a congruence spanning net and $\mathrm{Tc}(N_1;N_2) = P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$ for P defined as in Definition E.1.3.*

Proof. It is easy to see, using the preceding lemma, that $P \triangleright \mathsf{seq}\,2{:}\mathrm{Tc}(t_2)$ is the unique reachable marking in which no event has concession. The lemma is also needed to show that the net $N_1;N_2$ is well-terminating. All the other requirements follow immediately from the nets N_1 and N_2 being congruence spanning nets. □

We can now show that the existence of an open map is preserved by the operation of sequencing.

Lemma E.1.5. *Suppose that there exist synchronous* **Pom***-open morphisms $f_1{:}N_1 \to N_1'$ and $f_2{:}N_2 \to N_2'$ between congruence spanning nets N_1, N_1', N_2 and N_2'. Then there exists a synchronous* **Pom***-open morphism $f{:}N_1;N_2 \to N_1';N_2'$.*

Proof. The proof is rather technical and detailed, so the reader may wish to pass over it on first reading.

Let $f_1 = (\eta_1, \beta_1)$ and $f_2 = (\eta_2, \beta_2)$. Define $P = \mathsf{seq}\,1{:}\mathrm{Tc}(N_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(N_2)$ and define $P' = \mathsf{seq}\,1{:}\mathrm{Tc}(N_1') \times \mathsf{seq}\,2{:}\mathrm{Ic}(N_2')$. The morphism $f = (\eta, \beta)$ is defined as follows:

$$\begin{aligned}
\eta(P \triangleleft \mathsf{seq}\,1{:}e_1) &= P' \triangleleft \mathsf{seq}\,1{:}f_1(e_1) \\
\eta(P \triangleright \mathsf{seq}\,2{:}e_2) &= P' \triangleright \mathsf{seq}\,2{:}f_2(e_2)
\end{aligned}$$

$$\begin{aligned}
\beta(\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,1{:}b_1') &\iff b_1' \notin \mathrm{Tc}(N_1') \;\&\; \beta_1(b_1, b_1') \\
\beta(\mathsf{seq}\,1{:}b_1, (\mathsf{seq}\,1{:}b_1', \mathsf{seq}\,2{:}b_2')) &\iff b_1' \in \mathrm{Tc}(N_1') \;\&\; b_2' \in \mathrm{Ic}(N_2') \\
&\qquad \&\; \beta_1(b_1, b_1') \\
\beta(\mathsf{seq}\,2{:}b_2, \mathsf{seq}\,2{:}b_2') &\iff b_2' \notin \mathrm{Ic}(N_2') \;\&\; \beta_2(b_2, b_2') \\
\beta(\mathsf{seq}\,2{:}b_2, (\mathsf{seq}\,1{:}b_1', \mathsf{seq}\,2{:}b_2')) &\iff b_2' \in \mathrm{Ic}(N_2') \;\&\; b_1' \in \mathrm{Tc}(N_1') \\
&\qquad \&\; \beta_2(b_1, b_1') \\
\beta((\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2), (\mathsf{seq}\,1{:}b_1', \mathsf{seq}\,2{:}b_2')) & \\
&\iff b_1 \in \mathrm{Tc}(N_1) \;\&\; b_2 \in \mathrm{Ic}(N_2) \\
&\quad \&\; b_1' \in \mathrm{Tc}(N_1') \;\&\; b_2' \in \mathrm{Ic}(N_2') \\
&\quad \&\; \beta_1(b_1, b_1') \;\&\; \beta_2(b_2, b_2')
\end{aligned}$$

We first show that $f{:}N_1;N_2 \to N_1';N_2'$ is a morphism. This will follow immediately from the fact that, for any C_1 a subset of conditions of N_1 and C_2 a subset of conditions of N_2 on which β_1 and β_2 are, respectively, locally injective:

$$\begin{aligned}
\beta(P \triangleleft \mathsf{seq}\,1{:}C_1) &= P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1 \\
\beta(P \triangleright \mathsf{seq}\,2{:}C_2) &= P' \triangleright \mathsf{seq}\,2{:}\beta_2 C_2
\end{aligned}$$

We shall show only the first equation; the second is similar.

$\underline{\beta(P \triangleleft \mathsf{seq}\,1{:}C_1) \subseteq P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1}$: Suppose that $b \in P \triangleleft \mathsf{seq}\,1{:}C_1$ and $\beta(b, b')$. We must show that $b' \in P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1$.

First take $b = \mathsf{seq}\,1{:}b_1$ for some b_1. We have $b_1 \in C_1 \setminus \mathrm{Tc}(N_1)$. According to the definition of β, there are two cases for b':

- $b' = \mathsf{seq}\,1{:}b'_1$ for some $b'_1 \notin \mathrm{Tc}(N'_1)$ and $\beta_1(b_1, b'_1)$:
  We have $b'_1 \in \beta_1 C_1$, so $\mathsf{seq}\,1{:}b'_1 \in P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1$ as required.
- $b' = (\mathsf{seq}\,1{:}b'_1, \mathsf{seq}\,2{:}b'_2)$ for some $b'_1 \in \mathrm{Tc}(N'_1)$ and $b'_2 \in \mathrm{Ic}(N'_2)$ and $\beta_1(b_1, b'_1)$:
  We have $b'_1 \in \beta_1 C_1$, so $(\mathsf{seq}\,1{:}b'_1, \mathsf{seq}\,2{:}b''_2) \in P' \triangleleft \mathsf{seq}\,1{:}C_1$ for all $b''_2 \in \mathrm{Ic}(N'_2)$, as required.

Now take $b = (\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2)$ for some b_1 and b_2. We have $b_1 \in C_1 \cap \mathrm{Tc}(N_1)$ and $b_2 \in \mathrm{Ic}(N_2)$. From the definition of β, it must be the case that $b' = (\mathsf{seq}\,1{:}b'_1, \mathsf{seq}\,2{:}b'_2)$ for some $b'_1 \in \mathrm{Tc}(N'_1)$ and $b'_2 \in \mathrm{Ic}(N'_2)$ satisfying $\beta_1(b_1, b'_1)$ and $\beta_2(b_2, b'_2)$. Since $b'_1 \in \beta_1 C_1$, we have $b' \in P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1$ as required.

$\underline{\forall b' \in P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1\ \exists! b \in P \triangleleft \mathsf{seq}\,1{:}C_1\ :\ \beta(b, b')}$: Suppose that $b' \in P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1$. There are two cases for b'.

First take $b' = \mathsf{seq}\,1{:}b'_1$ for $b'_1 \in (\beta_1 C_1) \setminus \mathrm{Tc}(N'_1)$. There exists unique $b_1 \in C_1$ such that $\beta_1(b_1, b'_1)$. We have $b_1 \notin \mathrm{Tc}(N_1)$ since otherwise $b'_1 \in \mathrm{Tc}(N'_1)$ as open maps preserve terminal markings. Hence $\beta(\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,1{:}b'_1)$ and $b_1 \in P \triangleleft \mathsf{seq}\,1{:}C_1$. Uniqueness follows from uniqueness of b_1.

Now take $b' = (\mathsf{seq}\,1{:}b'_1, \mathsf{seq}\,2{:}b'_2)$ for $b'_1 \in (\beta_1 C_1) \cap \mathrm{Tc}(N'_1)$ and $b'_2 \in \mathrm{Ic}(N'_2)$. There exists a unique $b_1 \in C_1$ such that $\beta_1(b_1, b'_1)$.

If $b_1 \notin \mathrm{Tc}(N_1)$ then $\mathsf{seq}\,1{:}b_1 \in P \triangleleft \mathsf{seq}\,1{:}C_1$ and $\beta(\mathsf{seq}\,1{:}b_1, b')$.

If $b_1 \in \mathrm{Tc}(N_1)$ then there exists unique $b_2 \in \mathrm{Ic}(N_2)$ such that $\beta_2(b_2, b'_2)$ since β_2 is a morphism (here, the proof of the other equation uses the fact that $\beta_2 \mathrm{Tc}(N_2) = \mathrm{Tc}(N'_2)$ since open maps preserve terminal markings). We have $(\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2) \in P \triangleleft \mathsf{seq}\,1{:}b_1$ and, from the definition of β, it is the case that $\beta((\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2), b')$.

Uniqueness follows, in both cases, from the uniqueness of b_1 such that $b_1 \in C_1$ and $\beta_1(b_1, b'_1)$.

We now show that the morphism $f = (\eta, \beta)$ is **Pom**-open, by showing the following two properties:

- For any marking C reachable from $\mathrm{Ic}(N_1; N_2)$ in $N_1; N_2$, if $\beta C \xrightarrow{e'}\!\!\!\!\!\rightarrow$ then there exists e such that $C \xrightarrow{e}\!\!\!\!\!\rightarrow$ and $\eta(e) = e'$:
  Let π be the path $\mathrm{Ic}(N_1; N_2) \xrightarrow{\pi}\!\!\!\!\!\rightarrow C$. According to Lemma E.1.3, there are two cases for the marking C.

  $\underline{C = P \triangleleft \mathsf{seq}\,1{:}C_1}$: for some C_1 and there exists π_1 satisfying $\mathrm{Ic}(N_1) \xrightarrow{\pi_1}\!\!\!\!\!\rightarrow C_1$ and $\pi = P \triangleleft \mathsf{seq}\,1{:}\pi_1$.
  From the earlier result, we have $\beta C = \beta(P \triangleleft \mathsf{seq}\,1{:}C_1) = P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1$. According to the definition of the events of $N'_1; N'_2$, there are two cases for the event e':

If $e' = P' \triangleleft \mathsf{seq}\,1{:}e'_1$ for some e'_1, we have $P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1 \overset{P' \triangleleft \mathsf{seq}\,1:e'_1}{\longrightarrow\!\!\!\!\!\rightarrow}$. From Lemma 3.3.2, we therefore have $\beta_1 C_1 \overset{e'_1}{\longrightarrow\!\!\!\!\!\rightarrow}$ in N'_1. From the openness of (η_1, β_1), there therefore exists e_1 such that $\eta_1(e_1) = e'_1$ and $C_1 \overset{e_1}{\longrightarrow\!\!\!\!\!\rightarrow}$. From Lemma 3.3.2, it is therefore the case that $P \triangleleft \mathsf{seq}\,1{:}C_1 \overset{P \triangleleft \mathsf{seq}\,1:e_1}{\longrightarrow\!\!\!\!\!\rightarrow}$. Now, $\eta(P \triangleleft \mathsf{seq}\,1{:}e_1) = P' \triangleleft \mathsf{seq}\,1{:}\eta_1(e_1)$ by definition, so the case is complete.
If $e' = P' \triangleright \mathsf{seq}\,2{:}e'_2$ for some e'_2, we have

$$P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1 \overset{P' \triangleright \mathsf{seq}\,2:e'_2}{\longrightarrow\!\!\!\!\!\rightarrow} \quad \text{in } N'_1; N'_2.$$

By assumption, every event of N'_2 has a precondition, so there exists $b'_2 \in {}^\bullet e'_2$. Since the event has concession in $P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1$, it must be the case that $b'_2 \in \mathrm{Ic}(N_2)$. We must therefore have $(\mathsf{seq}\,1{:}b'_1, \mathsf{seq}\,2{:}b'_2) \in P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1$ for all $b'_1 \in \mathrm{Tc}(N'_1)$. Consequently, since the marking $\beta_1 C_1$ is reachable in N'_1 which is a well-terminating net, it must be the case that $\beta_1 C_1 = \mathrm{Tc}(N'_1)$ and therefore $C_1 = \mathrm{Tc}(N_1)$. (Otherwise, because N_1 is a congruence spanning net, there would exist an event with concession in C_1 according to Lemma 3.9.1, so there would be an event with concession in $\beta_1 C_1 = \mathrm{Tc}(N'_1)$, contradicting N'_1 being a congruence spanning net.) Hence

$$P' \triangleleft \mathsf{seq}\,1{:}\beta_1 C_1 = P' \triangleright \mathsf{seq}\,2{:}\mathrm{Ic}(N'_2) \overset{P' \triangleright \mathsf{seq}\,2:e'_2}{\longrightarrow\!\!\!\!\!\rightarrow}$$

and therefore, by Lemma 3.3.2, $\mathrm{Ic}(N'_2) \overset{e'_2}{\longrightarrow\!\!\!\!\!\rightarrow}$. By openness of (η_2, β_2), there exists e_2 such that $\eta_2(e_2) = e'_2$ and $\mathrm{Ic}(N_2) \overset{e_2}{\longrightarrow\!\!\!\!\!\rightarrow}$. Again by Lemma 3.3.2, we have

$$P \triangleright \mathsf{seq}\,2{:}\mathrm{Ic}(N_2) = P \triangleleft \mathsf{seq}\,1{:}\mathrm{Tc}(N_1) \overset{P \triangleright \mathsf{seq}\,2:e_2}{\longrightarrow\!\!\!\!\!\rightarrow} \quad \text{in } N_1; N_2,$$

which completes the case since $\eta(P \triangleright \mathsf{seq}\,2{:}e_2) = P' \triangleright \mathsf{seq}\,2{:}\eta_2(e_2) = e'$.

$\underline{C = P \triangleright \mathsf{seq}\,2{:}C_2}$: for some C_2 and there exist π_1, π_2 such that $\pi = (P \triangleleft \mathsf{seq}\,1{:}\pi_1) \cdot (P \triangleright \mathsf{seq}\,2{:}\pi_2)$ and

$$\mathrm{Ic}(N_1) \overset{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} \mathrm{Tc}(N_1) \qquad \mathrm{Ic}(N_2) \overset{\pi_2}{\longrightarrow\!\!\!\!\!\rightarrow} C_2.$$

Again, we consider the two cases for the event e' (it shall turn out that the first cannot be so).

If $e' = P' \triangleleft \mathsf{seq}\,1{:}e'_1$ for some e'_1, we have $P' \triangleright \mathsf{seq}\,2{:}\beta_2 C_2 \overset{P' \triangleleft \mathsf{seq}\,1:e'_1}{\longrightarrow\!\!\!\!\!\rightarrow}$. Following an argument similar to that in the case above, we must have $\beta_2 C_2 = \mathrm{Ic}(N'_2)$ and therefore $P' \triangleright \mathsf{seq}\,2{:}\beta_2 C_2 = P' \triangleleft \mathsf{seq}\,1{:}\mathrm{Tc}(N'_1)$. Hence, by Lemma 3.3.2, we have $\mathrm{Tc}(N'_1) \overset{e'_1}{\longrightarrow\!\!\!\!\!\rightarrow}$, contradicting terminality of $\mathrm{Tc}(N'_1)$.

If $e' = P' \triangleright \mathsf{seq}\,2{:}e'_2$ for some e'_2, we have $P' \triangleright \mathsf{seq}\,2{:}\beta_2 C_2 \overset{P' \triangleright \mathsf{seq}\,2:e'_2}{\longrightarrow\!\!\!\!\!\rightarrow}$. From Lemma 3.3.2, we obtain $\beta_2 C_2 \overset{e'_2}{\longrightarrow\!\!\!\!\!\rightarrow}$. From the openness of (η_2, β_2), we may deduce that there exists e_2 such that $C_2 \overset{e_2}{\longrightarrow\!\!\!\!\!\rightarrow}$ in N_2 and $\eta_2(e_2) = e'_2$. Hence, again by Lemma 3.3.2, $P \triangleright \mathsf{seq}\,2{:}C_2 \overset{P \triangleright \mathsf{seq}\,2:e_2}{\longrightarrow\!\!\!\!\!\rightarrow}$ in $N_1; N_2$, which completes this case since $\eta(P \triangleright \mathsf{seq}\,2{:}e_2) = P' \triangleright \mathsf{seq}\,2{:}\eta_2(e_2)$.

- For any marking C reachable from $\mathrm{Ic}(N_1;N_2)$ in $N_1;N_2$ and any e and e', if there exists C' such that $C \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} C' \stackrel{e'}{\longrightarrow\!\!\!\!\!\rightarrow}$ in $N_1;N_2$ and $\eta(e)I'\eta(e')$ then eIe', where I is the independence relation on $N_1;N_2$ and I' is the independence relation on $N_1';N_2'$.

  Suppose that there exists C reachable from $\mathrm{Ic}(N_1;N_2)$ by a path π in $N_1;N_2$ for which there exist C', e and e' such that $C \stackrel{e}{\longrightarrow\!\!\!\!\!\rightarrow} C' \stackrel{e'}{\longrightarrow\!\!\!\!\!\rightarrow}$ and $\eta(e)I\eta(e')$. According to the sequential path lemma, Lemma E.1.3, there are three cases:

  - there exist e_1 and e_1' such that $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ and $e' = P \triangleleft \mathsf{seq}\,1{:}e_1'$:
    Informally, the events e and e' are both in the N_1 part of $N_1;N_2$. Their consecutive occurrence in $N_1;N_2$ will yield the consecutive occurrence of the events in N_1, and the independence of $\eta(e)$ and $\eta(e')$ in N_1' will yield independence in N_1.

    Formally, in this case, according to Lemma E.1.3 there exist C_1, C_1', π_1 such that

    $$C = P \triangleleft \mathsf{seq}\,1{:}C_1 \qquad C' = P \triangleleft \mathsf{seq}\,1{:}C_1' \qquad \pi = P \triangleleft \mathsf{seq}\,1{:}\pi_1$$

    and $\mathrm{Ic}(N_1) \stackrel{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ in N_1. In addition, we also have $C_1 \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1' \stackrel{e_1'}{\longrightarrow\!\!\!\!\!\rightarrow}$ in N_1'. By definition, $\eta(e) = P' \triangleleft \mathsf{seq}\,1{:}\eta_1(e_1)$ and $\eta(e') = P' \triangleleft \mathsf{seq}\,1{:}e_1'$. We have $\beta_1 C_1$ reachable from $\mathrm{Ic}(N_1')$ and $\beta_1 C_1 \stackrel{\eta_1(e_1)}{\longrightarrow\!\!\!\!\!\rightarrow} \beta_1 C_1' \stackrel{\eta_1(e_1')}{\longrightarrow\!\!\!\!\!\rightarrow}$. Now, it is easy to see that $\eta(e)I'\eta(e')$ iff $\eta_1(e_1)I_1'\eta(e_1')$, where I_1' is the independence relation on events of N_1'. Since the map (η_1, β_1) is open, we have $e_1 I_1 e_1'$ and hence eIe'.
  - there exist e_1 and e_2 such that $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ and $e' = P \triangleright \mathsf{seq}\,2{:}e_2$:
    Informally, the event e is an event of N_1 and the event e' is an event of N_2. We shall show that $\eta(e)$ and $\eta(e')$ cannot be independent.

    Formally, in this case, according to Lemma E.1.3 we have $C' = P$ and there exist C_1 and π_1 such that

    $$C = P \triangleleft \mathsf{seq}\,1{:}C_1 \qquad \pi = P \triangleleft \mathsf{seq}\,1{:}\pi_1$$

    and $\mathrm{Ic}(N_1) \stackrel{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ in N_1. We also have $C_1 \stackrel{e_1}{\longrightarrow\!\!\!\!\!\rightarrow} \mathrm{Tc}(N_1)$ in N_1 and $\mathrm{Ic}(N_2) \stackrel{e_2}{\longrightarrow\!\!\!\!\!\rightarrow}$ in N_2. By assumption, $e_1^\bullet \neq \emptyset$ and ${}^\bullet e_2 \neq \emptyset$; suppose that $b_1 \in e_1^\bullet$ and $b_2 \in {}^\bullet e_2$. We must have $b_1 \in \mathrm{Tc}(N_1)$ and $b_2 \in \mathrm{Ic}(N_2)$. Hence $(\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2) \in e^\bullet \cap {}^\bullet e'$. Since $\mathrm{Tc}(N_1') \neq \emptyset$ and $\mathrm{Ic}(N_2') \neq \emptyset$, from the definition of β there exists b' such that $\beta((\mathsf{seq}\,1{:}b_1, \mathsf{seq}\,2{:}b_2), b')$. It follows that $b' \in \eta(e)^\bullet \cap {}^\bullet\eta(e)$, and therefore $\neg(\eta(e)I'\eta(e'))$.
  - The final case, where $e = P \triangleright \mathsf{seq}\,2{:}e_2$ and $e' = P \triangleright \mathsf{seq}\,2{:}e_2'$ is similar to first case. □

We can show similarly that if there are open morphisms $f_1{:}N_1 \to N_1'$ and $f_2{:}N_2 \to N_2'$ between congruence spanning nets then there is an open morphism from the net $\alpha_1.N_1 + \alpha_2.N_2$ to the net $\alpha_1.N_1' \to \alpha_2.N_2'$. We have again extended the notation for the nondeterministic guarded sum of terms to form the nondeterministic guarded sum of nets.

Similar results can be obtained for the other constructs. A slight technical complication is that we deal with nets representing closed terms, so it is not enough to say that if there is an open map from N to N' then there is an open map from `resource` w `do` N `od` to

$\texttt{resource}\ w\ \texttt{do}\ N'\ \texttt{od}$. Instead, we must show that if, for every $r \in \mathrm{Res}$, there is an open map $f_r{:}N_r \to N'_r$ for congruence spanning nets N_r and N'_r then there is an open map

$$f{:}\texttt{resource}\ \_\ \texttt{do}\ (N_r)_{r\in\mathrm{Res}}\ \texttt{od} \to \texttt{resource}\ \_\ \texttt{do}\ (N'_r)_{r\in\mathrm{Res}}\ \texttt{od}.$$

The control net $\texttt{resource}\ \_\ \texttt{do}\ (N_r)_{r\in\mathrm{Res}}\ \texttt{od}$ is defined in a manner similar to the definition of the net $\mathcal{C}\,[\![\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}]\!]$, with the net N_r taking the place of the control net for $[r/w]t$.

Lemma E.1.6. *Suppose that there are synchronous* **Pom***-open morphisms $f_1{:}N_1 \multimap N'_1$ and $f_2{:}N_2 \multimap N'_2$ between congruence spanning nets. Then there are* **Pom***-open morphisms*

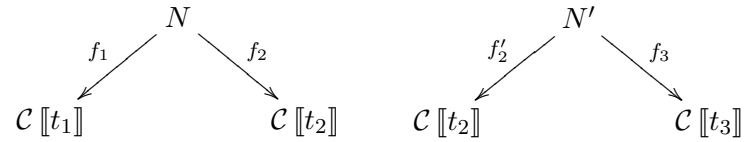
$$\begin{aligned} N_1 \parallel N_2 &\multimap N'_1 \parallel N'_2 \\ \alpha_1.N_1 + \alpha_2.N_2 &\multimap \alpha_1.N'_1 + \alpha_2.N'_2 \\ \texttt{while}\ b\ \texttt{do}\ N_1\ \texttt{od} &\multimap \texttt{while}\ b\ \texttt{do}\ N'_1\ \texttt{od} \\ \texttt{with}\ r\ \texttt{do}\ N_1\ \texttt{od} &\multimap \texttt{with}\ r\ \texttt{do}\ N'_1\ \texttt{od} \end{aligned}$$

Suppose that there is a synchronous **Pom***-open morphism $f_r{:}N_r \multimap N'_r$ between congruence spanning nets N_r and N'_r for each $r \in R$ for $R \subseteq \mathrm{Res}$. Then there is a* **Pom***-open morphism*

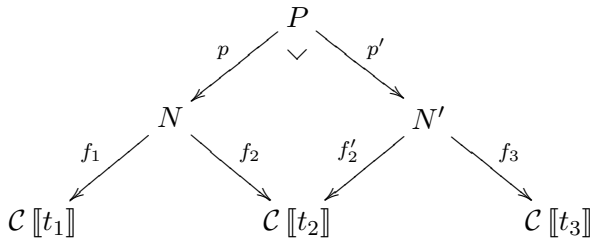
$$\texttt{resource}\ \_\ \texttt{do}\ (N_r)_{r\in R}\ \texttt{od} \multimap \texttt{resource}\ \_\ \texttt{do}\ (N'_r)_{r\in R}\ \texttt{od}.$$

The above open maps are demonstrated in the same way as the open map for the sequential composition described above. In fact, all but the nondeterministic sum are much less intricate.

Part of showing that open map bisimilarity $\sim_{\mathsf{cc}}$ on control nets representing terms is a congruence is showing that it is an equivalence relation, and in particular that it is transitive. The usual way of composing a span is to take a pullback. That is, if we have spans



exhibiting $t_1 \sim_{\mathsf{cc}} t_2$ and $t_2 \sim_{\mathsf{cc}} t_3$ then $t_1 \sim_{\mathsf{cc}} t_3$ is exhibited by the span in the following diagram, where (P, p, p') is the pullback of f_2 against f'_2:



It follows that, if $t_1 \sim_{\mathsf{cc}} t_2$ and $t_2 \sim_{\mathsf{cc}} t_3$, there exists a safe net P and **Pom**-open morphisms $f_1 \circ p{:}P \multimap N_1$ and $f_3 \circ p'{:}P \multimap N_3$. The morphisms are open because pullbacks of open maps are open ([JNW95, Proposition 3]). From the definition of the pullback, the pullback morphisms p and p' are synchronous since f_2 and f'_2 are, and therefore so are $f_1 \circ p$ and $f_3 \circ p'$. To show that $t_1 \sim_{\mathsf{cc}} t_3$, all that remains is to demonstrate that the net P is a congruence spanning net. This requires two lemmas to help us to understand the reachable markings of pullbacks, the first of which is a technical lemma that allows us to prove the second.

Suppose that we take the following pullback in **Safe** for synchronous morphisms f_1 and f_2.

$$\begin{array}{ccc} P & \xrightarrow{p_1} & N_1 \\ {\scriptstyle p_2}\downarrow & \lrcorner & \downarrow{\scriptstyle f_1} \\ N_2 & \xrightarrow[f_2]{} & N \end{array}$$

Let $P = (B_P, E_P, F_P, M_P)$. Let $N_i = (B_i, E_i, F_i, M_i)$ and $N = (B, E, F, M_0)$. Let the pullback morphisms $p_i = (\theta_i, \alpha_i)$ and morphisms being pulled-back be $f_i = (\eta_i, \beta_i)$.

Lemma E.1.7. *Let M be reachable from M_P in P with ${}^\bullet(e_1, e_2) \subseteq M$. For any $b_1 \in B_1$:*

(1) $b_1 \in {}^\bullet e_1 \implies [b_1]^{{}^\bullet e_1}_{{}^\bullet e_2} = [b_1]^{\alpha_1 M}_{\alpha_2 M}$

(2) $b_1 \notin {}^\bullet e_1 \implies [b_1]^{\alpha_1 M \setminus {}^\bullet e_1}_{\alpha_2 M \setminus {}^\bullet e_2} = [b_1]^{\alpha_1 M}_{\alpha_2 M}$

Now let M' be reachable from M_P in P with $(e_1, e_2)^\bullet \subseteq M'$. For any $b_1 \in B_1$:

(3) $b_1 \in e_1{}^\bullet \implies [b_1]^{e_1{}^\bullet}_{e_2{}^\bullet} = [b_1]^{\alpha_1 M'}_{\alpha_2 M'}$

(4) $b_1 \notin e_1{}^\bullet \implies [b_1]^{\alpha_1 M' \setminus e_1{}^\bullet}_{\alpha_2 M' \setminus e_2{}^\bullet} = [b_1]^{\alpha_1 M'}_{\alpha_2 M'}$

The symmetric statements for $b_2 \in B_2$ also hold.

Proof.
(1): Suppose first that $b_1 \in {}^\bullet e_1$. We shall show that $b_1 \sim^{{}^\bullet e_1}_{{}^\bullet e_2} b_2$ iff $b_1 \sim^{\alpha_1 M}_{\alpha_2 M} b_2$, from which it follows that $[b_1]^{e_1{}^\bullet}_{e_2{}^\bullet} = [b_1]^{\alpha_1 M}_{\alpha_2 M}$ by a simple induction following the inductive characterization of $\sim^{{}^\bullet e_1}_{{}^\bullet e_2}$. (Technically, the induction has to be performed simultaneously with the similar statement for b_2).

'Only if': We have $b_2 \in {}^\bullet e_2$ from the definition of $b_1 \sim^{{}^\bullet e_1}_{{}^\bullet e_2} b_2$. The events e_1 and e_2 have concession in $\alpha_1 M$ and $\alpha_2 M$, respectively, so $b_1 \in \alpha_1 M$ and $b_2 \in \alpha_2 M$. From the definition of $\sim^{\alpha_1 M}_{\alpha_2 M}$, we have $b_1 \sim^{\alpha_1 M}_{\alpha_2 M} b_2$.

'If': Now suppose that $b_1 \sim^{\alpha_1 M}_{\alpha_2 M} b_2$. We wish to show that $b_1 \sim^{{}^\bullet e_1}_{{}^\bullet e_2} b_2$. We have, by assumption, $b_1 \in {}^\bullet e_1$. Since $b_1 \sim^{\alpha_1 M}_{\alpha_2 M} b_2$, there exists $b_0 \in \beta_1 \alpha_1 M = \beta_2 \alpha_2 M$ such that $\beta_1(b_1, b_0)$ and $\beta_2(b_2, b_0)$. Since (η_1, θ_1) is a morphism, $b_0 \in {}^\bullet \eta_1(e_1) = {}^\bullet \eta_2(e_2)$. Since (η_2, β_2) is a morphism, there exists a unique $b_2' \in {}^\bullet e_2$ such that $\beta_2(b_2', b_0)$. The event e_2 has concession in $\alpha_2 M$, so $b_2' \in \alpha_2 M$. Since the marking $\alpha_2 M$ is reachable in N_2 and $b_2 \in \alpha_2 M$ from the definition of $b_1 \sim^{\alpha_1 M}_{\alpha_2 M} b_2$, the condition b_2 is the unique condition in $\alpha_2 M$ such that $\beta_2(b_2, b_0)$, and therefore $b_2 = b_2'$. Hence $b_1 \sim^{{}^\bullet e_1}_{{}^\bullet e_2} b_2$.
(2): Suppose that $b \notin {}^\bullet e_1$. As in the previous case, we shall show that $b_1 \sim^{\alpha_1 M \setminus {}^\bullet e_1}_{\alpha_2 M \setminus {}^\bullet e_2} b_2$ iff $b_1 \sim^{\alpha_1 M}_{\alpha_2 M} b_2$.

For the 'only if' direction, suppose that $b_1 \sim^{\alpha_1 M \setminus {}^\bullet e_1}_{\alpha_2 M \setminus {}^\bullet e_2} b_2$. We have $b_1 \in \alpha_1 M \setminus {}^\bullet e_1$ and $b_2 \in \alpha_2 M \setminus {}^\bullet e_2$, and there exists b_0 such that $\beta_1(b_1, b_0)$ and $\beta_2(b_2, b_0)$. We immediately obtain $b_1 \sim^{\alpha_1 M}_{\alpha_2 M} b_2$ from its definition.

For the 'if' direction, suppose that $b_1 \sim^{\alpha_1 M}_{\alpha_2 M} b_2$. There exists $b_0 \in \beta_1 \alpha_1 M = \beta_2 \alpha_2 M$ such that $\beta_1(b_1, b_0)$ and $\beta_2(b_2, b_0)$. Suppose, for contradiction, that $b_0 \in {}^\bullet \eta_1(e_1) = {}^\bullet \eta_2(e_2)$: There exists $b_1' \in {}^\bullet e_1$ such that $\beta_1(b_1', b_0)$ since (η_1, β_1) is a morphism. The event e_1 has concession in $\alpha_1 M$, so $b_1' \in \alpha_1 M$. However, we also have $b_1 \in \alpha_1 M$, so it must be the case that $b_1 = b_1'$ since (η_1, β_1) is a morphism. This contradicts $b_1 \notin {}^\bullet e_1$, so we can conclude

that $b_0 \notin \eta_1(e_1) = \eta_2(e_2)$. Furthermore, since (η_2, β_2) is a morphism we can also conclude that $b_2 \notin {}^\bullet e_2$. Hence $b_1 \sim^{\alpha_1 M \setminus {}^\bullet e_1}_{\alpha_2 M \setminus {}^\bullet e_2} b_2$, as required.

The proof of (3) is similar to the proof of (1), and the proof of (4) is similar to the proof of (2). □

Lemma E.1.8. *For any marking M reachable in P from M_P,*

$$M = \{[b_1]^{\alpha_1 M}_{\alpha_2 M} \mid b_1 \in \alpha_1 M\} \cup \{[b_2]^{\alpha_1 M}_{\alpha_2 M} \mid b_2 \in \alpha_2 M\}$$

Proof. The proof is by induction on the length of path to M. It is clearly the case that M_P satisfies the requirement from its definition. For the inductive case, suppose that M satisfies

$$M = \{[b_1]^{\alpha_1 M}_{\alpha_2 M} \mid b_1 \in \alpha_1 M\} \cup \{[b_2]^{\alpha_1 M}_{\alpha_2 M} \mid b_2 \in \alpha_2 M\}$$

and that there exists (e_1, e_2) such that $M \overset{(e_1,e_2)}{\longrightarrow\!\!\!\!\!\rightarrow} M'$. We must show that

$$M' = \{[b_1]^{\alpha_1 M'}_{\alpha_2 M'} \mid b_1 \in \alpha_1 M'\} \cup \{[b_2]^{\alpha_1 M'}_{\alpha_2 M'} \mid b_2 \in \alpha_2 M'\},$$

where $M' = M \setminus {}^\bullet(e_1, e_2) \cup (e_1, e_2)^\bullet$. The following chain holds:

$$\begin{array}{rcll}
M' & = & M \setminus {}^\bullet(e_1, e_2) \cup (e_1, e_2)^\bullet & \\
& = & \{[b]^{\alpha_1 M}_{\alpha_2 M} \mid b \in \alpha_1 M \cup \alpha_2 M\} \setminus \{[b]^{{}^\bullet e_1}_{{}^\bullet e_2} \mid b \in {}^\bullet e_1 \cup {}^\bullet e_2\} & \\
& & \cup \{[b]^{e_1^\bullet}_{e_2^\bullet} \mid b \in e_1^\bullet \cup e_2^\bullet\} & \text{by assumption and def} \\
& = & \{[b]^{\alpha_1 M}_{\alpha_2 M} \mid b \in \alpha_1 M \cup \alpha_2 M\} \setminus \{[b]^{\alpha_1 M}_{\alpha_2 M} \mid b \in {}^\bullet e_1 \cup {}^\bullet e_2\} & \\
& & \cup \{[b]^{e_1^\bullet}_{e_2^\bullet} \mid b \in e_1^\bullet \cup e_2^\bullet\} & \text{by Lemma E.1.7 (1)} \\
& = & \{[b]^{\alpha_1 M}_{\alpha_2 M} \mid b \in \alpha_1 M \setminus {}^\bullet e_1 \cup \alpha_2 M \setminus {}^\bullet e_2\} \cup \{[b]^{e_1^\bullet}_{e_2^\bullet} \mid b \in e_1^\bullet \cup e_2^\bullet\} & \\
& = & \{[b]^{\alpha_1 M \setminus {}^\bullet e_1}_{\alpha_2 M \setminus {}^\bullet e_2} \mid b \in \alpha_1 M \setminus {}^\bullet e_1 \cup \alpha_2 M \setminus {}^\bullet e_2\} & \\
& & \cup \{[b]^{e_1^\bullet}_{e_2^\bullet} \mid b \in e_1^\bullet \cup e_2^\bullet\} & \text{by Lemma E.1.7 (2)} \\
& = & \{[b]^{\alpha_1 M \setminus {}^\bullet e_1}_{\alpha_2 M \setminus {}^\bullet e_2} \mid b \in \alpha_1 M \setminus {}^\bullet e_1 \cup \alpha_2 M \setminus {}^\bullet e_2\} & \\
& & \cup \{[b]^{\alpha_1 M'}_{\alpha_2 M'} \mid b \in e_1^\bullet \cup e_2^\bullet\} & \text{by Lemma E.1.7 (3)} \\
& = & \{[b]^{\alpha_1 M'}_{\alpha_2 M'} \mid b \in \alpha_1 M \setminus {}^\bullet e_1 \cup \alpha_2 M \setminus {}^\bullet e_2\} & \\
& & \cup \{[b]^{\alpha_1 M'}_{\alpha_2 M'} \mid b \in e_1^\bullet \cup e_2^\bullet\} & \text{by safety and Lemma E.1.7 (4)} \\
& = & \{[b]^{\alpha_1 M'}_{\alpha_2 M'} \mid b \in \alpha_1 M' \cup \alpha_2 M'\} &
\end{array}$$

The required result follows immediately. □

We are now able to show that P is indeed a congruence spanning net.

Lemma E.1.9. *Let (P, p, p') be a pullback of* **Pom***-open synchronous morphisms $f_1{:}N_1 \multimap N$ and $f_2{:}N_2 \multimap N$. If N_1, N_2 and N are congruence spanning nets then so is P.*

Proof. For $i \in \{1, 2\}$, let $f_i = (\eta_i, \beta_i)$ and $p_i = (\theta_i, \alpha_i)$. Only two requirements on P are not very straightforward.

- *there exists a unique non-empty marking* $\mathrm{Tc}(P)$ *of P reachable from any marking reachable from* $\mathrm{Ic}(P)$ *such that there exists no e with concession in* $\mathrm{Tc}(P)$:
  Let C be any marking of P reachable from $\mathrm{Ic}(P)$. The marking $\beta_1\alpha_1 C$ is reachable in N from $\mathrm{Ic}(N)$. From the net N being a congruence spanning net, $\mathrm{Tc}(N)$ is reachable from $\beta_1\alpha_1 C$ and so there is a sequence $N{:}\beta_1\alpha_1 C \overset{\pi}{\longrightarrow\!\!\!\!\!\rightarrow} \mathrm{Tc}(N)$. Since (η_1, β_1) is **Pom**-open, there is a sequence $N_1{:}\alpha_1 C \overset{\pi_1}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ for some π_1 such that $\eta_1(\pi_1) = \pi$ and some

C_1 such that $\beta_1 C_1 = \mathrm{Tc}(N)$. We have $C_1 = \mathrm{Tc}(N_1)$ since otherwise, because N_1 is a congruence spanning net, there exists an event e_1 with concession in C_1 and hence the event $\eta_1(e_1)$ has concession in $\mathrm{Tc}(N)$. By the openness of (θ_1, α_1), there exists a path $P{:}C \xrightarrow{\pi_1'}\!\!\!\!\!\rightarrow\ C'$ for some C' such that $\alpha_1 C' = \mathrm{Tc}(N_1)$. By the same argument, there exists no event e of P such that $C' \xrightarrow{e}\!\!\!\!\!\rightarrow$. Therefore, for any marking C reachable from $\mathrm{Ic}(P)$ there is a marking C' reachable from C such that no event has concession in C'.

To see that there is a *unique* reachable marking in which no event has concession, let C' be any marking of P reachable from $\mathrm{Ic}(P)$ in which no event has concession. We must have $\alpha_1 C' = \mathrm{Tc}(N_1)$ by an argument similar to that in the paragraph above. Similarly, $\alpha_2 C' = \mathrm{Tc}(N_2)$. This uniquely determines C' according to Lemma E.1.8.

- *the net P is well-terminating*
  Let C be reachable in P from $\mathrm{Ic}(P)$. We must show that if $\mathrm{Tc}(P) \subseteq C$ then $C = \mathrm{Tc}(P)$.

  For contradiction, suppose that there exists $b \in C \setminus \mathrm{Tc}(P)$. Without loss of generality, recalling the definition of the conditions of P as being equivalence classes of conditions of N_1 and N_2, there exists a condition b_1 of N_1 such that $b_1 \in b$ and hence $\alpha_1(b, b_1)$.

  We have $b_1 \notin \mathrm{Tc}(N_1)$. To see this, suppose that $b_1 \in \mathrm{Tc}(N_1)$. There exists $b' \in \mathrm{Tc}(P)$ such that $\alpha_1(b', b_1)$ because $\alpha_1 \mathrm{Tc}(P) = \mathrm{Tc}(N_1)$ (as seen in the earlier part). The marking C is reachable in P, so b is the unique condition in C such that $\alpha_1(b, b_1)$. However, $\mathrm{Tc}(P) \subseteq C$ so $b = b'$. We assumed that $b \notin \mathrm{Tc}(P)$ but $b' \in \mathrm{Tc}(P)$, giving the contradiction required to show that $b_1 \notin \mathrm{Tc}(N_1)$.

  From the definition of application of a relation to a set, we have $\alpha_1 \mathrm{Tc}(P) \subseteq \alpha_1 C$, and from Lemma 2.3.1 the marking $\alpha_1 C$ is reachable from $\mathrm{Ic}(N_1)$. From Lemma E.1.2, we have $\mathrm{Tc}(N_1) = \alpha_1 \mathrm{Tc}(P)$ and hence $\mathrm{Tc}(N_1) \subseteq \alpha_1 C_1$. Since $b_1 \notin \mathrm{Tc}(N_1)$, the net N_1 is not well-terminating — a contradiction. Hence $\mathrm{Tc}(P) = C$. □

Now that we have dealt with these technical matters, we are able to show that $\sim_{\mathsf{cc}}$ is a congruence.

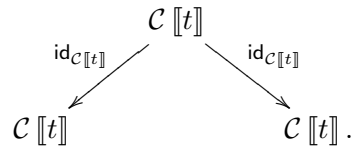
Theorem E.1 (Congruence)**.** *The relation $\sim_{\mathsf{cc}}$ is an equivalence relation and satisfies, for any closed terms t, t', t_1 and t_2 such that $t \sim_{\mathsf{cc}} t'$:*

$$
\begin{array}{rcl}
t; t_2 & \sim_{\mathsf{cc}} & t'; t_2 \\
t_1; t & \sim_{\mathsf{cc}} & t_1; t' \\
t \parallel t_2 & \sim_{\mathsf{cc}} & t' \parallel t_2 \\
t_1 \parallel t & \sim_{\mathsf{cc}} & t_1 \parallel t' \\
\alpha_1.t + \alpha_2.t_2 & \sim_{\mathsf{cc}} & \alpha_1.t' + \alpha_2.t_2 \\
\alpha_1.t_1 + \alpha_2.t & \sim_{\mathsf{cc}} & \alpha_1.t_1 + \alpha_2.t' \\
\texttt{while}\ b\ \texttt{do}\ t\ \texttt{od} & \sim_{\mathsf{cc}} & \texttt{while}\ b\ \texttt{do}\ t'\ \texttt{od} \\
\texttt{with}\ r\ \texttt{do}\ t\ \texttt{od} & \sim_{\mathsf{cc}} & \texttt{with}\ r\ \texttt{do}\ t'\ \texttt{od}
\end{array}
$$

For any terms t and t' with $\mathrm{res}(t) = \mathrm{res}(t')$ and $\mathrm{fv}(t) = \mathrm{fv}(t') = \{w\}$, if there exists $r \notin \mathrm{res}(t)$ such that $[r/w]t \sim_{\mathsf{cc}} [r/w]t'$ then

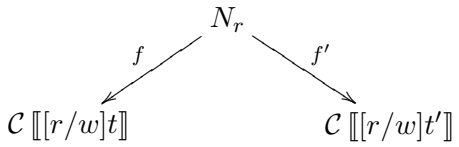
$$\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od} \sim_{\mathsf{cc}} \texttt{resource}\ w\ \texttt{do}\ t'\ \texttt{od}$$

Proof. The relation $\sim_{\mathsf{cc}}$ is, by definition, symmetric, and is transitive by taking the pull-back described above. It is clearly the case that $t \sim_{\mathsf{cc}} t$ for any closed term t since there is the span



The remainder of the result follows immediately from Lemmas E.1.5 and E.1.6 apart from the case for $\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}$.

Suppose that t and t' are terms with $\mathrm{fv}(t) = \mathrm{fv}(t') = w$ and that there exists $r \notin \mathrm{res}(t) \cup \mathrm{res}(t')$ for which there is a span

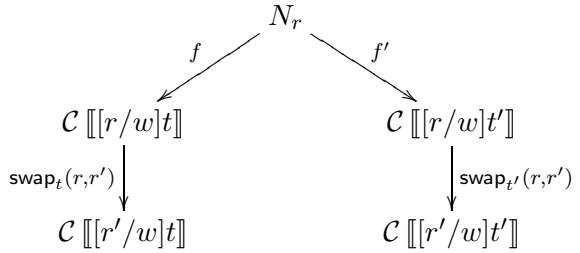


for some N_r, f and f'.

A simple induction on the size of terms shows the following property:

> for any term t with $\mathrm{fv}(t) = \{w\}$ and any pair of resources $r, r' \notin \mathrm{res}(t)$, the nets $\mathcal{N}[\![t[r/w]]\!]$ and $\mathcal{N}[\![t[r', w]]\!]$ are related through an isomorphism $\mathsf{swap}_t(r, r')$ that swaps r and r' in the obvious way.

We shall use the same notation for the ensuing isomorphism between the control nets. It follows that, for any $r' \notin \mathrm{res}(t) \cup \mathrm{res}(t')$ there is a span



It is easy to see from the definition that:

$$\begin{aligned}
\mathcal{C}[\![\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od}]\!] &= \texttt{resource}\ _\ \texttt{do}\ (\mathcal{C}[\![[r'/w]t]\!])_{r' \in \mathrm{Res} \setminus \mathrm{res}(t)}\ \texttt{od} \\
\mathcal{C}[\![\texttt{resource}\ w\ \texttt{do}\ t'\ \texttt{od}]\!] &= \texttt{resource}\ _\ \texttt{do}\ (\mathcal{C}[\![[r'/w]t']\!])_{r' \in \mathrm{Res} \setminus \mathrm{res}(t)}\ \texttt{od}
\end{aligned}$$

We may therefore apply Lemma E.1.6 to see that there exists a span exhibiting the bisimulation $\texttt{resource}\ w\ \texttt{do}\ t\ \texttt{od} \sim_{\mathsf{cc}} \texttt{resource}\ w\ \texttt{do}\ t'\ \texttt{od}$. □

E.2 Correspondence

Now that we have show that $\sim_{\mathsf{cc}}$ is a congruence, we progress to show that the net semantics for terms corresponds to the operational semantics introduced in Figure 3.3. The aim is to show the following:

- if $\langle t, \sigma\rangle \xrightarrow{\lambda} \langle t', \sigma'\rangle$ then there exists e such that $\mathcal{N}[\![t]\!] : (\mathrm{Ic}(t), \sigma) \xrightarrow{e} (C', \sigma')$ and $(\mathcal{C}[\![t]\!], C') \sim_{\mathsf{cc}} (\mathcal{C}[\![t']\!], \mathrm{Ic}(t'))$, and

- if $\mathcal{N}[\![t]\!] : (\mathrm{Ic}(t), \sigma) \xrightarrow{e} (C', \sigma')$ then there exists t' such that $\langle t, \sigma\rangle \xrightarrow{|e|} \langle t', \sigma'\rangle$ and $(\mathcal{C}[\![t]\!], C') \sim_{\mathsf{cc}} (\mathcal{C}[\![t']\!], \mathrm{Ic}(t'))$,

where $\sim_{\mathsf{cc}}$ is the congruence studied in the previous section.

There is some subtlety in the choice of the bisimulation to be $\sim_{\mathsf{cc}}$. Why, for instance, do we give a bisimulation between the control nets rather than a bisimulation between the embedded nets with their markings of state conditions,

$$(\mathcal{N}[\![t]\!], C', \sigma') \sim (\mathcal{N}[\![t']\!], \mathrm{Ic}(t'), \sigma') \ ?$$

The answer stems from the separate inductive proofs of the required properties. To see this, consider the second part of the correspondence theorem (the first has the same issue), and in particular the case for parallel composition. Part of the proof would involve showing that the net for $t_1 \parallel t_2$ in the state following the occurrence of an event e (of t_1, say) is bisimilar to the net for $t_1' \parallel t_2$ in its initial state. That is, we would need to exhibit a bisimulation

$$(\mathcal{N}[\![t_1 \parallel t_2]\!], \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}\mathrm{Ic}(t_2), \sigma') \sim (\mathcal{N}[\![t_1' \parallel t_2]\!], \mathrm{Ic}(t_1' \parallel t_2), \sigma'),$$

where C_1 is the marking of control conditions such that $(\mathrm{Ic}(t_1), \sigma) \xrightarrow{e_1} (C_1, \sigma')$ in $\mathcal{N}[\![t_1]\!]$ for e_1 the event such that $e = \mathsf{par}\,1{:}e_1$. From the induction hypothesis, we would obtain a bisimulation

$$(\mathcal{N}[\![t_1]\!], C_1, \sigma') \sim (\mathcal{N}[\![t_1']\!], \mathrm{Ic}(t_1'), \sigma'),$$

but, since this form of bisimulation is not a congruence, would not be able to use this to obtain the required bisimulation. It is for this reason that we employ the congruence $\sim_{\mathsf{cc}}$ on control nets, since we are able to obtain a bisimulation $(\mathcal{C}[\![t_1 \parallel t_2]\!], \mathsf{par}\,1{:}C_1 \cup \mathsf{par}\,2{:}\mathrm{Ic}(t_2)) \sim_{\mathsf{cc}} (\mathcal{C}[\![t_1' \parallel t_2]\!], \mathrm{Ic}(t_1' \parallel t_2))$ from $(\mathcal{C}[\![t_1]\!], C_1) \sim_{\mathsf{cc}} (\mathcal{C}[\![t_1']\!], \mathrm{Ic}(t_1'))$ because $\sim_{\mathsf{cc}}$ is a congruence.

Proving the properties above will involve exhibiting bisimulations between nets, and in particular congruence spans of **Pom**-open morphisms between control nets. The simplest forms of span are those comprising only one open morphism, the other being the identity: to exhibit a bisimulation $(\mathcal{C}[\![t_1]\!], C_1) \sim_{\mathsf{cc}} (\mathcal{C}[\![t_2]\!], C_2)$ between the control net for t_1 with marking of control conditions C_1 (reachable from $\mathrm{Ic}(t_1)$) and the control net for t_2 with marking of control conditions C_2 (reachable from $\mathrm{Ic}(t_2)$), it is sufficient to give a single synchronous **Pom**-open morphism

$$(\mathcal{C}[\![t_1]\!], C_1) \multimap (\mathcal{C}[\![t_2]\!], C_2).$$

The nets $(\mathcal{C}[\![t_1]\!], C_1)$ and $(\mathcal{C}[\![t_2]\!], C_2)$ are easily seen to be congruence spanning nets as a consequence of $(\mathcal{C}[\![t_1]\!], \mathrm{Ic}(t_1))$ and $(\mathcal{C}[\![t_2]\!], \mathrm{Ic}(t_2))$ being congruence spanning nets (Lemma E.1.1).

In demonstrating the correspondence of the operational and net semantics, we shall wish to stay in this simple setting as much as possible. The particular direction of the morphism to exhibit the bisimulations required in the introduction of this section shall be from $(\mathcal{C}[\![t']\!], \mathrm{Ic}(t'))$ to $(\mathcal{C}[\![t]\!], C')$. The reason for giving the open morphism in this direction is that, in the proofs of both of the above properties, when considering the action α we will need a bisimulation

$$(\mathcal{C}[\![\varepsilon]\!], \mathrm{Ic}(\varepsilon)) \sim_{\mathsf{cc}} (\mathcal{C}[\![\alpha]\!], \mathrm{Tc}(\alpha)).$$

There is no morphism of nets from $(\mathcal{C}[\![\alpha]\!], \mathrm{Tc}(\alpha))$ to $(\mathcal{C}[\![\varepsilon]\!], \mathrm{Ic}(\varepsilon))$, but there is an (open) morphism in the other direction. The most complicated open morphism that we shall require will be in the cases for the `while` construct, which we now consider.

Lemma E.2.1. *Let* $w = \texttt{while}\ b\ \texttt{do}\ t_0\ \texttt{od}$. *There is a synchronous* **Pom***-open morphism in the category* **Safe**

$$(\mathcal{C}[\![t_0; w]\!], \mathrm{Ic}(t_0; w)) \multimap (\mathcal{C}[\![w]\!], \mathsf{body}{:}\mathrm{Ic}(t_0))$$

Proof. Call the morphism (η, β). Let $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_0) \times \mathsf{seq}\,2{:}\mathsf{body}{:}\mathrm{Tc}(t_0)$. Informally, $(\eta, \beta){:}t_0; w \to w$ embeds the net for t_0 in $t_0; w$ into the body of the loop within w. On conditions in P, where the net for t_0 is joined to w, the morphism only relates $(\mathsf{seq}\,1{:}c, \mathsf{seq}\,2{:}\mathsf{body}{:}c)$ back to c for any $c \in \mathrm{Tc}(t)$.

Formally, the morphism is defined as:

$$\eta(P \triangleleft \mathsf{seq}\,1{:}e_0) = \mathsf{body}{:}e_0 \qquad \eta(P \triangleright \mathsf{seq}\,2{:}e_2) = e_2$$

$$\begin{array}{rcl} \beta(c, c') & \iff & \exists c_0 :\ c = \mathsf{seq}\,1{:}c_0\ \&\ c' = \mathsf{body}{:}c_0 \\ & \text{or} & \exists c_0 :\ c = (\mathsf{seq}\,1{:}c_0, \mathsf{seq}\,2{:}\mathsf{body}{:}c_0)\ \&\ c' = \mathsf{body}{:}c_0 \\ & \text{or} & c = \mathsf{seq}\,2{:}c_2\ \&\ c' = c_2 \end{array}$$

For any subsets of control conditions C_1 and C_2, we show the following. It will follow immediately that (η, β) satisfies the requirements for being a morphism.

1. $\beta(P \triangleleft \mathsf{seq}\,1{:}C_1) \subseteq \mathsf{body}{:}C_1$
   Let $c \in P \triangleleft \mathsf{seq}\,1{:}C_1$. There are two cases. First, if $c = \mathsf{seq}\,1{:}c_1$ for some c_1 then $c_1 \in C_1 \setminus \mathrm{Tc}(t_0)$. According to the definition of β, if $\beta(c, c')$ then $c' = \mathsf{body}{:}c_1$ and hence $c' \in \mathsf{body}{:}C_1$. Second, if $c = (\mathsf{seq}\,1{:}c_1, \mathsf{seq}\,2{:}\mathsf{body}{:}c_2)$ for some c_1 and c_2, then $c_1 \in C_1 \cap \mathrm{Tc}(t_0)$ and $c_2 \in \mathrm{Tc}(t_0)$. According to the definition of β, if $\beta(c, c')$ then $c_2 = c_1$ and $c' = \mathsf{body}{:}c_1$, so $c' \in \mathsf{body}{:}C_1$ as required.

2. $\forall c' \in (\mathsf{body}{:}C_1) \exists! c \in (P \triangleleft \mathsf{seq}\,1{:}C_1)$ such that $\beta(c, c')$
   Let $c' \in \mathsf{body}{:}C_1$. There exists $c'_1 \in C_1$ such that $c' = \mathsf{body}{:}c'_1$. If $c'_1 \notin \mathrm{Tc}(t_0)$, then $\mathsf{seq}\,1{:}c'_1 \in P \triangleleft \mathsf{seq}\,1{:}C_1$ and is the unique condition c such that $\beta(c, c')$. If $c'_1 \in \mathrm{Tc}(t_0)$ then $(\mathsf{seq}\,1{:}c'_1, \mathsf{seq}\,2{:}\mathsf{body}{:}c'_1) \in P \triangleleft \mathsf{seq}\,1{:}C_1$, and according to the definition of β must be the unique condition c such that $\beta(c, c')$.

3. $\beta(P \triangleright \mathsf{seq}\,2{:}C_2) \subseteq C_2$
   Let $c \in P \triangleright \mathsf{seq}\,2{:}C_2$. There are two cases. First, if $c = \mathsf{seq}\,2{:}c_2$ for some c_2 then $c_2 \in C_2 \setminus \mathrm{Ic}(w)$, noting that $\mathrm{Ic}(w) = \mathsf{body}{:}\mathrm{Tc}(t_0)$. According to the definition of β, if we have $\beta(c, c')$ then $c' = c_2$ and hence $c' \in C_2$. If $c = (\mathsf{seq}\,1{:}c_1, \mathsf{seq}\,2{:}c_2)$ for some c_1 and c_2 then $c_2 \in C_2 \cap \mathrm{Ic}(w)$ and $c_1 \in \mathrm{Tc}(t_0)$. According to the definition of β, if we have $\beta(c, c')$ then $c_2 = \mathsf{seq}\,2{:}\mathsf{body}{:}c_1$ and $c' = \mathsf{body}{:}c_1$. Since $c' = c_2$, we have $c' \in C_2$ as required.

4. $\forall c' \in C_2 \exists! c \in (P \triangleright \mathsf{seq}\,2{:}C_2)$ such that $\beta(c, c')$
   Let $c' \in C_2$. If $c' \notin \mathrm{Ic}(w)$ then $\mathsf{seq}\,2{:}c' \in P \triangleright \mathsf{seq}\,2{:}C_2$ and $\beta(\mathsf{seq}\,2{:}c_2, c_2)$. Uniqueness is immediate from the definition of β. If $c' \in \mathrm{Ic}(w)$ then $c' = \mathsf{body}{:}c_0$ for some $c_0 \in \mathrm{Tc}(t_0)$. We have $(\mathsf{seq}\,1{:}c_0, \mathsf{seq}\,2{:}\mathsf{body}{:}c_0) \in P$ and hence $(\mathsf{seq}\,1{:}c_0, \mathsf{seq}\,2{:}\mathsf{body}{:}c_0) \in P \triangleright \mathsf{seq}\,2{:}C_2$. Uniqueness again follows from the definition of β.

We now consider openness of (η, β). Suppose that C is reachable from $\mathrm{Ic}(t_0; w)$ in $\mathcal{C}[\![t_0; w]\!]$ and that there exist e' and C'' such that $\beta C \xrightarrow{e'} C''$ in $\mathcal{C}[\![w]\!]$. We shall first show that there exist e and C' such that $C \xrightarrow{e} C'$ in $\mathcal{C}[\![t_0; w]\!]$ and $\eta(e) = e'$. It will follow from (η, β) being a morphism that $C'' = \beta C'$.

According to Lemma 3.6.2, there are two distinct cases for the marking C:

For the first case, assume that $C = P \triangleleft \mathsf{seq}\,1{:}C_0$ for some $C_0 \neq \mathrm{Tc}(t_0)$ reachable from $\mathrm{Ic}(t_0)$ in $\mathcal{C}\,[\![t_0]\!]$. We have $\beta C = \mathsf{body}{:}C_0$ and $\mathsf{body}{:}C_0 \stackrel{e'}{\longrightarrow\!\!\!\!\!\bullet}\; C''$ by (1) and (2) above. Since the net $\mathcal{C}\,[\![t_0]\!]$ is well-terminating, there exists $c_0 \in \mathrm{Tc}(t_0) \setminus C_0$, so, considering the events in $\mathcal{C}\,[\![w]\!]$, we must have $e' = \mathsf{body}{:}e_0$ for some $e_0 \in \mathrm{Ev}(t_0)$; that is, the event e' is in the body of the loop, not a loop entry or exit event. It follows that $C'' = \mathsf{body}{:}C_0'$ for some C_0' such that $C_0 \stackrel{e_0}{\longrightarrow\!\!\!\!\!\bullet}\; C_0'$ in $\mathcal{C}\,[\![t_0]\!]$. From Lemma 3.3.2, it follows that $C = P \triangleleft \mathsf{seq}\,1{:}C_0 \stackrel{P \triangleleft \mathsf{seq}\,1:e_0}{\longrightarrow\!\!\!\!\!\bullet}\; P \triangleleft \mathsf{seq}\,1{:}C_0'$ in $\mathcal{C}\,[\![t_0; w]\!]$. We have, by definition, $\eta(P \triangleleft \mathsf{seq}\,1{:}e_0) = \mathsf{body}{:}e_0$, as required to complete the case.

For the second case for C, assume that $C = P \triangleright \mathsf{seq}\,2{:}C_2$ for some C_2 reachable from $\mathrm{Ic}(w)$ in $\mathcal{C}\,[\![w]\!]$. We have $\beta C = C_2$ by (3) and (4) above, and therefore $C_2 \stackrel{e'}{\longrightarrow\!\!\!\!\!\bullet}\; C''$ in $\mathcal{C}\,[\![w]\!]$. From Lemma 3.3.2, it follows that $C = P \triangleright \mathsf{seq}\,2{:}C_2 \stackrel{P \triangleright \mathsf{seq}\,2:e'}{\longrightarrow\!\!\!\!\!\bullet}\; P \triangleright \mathsf{seq}\,2{:}C''$ in $\mathcal{C}\,[\![t_0; w]\!]$. We have $\eta(P \triangleright \mathsf{seq}\,2{:}e') = e'$ by definition, as required to complete the case.

Finally, to complete the proof of openness and complete the theorem, we must show that (η, β) reflects consecutive independence. That is, for any marking C reachable from $\mathrm{Ic}(t_0; w)$, if $C \stackrel{e}{\longrightarrow\!\!\!\!\!\bullet}\; C' \stackrel{e'}{\longrightarrow\!\!\!\!\!\bullet}$ and $\eta(e) I_{\mathcal{C}[\![w]\!]} \eta(e')$ then $e I_{\mathcal{C}[\![t_0;w]\!]} e'$. The proof is straightforward if either there exist e_1 and e_1' such that $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ and $e' = P \triangleleft \mathsf{seq}\,1{:}e_1'$ or if there exist e_2 and e_2' such that $e = \mathsf{seq}\,2{:}e_2$ and $e' = \mathsf{seq}\,2{:}e_2'$. The final remaining case, according to Lemma 3.6.2, is if $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ and $e' = P \triangleright \mathsf{seq}\,2{:}e_2$. That is, inside $t_0; w$ the event e is an event from t_0 and e' is an event in the loop w. In this case, according to Lemma 3.6.2, $C' = P$ and $C = P \triangleleft \mathsf{seq}\,1{:}C_1$ for some C_1 such that $C_1 \stackrel{e_1}{\longrightarrow\!\!\!\!\!\bullet}\; \mathrm{Tc}(t_0)$ in $\mathcal{C}\,[\![t_0]\!]$, and $\mathrm{Ic}(w) \stackrel{e_2}{\longrightarrow\!\!\!\!\!\bullet}$ in $\mathcal{C}\,[\![w]\!]$. From Lemma 3.5.2, there exists $c_1 \in e_1{}^\bullet \cap \mathrm{Tc}(t_0)$. From the definition of the events of w, we have ${}^\bullet e_2 = \mathsf{body}{:}\mathrm{Tc}(t_0)$. It follows that, in $\mathcal{C}\,[\![w]\!]$, the events $\eta(e)$ and $\eta(e')$ are not independent. □

We now show that the transition semantics embeds into the net semantics, by showing that if $\langle t, \sigma\rangle \stackrel{\lambda}{\longrightarrow} \langle t', \sigma'\rangle$ then there exists an event e such that $\mathcal{N}\,[\![t]\!]{:}(\mathrm{Ic}(t), \sigma) \stackrel{e}{\longrightarrow\!\!\!\!\!\bullet}\; (C', \sigma')$ and $(\mathcal{C}\,[\![t]\!], C') \sim_{\mathsf{cc}} (\mathcal{C}\,[\![t']\!], \mathrm{Ic}(t'))$. As discussed, we wish to exhibit the bisimulation by giving just one open morphism

$$(\mathcal{C}\,[\![t']\!], \mathrm{Ic}(t')) \multimap (\mathcal{C}\,[\![t]\!], C').$$

Unfortunately, structural equivalence proves to be a slight obstacle here since it is not in general the case that if $t_1 \equiv t_2$ then there exist open morphisms in both directions between $(\mathcal{C}\,[\![t_1]\!], \mathrm{Ic}(t_1))$ and $(\mathcal{C}\,[\![t_2]\!], \mathrm{Ic}(t_2))$. To deal with structural equivalence, we shall therefore be forced to use spans of open maps. For the time being, however, we side-step this issue and prove the result 'up to' structural equivalence of terms.

Lemma E.2.2. *If $\langle t, \sigma\rangle \stackrel{\lambda}{\longrightarrow} \langle t', \sigma'\rangle$ then there exist t_0, t_0', C' and e such that $t \equiv t_0$ and $t' \equiv t_0'$ and $\lambda = |e|$ and $\mathcal{N}\,[\![t_0]\!]{:}(\mathrm{Ic}(t_0), \sigma) \stackrel{e}{\longrightarrow\!\!\!\!\!\bullet}\; (C', \sigma')$, and there is a synchronous* **Pom***-open morphism in the category* **Safe**

$$(\eta, \beta){:}(\mathcal{C}\,[\![t_0']\!], \mathrm{Ic}(t_0')) \multimap (\mathcal{C}\,[\![t_0]\!], C').$$

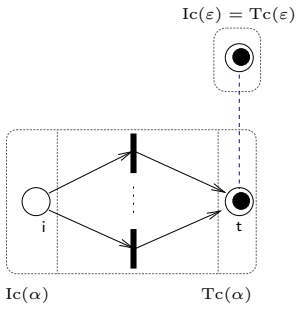
Proof. The proof is by induction on the rules for $\langle t, \sigma\rangle \stackrel{\lambda}{\longrightarrow} \langle t', \sigma'\rangle$.

(Act): We have $\langle \alpha, (D, L, R, N)\rangle \stackrel{\mathsf{act}(D_1, D_2)}{\longrightarrow} \langle \varepsilon, (D', L, R, N)\rangle$ for some D_1 and D_2 such that $(D_1, D_2) \in \mathcal{A}\,[\![\alpha]\!]$ and $D_1 \subseteq D$ and $D' = D \setminus D_1 \cup D_2$. Therefore, in the net

$\mathcal{N}[\![\alpha]\!]$ we have

$$(\mathrm{Ic}(\alpha),(D,L,R,N)) \overset{\mathsf{act}_{(\mathrm{Ic}(\alpha),\mathrm{Tc}(\alpha))}(D_1,D_2)}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathrm{Tc}(\alpha),(D',L,R,N))$$

The morphism depicted below from $(\mathcal{C}[\![\varepsilon]\!],\mathrm{Ic}(\varepsilon))$ to $(\mathcal{C}[\![\alpha]\!],\mathrm{Tc}(\alpha))$ is clearly open since no event in $\mathrm{Ev}(\alpha)$ has concession in the marking $\mathrm{Tc}(\alpha)$.



(Alloc),(Dealloc),(Rel),(End): Similar to (Act).

(Seq): Suppose that $\langle t_1;t_2,\sigma\rangle \xrightarrow{\lambda} \langle t_1';t_2,\sigma'\rangle$ because $\langle t_1,\sigma\rangle \overset{\lambda}{\longrightarrow\!\!\!\!\!\rightarrow} \langle t_1',\sigma'\rangle$. By induction, there exist t_0,t_0',C' and e such that $t_0 \equiv t_1$ and $t_0' \equiv t_1'$ and $|e| = \lambda$ and $\mathcal{N}[\![t_0]\!]:(\mathrm{Ic}(t_0),\sigma) \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} (C',\sigma')$. By Lemma 3.3.2, in $\mathcal{N}[\![t_0;t_2]\!]$ we have $(\mathrm{Ic}(t_0;t_2),\sigma) \overset{P\triangleleft\mathsf{seq}\,1:e}{\longrightarrow\!\!\!\!\!\rightarrow} (P\triangleleft\mathsf{seq}\,1{:}C',\sigma')$ where $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_0)\times\mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$. From the induction hypothesis, there also exists an open morphism $(\eta_0,\beta_0){:}(\mathcal{C}[\![t_0']\!],\mathrm{Ic}(t_0')) \multimap (\mathcal{C}[\![t_0]\!],C')$. From Lemma E.1.5, there exists an open morphism $(\eta,\beta){:}(\mathcal{C}[\![t_0';t_2]\!],\mathrm{Ic}(t_0';t_2)) \multimap (\mathcal{C}[\![t_0;t_2]\!],P\triangleleft\mathsf{seq}\,1{:}C')$. Since we have $t_1;t_2 \equiv t_0;t_2$ and $t_1';t_2 \equiv t_0';t_2$ from $\equiv$ being a congruence, the case is complete.

(Par-1): Suppose that $\langle t_1 \parallel t_2,\sigma\rangle \xrightarrow{\lambda} \langle t_1' \parallel t_2,\sigma'\rangle$ because $\langle t_1,\sigma\rangle \xrightarrow{\lambda} \langle t_1',\sigma'\rangle$. By induction, there exist t_0,t_0',C and e such that $t_0 \equiv t_1$ and $t_0' \equiv t_1'$ and $|e| = \lambda$ and

$$\mathcal{N}[\![t_0]\!]:(\mathrm{Ic}(t_0),\sigma) \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} (C,\sigma'),$$

and there is a synchronous open morphism

$$(\mathcal{C}[\![t_0']\!],\mathrm{Ic}(t_0')) \multimap (\mathcal{C}[\![t_0]\!],C).$$

From Lemma E.1.6, there is a synchronous open morphism

$$(\mathcal{C}[\![t_0' \parallel t_2]\!],\mathrm{Ic}(t_0' \parallel t_2)) \multimap (\mathcal{C}[\![t_0 \parallel t_2]\!],\mathsf{par}\,1{:}C \cup \mathsf{par}\,2{:}\mathrm{Ic}(t_2)).$$

Since $\equiv$ is a congruence, we have $t_1 \parallel t_2 \equiv t_0 \parallel t_2$ and $t_1' \parallel t_2 \equiv t_0' \parallel t_2$, so the case is complete.

(Par-2): Similar.

(Sum-1): Let $\sigma = (D,L,R,N)$ and, for brevity, let t denote the term $\alpha_1.t_1 + \alpha_2.t_2$. Suppose that $\langle t,\sigma\rangle \xrightarrow{\lambda} \langle t_1,\sigma'\rangle$ because $\langle \alpha_1,\sigma\rangle \xrightarrow{\lambda} \langle \varepsilon,\sigma'\rangle$. An induction on the rules for the transition relation $\xrightarrow{\cdot}$ gives the following property:

For any t_0 and $\sigma_0 = (D_0, L_0, R_0, N_0)$, if $t_0 \equiv \alpha$ and $\langle t_0, \sigma_0\rangle \xrightarrow{\lambda_0} \langle t_0', \sigma_0'\rangle$ then there exist D_1 and D_2 such that $(D_1, D_2) \in \mathcal{A}[\![\alpha]\!]$ and $\lambda_0 = \mathsf{act}(D_1, D_2)$ and $D_1 \subseteq D_0$ and $\sigma_0' = (D_0 \setminus D_1 \cup D_2, L_0, R_0, N_0)$.

From the above claim, there exist D_1 and D_2 such that $(D_1, D_2) \in \mathcal{A}[\![\alpha]\!]$ and $\lambda = \mathsf{act}(D_1, D_2)$ and $D_1 \subseteq D$ and $\sigma' = (D \setminus D_1 \cup D_2, L, R, N)$. Let $P = \mathsf{sum}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{sum}\,2{:}\mathrm{Tc}(t_2)$. In the net $\mathcal{N}[\![t]\!]$, we have

$$(\mathrm{Ic}(t), \sigma) \overset{\mathsf{act}_{(\mathrm{Ic}(t), P \triangleleft \mathsf{sum}\,1:\mathrm{Ic}(t_1))}(D_1, D_2)}{\longrightarrow\!\!\!\!\!\rightarrow} (P \triangleleft \mathsf{sum}\,1{:}\mathrm{Ic}(t_1), \sigma').$$

We need an **Pom**-open morphism

$$(\eta, \beta){:}(\mathcal{C}[\![t_1]\!], \mathrm{Ic}(t_1)) \multimap (\mathcal{C}[\![t]\!], P \triangleleft \mathsf{sum}\,1{:}\mathrm{Ic}(t_1)).$$

Define $\eta(e) = P \triangleleft \mathsf{sum}\,1{:}e$ and

$$\beta(c, c') \iff \begin{cases} c \notin \mathrm{Tc}(t_1) \;\&\; c' = \mathsf{sum}\,1{:}c \\ c \in \mathrm{Tc}(t_1) \;\&\; \exists c_2 \in \mathrm{Tc}(t_2) :\; c' = (\mathsf{sum}\,1{:}c, \mathsf{sum}\,2{:}c_2) \end{cases}.$$

It is an easy calculation to show that $\beta C = P \triangleleft \mathsf{sum}\,1{:}C$ for any subset of control conditions C, and therefore that (η, β) is a morphism.

For openness of the morphism (η, β), suppose that C is reachable from $\mathrm{Ic}(t_1)$ in $\mathcal{C}[\![t_1]\!]$ and $\beta C = P \triangleleft \mathsf{sum}\,1{:}C \overset{e'}{\longrightarrow\!\!\!\!\!\rightarrow} C'$ for some e' and C'. It follows from Lemma 3.6.7 that there exist e and C_1 such that $e' = P \triangleleft \mathsf{seq}\,1{:}e$ and $C \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} C_1$ in $\mathcal{C}[\![t_1]\!]$ and $C' = P \triangleleft \mathsf{sum}\,1{:}C_1 = \beta C_1$. The morphism (η, β) reflects the independence of consecutive events since $e I_{\mathcal{C}[\![t_1]\!]} e'$ iff $(P \triangleleft \mathsf{sum}\,1{:}e) I_{\mathcal{C}[\![t]\!]} (P \triangleleft \mathsf{sum}\,1{:}e')$.

(Sum-2): Symmetric.

(While): Let t abbreviate the term `while` b `do` t_0 `od` and let $\sigma = (D, L, R, N)$. Suppose that $\langle$`while` b `do` t_0 `od`$, \sigma\rangle \xrightarrow{\lambda} \langle t_0; t, \sigma'\rangle$. As in the case for (Sum-1) but recalling that b ranges over booleans, it must be the case that there exists D_0 such that $\lambda = \mathsf{act}(D_0, D_0)$ and $D_0 \subseteq D$ and $\sigma = \sigma'$. In the net $\mathcal{N}[\![t]\!]$, we therefore have

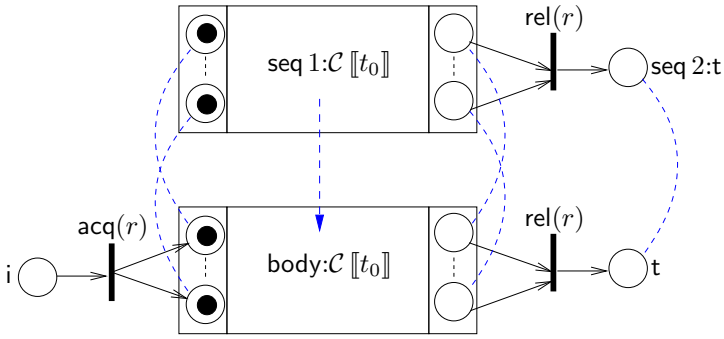
$$(\mathrm{Ic}(t), \sigma) \overset{\mathsf{act}_{(\mathrm{Ic}(t), \mathsf{body}:\mathrm{Ic}(t_0))}(D_0, D_0)}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{body}{:}\mathrm{Ic}(t_0), \sigma).$$

The required morphism $(\mathcal{C}[\![t_0; t]\!], \mathrm{Ic}(t_0; t)) \multimap (\mathcal{C}[\![t]\!], \mathsf{body}{:}\mathrm{Ic}(t_0))$ is described in Lemma E.2.1.

(With): Let $t =$ `with` r `do` t_0 `od` and suppose that $\langle t, \sigma\rangle \xrightarrow{\lambda} \langle t_0;$`rel` $r, \sigma'\rangle$ for $\sigma = (D, L, R, N)$ and $\sigma' = (D, L, R \setminus \{r\}, N)$ and $r \in R$. We have

$$\mathcal{N}[\![t]\!] : (\mathrm{Ic}(t), \sigma) \overset{e}{\longrightarrow\!\!\!\!\!\rightarrow} (\mathsf{body}{:}\mathrm{Ic}(t_0), \sigma')$$

for $e = \mathsf{acq}_{(\mathrm{Ic}(t), \mathsf{body}:\mathrm{Ic}(t_0))}(r)$. The net $\mathcal{C}[\![t]\!]$ in initial marking $\mathsf{body}{:}\mathrm{Ic}(t_0)$ with the event e removed is easily seen to be isomorphic to the net $\mathcal{C}[\![t_0;$`rel` $r]\!]$. It is easy to see that this gives rise to a synchronous open morphism since, by Lemma 3.6.13, the event e cannot occur in any marking reachable from $\mathsf{body}{:}\mathrm{Ic}(t_0)$ in $\mathcal{C}[\![t]\!]$. The morphism may be drawn as:



(RES): Similar to (WITH).

(EQUIV): Suppose that $\langle t, \sigma\rangle \xrightarrow{\lambda} \langle t', \sigma'\rangle$ because there exist t_0 and t_0' such that $t \equiv t_0$ and $t' \equiv t_0'$ and $\langle t_0, \sigma\rangle \xrightarrow{\lambda} \langle t_0', \sigma'\rangle$. By induction, there exist t_1 and t_1' such that $t_0 \equiv t_1$ and $t_0 \equiv t_1'$ and $\langle t_1, \sigma\rangle \xrightarrow{\lambda} \langle t_1', \sigma'\rangle$. Structural equivalence $\equiv$ is transitive, so we have $t \equiv t_1$ and $t' \equiv t_1'$, completing the case and the proof. □

We now show the reverse property, showing that the net semantics embeds into the transition semantics. This time, we do not need to work up to structural equivalence of terms.

Lemma E.2.3. *If* $\mathcal{N}[\![t]\!]:(\mathrm{Ic}(t), \sigma) \xrightarrow{e} (C', \sigma')$ *then there exists* t' *such that* $\langle t, \sigma\rangle \xrightarrow{|e|} \langle t', \sigma'\rangle$*, and there is a* **Pom***-open morphism in the category* **Safe**

$$(\eta, \beta):(\mathcal{C}[\![t']\!], \mathrm{Ic}(t')) \multimap (\mathcal{C}[\![t]\!], C').$$

Proof. Let $\sigma = (D, L, R, N)$ and $\sigma' = (D', L', R', N')$. The proof is by induction on the size of terms, considering each case for the term t separately. All cases but that for sequential composition are straightforward analyses of the events in the net $\mathcal{N}[\![t]\!]$, using the open morphisms defined in Lemma E.2.2. We therefore consider only the sequential composition.

Recall that $\mathrm{Ic}(t_1; t_2) = P \triangleleft \mathsf{seq}\,1{:}\mathrm{Ic}(t_1)$, where $P = \mathsf{seq}\,1{:}\mathrm{Tc}(t_1) \times \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$. Suppose that $t = t_1; t_2$ and that $\mathcal{N}[\![t_1; t_2]\!]:(\mathrm{Ic}(t_1; t_2), \sigma) \xrightarrow{e} (C', \sigma')$. According to the definition of $\mathrm{Ev}(t_1; t_2)$, there are two cases to consider:

If $e = P \triangleleft \mathsf{seq}\,1{:}e_1$ for some $e_1 \in \mathrm{Ev}(t_1)$ then $|e| = |e_1|$ and, according to Lemma 3.6.1 with Lemma 3.7.1, there exists C_1' such that $\mathcal{N}[\![t_1]\!]:(\mathrm{Ic}(t_1), \sigma) \xrightarrow{e_1} (C_1', \sigma')$ and $C' = P \triangleleft \mathsf{seq}\,1{:}C_1'$. By induction, there exists t_1' such that $\langle t_1, \sigma\rangle \xrightarrow{|e_1|} \langle t_1', \sigma'\rangle$ and there is a **Pom**-open map $(\eta_1, \beta_1):(\mathcal{C}[\![t_1']\!], \mathrm{Ic}(t_1')) \multimap (\mathcal{C}[\![t_1]\!], C_1')$. By (SEQ), we have $\langle t_1; t_2, \sigma\rangle \xrightarrow{|e_1|} \langle t_1'; t_2, \sigma'\rangle$, and by Lemma E.1.5 there is an open map $(\eta, \beta):(\mathcal{C}[\![t_1'; t_2]\!], \mathrm{Ic}(t_1'; t_2)) \multimap (\mathcal{C}[\![t_1; t_2]\!], P \triangleleft \mathsf{seq}\,1{:}C_1')$, as required.

If $e = P \triangleright \mathsf{seq}\,2{:}e_2$ for some $e_2 \in \mathrm{Ev}(t_2)$ then $|e| = |e_2|$ and, according to Lemma 3.6.1, we must have $\mathrm{Ic}(t_1) = \mathrm{Tc}(t_1)$. It follows that $\mathrm{Ic}(t_1; t_2) = P \triangleright \mathsf{seq}\,2{:}\mathrm{Ic}(t_2)$. It also follows from Lemma 3.6.1 that there exists C_2' such that $\mathcal{N}[\![t_2]\!]:(\mathrm{Ic}(t_2), \sigma) \xrightarrow{e_2} (C_2', \sigma')$. By induction, there exists t_2' such that $\langle t_2, \sigma\rangle \xrightarrow{|e_2|} \langle t_2', \sigma'\rangle$ and there is a **Pom**-open morphism $(\eta_2, \beta_2):(\mathcal{C}[\![t_2']\!], \mathrm{Ic}(t_2')) \multimap (\mathcal{C}[\![t_2]\!], C_2')$. Since $\mathrm{Ic}(t_1) = \mathrm{Tc}(t_1)$, we have $t_1 \equiv \varepsilon$ by Lemma 3.5.3 and so $t_1; t_2 \equiv t_2$. Hence, using rule (EQUIV), we may derive $\langle t_1; t_2, \sigma\rangle \xrightarrow{|e_2|} \langle t_2', \sigma'\rangle$. To complete the proof, it can be shown that $(\eta, \beta):(\mathcal{C}[\![t_1; t_2']\!], \mathrm{Ic}(t_1; t_2')) \multimap (\mathcal{C}[\![t_1; t_2]\!], P \triangleright$

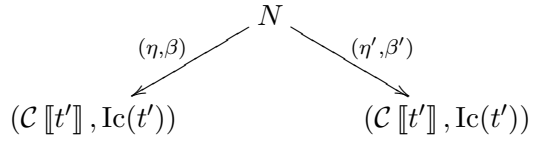
$\mathsf{seq}\,2{:}C_2'$) defined as

$$\begin{array}{rcl} \eta(e) & = & P \triangleright \mathsf{seq}\,2{:}e \\ \beta(c,c') & \text{iff} & \left\{ \begin{array}{ll} \text{either} & \exists c_2 :\ \beta_2(c,c_2)\ \&\ c' = \mathsf{seq}\,2{:}c_2\ \&\ c_2 \notin \mathrm{Ic}(t_2) \\ \text{or} & \exists c_1, c_2 :\ \beta_2(c,c_2)\ \&\ c' = (\mathsf{seq}\,1{:}c_1, \mathsf{seq}\,2{:}c_2) \\ & \qquad\qquad \&\ c_1 \in \mathrm{Tc}(t_1)\ \&\ c_2 \in \mathrm{Ic}(t_2) \end{array} \right. \end{array}$$

is a **Pom**-open morphism; the proof is similar to but easier than the proof of Lemma E.1.5. □

The final requirement before concluding this section is to show that structurally equivalent terms are bisimilar. Here, as mentioned earlier, we are forced to show the existence of a span of open morphisms rather than just show the existence of one open morphism.

Lemma E.2.4. *If $t \equiv t'$ then there is a span of synchronous* **Pom***-open morphisms*



from a congruence spanning net N, i.e. $\mathcal{C}\,[\![t]\!] \sim_{\mathsf{cc}} \mathcal{C}\,[\![t']\!]$

Proof. In Definition 3.2.1, structural equivalence $\equiv$ is defined inductively. The proof is performed by induction on the definition.

The only interesting rule for the relation $\equiv$ being an equivalence relation is transitivity, *viz.* if $t \equiv t'$ because there exists t_0 such that $t \equiv t_0$ and $t_0 \equiv t'$. In this case, the span for t and t_0 is composed with the span for t_0 and t' by taking the pullback of the morphisms into $(\mathcal{C}\,[\![t_0]\!], \mathrm{Ic}(t_0))$. The pullback obtained is a congruence spanning net with synchronous morphisms according to Lemma E.1.9.

Symmetry and associativity of the operators is easily dealt with since the nets are isomorphic (*e.g.* for associativity of sequential composition, there exists a span relating $\mathcal{C}\,[\![(t_1;t_2);t_3]\!]$ and $\mathcal{C}\,[\![t_1;(t_2;t_3)]\!]$ since the two nets are isomorphic).

The 'congruence' rules for $\equiv$ follow from Lemmas E.1.5 and E.1.6.

The only two remaining cases are $\varepsilon;t \equiv t$ and $\varepsilon \parallel t \equiv t$. The first span arises from the isomorphism $\mathcal{C}\,[\![\varepsilon;t]\!] \cong \mathcal{C}\,[\![t]\!]$. The second span arises from the morphism from the net $\mathcal{C}\,[\![\varepsilon \parallel t]\!]$ into $\mathcal{C}\,[\![t]\!]$ being essentially isomorphic apart from the net $\mathcal{C}\,[\![\varepsilon \parallel t]\!]$ having an initially-marked condition that is a not a pre- or post-condition of any event in $\mathcal{C}\,[\![t]\!]$. There is therefore a **Pom**-open morphism

$$(\mathcal{C}\,[\![\varepsilon \parallel t]\!], \mathrm{Ic}(\varepsilon \parallel t)) \multimap (\mathcal{C}\,[\![t]\!], \mathrm{Ic}(t))$$

that relates this redundant condition to no condition in $\mathcal{C}\,[\![t]\!]$. □

We briefly observe the following simple property about open maps of nets, in which (N, M) means the net with conditions and events as in N but with initial marking M.

Lemma E.2.5. *Let $(\eta,\beta){:}(N_1,M_1) \to (N_2,M_2)$ be a* **Pom***-open synchronous morphism. If $N_1{:}M_1 \xrightarrow{e_1} M_1'$ and $N_2{:}M_2 \xrightarrow{\eta(e_1)} \beta M_1'$ then $(\eta,\beta){:}(N_1,M_1') \to (N_2,M_2')$ is a* **Pom***-open synchronous morphism.*

Proof. Follows straightforwardly from Lemmas 2.3.2 and 2.4.1. □

We now arrive at the key result of this section, that the operational semantics corresponds to the net semantics.

Theorem E.2 (Correspondence)**.** *For any closed term t:*

- *If $\langle t, \sigma\rangle \xrightarrow{\lambda} \langle t', \sigma'\rangle$ then there exist C' and e such that $|e| = \lambda$ and $\mathcal{N}[\![t]\!]:(\mathrm{Ic}(t), \sigma) \xrightarrow{e} (C', \sigma)$ and $(\mathcal{C}[\![t']\!], \mathrm{Ic}(t')) \sim_{\mathsf{cc}} (\mathcal{C}[\![t]\!], C')$.*
- *If $\mathcal{N}[\![t]\!]:(\mathrm{Ic}(t), \sigma) \xrightarrow{e} (C', \sigma')$ then there exists a closed term t' such that $\langle t, \sigma\rangle \xrightarrow{|e|} \langle t', \sigma'\rangle$ and $(\mathcal{C}[\![t']\!], \mathrm{Ic}(t')) \sim_{\mathsf{cc}} (\mathcal{C}[\![t]\!], C')$.*

Proof. The second part is a re-statement of Lemma E.2.3.

For the first part, suppose that $\langle t, \sigma\rangle \xrightarrow{\lambda} \langle t', \sigma'\rangle$. According to Lemma E.2.2, there exist t_0 and t_0' such that $t \equiv t_0$ and $t' \equiv t_0'$ and $\mathcal{N}[\![t_0]\!]:\mathrm{Ic}(t_0) \xrightarrow{e_0} C_0'$ for some e_0 and C_0' such that $|e_0| = \lambda$ and there exists a synchronous **Pom**-open map $(\pi, \gamma):(\mathcal{C}[\![t_0']\!], \mathrm{Ic}(t_0')) \multimap (\mathcal{C}[\![t_0]\!], C_0')$. From Lemma E.2.4, there are spans of synchronous morphisms from congruence spanning nets

$$(\mathcal{C}[\![t]\!], \mathrm{Ic}(t)) \xleftarrow{(\eta,\beta)} N \xrightarrow{(\eta_0,\beta_0)} (\mathcal{C}[\![t_0]\!], \mathrm{Ic}(t_0)) \qquad (\mathcal{C}[\![t']\!], \mathrm{Ic}(t')) \xleftarrow{(\eta',\beta')} N' \xrightarrow{(\eta_0',\beta_0')} (\mathcal{C}[\![t_0']\!], \mathrm{Ic}(t_0')).$$

From Lemma E.2.5, since $\mathcal{N}[\![t_0]\!]:\mathrm{Ic}(t_0) \xrightarrow{e_0} C_0'$, there exists C and an event e such that $N:\mathrm{Ic}(N) \xrightarrow{e} C$ and $\eta_0(e) = e_0$ and $\beta_0 C = C_0'$ and $(\eta_0, \beta_0):(N, C) \multimap (\mathcal{C}[\![t_0]\!], C_0')$ is **Pom**-open. Recall that (N, C) is the net N with initial marking C. It is easy to see from Definition E.1.1 that (N, C) is a congruence spanning net since N is. We therefore have a congruence span

$$(\mathcal{C}[\![t]\!], \beta C) \xleftarrow{(\eta,\beta)} (N, C) \xrightarrow{(\eta_0,\beta_0)} (\mathcal{C}[\![t_0]\!], C_0').$$

The required span is then obtained by composing (by taking successive pullbacks) the following spans:

$$(\mathcal{C}[\![t]\!], \beta C) \xleftarrow{(\eta,\beta)} (N, C) \xrightarrow{(\eta_0,\beta_0)} (\mathcal{C}[\![t_0]\!], C_0') \xleftarrow{(\pi,\gamma)} (\mathcal{C}[\![t_0']\!], \mathrm{Ic}(t_0')) \xrightarrow{\mathsf{id}} (\mathcal{C}[\![t_0']\!], \mathrm{Ic}(t_0')) \xleftarrow{(\eta_0',\beta_0')} N' \xrightarrow{(\eta',\beta')} (\mathcal{C}[\![t']\!], \mathrm{Ic}(t'))$$

Pullbacks of congruence spanning nets are congruence spanning nets according to Lemma E.1.9, so we have $(\mathcal{C}[\![t]\!], \beta C) \sim_{\mathsf{cc}} (\mathcal{C}[\![t']\!], \mathrm{Ic}(t')))$, as required. □

As discussed in Chapter 2, it is shown in [NW96] that **Pom**-open map bisimilarity of nets $(\mathcal{C}[\![t_1]\!], C_1)$ and $(\mathcal{C}[\![t_2]\!], C_2)$ is equivalent to them being related through a strong history-preserving bisimulation. Strong history-preserving bisimulation is a strengthening of the standard form of bisimulation for transition systems to account for the independence of events. It follows immediately that the transition systems arising from the nets $(\mathcal{N}[\![t_1]\!], C_1, \sigma)$ and $(\mathcal{N}[\![t_2]\!], C_2, \sigma)$ are bisimilar in the usual sense.

Let us write $\langle t, \sigma\rangle \sim \langle t', \sigma'\rangle$ if the transition system derived according to the transition semantics from $\langle t, \sigma\rangle$ is bisimilar, in the usual sense, to that obtained from $\langle t', \sigma'\rangle$. From the correspondence theorem and the fact that strong history-preserving bisimulation is stronger than standard bisimulation, we obtain adequacy of our semantics:

Corollary E.3 (Adequacy)**.** *Let t and t' be closed terms. If $(\mathcal{C}[\![t]\!], \mathrm{Ic}(t)) \sim_{\mathsf{cc}} (\mathcal{C}[\![t']\!], \mathrm{Ic}(t'))$ then $\langle t, \sigma\rangle \sim \langle t', \sigma'\rangle$ for any consistent state σ.*

The converse property with respect to $\sim$ on the operational semantics fails if we keep the equivalence $\sim_{\mathsf{cc}}$ on nets. For instance, for any σ we have

$$(\alpha_1 \parallel \alpha_2, \sigma) \sim (\alpha_1.\alpha_2 + \alpha_2.\alpha_1, \sigma).$$

However, we can see that

$$(\mathcal{C}[\![\alpha_1 \parallel \alpha_2]\!], \mathrm{Ic}(\alpha_1 \parallel \alpha_2)) \not\sim_{\mathsf{cc}} (\mathcal{C}[\![\alpha_1.\alpha_2 + \alpha_2.\alpha_1]\!], \mathrm{Ic}(\alpha_1.\alpha_2 + \alpha_2.\alpha_1)).$$

The reason why the property fails, apart from $\sim$ not being a congruence, is that the transition system does not capture the independence of actions whereas the equivalence $\sim_{\mathsf{cc}}$ does. This would be resolved by defining independence alongside the operational semantics, thereby defining a transition system with independence [WN95]. The purpose of this section has, however, been to relate the net semantics to the standard transition semantics for the language, so we shall refrain from doing this at the present time.