

Events in Computation

Glynn Winskel

Degree: Doctor of Philosophy
University of Edinburgh
1980



ABSTRACT

This thesis demonstrates how general and fundamental is the notion of event in the theory of computation. It points the way to a more complete theory of events.

The central idea is that of event structures consisting of relations on sets of events. Event structures are accompanied by an idea of state called configuration. They model the behaviour of computations in time. To reflect this finiteness restrictions are appropriate.

Using event structures as an intermediary the approaches of net theory and denotational semantics are related. This is formalised by representation theorems which express mathematically the translation between equivalent though apparently very different descriptions. In this way, for example, the net theoretic notion of confusion is related to concrete domains while using natural ideas of state of event structures Petri's finiteness axiom of K-density on causal nets is assessed as too restrictive and accordingly his formulation of state, as a case, too wide.

Apart from their unifying role event structures are important in themselves because of their abstract yet intuitive and operational nature. Their range of importance is widened considerably by the demonstration that event structures may represent functions of arbitrary type - rather abstract objects - while still preserving their operational nature. This is achieved by relating event structures to the bidomains of Berry.

Acknowledgements

It is hard to express sufficiently how much I owe to my supervisor Gordon Plotkin; his guidance, encouragement, advice and more detailed suggestions have had a fundamental influence on the development of this thesis. I am grateful to Mogens Nielsen for many helpful ideas. Mogens, Gordon and I collaborated in writing [Nie] (see the references) and its follow-up, yet to be published. In addition I have benefited from ideas suggested by Gérard Berry, Pierre Louis Curien, Robin Milner and Pippo Torrigiani. More generally I would like to thank all those involved in the theory of computation post-graduate course (77-78) here in Edinburgh which as well as providing a broad background knowledge, delineated the boundaries of research, so important to a student beginning research. Thank you as well to DAIMI, Aarhus for their hospitality when Aarhus became a second home. I am very thankful to Eleanor Kerse for her skill and patience in typing a long difficult manuscript and to Heather Carlin for finishing the job. Thank you so much to Kirsten whose sympathetic understanding helped me through the difficult patches and made the writing easier.

The work here was supported in part by an SRC studentship and in part by an SRC grant under the direction of Robin Milner and Gordon Plotkin.

Declaration

This thesis was composed by myself. Chapter 4 and some parts of chapter 5 are essentially [Nie] a paper produced in collaboration with Mogens Nielsen and Gordon Plotkin; otherwise the work is my own, under the guidance of my supervisor Gordon Plotkin.

Contents

<u>Chapter 1</u>	<u>Introduction</u>	1
1.1	Basic ideas	1
1.2	Events in context	5
1.3	Summary	9
<u>Chapter 2</u>	<u>Introduction to Petri nets</u>	16
2.1	Basic definitions	16
2.2	Transition nets	17
2.3	Examples of modelling computations by transition nets	22
2.3A	Modelling Milner nets by transition nets	22
2.3B	Transition nets as datatypes	29
2.3C	Modelling Kahn-MacQueen networks by transition nets	31
2.4	Causal nets, cases and K-density	38
2.5	Net morphisms	45
<u>Chapter 3</u>	<u>Introduction to concrete domains and sequentiality</u>	51
3.1	Background material	51
3.2	Concrete domains, matrices and sequential functions	61
3.3	The representation theorem	75
<u>Chapter 4</u>	<u>Petri nets give Scott domains</u>	91
4.1	Causal nets, elementary event structures and lattices	91
4.2	Occurrence nets, event structures and domains	102
<u>Chapter 5</u>	<u>States and observable states</u>	116
5.1	Observers, states and observable states	116
5.2	Distance measures on events and states	120
5.3	Event structures with finite width and finite branching	125
5.4	States of occurrence nets and K-density	140
5.5	Confusion and concrete domains	145
5.6	Alternative axioms on event structures and other ideas of observable state	150
<u>Chapter 6</u>	<u>Conditions</u>	156
6.1	Conditions of an event structure	156
6.2	Expressiveness	175
6.3	The constructions \mathcal{N} and \mathcal{N} give maximally expressive nets	185
6.4	Restless events	192

<u>Chapter 7</u>	<u>Event structures with infinite pasts</u>	197
7.1	Observers and observable states	198
7.2	Reachability classes	205
7.3	An axiomatisation of the reachability class	209
7.4	Causal nets representing processes with infinite pasts and K-density	211
<u>Chapter 8</u>	<u>The full-abstractness problem for PCF - an introduction</u>	213
8.1	The problem	213
8.2	The work of Gérard Berry	218
<u>Chapter 9</u>	<u>Higher-type event structures</u>	230
9.1	Introducing higher type event structures	230
9.2	Stable event structures	237
9.3	Stable configurations	239
9.4	Images of M are configurations, some "staircase" orderings	244
9.5	The structure $(R(E), \sqsubseteq^R)$	250
9.6	The structure $(R(E), \sqsubseteq^L)$	253
9.7	Stable exponentiation and product of event structures	258
9.8	The category of stable event structures	265
9.9	Cartesian-closed categories of domains	274
9.10	Sequential configurations	278
<u>Chapter 10</u>	<u>Conclusion</u>	281
10.1	Achievements	281
10.2	Problems	282
10.3	Future work	284
<u>References</u>		287

Chapter 1. Introduction

The idea of an event in computer science arises in the work of many different authors sometimes with different aims in mind (for example in distributed computing with [Pet], [Hew] and [Lam], and in denotational semantics with [Kah and Plo]). This thesis examines the role of events, teasing-out the concept where it occurs implicitly and relating sometimes apparently divergent approaches. In nature the thesis is exploratory, and consequently a little unbalanced, but it is hoped that it will at least help towards an appreciation of the important role events can play in the theory of computation. I see the work here as a step on the way to a theory of events in computation. Such a theory, important in its own right, would have a strong unifying influence in the theory of computation.

1.1 Basic ideas

This section is an informal introduction to those basic and general ideas which guide and continually appear as this thesis develops.

What is an event in computation? Many examples will be given; typical are acts of synchronisation between computing agents operating concurrently, and atomic actions of input or output. Just as in physics, what is considered to be an event depends on how abstract is the level of description. The creation of a supernova, the collision of two billiard balls, the communication of two agents in a Milner net are all regarded as events but at very different levels of abstraction. A shared property is that once started they must finish; strengthened a little we might suppose they have connected compact duration in time. The naive view is that an event is essentially an instantaneous action. More accurately, according to this view an event is atomic, that is has no internal structure (at that level of description), and an all-or-nothing character; at any time it either has or has not occurred. An event, still atomic, but with a duration in time can be reduced to this case by splitting it into ^{an} instantaneous beginning and a subsequent end event. We mention another possible view of events. Keep the view that an event once started must end but drop atomicity. Accordingly then an event might have connected compact duration in time and also internal structure, events inside so to speak; defined in this way events could

be called episodes. It would be possible for episodes to overlap and have subepisodes. Unfortunately here we do not follow up this line. For most of our work the naive view suffices. (In chapter 9 though, the orders on higher type events, associated with functions and functionals, express relations on the internal structure of events.)

We are concerned with how computations can be modelled by relations on events. The events with relations are called event structures. An event structure is an abstract description of a computation picking out certain events related to the computation and describing the possible courses the computation may follow. Event structures take several forms. Typical are (E, \leq, \times) and more generally (E, \vdash, \times) . The set E of events possesses a causality relation \leq , a partial order on E , or \vdash a subset of $P(E) \times E$. In the case of (E, \leq, \times) an event e cannot occur until the events in $\leq^{-1}\{e\}$ have all occurred whereupon it may occur. The causality relation \vdash is a little more general; it allows an event to occur in different ways. For (E, \vdash, \times) an event e can occur once all the events in any of $\vdash^{-1}\{e\}$ have occurred. The relation \times expresses an incompatibility between events; certain events occurring exclude certain others. Often \times will be a binary symmetric relation on E so events mutually exclude each other in a pairwise fashion.

This is really only half the picture. We must somehow express the dynamic behaviour of event structures. Alongside an event structure we should specify those states or configurations of events which can occur in the computation; this expresses formally what the two relations on events mean. For event structures of the form (E, \leq, \times) configurations, which are sets of events which have occurred, will at least be \leq -left-closed in accord with the intuition of \leq . Some consistency requirement will be imposed by \times too; for \times a binary and symmetric relation a configuration cannot include two events in that relation.

Scott domains of information can be represented by event structures with the construction above. Less information about the computation corresponds to less events having occurred, so configurations are naturally ordered by inclusion which, it turns out, gives a domain. In fact event structures represent suitable classes of

domains, generally specified by axioms; not only do event structures yield the class of domains but also from a domain D in the class an event structure can be recovered naturally so that its domain of configurations is isomorphic to D . This is the form of a representation theorem. It expresses that two classes of descriptions are equivalent and provides a means of translating back and forth between the two equivalent descriptions. Typical examples of representation theorems appear in group theory and lattice theory: for example rings of sets correspond to distributive lattices and fields of sets to boolean lattices etc. ([Bir], [Gra]). Event-structure representations of domains are generally far simpler and more intuitive than the represented domain.

In addition Petri nets represent event structures, with some qualification (see chapter 4). Thus representation results are a fundamental tool in relating theories with radically different vocabularies. Coupled to a theory of events they could sometimes justify or falsify an assumption of another theory perhaps through checking its physical feasibility or relating it to something more intuitive and acceptable. (This is just begun here, though see the appraisal of K-density - chapters 2 and 5 - and § 5.6 where Scott's thesis - "computable functions are continuous" - has implications for event structures.)

An important fact about event structures is that they model possible behaviour in time in an intuitive way. They have an operational yet simple nature. If an event is to occur it must occur at finite time. This will impose finiteness restrictions on the way in which an event is caused. In this thesis we use a variety of finiteness restrictions; the one natural to net theory where a finite set of events can occur concurrently to cause another is less restrictive than that appropriate to denotational semantics. Here is one we use a lot for event structures of the form (E, \leq, \ast) :

$$\leq^{-1}\{e\} \text{ is finite.}$$

An event need only wait for finitely many events in order to occur. For event structures of the form (E, \vdash, \ast) the corresponding restriction will be on the definition of configurations; in any configuration an event must have depended on only finitely many events to occur so every set of possible immediate causes of e in $\vdash^{-1}\{e\}$ can be assumed finite.

An event structure represents possible behaviour in time. Correspondingly, in the associated domain of information the partial order on configurations represents possible later behaviour in time. In this sense the domain has the same nature as those "real world" datatypes consisting of basic input or output values such as the domains of integers, booleans or infinite tapes. It is unlike domains of functions or functionals ordered pointwise; here the ordering no longer reflects later behaviour.

In chapter 9 we shall come to represent domains of functions of arbitrary type by event structures which will still stand for behaviour in time. This is only at the cost of imposing extra structure on event structures which among other things distinguishes higher-type event structures from those representing basic input or output values like the booleans. The extra structure is very closely related to Berry's stable ordering on functions which he has long recognised to be an ordering on behaviours of functions. The argument for adding extra structure to a theory runs like this: Whenever in a mathematical theory isomorphic objects model two situations which one wishes distinguished the theory must be extended to include some extra structure so the original objects are no longer isomorphic. So simple it seems silly! However we shall use the idea a lot. Of course it says very little, it does not say what extra structure, just "start looking". The search generally starts when a theory is formally applicable to a greater range of situations than originally envisaged even though the new situations are basically different; extending the range of the theory often involves very different phenomena being treated identically. For example one can produce two event structures with identical structure, one representing an input domain where the ordering reflects later behaviour and the other a function domain where the pointwise ordering does not. As a consequence the event structure representing the function domain does not capture behaviour in time; extra structure is called for to distinguish it from the other. In chapter 9 the extra structure is chosen to re-instate a behavioural nature to event structures representing functions; finiteness restrictions provided a useful guide.

We give an example where we have not yet found convincing extra

structure. As described an event structure stands for all possible courses a computation may follow. (It represents a datatype.) An event is under no obligation to occur even when it is given unbounded time to do so. For some computations naturally associated with such an event structure this may well not be the case for certain events (see 2.3), an example where the same event structure represents two situations we would like to distinguish formally. (An attempt is made using restless events in 6.4.)

Finally I should apologise for one big omission. There is no chapter dealing with morphisms on event structures, although morphism-like constructions are occasionally used. This is largely because of lack of time and partly because it is still unclear what extra structure to put on event structures to "force" event-occurrences. (The natural idea of contracting a convex set of events in (E, \leq, \otimes) to an event depends on this issue.)

1.2 Events in context

A major aim of theoretical computer science is the development of a mathematical theory in which to model reasonably completely the world of concepts and ideas in computer science. Such a theory must be both broad enough in scope and rich enough in its power of abstraction to handle the full range of phenomena at appropriate levels of detail. Two main theories of this nature are denotational semantics [Sco] initiated by Scott and Strachey and net theory [NProc] started by Petri. As indicated in section 1.1 we can relate the two theories using representation results and the intermediate concept of event structures.

Roughly Petri nets are a generalisation of flowdiagrams to allow concurrent activity and non-determinism. The emphasis is on modelling control through focussing on how actions (interpreted by events in the theory) and local states (interpreted by conditions) depend on previous occurrences of actions or states holding. Nets highlight the pattern of behaviour in time which in the case of transition nets is simulated by playing the "token game" on markings. Concurrency is represented more naturally than in alternative approaches where it is generally represented as non-deterministic interleaving. Net theory is a useful pragmatic tool in the understanding and design of distributed systems and hardware; it includes techniques to prove

properties of such systems. In addition the graphical representation of nets guides the mind's eye in design and makes them attractive to those involved in the pragmatic side of computing.

The mathematical approach in denotational semantics, originated by Scott, is more abstract. In denotational semantics a programming construct is attributed with a mathematical meaning; it is denoted by an element in a partially ordered domain of information. The denotations of compound constructs are built-up by operations on the denotations of the sub-constructs. Only for domains corresponding to basic datatypes such as the booleans does the information order directly reflect the idea of later behaviour in time. Nevertheless some idea of behaviour in time is captured by formalising the notion of those points of information which may be realised by a computing agent in finite time and by requiring that computable functions between domains be continuous - this expresses that eventual behaviour in time is exactly the "limit" of the finite behaviours. Denotational semantics has been very successful in giving a formal meaning to a wide class of programming languages thus enabling proofs of properties of programs. It has the advantage over more operational methods of giving semantics in that it cuts down on the arbitrary detail such semantics often possess.

We now discuss deficiencies in the two theories at their present stage of development. The general line is: denotational semantics is sometimes not operational enough while net theory is sometimes not abstract enough. Where possible we point out how a mathematical theory of events should help and how the issues raised in section 1.1 have a bearing.

For net theory I think it is fair to say that the mathematical foundations have not been worked out very thoroughly, and it is the more foundational aspects which concern us here. I believe there is a reason. Net theory, and the foundational work in particular, attempts to be very general. In practice when a net is used to model some situation it bears inscriptions as part of the modelling process. The inscriptions relate the net to the situation described, sometimes serving to interpret the conditions and events or detailing when events may or must occur. Such inscriptions play an essential role in the modelling.

However they appear to be ignored in the foundations (see the treatments of K-density and morphisms in [N.Proc] for example). There very little commitment is made to the range of interpretations in mind. Once the range of interpretation is unclear it becomes very hard to recognise when and what extra structure is required; it is difficult for the theory to recognise its limitations and grow. This may be one reason why the theory of net morphisms is so weak.

Unfortunately we say little on morphisms in this thesis. However we can be more constructive in our appraisal of causal nets and K-density where again I believe lack of commitment has misled. Causal nets were chosen by Petri to represent the net analogue of history or partial history; they are chosen to represent a course a computation may follow. As such their events and conditions are regarded as having occurred or as being inevitable. This is not true of events generally. This cries out for extra structure. Petri has insisted that causal nets be K-dense, imposed as a finiteness restriction. (It is intended to ban Zeno machines for instance.) Using a simple theory of states of event structures and representation results we shall give a critical appraisal of K-density, conclude that the present formulation is too restrictive, while proving a restricted form of K-density does hold. In other words, we agree with the spirit of K-density but not with its exact statement. This disagreement stems from Petri's formalisation of the idea of (global) state (taken to be a case - a maximal cut across the net) so it is quite fundamental.

We now present some limitations of denotational semantics which are fairly well-defined.

Denotational semantics does not, as yet, handle concurrent computations in a natural way. Successful treatments have depended on simulating concurrency by non-deterministic interleaving of uninterruptable actions often atomic events (see [Plo2] and [Mil]). We call attention to Milner's book [Mil] which sets a paradigm for future work on concurrent computation because of the ideas it introduces and the "scientific approach" it adopts. Algebraic laws on the communication of computing agents are justified by notions of observational equivalence; even interleaving is shown appropriate once

observations are restricted to being serial. It is hoped that by using event structures, the ideas there can be brought closer in spirit to net theory, and concurrency treated more naturally.

There is no uniform way in which to treat problems associated with "fairness". Particular fair implementations can generally be modelled; the problem is to find a denotation which is both an abstraction from all possible implementations and still expresses that certain events will occur eventually. Perhaps event structures are an appropriate framework in which to express ideas of abstraction and inevitability of occurrence (see the relations \sqsubseteq , \sqsubset of 5.3 and restless events in 6.4).

Related to the fairness issue are technical problems associated with infinite non-determinism when generalising Plotkin's power-domain construction based on finite non-determinism [Flo 2]. From the work of Park [Par] and Plotkin (unpublished) it appears that continuity should be generalised to model infinite non-determinism successfully. As continuous functions have been a basis for a successful theory, domains of information associated with infinite non-determinism should carry extra structure to distinguish them from those used formerly. Event-structure ideas may help here. Interestingly continuity can be rescued for infinite non-determinism by "padding-out" denotations with extra operational detail (e.g. in [Bac] taking denotations built from sets of histories does this). This can be seen as part of a general trend to add details of a more operational nature to denotations in order to model situations correctly.

The correctness of a denotational semantics with respect to operational ideas is determined by the criterion of full-abstractness; a semantics is fully-abstract if denotations are identified iff they are operationally equivalent. This notion enables one to home-in on inadequacies of denotational semantics, highlighting those operational features which it does not and should treat explicitly. For example the full-abstractness problem for PCF (see chapter 8) led Berry to an important new ordering on domains of functions, the stable ordering. It is an ordering on behaviours of functions and viewed in an event-structure setting

with functions regarded as configurations (chapter 9) it is associated with finiteness restrictions. This is new but back in '75 Kahn and Plotkin recognised the need for some kind of event-structure representation of basic input and output domains in order to define the notion of sequential function, involved in the PCF-problem. I was led to study event structures by the problem of injecting time into domains so that denotations also included the time complexity.

It is hoped that event structures associated with domains by representation results will prove fruitful in semantics by capturing operational ideas in a natural, intuitive way.

A word on work outside the two main streams of net theory and denotational semantics: Hewitt's actor model of distributed computing [Hew] uses the concept of an event - *for Hewitt an event is the receipt of a message by an actor*; he presents some finiteness restrictions on a form of event structure. Lamport's paper [Lam] constructs an event structure from deterministic processes communicating; his ideas on logical clocks and time-stamps implicitly impose finiteness restrictions *on event structures* (see 5.6).

1.3 Summary

We summarise the work in the thesis.

In chapter 2 we introduce net theory. The manner of introduction has been motivated by the future issues with which we shall be concerned; for this reason it is not unbiased or uncritical.

Initially we show how nets, structures built-up from events and conditions (2.1) may be given a dynamic behaviour (the "token game" on transition nets) in terms of markings (subsets of conditions) changing according to the firing rule which determines those concurrent occurrences of events which are possible (2.2). In particular we define and illustrate the notions of concession (that situation in which an event may occur), conflict (when event occurrences are mutually exclusive) and confusion a phenomenon due to conflict not being localised. Starting from an initial marking repeated application of the firing rule yields the forwards-

reachable markings. We then illustrate how transition nets with initial marking can be used to model computations such as those described by Milner nets, Kahn-MacQueen networks and datatypes like the integers or infinite tapes (2.3). These illustrate how events may be interpreted as atomic actions and conditions as local states. For Milner nets and Kahn-MacQueen networks there are inadequacies in the modelling by nets. This is traced to an ambiguity in the firing rule; occasionally one does not wish events to have concession forever - some events must occur or lose the ability to occur eventually (the idea of restless events).

Petri defined causal nets (see 2.4) in order to formalise the idea of a course that a computation may follow. Causal nets are the net-analogue of history or run and can be associated with particular plays of the token game. Petri has imposed a kind of finiteness restriction on them called K-density based on an idea of state for causal nets, formalised as a case. We present a precise though informal argument for K-density based on evidence in the literature ([Pet 1], [Bes]) and fair I hope, as using it we find a point to disagree; we take issue with Petri's formalisation of state as case. (Later, chiefly in chapter 5, we present more detailed evidence).

Finally we introduce and examine net morphisms a little (2.5), defining and illustrating concepts such as subnet and folding morphisms.

Chapter 3 deals chiefly with the concrete domains [Kah and Plot]. They are, I believe, the first example where events came to be treated explicitly in denotational semantics.

We start with a racy summary of the main definitions and ideas in denotational semantics, presenting such concepts as complete partial order, isolated element ω -algebraic domain and continuous function informally relating them to computations.

Concrete domains are domains of basic input or output information which support a definition of sequential function. As part of the process of axiomatising the domains Kahn and Plotkin required a representation theorem for them. A concrete domain is

represented by an event structure in the form of a matrix (3.2.3) rather like a Petri net. The domains consist of information about what events have occurred. The events are localised to occur at places. When an appropriate set of events (not necessarily unique) has occurred a place is allowed to be occupied by one of a set of mutually exclusive events. The representation theorem recovers events and places "hidden" in a concrete domain- they are recovered as equivalence classes of prime intervals (3.2.17) based on the covering relation (3.2.12). As a sort of appendix to chapter 3 we present in section 3.3 an improvement of the proof of the representation theorem in [Kah and Plo]; the proof is also a little more general - it works for a broader class of event structures than matrices.

One notable axiom of concrete domains is axiom F (3.2.11) saying that an isolated element only dominates a finite number of elements. In terms of the representation this means an occurrence of an event is only dependent on a finite number of events having occurred. Axiom F is a form of finiteness restriction. (In section 5.6 we present an argument for it based on Scott's thesis that computable functions are continuous).

In chapter 4 we give the basic machinery for translating back-and-forth between nets, event structures and domains. We generalise Petri's causal nets to yield the class of occurrence nets, so-called because in an occurrence net events and conditions stand for unique occurrences - not so for nets in general. The definition of case generalises easily too. However, surprisingly perhaps Petri's definition of sequential process does not. We then define the unfolding of a transition net to be that occurrence net which describes all possible courses the token game may follow. We associate an event structure with an occurrence net essentially by forgetting the conditions but remembering the conflict they incur. Such event structures have the simple form (E, \leq, \otimes) where \otimes is the conflict relation and \leq , the causality relation, is a partial order, corresponding to the fact that an event can occur in a unique way. Consequently when we pass over to domains events manifest themselves in a particularly simple way, in fact as complete primes. Accordingly there is a very simple representation

theorem in terms of complete primes rather than equivalence classes of prime intervals.

Chapter 5 provides event structures with a theory of states. We work chiefly with fairly general definitions chosen to reflect net-theoretic intuitions in order to extend the translation begun in chapter 4.

Our definitions of state are based on the concept of an observer for an event structure; intuitively an observer stands for a run or history of a computation. The definition of observer (5.1.4) depends on two assumptions about the nature of the computations described which are called the initiality and discreteness restrictions. The definition allows an infinite set of causally unrelated events to occur within finite time. An observable state is defined to be the set of events some observer records in finite time while for a state time may be unbounded. It is observable states which capture those intuitions motivating Petri's definition of a case. We characterise both forms of state using a metric (5.2) closely allied to the idea of reachable markings of a net. The finiteness restriction of finite depth (5.2.11) on event structures follows from the definition of observer.

Using the techniques of chapter 4 the notions of state are transferred to nets (5.4). Observable states transfer to a subset, generally proper, of cases of an occurrence net. We call them observable cases. In the situation where the occurrence net is an unfolding of a transition net, reachable markings are precisely the images of the observable cases under the folding map (5.4.4).

Only in the situation where cases are observable would one expect K-density to apply and in fact restricting cases to being observable we prove a restricted form of K-density (5.4.7). Under certain conditions we prove a neat equivalent of K-density (5.4.8).

The translation of the concept of confusion in nets is far more direct and less qualified. We show in section 5.5 how it connects with concrete domains. Confusion turns out to be a property of event structures; conditions play no role other than to express conflict. A major result of this chapter is that the domain associated with a net is concrete iff the net is confusion-

free. (5.5.9).

We examine an idea of computationally feasible which induces a further finiteness restriction, that of finite width (5.3.2). This is intended to capture the idea that only a finite number of computing agents can be in operation at a finite time. It is based on the definition of observer which is determined solely by the causality and conflict relations \leq and \bowtie . We introduce relations \leq and \leq_{\circ} between event structures to express ideas of implementation (5.3.12) particularly by finite-width event structures. Following how states go through the implementation relations suggests a more abstract definition of observer closer in spirit to denotational semantics (5.3.18). In short, section 5.3 shows how constructions based on ideas of abstraction, natural for net theory, yield a more abstract notion of state like that in semantics.

The final section of chapter 5 deals with alternative finiteness restrictions and definitions of states as expressed by other authors. We briefly look at restrictions imposed by Hewitt's [Hew] and Lamport's [Lam] approaches and in a little more detail how the ideas of denotational semantics relate. We translate Scott's thesis ("computable functions are continuous", [Sco]) to a finiteness restriction on event structures (5.6.5).

Chapters 6 and 7 are concerned with following-up our ideas in net theory.

In chapter 6 we are concerned with conditions. When we pass from nets to event structures they are ignored; many different occurrence nets may induce the same event structure. Here we are concerned with what, if anything, is lost in this process. This involves considering how conditions are to be interpreted; we regard them as local assertions having extents in time.

The work begins by noting that with an extensionality principle on conditions one may recover the conditions of occurrence nets inducing an event structure from the event structure alone. Then using the simple machinery on states we have developed it is possible to define natural relations on the conditions of an event structure. One particularly useful relation formalises the situation where one condition holding implies that another holds (6.1.6).

Using such relations given an event structure we may define a net which is in some sense the minimum net inducing that event structure. We can also define the maximum net associated with an event structure (rather trivially this time). For such occurrence nets we show K-density results (6.1.32) which are close to Petri's original ideas.

In section 6.2, regarding conditions as sets of assertions, we introduce a relation between nets which compares their degree of expressiveness. This relation enables us to characterise (6.3) the two constructions of nets from an event structure. They will both be in the class of nets of maximum expressive power, one being included in and the other including all nets of this class.

Finally in section 6.4 we look briefly at restless events of an event structure. They express an idea of inevitability. The topic appears to involve generalising Petri's conditions.

In chapter 7 we take another look at observers for an event structure. This time we do not insist on the initiality restriction - generally net theory does not. The results translate to causal nets. We determine when (countable) event structures have a total observer (7.1.7) - so all events are recorded at some time. Observers determine a reachability relation on observable states as in chapter 5. However now there may be more than one equivalence class of reachable states. We characterise those (countable) event structures with one and only one (7.2.7). Then the event structure (or causal net inducing it) can alone be regarded as describing a course of computation (this is close to a remark by Petri motivating K-density in [Pet 2]). The mathematics involves such ideas as collapsing a convex subset of events to an "event", a kind of quotienting operation (7.1.10). As usual a restricted-K-density result applies (7.4.3).

In chapter 8 we introduce an as yet open problem in denotational semantics, the full-abstractness problem for PCF.

In chapter 9 we define higher-type event structures in which configurations represent functions. We produce a cartesian-closed category of event structures which is naturally equivalent to a full

subcategory of Berry's bidomains, ~~his~~ major step on the way to a solution of the PCF problem. Finally we indicate how by strengthening the axioms and restricting configurations a fully-abstract model might be produced.

Chapter 2. Introduction to Petri nets

In this chapter we introduce Petri nets and outline net theory in so far as it connects with our later work. A Petri net models a computation. Thus we shall be concerned with two aspects, the formal definitions and properties of the nets themselves, and, how they model computations. We use the word "computation" in a slightly vague way. We shall say more on this later. For the time being we note that what one thinks of as being a computation depends on what theory one has in mind. For instance one might sometimes think of a computation as merely a partial function from input to output. In net theory one is concerned with how computations proceed focussing on such properties as concurrency and conflict. Of course every theory automatically stakes out its own territory by virtue of what primitives it takes and what basic assumptions it makes thus determining what it can and cannot describe. Net theory takes events, conditions and causal dependency as its primitives and views the world accordingly. Nets have proved very useful as models of control.

2.1 Basic definitions

We shall take a slightly more general definition of a Petri net than is customary.

Definition 2.1.1

A Petri net N is a tuple (B, E, F) where:

B is a set of conditions

E is a set of events

$F \subseteq (B \times E) \cup (E \times B)$ is the causal dependency relation satisfying: $N1. B \cap E = \emptyset$.

Notation 2.1.2

Let N be a Petri net. If $x \in B \cup E$ we write $\bullet x$ (respectively x^\bullet) for $\{y \mid yFx\}$ (respectively $\{y \mid xFy\}$). If $x \in E$ we call $\bullet x$ the preconditions and x^\bullet the postconditions of x . If $x \in B$ we call $\bullet x$ the preevents and x^\bullet the postevents of x .

The definition of a Petri net is more general than usual because we allow F to be null and do not insist that the field of F , $\{x \in B \cup E \mid \exists y \in B \cup E xFy \text{ or } yFx\}$, is $B \cup E$. Thus we allow a net to consist of a single condition or event. We recall the standard

graphical representation of Petri nets in which events are represented by squares " \square " and conditions by circles " \circ " and the relation F by oriented arcs " \rightarrow ". Note that with this representation we allow .

Later we shall sometimes impose a further axiom on nets which ensures conditions are extensional in the sense that two conditions with the same pre and post events are identical (N2 below). It is convenient to define another axiom (N3) too. We shall not use either till chapter 4.

Definition 2.1.3

Let $N = (B, E, F)$ be a Petri net. N satisfies N2 iff

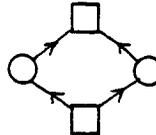
N2.: $\forall b_1, b_2 \in B \cdot b_1 = \bullet b_2 \ \& \ b_1^\circ = b_2^\circ \Rightarrow b_1 = b_2$.

If N satisfies N2 it is condition-extensional.

N satisfies N3 iff

N3: $\forall e \in E \cdot e \neq \emptyset \ \& \ e^\circ \neq \emptyset$.

This net satisfies neither N2 nor N3:



2.2 Transition nets

Perhaps the most familiar part of Net theory is the "token game" in which markings of conditions in the net change as events fire. We deal with this now. We should remark that within net theory there is a semiformal idea of level of net description, the higher the level of the net the more abstract is the net description. The token game occurs at the level of transition nets. Here the events are usually called transitions and the conditions places. At this level nets are endowed with a dynamic behaviour in which markings change according to the firing rule. A marking is a subset of conditions usually represented by a distribution of tokens on a graphical representation of the net. (Only a single token is allowed on each condition of the marking.)

Definition 2.2.1

Let $N = (B, E, F)$ be a net. A marking of N is a subset of B .

The firing rule depends on two notions, concession and conflict. An event may fire only when it has concession.

Definition 2.2.2 (concession)

Let $N = (B, E, F)$ be a net. Suppose M is a marking of N and $e \in E$.

Then e has concession at M iff $\bullet e \subseteq M$ & $e^\circ \cap M = \emptyset$.

Definition 2.2.3 (conflict)

Let $N = (B, E, F)$ be a net. Suppose M is a marking of N and e_0, e_1 are in E . Then e_0 and e_1 are in forwards conflict at M iff they both have concession and $\bullet e_0 \cap \bullet e_1 \neq \emptyset$. They are in backwards conflict at M iff they both have concession and $e_0^\circ \cap e_1^\circ \neq \emptyset$. They are in conflict at M iff they are in forwards or backwards conflict at M .

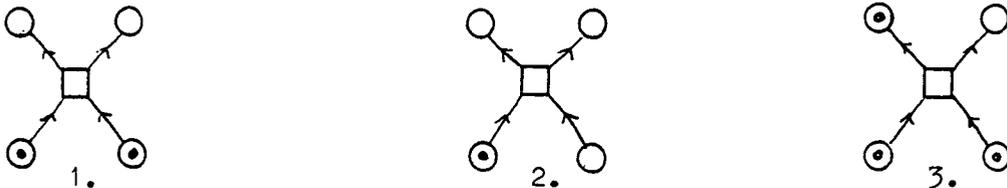
Now we can give the firing rule which specifies when a subset of events may fire concurrently.

Definition 2.2.4 (The firing rule)

Let $N = (B, E, F)$ be a net. Suppose M and M' are markings of N and that $X \subseteq E$. Define $M[X]M'$ iff (i) each member of X has concession at M , (ii) no two members of X are in conflict at M , (iii) $M' = (M \setminus \bigcup \{\bullet e \mid e \in X\}) \cup \bigcup \{e^\circ \mid e \in X\}$. (Then events in X are said to fire concurrently.)

Thus the firing rule gives a "one-step forwards" reachability relation between markings. Note if two events are in conflict one excludes the other from firing.

Example 2.2.5 (illustrating concession)



Here e has concession in 1 but not in 2 and 3.

Example 2.2.6 (illustrating conflict)



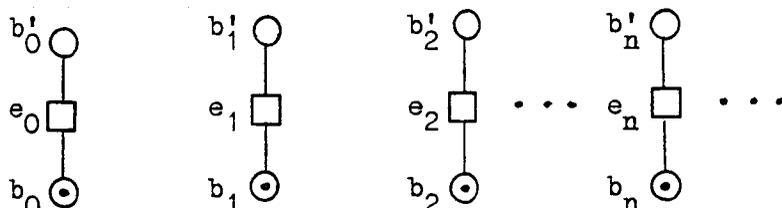
N_1 , Forwards conflict

N_2 , Backwards conflict

In the above net N_1 , e_0, e_1 are in forwards conflict for the marking shown as they both have concession and share a common precondition. In N_2 , e_0 and e_1 are in backwards conflict for the

marking as they both have concession and share a common post-condition. Referring to the firing rule note that in either case only one of the events e_0, e_1 can fire. Thus implicit in the firing rule is: The change in a condition-holding that takes place as a result of an event occurrence is associated uniquely with that occurrence.

Example 2.2.7



In this example the net is infinite. As the firing rule does not require that only one event fires at a time the marking $\{b'_n \mid n \in \omega\}$ is reachable from the marking shown through the concurrent firing of $\{e_n \mid n \in \omega\}$.

So far we have only dealt with one application of the firing rule. Repeated applications of it give a forwards reachability relation between markings. The precise nature of this reachability relation depends on how fast one is allowed to play the token game (see §4.2). However the following definition seems to be accepted.

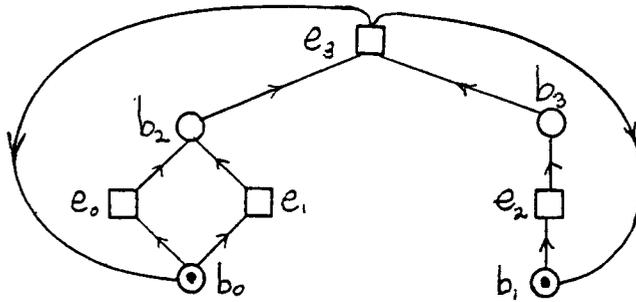
Definition 2.2.8

Let $N = (B, E, F)$ be a net. Suppose M and M' are markings. Write $M \rightarrow' M'$ iff $\exists X \subseteq E \ M \{X\} M'$. Define \rightarrow to be the transitive closure of \rightarrow' . If $M \rightarrow M'$ say M' is forwards reachable from M .

Net theory generally deals with a symmetric reachability relation (the symmetric closure of \rightarrow) so it is also concerned with backwards reachability. However in our work we shall generally assume transition nets have an initial marking from which the forwards reachable markings are obtained by the firing rule.

Definition 2.2.9

Define a transition net with initial marking to be a pair (N, M) consisting of a Petri net N together with a marking M . The (forwards) reachable markings of (N, M) are all markings M' such that $M \rightarrow M'$.

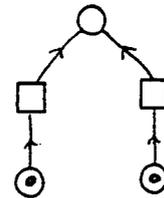
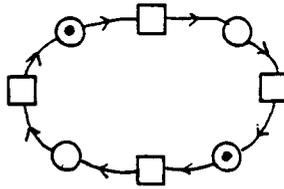
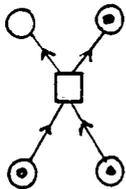
Example 2.2.10

Here the initial marking $\{b_0, b_1\}$ is marked. The events e_0, e_1 are in conflict. Either e_0 or e_1 can fire to yield the marking $\{b_1, b_2\}$. One of them may fire concurrently with e_2 to yield the marking $\{b_2, b_3\}$. The further firing of e_3 would then return us to the initial marking and the cycle could be repeated.

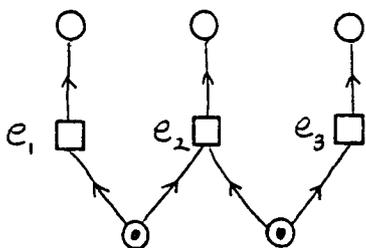
Later we shall be concerned with contact-free transition nets with initial marking.

Definition 2.2.11

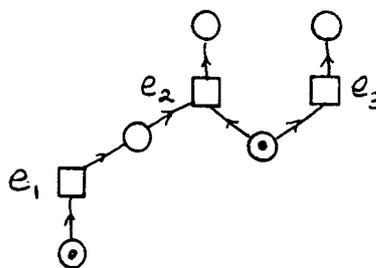
Let (N, M) be a transition net with initial marking. The (N, M) is contact-free iff for any reachable marking M and event e we have $\bullet e \subseteq M \Rightarrow e^\bullet \cap M = \emptyset$.

Example 2.2.12 (nets which are not contact-free)

We shall also be concerned with the concept of confusion in transition nets. Confusion can occur in two forms, symmetric and asymmetric. We illustrate these below deferring the formal definition until after.

Example 2.2.13 (confusion)Symmetric confusion

In the case of symmetric confusion at a marking two events e_1 and e_3 can occur concurrently. Through the occurrence of e_1 , e_3 is brought out of conflict with e_2 ; through the occurrence of e_3 , e_1 is brought out of conflict with e_2 .

Asymmetric confusion

In the case of asymmetric confusion at a marking e_1 and e_3 can occur concurrently. Through the occurrence of e_1 , e_3 is brought into conflict with e_2 .

For simplicity we define confusion for a contact-free transition net with initial marking.

Definition 2.2.14 (confusion)

Let (N, M_0) be a contact-free transition net with initial marking. Let M be a reachable marking.

Say N is symmetrically confused at M iff there are events e_1, e_2, e_3 such that e_1 and e_2 are in conflict and e_2 and e_3 are in conflict at M but e_1 and e_3 are not in conflict at M .

Say N is asymmetrically confused at M iff there are events e_1, e_2, e_3 such that e_1, e_3 but not e_2 have concession at M and $M[e_1 \rangle M'$ so that e_2 and e_3 are in conflict at M .

Say (N, M_0) is symmetrically (asymmetrically) confused iff for some reachable marking M we have N is symmetrically (asymmetrically) confused at M .

Say (N, M_0) is confused iff it is symmetrically or asymmetrically confused.

In net theory it is said that "resolution of conflict is not objective" when confusion occurs. The following informal argument is used. It uses the idea of an observer - we shall make the

explanation more solid in the next section where we discuss one possible notion of observer. We sketch the argument: In the case of symmetric confusion in example 2.2.13 if e_1 and e_3 occur concurrently one regards this as meaning they can occur at any time relative to each other according to an observer. Thus it depends on the observer whether conflict has been resolved between e_2 and e_3 . Similarly for asymmetric confusion it will depend on the observer whether or not conflict is resolved between e_2 and e_3 [N.proc p.34].

2.3 Examples of modelling computations by transition nets

In the previous section we have outlined the dynamic behaviour of transition nets (the token game) and illustrated some of the basic concepts such as concession, conflict and the more obscure notion of confusion. This was discussed purely within the theory of transition nets. In this section we illustrate how transition nets may be used to model situations in computer science. The examples will necessarily be limited; we refer the interested reader to the literature (in particular see [N.Proc]), pointing out that net theory is a growing subject consisting of far more than will be mentioned in this thesis. Nevertheless we see the theory of transition nets as a keystone of net theory, from which more recent work has been done in securing it by examining assumptions to be made on lower level nets [Pet 1] and also extending it to higher levels as in the work of Genrich and Lautenbach, and Jensen ([Jen], [Gen]). Thus the examples will illustrate some basic issues.

A. Modelling Milner nets by transition nets

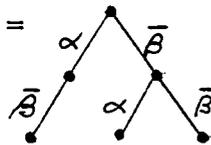
We first dwell a little on Milner nets. These are fairly easy to understand intuitively as computations although there are many subtleties which we shall gloss over. Our use of them here is the modest one of providing a (for us) semiformal description of some computations which we can model by transition nets. The interested reader is referred to the fast-growing literature on Milner nets (e.g. [Mil]). Milner nets are constructed by "wiring together" a collection of computing agents each with its own internal program determining its behaviour following the communications it makes with its fellow agents. An agent has ports at which it may communicate. These are labelled. From the outside, an agent A may look like this:



The label α indicates that A *may* make an α -communication with another agent with port labelled by $\bar{\alpha}$. (called the co-label of α). (Thus A above could make a β -communication with another agent labelled with a β .) Here we shall assume that the communication is purely one of synchronisation (a "handshake" between agents). After making a communication an agent will move into a new state determining whether and how it is prepared to communicate. At any stage an agent may be prepared to make several communications. However, significantly, it is only allowed to make at most one; thus an agent is not allowed to make two communications concurrently. Given these constraints the internal program of an agent may be cast in algebraic form as a synchronisation tree or its equivalent algebraic expression. For the agent A above an example program p would be:

$$p = \alpha:\bar{\beta}:NIL + \bar{\beta}:(\alpha:NIL + \bar{\beta}:NIL)$$

or drawn as a synchronisation tree, p =

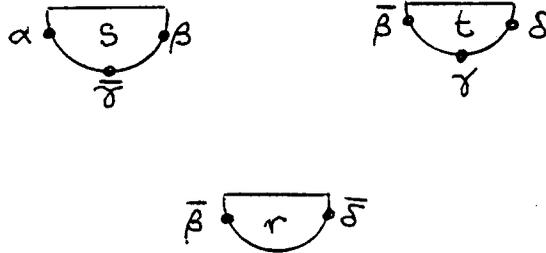


Thinking of a program as a tree the nodes of the tree determine states, the future behaviour from a node being given by the subtree with it as root. The program NIL, represented as a tree with one node "•", says no future communication will occur. The program p above means that the agent is prepared to make either an α or a β communication. If the external world of other agents is such that it performs an α -communication then it may do a β -communication whereupon it loses interest in future interaction with any other agents there may be. On the other hand the external world may provide a β -communication. Then it is prepared to do an α or a β communication, not both, before losing interest.

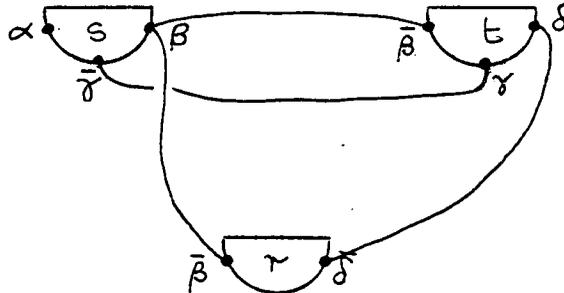
It remains to describe the operations on agents. For Milner et al these operations yield agents - remember an agent has a particularly simple internal program. This is achieved by simulating parallelism by interleaving so a compound agent formed by setting two agents in parallel still possesses an internal program of this simple form. In fact congruence classes of programs then

form a natural domain of denotations once one has settled on a suitable notion of equivalence of behaviours. However our concern is different; we wish to associate a transition net with the compound agent to exhibit any concurrency it may possess. We will have two operations derived from Milner's: one will take a set of agents and link them together in parallel; the other will screen-off certain labelled ports. Both these operations use the labelling on ports.

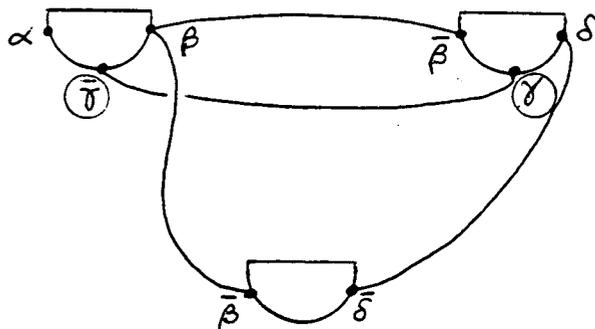
Think of the operations as being done physically on the agents. Picture three agents:



Combining them in parallel yields the following picture of a compound agent; call it $\text{par}\{s,t,r\}$.



The link between $\bar{\gamma}$ and γ for instance shows that s and t may communicate via their respective $\bar{\gamma}$ and γ ports. Of course, how the compound agent behaves depends on the internal programs of s, t and r . Having set up such an agent one may wish to screen-off certain ports. For example at present s can still make a communication with the external world via its $\bar{\gamma}$ port. If we wish to prevent this we can remove the labels γ and $\bar{\gamma}$ to form the new compound agent $\text{par}\{s,t,r\} \setminus \{\gamma\}$, which has $\bar{\gamma}, \gamma$ ports hidden from view. We can picture this as



Similarly we can screen-off any set of labels.

Well, how do we associate a transition net with such compound agents? It is natural to take the communications as events. For the conditions we take states of the agents; thus we interpret conditions as local elements of a global state. The state of an agent is altered by the occurrence of a communication; this induces the causal dependency relation. A little care is needed to ensure that the token game is correct. For example suppose we have an agent which starts in some state from which it may communicate to return immediately to the same state. In some appropriate compound agent this will yield an event with a precondition and postcondition in common which will be marked initially. According to the token game the event will not have concession whereas from the Milner net point of view we would like it to be able to fire. I see three ways out. One is to change the definition of concession so that it differs from the usual one (say an event e has concession for a marking M iff $\cdot e \subseteq M$ and $(e^\bullet \setminus \cdot e) \cap M = \emptyset$). Another is to distinguish different occurrences of holdings of the same place. Finally (a sly trick!) we could choose our agents so this can never occur. We pick the latter by assuming in examples that our agents have finite internal programs.

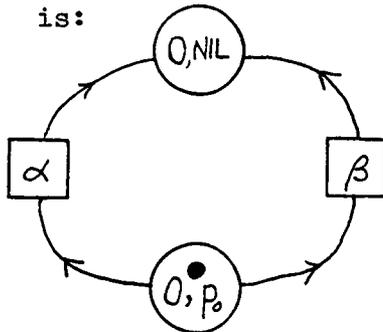
We give some examples showing how a transition net with initial marking is associated with a Milner net. In fact the transition nets have a bit of extra structure due to labelling the events. This is because there are essentially two different kinds of event. There are "external events" (which we label by α or $\bar{\alpha}$ for example) corresponding to possible communication with an external agent (ports labelled $\bar{\alpha}$ or α) not in the Milner net. There are "internal events" which we label by τ (as in [Mil 1]) corresponding to internal

communications between agents in the Milner net.

Example 2.3.1



For the single agent 0 with internal program $p_0 = \alpha:Nil + \beta:NIL$ the corresponding transition net is:

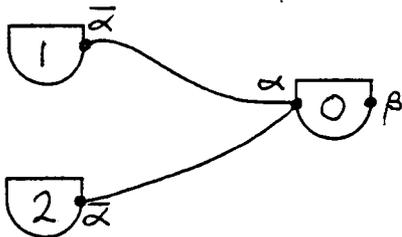


Note the conditions are associated with the states of the agent 0 - they are pairs consisting of the agent and one of its possible states. The initial state of 0 is marked. The agent is initially prepared to make an α or a β communication.

When the agent 0 above is set-up in parallel with other agents we may get internal communications as the next example illustrates.

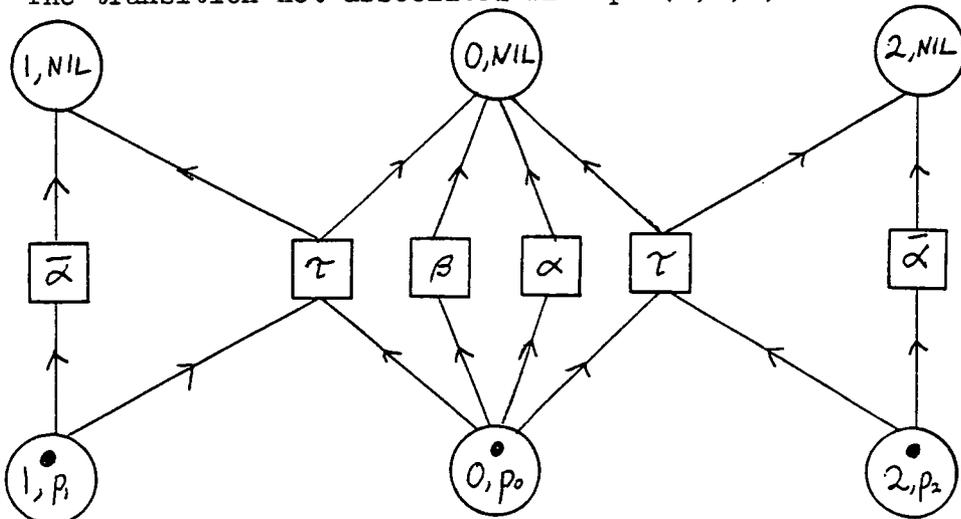
Example 2.3.2

Suppose the agent of 2.3.1 is set in parallel with two other agents, 1 and 2 with programs p_1 and p_2 as shown:



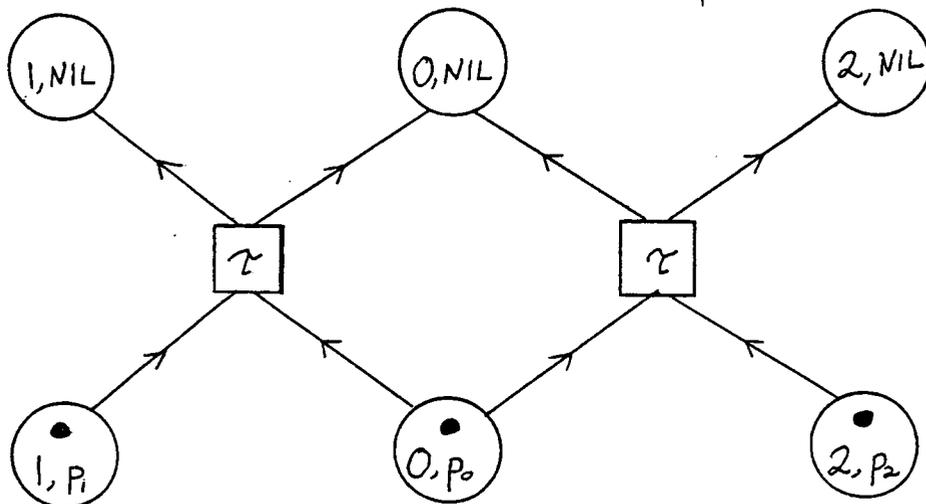
$p_0 = \alpha:NIL + \beta:NIL$
 $p_1 = \bar{\alpha}:NIL$
 $p_2 = \bar{\alpha}:NIL.$

The transition net associated with $par\{0,1,2\}$ is:



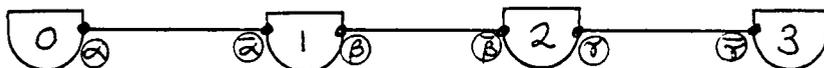
This time 0 may make a communication with 1 or 2. The corresponding events are labelled τ - they are internal to the Milner net above.

If α and β ports were screened-off from external communication those events labelled by $\alpha, \bar{\alpha}, \beta$ could never occur. This is reflected by omitting these events from the net. Thus the transition net associated with $\text{par}\{0,1,2\} \setminus \{\alpha,\beta\}$ is:



In the next example we show how confusion can arise from Milner nets. To make the drawings simple we only consider internal communications.

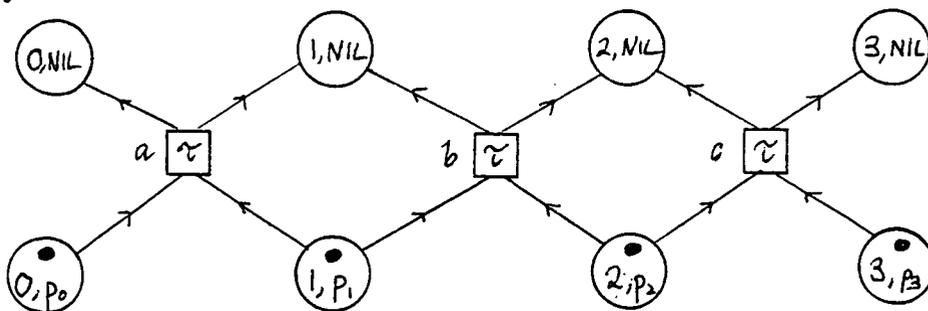
Example 2.3.3 (How symmetric confusion can arise from Milner nets)



Consider the above compound agent consisting of four agents 0,1,2,3 linked in parallel. We can write it as $\text{par}\{0,1,2,3\} \setminus \{\alpha,\beta,\gamma\}$. The respective programs are:

$$\begin{aligned} p_0 &= \alpha:\text{NIL} \\ p_1 &= \bar{\alpha}:\text{NIL} + \beta:\text{NIL} \\ p_2 &= \bar{\beta}:\text{NIL} + \gamma:\text{NIL} \\ p_3 &= \bar{\gamma}:\text{NIL} \end{aligned}$$

The corresponding transition net below is an example of symmetric confusion



From left to right the three events a,b,c labelled τ , correspond to 0 and 1, 1 and 2, 2 and 3 communicating.

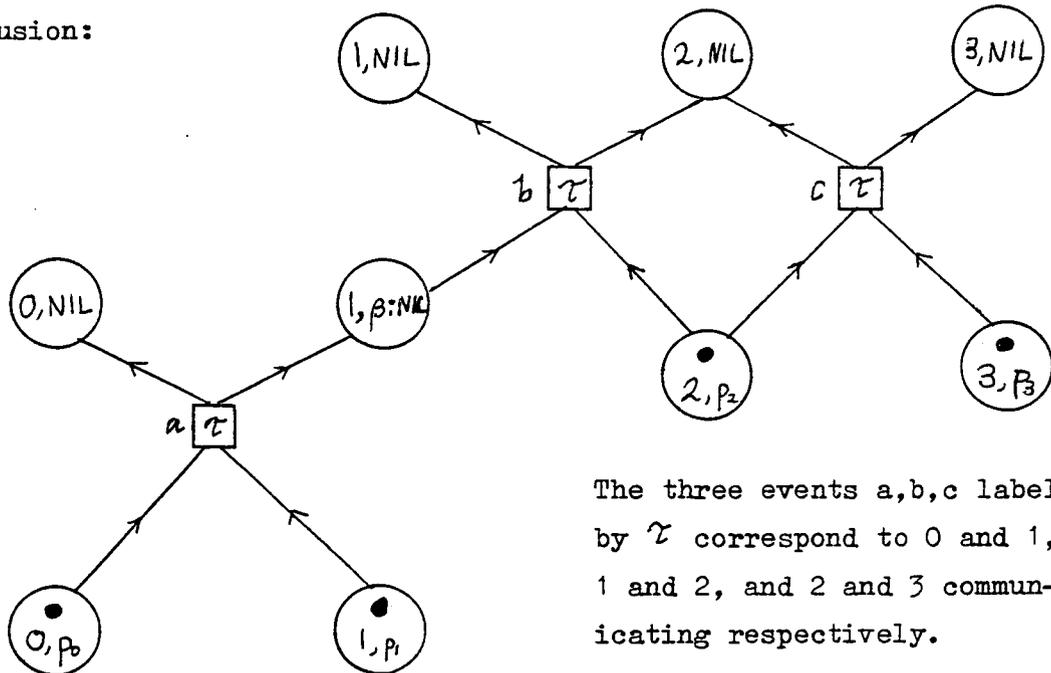
Example 2.3.4 (How asymmetric confusion can arise from Milner nets)



This time the compound agent $\text{par}\{0,1,2,3\} \setminus \{\alpha,\beta,\gamma\}$ is formed from four agents 0,1,2,3 with respective internal programs:

- $p_0 = \alpha:\text{NIL}$
- $p_1 = \bar{\alpha}:\beta:\text{NIL}$
- $p_2 = \bar{\beta}:\text{NIL} + \gamma:\text{NIL}$
- $p_3 = \bar{\gamma}:\text{NIL}$

Our associated transition net is now an example of asymmetric confusion:



The three events a,b,c labelled by τ correspond to 0 and 1, 1 and 2, and 2 and 3 communicating respectively.

Recall that in the previous section we gave the traditional net theoretic analysis of confusion in which it is said that confusion occurs when conflict resolution is not objective i.e. it depends on the observer if and between what events conflict is resolved. We left, somewhat up in the air, the idea of what an observer is. One possible idea is that of a run or history of the computation by which is meant a record of what events happened and when they happened. In a particular run of the Milner nets in examples 2.3.3 and 2.3.4, because we know nothing of the relative speeds, conflict between b and c may or may not occur even when c certainly occurs sometimes during

the run.

A Petri net can be regarded as determining a set of possible runs or histories, as above. However this intrudes on another issue, one which we have deliberately left ambiguous till now and which we shall only mention here. In the Milner nets of examples 2.3.3 and 2.3.4 a,b,c the events labelled τ have been screened-off from interruption by the outside world. For this reason (see [Mil 1]) in the Milner net of 2.3.3 either b or a and c communications will eventually occur and in the net of 2.3.4 either a and c or a and b will eventually occur. The Petri nets modelling Milner nets do not express this. In examples 2.3.3 and 2.3.4 all the events are internalised so one could make the token game behave correctly for these examples by appending another rule which ensures a kind of fairness:

No event can have concession forever; it must either eventually fire or lose its concession through a conflicting event firing.

Of course in general a Milner net will include a mixture of internal and external communications. To reflect this the associated transition net must bear extra structure. One idea is to distinguish a subset of events, perhaps called restless events, such that no event in the subset can have concession forever; it must either eventually fire or lose its concession through a conflicting event firing.

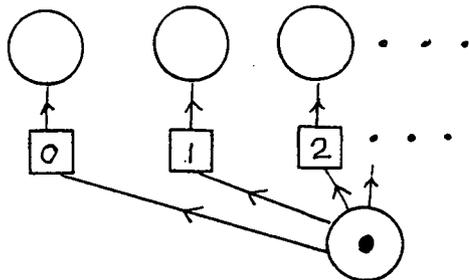
Our chief aim was to illustrate how transition nets can model the computations associated with Milner nets. For this reason our approach was very informal. Undoubtedly it could be made more systematic and general. For example Mogens Nielsen has given a formal semantics for Milner nets (like the ones we have used) in terms of labelled event structures. Importantly then an agent can communicate concurrently.

B. Transition nets as datatypes

The issue of restless events above suggest another class of computations described by transition nets, namely those in which no events are restless. Such computations correspond naturally to datatypes. A datatype is a possible set of values associated with a computation (the set may have a lot of structure of course). Typical datatypes are the Booleans, the integers, finite and infinite strings or tapes and, if we are prepared to go to higher types, partial

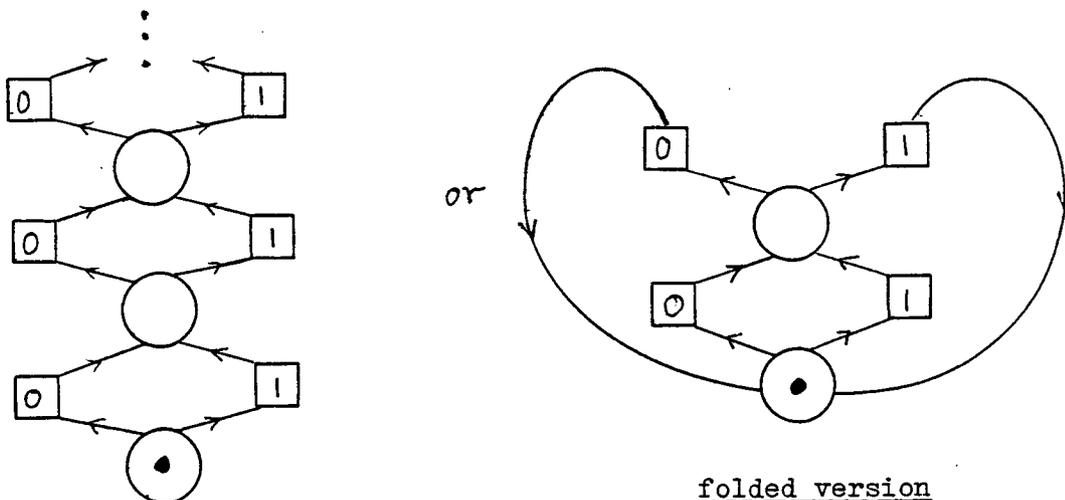
functions and functionals. (It might be thought that causality structures such as transition nets are so inherently "low-type" that the latter are beyond their range; however see chapter 9 on event-structures of higher type.)

Example 2.3.5 (The integers)



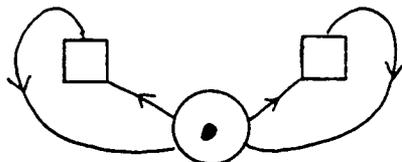
Here at most one value, an integer, can appear. Thinking of this as occurring at some place, such as a square on a tape, one can give a physical interpretation of the conditions. The bottom condition corresponds to no value having occurred there and the upper conditions to particular values having occurred. Imagining this net to occur as part of a computation which may yield an integer value, it is possible that no integer is ever produced through the computation diverging; then the bottom condition would hold forever.

Example 2.3.6 (Possibly-infinite tapes or strings over $\{0,1\}$)



Looking at the figure on the left it is easy to see how arbitrary tapes over $\{0,1\}$ including the null tape can be generated by playing the token game; the null tape corresponds to the token getting stuck forever in the initial place and infinite tapes to infinite games. Regarded as part of a computation yielding tapes as output the token getting stuck forever at some place corresponds to

the computation diverging at this stage. To the right we have drawn a folded version of this net in which even occurrences and odd occurrences have been collapsed together.



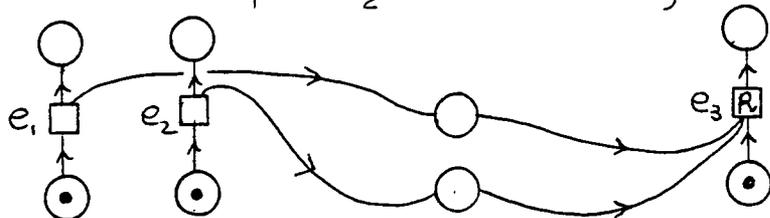
as a folded version and keep the standard notion of concession (another reason for changing the definition of concession?)

Frequently datatypes will be associated with possible input or output values for a computation. As such they may be represented by "subnets" (we give a precise definition in 2.5) of the net associated with the entire computation. Again in general this will give rise to a transition net where some events will be restless and some not. The events associated with input will not be restless; the choice of input and whether or not there is to be any is decided by the outside environment. The remaining events may well be restless in the net corresponding to the entire computation. We give a simple example.

Example 2.3.7



Regard N_1 as the input datatype and N_2 as the output datatype in the following computation in which one event e_3 is restless so marked by an "R". When e_1 and e_2 occur as input e_3 eventually occurs as output.

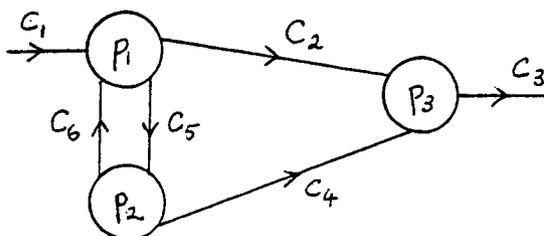


C. Modelling Kahn-MacQueen networks by transition nets

We now sketch how to model Kahn-MacQueen networks [Kah and Mac] by Petri nets. They provide examples of a process interacting with datatypes. Kahn-MacQueen networks consist of processes which may communicate through channels able to queue arbitrarily long sequences of values. The processes are deterministic and the states of the channels can be regarded as forming a datatype. For simplicity we

assume that in a network distinct processes cannot share a common channel to output or input to, and that the values exchanged are always from a set V . The act of outputting a value to a channel we call writing, the act of inputting from a channel reading. Then our assumption implies each channel c has at most one process writing to it; call it $w(c)$ if it exists in the network. Similarly each channel c has at most one process reaching from it; call it $r(c)$ if it exists in the network. It is customary to draw diagrams like the following to represent Kahn-MacQueen networks.

Example 2.3.8



This diagram represents a network consisting of three processes p_1 , p_2, p_3 connected to six channels marked as arcs directed to show how information flows. We have $w(c_4) = p_2$ and $r(c_4) = p_3$. Note we do not insist on each channel having both a writer and a reader - the "processes" $w(c_1)$ and $r(c_3)$ are in the external environment.

Rather than describing a programming language to determine the internal programs of the processes we give them an informal semantics. Call the semantic denotation of a process a behaviour. As with Milner nets we have the behaviour of doing nothing evermore which we call "NIL". Otherwise a process may be in a reading state, when it is about to read from a definite channel if it can, or in a writing state, when it is about to output to a definite channel. After accomplishing these actions it will follow some subsequent behaviour. Of course, if the action is that of reading a value its subsequent behaviour will depend on the value in general. Thus a behaviour b of a process p has three forms according to p 's state:

(reading state) $b = (c, f)$ where c is a channel s.t. $r(c) = p$
and f is a function from V to behaviours.

(writing state) $b = (c, (v, b'))$ where c is a channel s.t. $w(c) = p$, $v \in V$
and b' is a behaviour.

(null state) $b = \text{NIL}$

(This can be regarded as an inductive definition of a set of finite behaviours or alternatively behaviours may be thought of as elements

of a recursively defined domain. Here we do not care, though the latter would be necessary for infinite or non-terminating behaviours.)

Now we show how to construct a transition net with initial marking modelling a network satisfying our assumptions. The events will be actions of reading or writing. Conditions will correspond to states of processes and local states of the channels.

Process-conditions will be of the form:

(p, b)

where p is a process and b is a behaviour.

Of these conditions those in which b is the initial behaviour of p will be marked initially.

Essentially a channel is a queue of values. A process writes the latest value onto the queue and reads (and removes) the earliest. Roughly we shall represent the queue as the (temporal) sequence of values written to the channel (the temporal order is indexed by t in ω below) with additional constraints. The constraints ensure that the sequence behaves like a queue in that a process may only read in order from the beginning and write in order onto the end.

Associated with a channel c we have three kinds of place. The temporal position of a value written is represented by places

$(c, t, -)$

where $t \in \omega$.

This means the t^{th} value has not yet been written to c but all previous values have been written to c . Accordingly the place

$(c, 0, -)$

is marked initially.

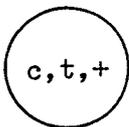
To keep track of what values have been written to c , for future reading we have places

(c, t, v)

where $t \in \omega$ and $v \in V$.

This means the t^{th} value has been written to c , it is v , and it has not yet been read from c .

Lastly, we have a further set of places to guarantee a process reads in order from the beginning of the queue. These are



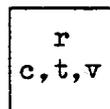
where $t \in \omega$.

This means the t^{th} value has been written to c and read from c .

The events will be of two forms. We have, for c a channel, $t \in \omega$ and $v \in V$,

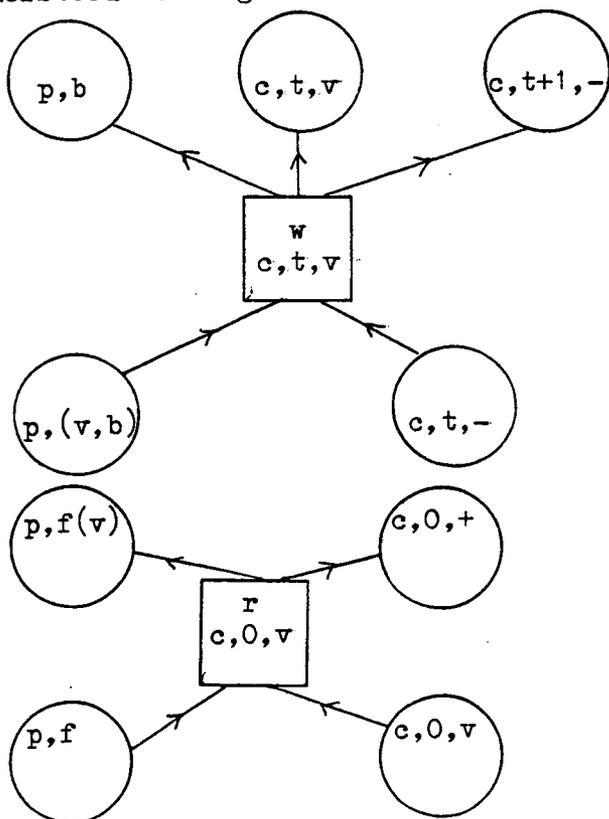


and



corresponding to the actions of writing and reading value v as the t^{th} value of c .

The transition net with initial marking is determined by the pre and post conditions of the events. We draw these now, but only for those channels c such that $w(c)$ and $r(c)$ exist; otherwise simply omit places referring to the non-existent process. The variables used are understood to range over the obvious sets.

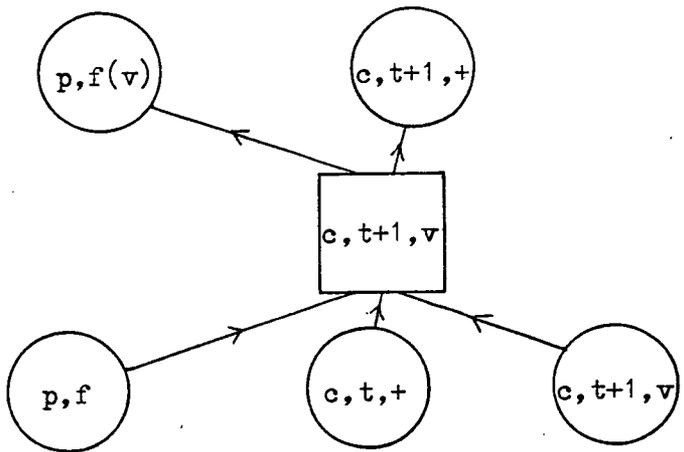


Writing

A process can write to a t^{th} value v to a channel c only if the channel has had the previous value written to it.

Initial reading

A process can read the initial value provided it has not yet been read off.



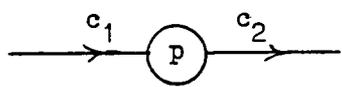
Further reading

A process can read off the $t+1^{\text{th}}$ value once the $t+1^{\text{th}}$ value has *been* written to c and the t^{th} value has been read off c .

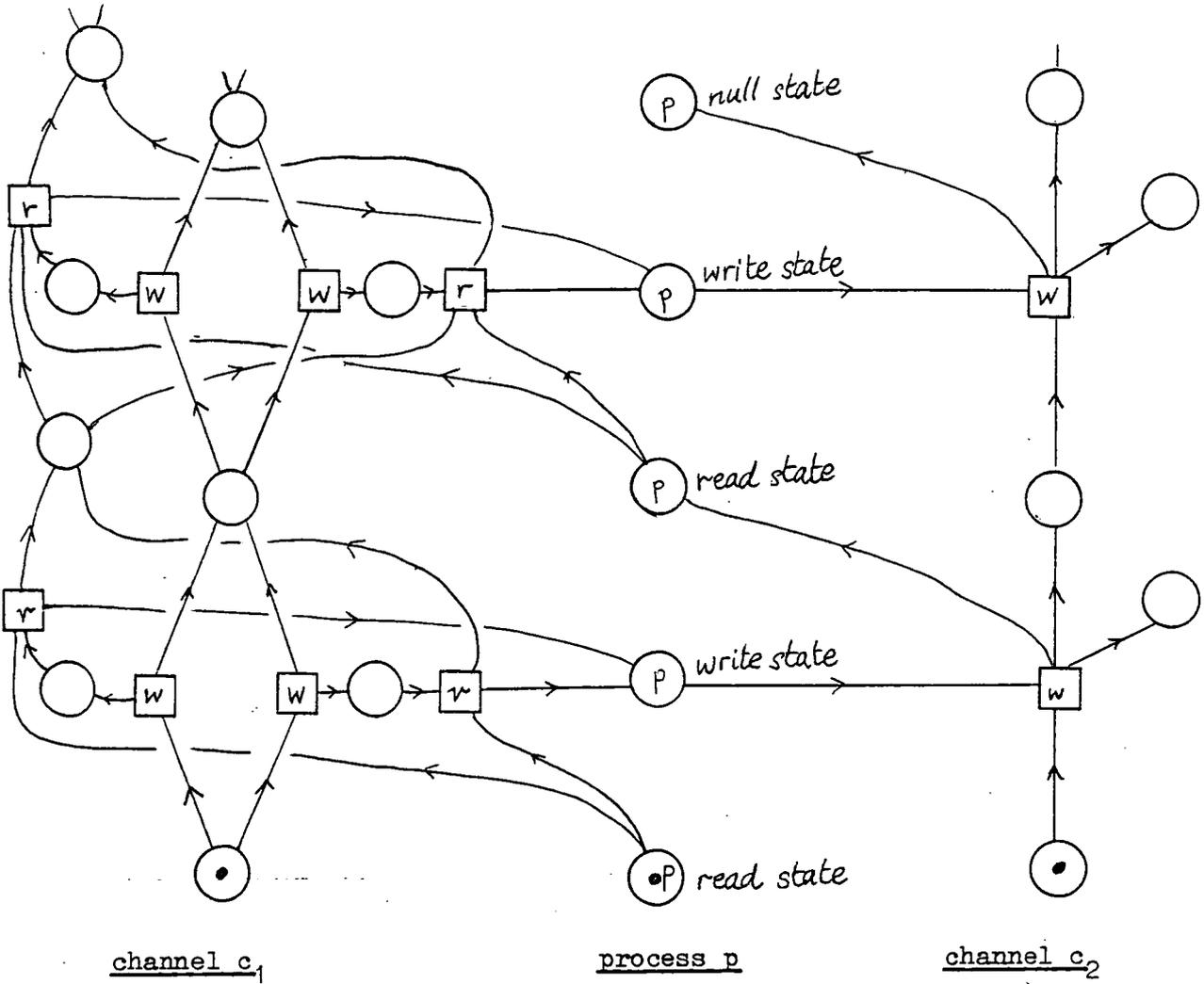
Again, as with our transition net models of Milner nets, we have problems with the standard definition of concession. It is possible for an event, which we would like to be able to fire, to have a place which is both a pre and post condition. This occurs for example if a process has behaviour f with $f(v) = f$ for some value v . (Then f will be an infinite behaviour.) Here again the revised definition of concession is appropriate. Recall this says an event e has concession for a marking M iff $\bullet e \subseteq M$ and $(e^\bullet \setminus \bullet e) \cap M = \emptyset$. In the following example, where the process has finite behaviour, the standard definition of concession works.

Example 2.3.9

In this example a process p reads, outputs, reads again then outputs again before going into the null state. The network is



where c_1 takes values 0 or 1 and c_2 takes only 0's as values. We draw the associated transition net derived from our construction, marking those conditions which represent the states of p and whether events are reading (r) or writing (w) actions. We first draw the net so as to exhibit the subnets corresponding to c_1 , p and c_2 . We also draw the subnet of c_1 so as to separate the writing-part and reading-part. The reading-events of c_1 are identified with reading transitions of p and the writing-events of p are identified with writing-events of c_2 . The identification is marked by a dotted line. Note the writing events of c_1 depend on the external environment.



The above example illustrates a computation which can be viewed more abstractly as determining a function from an input datatype (associated with c_1) to an output datatype (associated with c_2). The process will read a value if it is in a read-state and there is a value to read. Also it will write a value if it is a write-state. The corresponding transitions are thus restless. However the write-transitions of c_1 are not; they depend on the outside world.

In the examples we have given particular constructions of transition nets modelling computations. In example 2.3.9 many other transition net descriptions are possible even once the interpretation of transitions has been fixed. One would like a means of expressing the relationship between net descriptions which in particular induces notions of equivalence (the latter corresponding

to "are essentially the same description of a computation").

2.4 Causal nets, cases and K-density

Historically transition nets came first in the development of net theory. Later Petri, in particular, has attempted to develop the foundations of net theory by analysing the assumptions to be made at "lower conceptual levels" [Pet 1]. It is hoped that a theory of morphisms (see section 2.5) will make this precise.

Causal nets [Pet 1] appear at the "first conceptual level". A transition net description of a computation determines a set of possible courses (called "processes" by Petri in [Pet 1]) the computation may take. (We avoid the words "history" or "run" as for us they invoke a time-scale.) Petri requires a type of net to formalise the idea of a course of computation. At the very least he requires such nets to be causal nets. In addition he also requires them to be K-dense. Petri has said that the set of causal nets associated with a transition net constitutes its semantics [Pet 2].

There are difficulties with the formalisation of the idea of course of a computation by causal nets. A causal net is being used as a net-analogue of history. As such the events are regarded as eventually occurring so we encounter the restless events issue again. It appears courses are allowed to have infinite pasts which introduces some subtleties (see chapter 7). Also, importantly, K-density seems far too restrictive an axiom. As we shall argue against it later (see chapter 5) we shall spell out the arguments given for K-density in [Pet 1] and [Bes]. The axiom of K-density involves the net-theoretic idea of state of a causal net, called a case.

As we mentioned, the courses of a computation must at least be representable by causal nets. As net analogues of histories they do not possess conflict. However causal nets are not marked so this is banned in a formal way by the axioms N4 and N5. In order that the events and conditions of a causal net correspond to occurrences loops in F^* are also disallowed (axiom N6). (Note as our definition of a Petri net is a little more general than usual so too is our definition of a causal net.)

Definition 2.4.1

A Petri-net $N = (B, E, F)$ is a causal net iff

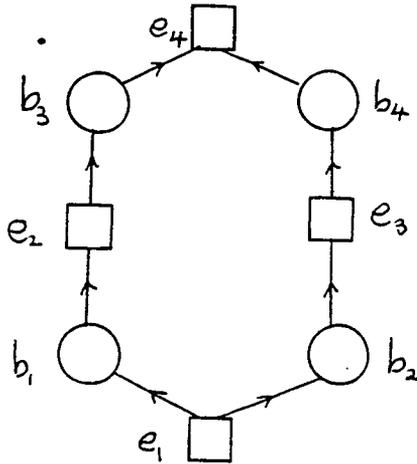
N4. $\forall b \in B \ |b^\bullet| \leq 1$

N5. $\forall b \in B \ |\bullet b| \leq 1$

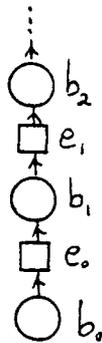
N6. F^+ is irreflexive.

The following are examples of causal nets which we shall refer to later.

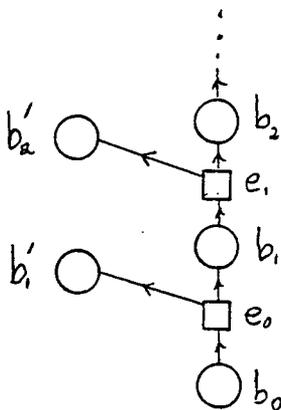
Example 2.4.2

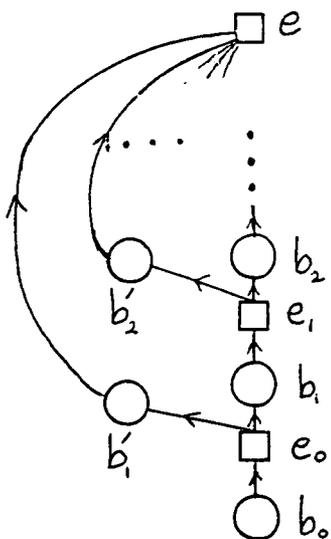
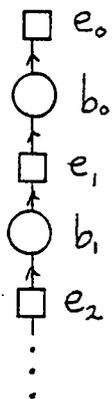
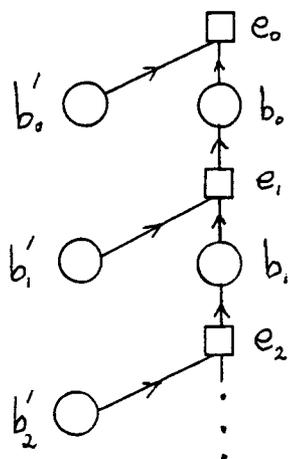


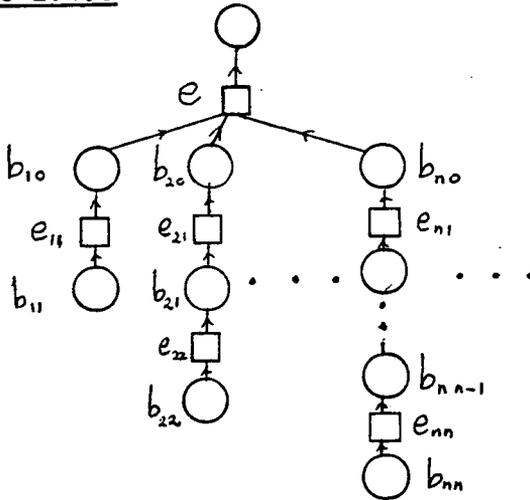
Example 2.4.3



Example 2.4.4



Example 2.4.5Example 2.4.6Example 2.4.7

Example 2.4.8

Note in example 2.4.5 an event e is dependent on an infinite chain of events e_0, e_1, \dots . In examples 2.4.6 and 2.4.7 the event e_0 is dependent on an infinite chain of events e_1, e_2, \dots stretching into the past. In example 2.4.8 the event e depends on chains of events of unbounded length.

For a causal net it is easy to define a concurrency relation, representing causal independence between events and conditions; it is simply the complement of the causal dependency $F^+ \cup F^{+ -1}$.

Definition 2.4.9

For a causal net $N = (B, E, F)$ the concurrency relation $co_N \subseteq (B \cup E) \times (B \cup E)$ is defined by

$$co_N = (B \cup E) \times (B \cup E) \setminus (F^+ \cup (F^+)^{-1})$$

From our axioms on causal nets it follows that co_N is symmetric and reflexive and that any two elements of $B \cup E$ are either causally dependent or concurrent.

The concurrency relation is used in defining the net-theoretic notion of state. This is taken to be a maximal subset of $B \cup E$ pairwise related under co_N , and is called a case. This form of definition occurs frequently in dealing with nets so we spend a little time on notation.

Proposition 2.4.10

Let X be a set with binary relation R s.t. $R \supseteq 1_X$ (the identity on X). Then a ken of R in X is defined to be a maximal subset of pairwise R -related elements of X . Note, for $Y \subseteq X$, Y is

a ken of R in X iff the following holds:

$$\forall x \in X (\forall y \in Y xRy \iff x \in Y).$$

Definition 2.4.11

Let N be a causal net (B,E,F) with concurrency relation CO_N. A case of N is defined to be a ken of CO_N in B ∪ E.

The definition of case (only defined for causal nets) is intended to formalise some notion of global state. In example 2.4.2 {e₄}, {b₂,b₃} and {e₂,b₂} are some of the cases. In examples 2.4.4 and 2.4.5 {b₀}, {b₁,b'₁}, {e₁,b'₁}, {b'₁,b'₂,...,b'_n,b_n} as well as the infinite set {b'_n | n = 1,2,...} are cases.

To state the axiom of K-density we need a further definition.

Definition 2.4.12

Let N = (B,E,F) be a causal net. A sequential process of N is a ken of (F* ∪ F*⁻¹) in B ∪ E.

The name "sequential process" is apt for the "subnets" corresponding to Milner's agents or Kahn-MacQueen processes when there is no conflict. Note sequential processes may possess a variety of order-types. In examples 2.4.6 and 2.4.7 the sequential process ({e_i | i ∈ ω} ∪ {b_i | i ∈ ω}) has order-type ω. In example 2.4.5 the sequential process ({b_i | i ∈ ω} ∪ {e_i | i ∈ ω} ∪ {e}) has order-type ω + 1.

Now we state the axiom of K-density giving our intuitive interpretation of it later. It says any case determines a unique "local state" of a sequential process.

Definition 2.4.13 (The axiom of K-density)

Let N = (B,E,F) be a causal net. The net N is said to be K-dense iff every case intersects every sequential process.

Notice that because of the properties of CO any non-null intersection of a case and a sequential process is a singleton. As Petri noted, any finite net is K-dense. Also the nets in examples 2.4.2, 2.4.3, 2.4.6 and 2.4.8 are K-dense. However the nets of examples 2.4.4, 2.4.5 and 2.4.7 are not. In examples 2.4.4 and 2.4.5 the cases described by {b'_n | n = 1,2,...} do not meet the sequential processes ({b_i | i ∈ ω} ∪ {e_i | i ∈ ω}) and

$(\{b_i \mid i \in \omega\} \cup \{e_i \mid i \in \omega\} \cup \{e\})$ respectively. In example 2.4.7 the case $\{b_i^i \mid i \in \omega\}$ does not meet the sequential process $(\{e_i \mid i \in \omega\} \cup \{b_i \mid i \in \omega\})$.

In [Pet 1] K-density is announced as a thesis; there it is stated that a causal net representing a course of computation is K-dense. Thus the nets of 2.4.5 and 2.4.7 and the seemingly inoffensive net of 2.4.4 are banned from representing courses of a computation. Examples 2.4.3 and 2.4.4 show that the property of being K-dense or not depends crucially on what conditions are included. As later we shall deal with event structures, essentially nets without conditions, it is important we understand at least the intuition behind K-density. In fact we shall disagree with it. To us the net of 2.4.4 seems reasonable even though, incidentally it cannot be associated with the course of a finite transition net. For instance the conditions b_n^i of 2.4.4 might correspond to resource n being made available by an agent on transition e_{n-1} from state b_{n-1} to b_n . Thus we must find a point on which to disagree.

It is hard to argue directly with the thesis in [Pet 1] or the "simplicity" - and-"attractiveness" argument in [Bes]. In contrast we sketch how K-density may be deduced once certain assumptions are made. The assumptions are based on discussion of examples in [Pet 1] and [Bes]. In representing a course of computation by a causal net we assume all conditions and events occur sometime. This can be made precise using the idea of an observer (see 5.1 and 7 for formal uses of this concept). An observer is a projection of the entire course of computation onto a time-scale; accordingly all the events and conditions of the associated causal net are ascribed extents of time consistent with the causal dependency relation. Our first assumption can be replaced by: there is an observer for the causal net. An observable state can now be defined as the set of conditions which hold and events which fire at one time according to some observer. We mentioned that cases represented a notion of global state. From [Pet 1] and [Bes] it seems that cases are observable states, our next assumption. Our final assumption may be summarised as infinite sequential processes take infinite time according to observers. By this we mean an infinite chain $x_0 F x_1 F x_2 \dots F x_n F \dots$ is never completed at any

finite future time according to an observer. Also an infinite chain $x_0 F^{-1} x_1 F^{-1} \dots F^{-1} x_n F^{-1} \dots$ never begins at a finite time in the past according to an observer.

We examine the examples to see if they are consistent with the assumptions, before deriving K-density from them. In example 2.4.4 the sequential process $(\{e_i \mid i \in \omega\} \cup \{b_i \mid i \in \omega\})$ can never be observed completed at finite time. Thus the case $\{b'_n \mid n = 1, 2, \dots\}$ is not an observable state contradicting our second assumption. Thus the net of example 2.4.4 cannot represent courses of computation according to the assumptions. We have already seen that it is not K-dense. Similarly the net of 2.4.5 fails the assumptions. (In addition the event e could never be observed.) The non-K-dense net of example 2.4.7 has a case which can only hold in the infinite past, again contradicting the assumptions. The remaining examples of causal nets do not contradict any assumptions.

We now outline the argument for K-density. Suppose a causal net were not K-dense. That is, suppose some sequential process did not meet some case for N. Eike Best has shown that this implies one or other of the following situations [Bes]. Either there is a case C above an infinite F^* -chain i.e. there is $x_0 F x_1 F x_2 \dots F x_n F \dots$ in N with $\forall x_i \exists c \in C x_i F^+ c$ or there is a case C below an infinite F^{*-1} -chain i.e. there is $x_0 F^{-1} x_1 F^{-1} x_2 \dots F^{-1} x_n F^{-1} \dots$ with $\forall x_i \exists c \in C c F^+ x_i$.

The two situations can be seen in examples 2.4.4 and 2.4.7 respectively. In the first situation the case can only be seen by an observer in the infinite future while, for the second, it can only be seen in the infinite past. In both situations we contradict at least one of our assumptions.

Whether or not the above assumptions are acceptable to net-theorists, in rejecting K-density we must reject at least one assumption. In future we shall not assume cases are observable states. If our analysis is correct our disagreement with Petri's foundational work on net theory is as fundamental as the notion of state. Of course, there is something correct in the spirit of K-density; for the most part one does rule out courses of computation like that described by 2.4.5 in which an event depends on an infinite chain. (Such computations represent Zeno machines

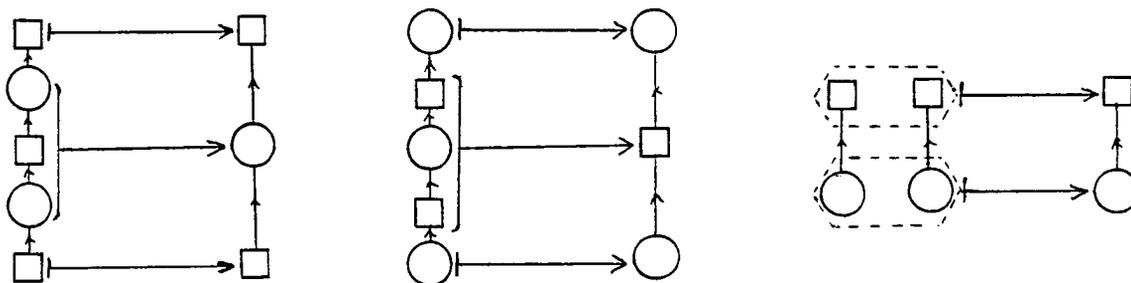
[Hew]). Also note we expect a revised form of K-density to hold when cases are restricted to being observable.

2.5 Net morphisms

Net morphisms are intended to provide a framework for operations on nets like refinement, contraction, extension, restriction and completion (see [N.Proc] - we shall illustrate some of them). The current definition of net morphism ([Pet 1] and [N.Proc]) does not take into account markings, cases or any other representation of the idea of state. Roughly it is a local definition based on the idea that conditions and events are generalisations of respectively open and closed connected intervals of time. We try to explain the idea of it before giving the formal definition. Firstly assume a morphism from a net N_0 to N_1 is a function f from the elements of N_0 to the elements of N_1 . It is reasonable that it should be F-respecting that is:

$$x F_0 y \Rightarrow f(x) F_1 \cup 1 f(y)$$

Thus maps like these are allowed so far:



The first two "collapse" part of the net while the third "identifies" elements of the net. However note at present the following maps are allowed too:



Taking composition as the usual function composition gives the nets  and  are isomorphic. In this sense we fail to account for the different nature of events and conditions. The net-topology is intended to do this. In the topology singletons of conditions are open and singletons of events closed.

Proposition 2.5.1

Let $N = (B, E, F)$ be a net. Taking as open sets those subsets X of $B \cup E$ satisfying $\forall e \in X \cap E \ 'e \subseteq X \ \& \ e' \subseteq X$ gives a topology (the net topology). Closed sets are characterised as being subsets X such that $\forall b \in B \cap X \ 'b \subseteq X \ \& \ b' \subseteq X$.

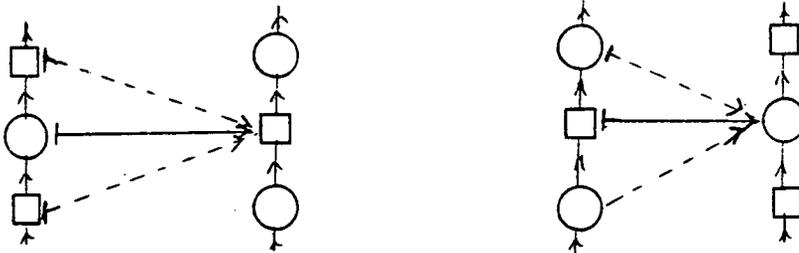
Thus if an open set contains an event it must include its pre and post conditions. If a closed set contains a condition it must also include its pre and post events. (Note the symmetry in the definitions of open and closed - the closed sets also form a topology.)

Currently a morphism is defined to be a map which is F -representing and continuous with respect to the topology.

Definition 2.5.2

Let $N_i = (B_i, E_i, F_i)$ for $i = 0, 1$ be two Petri nets. Then a net morphism from N_0 to N_1 is defined to be a map $f: B_0 \cup E_0 \rightarrow B_1 \cup E_1$ which is such that (i) $x F_0 y \Rightarrow f(x) F_1 \cup \mid f(y)$
(ii) f is continuous with respect to the net topology.

Diagrammatically, continuity implies the dotted arrows follow from the solid arrows in "building-up" the two morphisms below:



The further property of respecting F guarantees that the causal dependency relation cannot switch direction under a morphism.

In fact morphisms may be defined in an alternative way as those maps respecting the F -relation and an adjacency relation (generally denoted P) which we now define.

Definition 2.5.3

Let $N = (B, E, F)$ be a net. Define the adjacency relation P to be the relation $B \times E \cap (F \cup F^{-1})$.

Lemma 2.5.4

Let $N_i = (B_i, E_i, F_i)$ for $i = 0, 1$ be two Petri nets with adjacency relations P_0 and P_1 as defined above. Then a map

$f: B_0 \cup E_0 \rightarrow B_1 \cup E_1$ is a net morphism iff

(i) $x F_0 y \Rightarrow f(x) F_1 \cup 1f(y)$ and

(ii) $x P_0 y \Rightarrow f(x) P_1 \cup 1f(y)$.

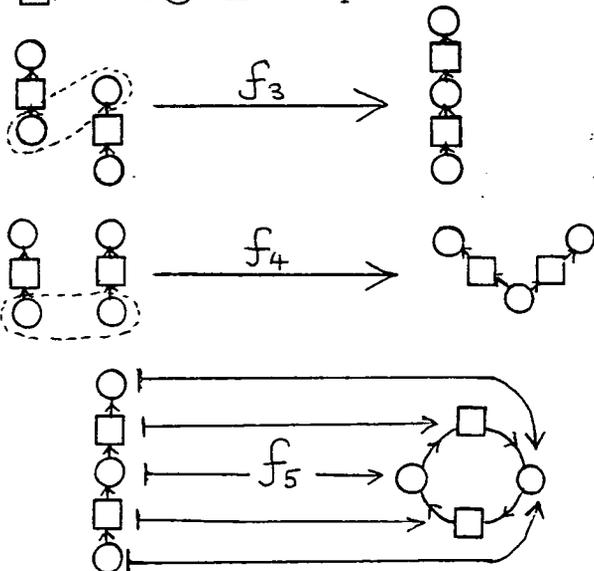
Proof

Suppose f is a net morphism $N_0 \rightarrow N_1$. We require f to be P -respecting. Suppose $x P_0 y$. Then for some $b \in B_0$ and $e \in E_0$ either $b F_0 e$ or $e F_0 b$ if $b F_0 e$ then $f(b) F_1 \cup 1f(e)$. Thus if $f(b) \in B_1$ we have $f(b) P_1 \cup 1f(e)$ as required. Otherwise $f(b) \in E_1$. Then as f is continuous closed sets pull back to closed sets under f^{-1} . This means as $b \in f^{-1}\{f(b)\}$ we must have $e \in f^{-1}\{f(b)\}$ i.e. $f(e) = f(b)$. Thus $f(b) P_1 \cup 1f(e)$ as required. Similarly if $e F_0 b$.

Suppose f is a map $B_0 \cup E_0 \rightarrow B_1 \cup E_1$ such that (i) and (ii) above hold. We check f is continuous. Suppose $e \in f^{-1} X$ i.e. $f(e) \in X$. If $e F_0 b$ then $f(e) F_1 \cup 1f(b)$. Thus assuming $f(e) \in E_1$ gives $f(b) \in X$ i.e. $b \in f^{-1} X$. Otherwise $f(e) \in B_1$ in which case as $b P_0 e$ we have $f(b) P_1 \cup 1f(e)$ so $f(b) \in X$ too. By the defn. of P_1 , $f(b) = f(e)$ so $b \in f^{-1} X$. Thus $e \in f^{-1} X$ implies $e \subseteq f^{-1} X$. Similarly $e \in f^{-1} X \Rightarrow \bullet e \subseteq X$. This means $f^{-1} X$ is open as required for f to be continuous. ■

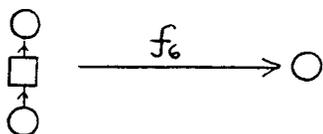
Example 2.5.5 (Some morphisms)

Recall we allow nets to be singletons so $f_1: \bigcirc \mapsto \square$ and $f_2: \square \mapsto \bigcirc$ are morphisms. So are these:



The maps f_3 and f_4 "pinch together" the encircled conditions.

The map f_5 introduces a loop by identifying top and bottom conditions.



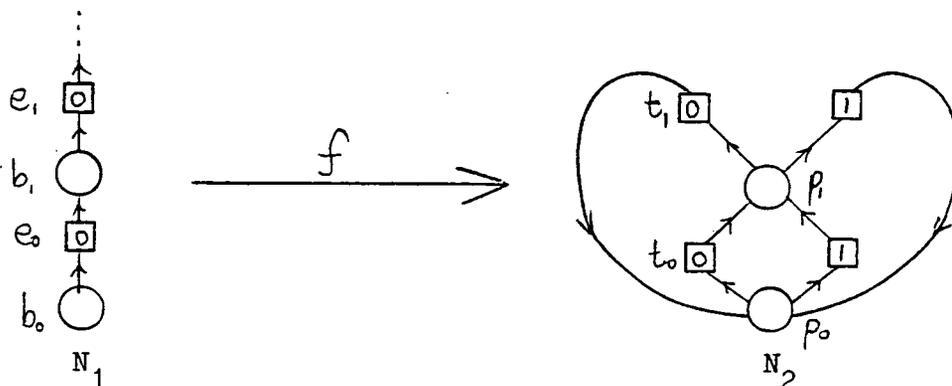
The map f_6 "collapses" the small net on the left to a condition.

It is hard to see a uniform intuitive interpretation of the above morphisms. (For example the obvious maps induced on markings by f_3 and f_6 are in opposite directions.)

There are possible criticisms of the above definition of morphism. There may not be an intuitively acceptable "morphism" which fails either of the properties (i) or (ii) in 2.5.2. However the definition is perhaps too general in that it allows morphisms which are hard to justify intuitively. As remarked a morphism as defined in 2.5.2 takes no account of markings and markings are crucial to the dynamic behaviour of the token game.

We look at some specific intended uses of net morphisms. According to their use we expect further restrictions in their definition. Recall that certain types of causal net are the net-theoretic representation of the possible courses of a computation described by a transition-net (section 2.4). The fact that a causal net N_1 is the course of a computation described by a transition net N_2 is represented by a special form of morphism from N_1 to N_2 called a folding. Example 2.3.6 showed a folding. Before the formal definition of a folding we give a further example where the net folded is a causal net. Petri has said that the class of causal nets which fold into a transition net constitute its semantics [Pet 2].

Example 2.5.6



Here the net N_1 corresponds to an infinite tape of 0's while the net N_2 represents the datatype consisting of possibly infinite tapes of 0's and 1's. The net N_1 might be the output from a computation with possible outputs represented by N_2 . The map f is defined by:

$$\begin{aligned} f(b_i) &= p_0 \text{ if } i \text{ is even, } p_1 \text{ otherwise} \\ f(e_i) &= t_0 \text{ if } i \text{ is even, } t_1 \text{ otherwise.} \end{aligned}$$

The map f is an example of a folding. We have ignored initial markings and the fact that all the events of N_1 are supposed to occur eventually (they are restless).

Definition 2.5.7

Let N_0 and N_1 be nets. Then a map $f: B_0 \cup E_0 \rightarrow B_1 \cup E_1$ is a folding iff

- (i) $x F_0 y \Rightarrow f(x) F_1 f(y)$
- (ii) $f B_0 \subseteq B_1$ & $f E_0 \subseteq E_1$

This differs from the definition in [Pet] where instead of (ii) there is the property f preserves P . However when the field of F is $B \cup E$, an assumption generally made on nets, (i) gives that (ii) above is equivalent to f being P -preserving.

In modelling Kahn-MacQueen networks by transition nets we saw how nets representing datatypes were, in some sense, subnets of nets giving a more complete description of the computation. We give a formal definition of the idea of subnet now.

Definition 2.5.8

Let $N_i = (B_i, E_i, F_i)$ be nets, for $i = 0, 1$. Then a map $f: B_0 \cup E_0 \rightarrow B_1 \cup E_1$ is a subnet morphism iff f is a 1-1 net morphism sending conditions to conditions, events to events and such that $f(x) F_1 f(y) \Rightarrow x F_0 y$.

If f is the inclusion map then N_1 is a subnet of N_2 .

We confess that the extra restriction of preserving events and conditions is redundant in the presence of the assumption generally made on nets $N = (B, E, F)$ that $B \cup E$ equals the field of F i.e.

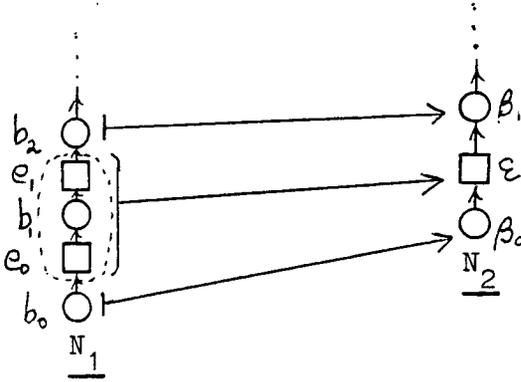
$$\forall x \in B \cup E \exists y \in B \cup E \text{ } x F y \text{ or } y F x.$$

Then the assumption of f being a 1-1 morphism implies f preserves

events and conditions; it does not imply f is a subnet morphism, however.

We illustrate another type of morphism which seems important though we shall not give it a formal definition because there appear to be difficulties.

Example 2.5.9 (Contraction)



The map f drawn schematically above contracts the "boxed-off" part of N_1 to a single event of N_2 .

The map f of the above example is certainly a morphism. It has a seemingly natural interpretation: N_2 is a more coarsely grained description than N_1 with event ϵ standing for the subcomputation described by $e_0 F b_1 F e_1$. With this interpretation there is a problem if e_0 occurs but e_1 never occurs. Then correspondingly the event would begin but never end firing. This contradicts one intuition about events namely that occurrences of events should take up extents of time which are compact connected intervals. The situation can be remedied for example 2.5.9 by ensuring that e_1 will occur once e_0 has occurred. However the extra structure is necessary to reflect this fact and ensure f does not violate our intuitions about events.

Of course, for another interpretation of f the above argument may not even make sense. For instance one could think of f as standing for a computation from an input datatype described by N_1 to an output datatype described by N_2 ; the map f then determines the output values produced by input values (cf. examples 2.3.7 and 2.3.9). This points out the danger of not having a precise interpretation in mind; non-commitment to a particular interpretation can lead to at best vagueness ~~at~~ worst error and rarely to a theorem.

Chapter 3. Introduction to concrete domains and sequentiality

In this chapter we see how the idea of events came to be treated formally and explicitly within denotational semantics. This arose through the collaborative work of Gilles Kahn and Gordon Plotkin in formalising the idea of concrete datatypes (or domains) and sequential functions in the autumn of 1975 ([Kah and Plo]). Concrete domains are domains of information about "basic" input or output which also support a general and natural notion of sequential function. Kahn and Plotkin discovered that their concrete domains were represented by matrices, objects similar in form to Petri nets.

In the first section we give some background results from denotational semantics with some illustrations of Dana Scott's idea of information ([Sco]). The presentation is inevitably rather "racy"; for further background see [Gor] for applications and [Wad] for theory and practice.

In the second section we outline in fair detail the fundamental results on concrete domains, how they are represented by matrices (the representation theorem) and the definition of sequential function. The relevant work here is [Kah and Plo], [Cur] and [Ber and Cur]. In the latter, Gerard Berry and Pierre-Louis Curien produce a cartesian closed category of concrete domains taking algorithms (an abstract form of deterministic program) as morphisms. They show sequential functions are precisely those functions realised by algorithms. We omit the category theoretic aspects of concrete domains, in particular rigid embeddings, which enable concrete-domain solutions to a restricted form of recursive domain equation.

In the final section, a kind of appendix, we prove the representation theorem in detail. (In fact we prove a more general result for a kind of event structure.)

3.1 Background material

In denotational semantics the meaning of a programming construct such as a procedure or command is denoted by an element of a particular form of partial order called a domain. The partial ordering reflects an idea of information.

Definition 3.1.1

A partial order (D, \sqsubseteq) is composed of a set D and an ordering



relation \subseteq on D that is a binary relation \subseteq satisfying

- (i) $\forall x \in D \ x \subseteq x$ (reflexivity)
- (ii) $\forall x, y \in D \ x \subseteq y \ \& \ y \subseteq x \Rightarrow x = y$ (antisymmetry)
- (iii) $\forall x, y, z \in D \ x \subseteq y \ \& \ y \subseteq z \Rightarrow x \subseteq z$ (transitivity)

We write $x \subset y$ for $x \subseteq y \ \& \ x \neq y$. We sometimes write $x \supseteq y$ for $y \subseteq x$. Two elements x and y are comparable when $x \subseteq y$ or $y \subseteq x$; otherwise they are incomparable. If $x \subseteq y$ we sometimes say y dominates x .

Notation

Let (D, \subseteq) be a partial order, X a subset of D and y a member of D . Then y is an upper bound of X iff $\forall x \in X \ x \subseteq y$ (we abbreviate this to $X \subseteq y$); similarly y is a lower bound of X iff $\forall x \in X \ y \subseteq x$ (abbreviated to $y \subseteq X$). The supremum of X , written $\sqcup X$, is an upper bound which is dominated by all upper bounds of X . The infimum of X , written $\sqcap X$, is a lower bound which dominates all lower bounds of X . If X is $\{a, b\}$ we write $a \sqcup b$ and $a \sqcap b$ for $\sqcup X$ and $\sqcap X$ respectively. If X possesses an upper bound we say X is compatible (and write $X \uparrow$) *otherwise* incompatible (and write $X \narrow$). If X is $\{x, y\}$ we write $X \uparrow$ as $x \uparrow y$ and $X \narrow$ as $x \narrow y$.

Definition 3.1.2

In a particular order (D, \subseteq) a subset S of D is directed iff S is non-null and $\forall s_1, s_2 \in S \ \exists s_3 \in S \ s_1 \subseteq s_3 \ \& \ s_2 \subseteq s_3$.

For example an ω -chain $x_1 \subseteq x_2 \subseteq \dots \subseteq x_n \subseteq \dots$ is directed.

Definition 3.1.3

A partial order (D, \subseteq) is a complete partial order (cpo) iff

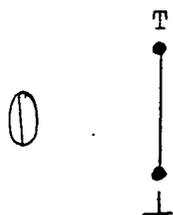
- (i) D has a minimum element \perp
- (ii) All directed subsets of D have a supremum in D .

Cpos are the objects in which denotations are taken. They are often called (semantic) domains. In a cpo the elements of a directed set S can be thought of as earlier approximations to the element $\sqcup S$ which the directed set eventually determines. There is another possible definition of cpo in terms of ω -chains which is perhaps more intuitive. In the presence of natural restrictions the two notions coincide. We choose to work with directed sets simply

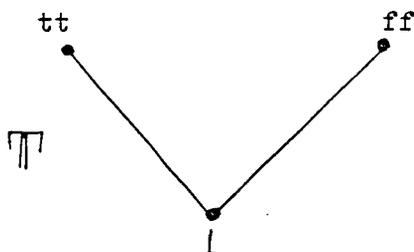
because this is the most common approach in the literature.

Example 3.1.4 ($\mathbb{0}$)

$\mathbb{0}$ is a very useful little domain consisting of 2 elements \perp and \top with $\perp \sqsubseteq \top$. It looks like this:

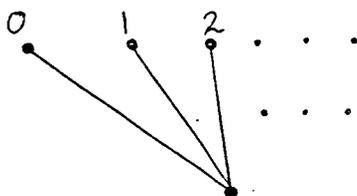


Example 3.1.5 (\mathbb{T} - the domain of truth values or Booleans)



The domain \mathbb{T} is represented above; it consists of a set $\{\perp, tt, ff\}$ with $\perp \sqsubseteq tt$ and $\perp \sqsubseteq ff$. The symbol tt denotes true and ff false. The set of tt and ff is incompatible. We give an idea of the intuition behind the ordering. Suppose a computation may give a single truth value as output. Before it has terminated with a value we have information \perp about the output i.e. no information at all. Once it terminates with value true we have information tt and similarly if it terminates with false we have information ff . If it should diverge (never terminate) we always have information \perp about the output. The information \perp may grow into the information tt or the information ff .

Example 3.1.6 (\mathbb{N} - the domain of integers)



\mathbb{N} consists of $\perp \cup \{\omega\}$ (where ω denotes the natural numbers) ordered by $\perp \sqsubseteq n$ for all n in ω . The intuition of the ordering

is like that for \mathbb{T} . All the domains \mathcal{O} , \mathbb{T} and \mathbb{N} are examples of discrete (or flat) cpos. They are formed by adjoining the element \perp below a set. In them information has an all-or-nothing character; in \mathbb{T} for example the information is either a truth value or nothing at all \perp . \int These two properties of domains crop up frequently:

Definition 3.1.7

Let (D, \sqsubseteq) be a cpo.

It is consistently complete iff for all compatible subsets X we have the supremum $\bigsqcup X$ exists in D .

Say X a subset of D is pairwise-compatible iff for all x, y in X we have x and y are compatible. The cpo (D, \sqsubseteq) is coherent iff every pairwise-compatible subset X has a supremum $\bigsqcup X$ in D .

Example 3.1.8



The first domain is not consistently complete while the second is but is not coherent. Thus coherence is strictly stronger than consistent completeness.

Consistent completeness has this characterisation:

Lemma 3.1.9

A cpo (D, \sqsubseteq) is consistently complete iff all compatible pairs $x \uparrow y$ have a supremum $x \sqcup y$.

Proof Suppose all compatible pairs of D have suprema.

Suppose $X \subseteq D$. If $X = \emptyset$ then $\bigsqcup X = \perp$. If X is non-null take S to consist of elements $x_1 \sqcup x_2 \dots \sqcup x_n$ for x_1, \dots, x_n in X . (We get $x_1 \sqcup \dots \sqcup x_n$ exist in D by a simple induction.) Then S is directed so $\bigsqcup S$ exists and is easily checked to be $\bigsqcup X$. The converse is trivial. ■

Consistent completeness implies infima always exist for non-null subsets.

Lemma 3.1.10

Let D be a consistently complete cpo. Then for all non-null subsets X of D , $\bigsqcap X$ exists in D .

Proof

Let X be a non-null subset of D . Define $Y = \{y \in D \mid y \in X\}$. Then $\bigsqcup Y$ exists and may be checked to be $\bigsqcap X$. ■

We now look at functions between partial orders.

Definition 3.1.11

Let (D_i, \sqsubseteq_i) for $i = 0, 1$ be two partial orders. A function $f: D_0 \rightarrow D_1$ is monotonic iff $\forall x, y \in D_0 \ x \sqsubseteq_0 y \Rightarrow f(x) \sqsubseteq_1 f(y)$. The function f is an order isomorphism iff there is a monotonic $g: D_1 \rightarrow D_0$ such that $g \circ f = 1_{D_0}$ and $f \circ g = 1_{D_1}$. (This is equivalent to f being 1-1, monotonic and $f(x) \sqsubseteq_1 f(y) \Rightarrow x \sqsubseteq_0 y$ for x, y in D_0 .) Then D_0 and D_1 are (order-) isomorphic. We are interested in computable functions. Suppose a computation gives output according to input. For more input information it will give more output information. Thus it will correspond to a function f between the domains of information which is monotonic. The input information may be presented over time (possibly unbounded) as a chain $x_0 \sqsubseteq x_1 \sqsubseteq \dots \sqsubseteq x_n \sqsubseteq \dots$ which has supremum $\bigsqcup \{x_n \mid n \in \omega\}$. The corresponding output information will be $f(x_0) \sqsubseteq f(x_1) \dots \sqsubseteq f(x_n) \sqsubseteq \dots$ with supremum $\bigsqcup \{f(x_n) \mid n \in \omega\}$. We expect the eventual output for the eventual input $\bigsqcup \{x_n \mid n \in \omega\}$ to be no more than the supremum $\bigsqcup \{f(x_n) \mid n \in \omega\}$. This means we require $f(\bigsqcup \{x_n \mid n \in \omega\}) = \bigsqcup \{f(x_n) \mid n \in \omega\}$. It is this intuition which the continuity restriction on functions expresses. (See [Sco], [Wad]).

We give the definition in terms of directed sets rather than ω -chains because this is the most common approach. (For ω -algebraic domains for instance the two definitions agree.)

Definition 3.1.12

Let (D_0, \sqsubseteq_0) and (D_1, \sqsubseteq_1) be two cpos. A function $f: D_0 \rightarrow D_1$ is continuous iff it is monotonic and for all directed sets S of D_0 $f(\bigsqcup_0 S) = \bigsqcup_1 \{f(s) \mid s \in S\}$.

Proposition 3.1.13

The continuity property is preserved by the usual function composition. If D is a cpo the identity function 1_D is continuous.

This means cpos and continuous functions form a category. In fact it is a cartesian closed category with product and exponentiation objects given by the following constructions.

Definition 3.1.14

Let (D_0, \sqsubseteq_0) , (D_1, \sqsubseteq_1) be two cpos. Define their product $D_0 \times D_1$ to be all pairs $D_0 \times D_1$ ordered co-ordinatewise by

$(x_0, x_1) \sqsubseteq (y_0, y_1)$ iff $x_0 \sqsubseteq_0 y_0$ & $x_1 \sqsubseteq_1 y_1$. Define their function space $[D_0 \rightarrow D_1]$ to consist of all continuous functions $f: D_0 \rightarrow D_1$ ordered pointwise by $f \sqsubseteq f'$ iff $\forall x \in D_0 f(x) \sqsubseteq_1 f'(x)$. (The definition of product generalises to arbitrary sets of cpos.)

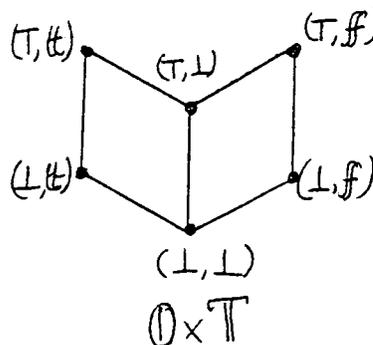
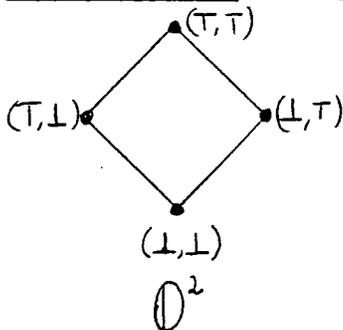
Proposition 3.1.15

The product $D_0 \times D_1$ of two cpos D_0 and D_1 is a cpo with minimum element $\perp = (\perp_0, \perp_1)$; the supremum of a directed set S of $D_0 \times D_1$ is $(\bigsqcup_0 S_0, \bigsqcup_1 S_1)$ where $S_0 = \{x_0 \mid \exists x_1 (x_0, x_1) \in S\}$ and $S_1 = \{x_1 \mid \exists x_0 (x_0, x_1) \in S\}$.

The function space $[D_0 \rightarrow D_1]$ of two cpos D_0 and D_1 is a cpo with minimum element $\perp: x_0 \mapsto \perp_1$; the supremum of a directed set S of $[D_0 \rightarrow D_1]$ is the function $x \mapsto \bigsqcup_1 \{f(x) \mid f \in S\}$.

A function f from $D_0 \times D_1$ is continuous iff it is continuous in each argument separately (i.e. the function $\lambda x_0. f(x_0, x_1)$ is continuous for all x_1 and $\lambda x_1. f(x_0, x_1)$ is continuous for all x_0).

Of course, the function space generally includes far more functions than the computable ones. To see how the theory of computability can be grafted onto domains see [Smy] for example.

Example 3.1.16 (Two products)Example 3.1.17 ($[\mathbb{N} \rightarrow \mathbb{N}]$)

The continuous functions $\mathbb{N} \rightarrow \mathbb{N}$ form the domain $[\mathbb{N} \rightarrow \mathbb{N}]$. Here all monotonic functions $\mathbb{N} \rightarrow \mathbb{N}$ are continuous and the pointwise ordering gives $f \sqsubseteq f'$ iff

$$\forall x \in \mathbb{N} f(x) = n \in \omega \Rightarrow f'(x) = n.$$

Thus $f \subseteq f'$ means "less defined than". Some maximal functions of $[\mathbb{N} \rightarrow \mathbb{N}]$ are of the form $f: x \mapsto n$ for all x in \mathbb{N} and some fixed $n \in \omega$; then $f(\perp) = n$ so the function "disregards" the input and always outputs n . The other maximal functions induce total functions $\omega \rightarrow \omega$ and must act so $\perp \mapsto \perp$ to guarantee monotonicity. Clearly there are many more continuous functions $\mathbb{N} \rightarrow \mathbb{N}$ than there are computable functions.

The least-fixed-point operator is used to give a denotation to recursively defined functions or procedures and iterative constructs like while loops. If D is a domain and f is a function in $[D \rightarrow D]$ then the least-fixed-point operator acts on f to give its least fixed point.

Proposition 3.1.18

Let D be a cpo.

- (i) If $f \in [D \rightarrow D]$ then the least fixed point of f exists and is $Y(f) =_{\text{def}} \bigsqcup \{f^n(\perp) \mid n \in \omega\}$
- (ii) The function $Y: [D \rightarrow D] \rightarrow D$ given above is continuous.

Proof

We shall only prove (i). For f in $[D \rightarrow D]$ it is clear that $\perp = f^0(\perp) \subseteq f(\perp) \subseteq \dots \subseteq f^n(\perp) \subseteq \dots$ is an ω -chain and so forms a directed set. Continuity of f gives $f(Y(f)) = Y(f)$ so $Y(f)$ is a fixed point. Suppose x is another fixed point of f i.e. $f(x) = x$. Then as $\perp \subseteq x$ we get $f^n(\perp) \subseteq f^n(x) = x$ by repeated application of the monotonic function f . Thus $Y(f) = \bigsqcup \{f^n(\perp) \mid n \in \omega\} \subseteq x$ so $Y(f)$ is the least fixed point. ■

Example 3.1.19

We indicate how the fixed point operator is used to give denotations of recursive procedures. In a programming language a procedure giving the factorial function might be defined by:

$$f(x) = \text{if } x=1 \text{ then } 1 \text{ else } x \times f(x-1).$$

Assume for definiteness that evaluation of f is call-by-name and that $x < y$ is 0 if $x < y$. If f is called for argument an expression t , then the expression is passed to the defining body of f . The test ("if $x=1$ ") attempts to evaluate t . If and only if this terminates

the appropriate branch of the conditional is selected. In general this will lead to f being called again and if t evaluates to 0 to f being called an infinite number of times. Define semantic versions of conditional, test, multiplication and subtraction by:

$$\begin{aligned} \text{cond: } \mathbb{N} \times \mathbb{N}^2 &\rightarrow \mathbb{N} \\ \text{cond}(\perp, n, m) &= \perp \\ \text{cond}(tt, n, m) &= n, \text{cond}(ff, n, m) = m \\ \text{eq: } \mathbb{N}^2 &\rightarrow \mathbb{N} \\ \text{eq}(n, m) &= \perp \text{ if } n = \perp \text{ or } m = \perp \\ &= tt \text{ if } n, m \neq \perp \ \& \ n = m \\ &= ff \text{ otherwise} \\ \text{p: } \mathbb{N}^2 &\rightarrow \mathbb{N} \\ \text{p}(n, m) &= \perp \text{ if } n = \perp \text{ or } m = \perp \\ &= n \times m \text{ otherwise.} \end{aligned}$$

Subtraction s is similar.

Then the recursive definition determines a continuous function $\Gamma: [\mathbb{N} \rightarrow \mathbb{N}] \rightarrow [\mathbb{N} \rightarrow \mathbb{N}]$. $\Gamma(f) = \lambda x. \text{cond}(\text{eq}(x, 1), 1, \text{p}(x, f(s(x, 1))))$. Each iterate $\Gamma^n(\perp)$ agrees with the factorial function on $1, 2, \dots, n$ in \mathbb{N} and is \perp elsewhere. Roughly an iterate gives the information about f which may be got in a certain finite time. The procedure f is denoted by the least fixed point $\Upsilon(\Gamma)$ in $[\mathbb{N} \rightarrow \mathbb{N}]$ which is all the information which may be got ever.

Algebraic domains are those domains of chief importance in denotational semantics at the moment. They are determined by their isolated elements which form a basis.

Definition 3.1.20

Let D be a cpo. Say x in D is isolated iff for all directed sets S in D

$$x \sqsubseteq \bigcup S \Rightarrow \exists s \in S \ x \sqsubseteq s.$$

Denote the set of isolated elements by D^0 .

Definition 3.1.21

Let D be a cpo. Then D is algebraic if for all x in D we have $\{y \in D^0 \mid y \sqsubseteq x\}$ is directed and $x = \bigcup \{y \in D^0 \mid y \sqsubseteq x\}$. D is ω -algebraic iff it is algebraic and D^0 is countable.

Lemma 3.1.22

Let D be a cpo. Then $\perp \in D^0$. Suppose $x, y \in D^0$. Then if $x \sqcup y$ exists $x \sqcup y \in D^0$.

Proof

We have $\perp \in D^0$ as directed sets are non-null. Suppose $x, y \in D^0$ with $x \sqcup y$ in D . Let S be a directed set with $x \sqcup y \sqsubseteq \bigsqcup S$. Then $x \sqsubseteq s$ and $y \sqsubseteq t$ for some s and t in S . Thus $x \sqcup y \sqsubseteq u$ for some u in S by the definition of directed. Thus $x \sqcup y \in D^0$. ■

Proposition 3.1.23

Let (D, \sqsubseteq) be an algebraic cpo. Define $\mathcal{S}(D^0)$ to consist of \sqsubseteq -left closed directed subsets of D^0 ordered by inclusion. ($S \subseteq D^0$ is \sqsubseteq -left-closed iff $\forall x, y \in D^0 \ x \sqsubseteq y \in S \Rightarrow x \in S$). Then $D \cong \mathcal{S}(D^0)$ under the map $x \mapsto \{y \in D^0 \mid y \sqsubseteq x\}$. Thus D is determined by (D^0, \sqsubseteq) to within isomorphism.

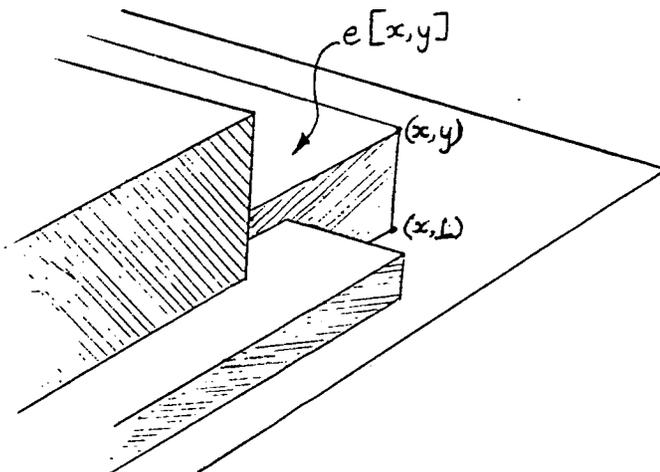
Provided domains are consistently complete algebraicity is preserved by the function space and product constructions. The isolated elements of the function space are step-functions.

Definition 3.1.24

Let (D_0, \sqsubseteq_0) , (D_1, \sqsubseteq_1) be algebraic cpos. Define the function $e[x, y]$ for $x \in D_0^0$ and $y \in D_1^0$ by $e[x, y](z) = y$ if $x \sqsubseteq_0 z$
 $= \perp$ otherwise.

A step-function in $[D_0 \rightarrow D_1]$ is a function of the form $e[x_0, y_0] \sqcup \dots \sqcup e[x_n, y_n]$ for x_i in D_0^0 and y_i in D_1^0 .

Step functions can be drawn to look like steps. The vertical direction represents increasing information in the range D_1 and in the horizontal direction (right to left) increasing information in the domain D_0 .



Proposition 3.1.25

Suppose (D_0, \sqsubseteq_0) and (D_1, \sqsubseteq_1) are consistently complete $(\omega-)$ algebraic cpos. Then

- (1) $D_0 \times D_1$ is consistently complete and $(\omega-)$ algebraic;
 $(D_0 \times D_1)^0 = D_0^0 \times D_1^0$.
- (2) $[D_0 \rightarrow D_1]$ is consistently complete and $(\omega-)$ algebraic,
 $[D_0 \rightarrow D_1]^0$ is precisely the set of step functions.

The domains \mathbb{T} , \mathbb{N} and $[\mathbb{N} \rightarrow \mathbb{N}]$ are ω -algebraic and consistently complete. We have

$$\mathbb{T}^0 = \mathbb{T}$$

$$\mathbb{N}^0 = \mathbb{N}$$

$$[\mathbb{N} \rightarrow \mathbb{N}]^0 = \{ f \in [\mathbb{N} \rightarrow \mathbb{N}] \mid f(\perp) \in \omega \text{ or } f^{-1}\omega \text{ is finite} \}.$$

Intuitively an isolated element of an algebraic domain corresponds to the information a computing agent may extract or produce in finite time through performing a finite number of actions.

The following types of function are of particular importance. We shall use them later.

Definition 3.1.26

Let D_0 and D_1 be cpos. Suppose $\psi \in [D_0 \rightarrow D_1]$. Then ψ is strict iff $\psi(\perp) = \perp$.

ψ is a projection iff $\exists \phi \in [D_1 \rightarrow D_0]$
 $\psi \circ \phi = 1_{D_0}$ & $\phi \circ \psi \sqsubseteq 1_{D_1}$

(then ϕ is called an embedding).

Embedding-projection pairs are used in solving recursive domain equations. Roughly they give the relation of one domain "approximating" another. Strict functions are necessary to give semantics for call-by-value evaluation.

We shall often be concerned with distributive domains.

Definition 3.1.27

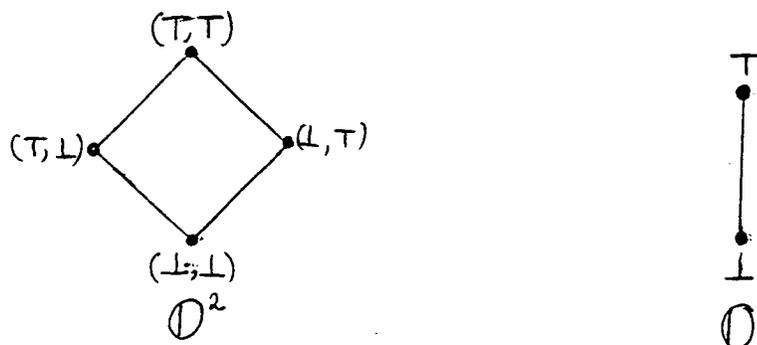
Let D be a consistently complete cpo. Then D is distributive iff

$$y \uparrow z \Rightarrow x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z).$$

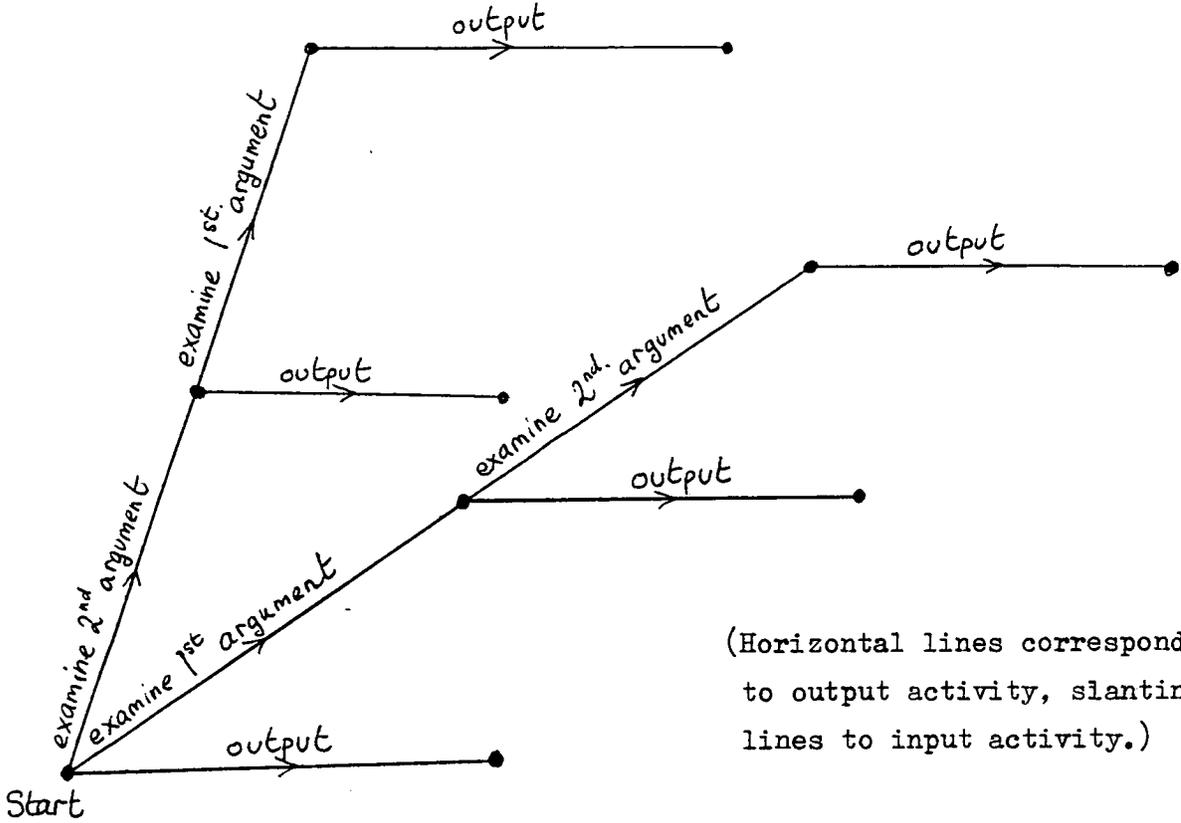
3.2 Concrete domains, matrices and sequential functions

Continuity is a general restriction on functions between domains which have a chance of being computable. It is natural to ask for a general restriction on functions which have a chance of being computable in "a deterministic way", that are in this intuitive sense sequential. (Note all the functions are determinate; they can only yield one value for one argument. We are concerned with whether or not such functions can be realised by a deterministic computation.) Some care is needed with the idea of deterministic. For example we would not allow the computation to depend on information about time not present in the domains; if this were allowed we could simulate parallel evaluation of the arguments. We wish any current (single) activity of the computation (its "flow of control") to be determined solely by information in the domains. (The algorithms of Pierre-Louis Curien ([Cur], [Ber and Cur]) provide one way of formalising this idea.)

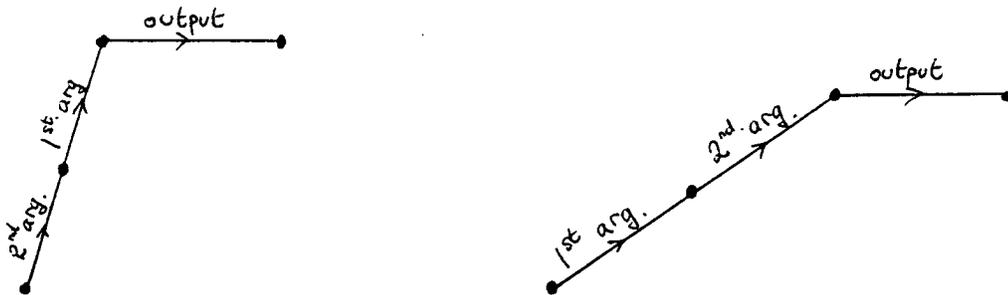
Example 3.2.1



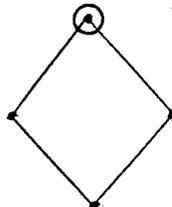
Regard the functions in $[D^2 \rightarrow D]$ as being on two arguments (x,y) in D^2 . A deterministic computation from input D^2 to output D should proceed according to the following general scheme (borrowing ideas from [Cur]).



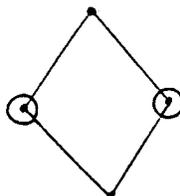
A deterministic computation will determine any partial branch beginning at start. Thus initially at its start the computation either examines a particular argument or ignores the arguments and perhaps, but not necessarily, outputs. Any completely slanting branch (including the single node "start") realises the function \perp in $[0^2 \rightarrow 0]$. The two maximal branches



both correspond to the least monotonic function giving $(T, T) \mapsto T$, which we can draw on 0^2 as:



Consider the least monotonic function giving $(T, \perp) \mapsto T$ and $(\perp, T) \mapsto T$ drawn on 0^2 as:



This cannot be realised according to the scheme above; it examines its two arguments in parallel. It should not be a sequential function.

We seek a definition of sequential function between domains based solely on the structure of the domains themselves. Two early definitions of sequential function were proposed independently by Robin Milner and Jean Vuillemin. These depend on viewing a function $f: \prod_{i < n} D_i \rightarrow E$ as being of n arguments (viewed as being more or less arguments may change its character according to these definitions!)

Definition 3.2.2

Let D_0, \dots, D_{n-1}, E be cpos. Let f be a continuous function: $\prod_{i < n} D_i \rightarrow E$. Then f is M-sequential (Milner) iff either it is constant or there is an integer i (with $0 \leq i < n$) such that f is strict in its i^{th} argument ($(x)_i = \perp \Rightarrow f(x) = \perp$) and the function obtained by fixing its i^{th} argument ($\lambda x_0, \dots, x_{i-1}, x_{i+1}, \dots, x_{n-1}. f(x_0, \dots, x_{i-1}, x_i, x_{i+1}, \dots, x_{n-1})$) is M-sequential.

Also f is V-sequential (Vuillemin) iff it is a constant or there is an integer i (with $0 \leq i < n$) such that $y \sqsupseteq x$ and $(y)_i = (x)_i$ implies $f(y) = f(x)$.

The two above definitions of sequential do not agree in general. However importantly they do coincide and appear correct in the situation where D_0, \dots, D_{n-1} and E are flat cpos. Note their dependence on argument places.

Gilles Kahn and Gordon Plotkin sought a very general definition of sequential function which unlike M and V-sequentiality was independent of the way that the function was viewed as having arguments. Reasonably, the definition should agree with M and V-sequentiality in the case where the domain and codomain were of the form $\prod_{i < n} D_i$ and E respectively for flat domains D_i and E . They achieved this by axiomatising a wide class of domains for which there was a natural definition of places accessible from a point. Places are a generalisation of argument-places which can take values from a flat cpo. Unlike argument places, however, places are defined independently of the way the domain is viewed as a product. Their definition of sequential then agrees locally with M or V-sequentiality. Recognising that the notion of sequential depended

on the nature of the objects denoted in the domains they chose to axiomatise only those domains corresponding to basic input or output values. Certainly integers, truth values, tapes and trees are basic and almost physical (their names often suggest it too!) whereas functions are not. In a computation a function must be represented for instance by the text of a procedure whereas basic values present themselves directly and concretely. Concrete domains are domains representing basic values and supporting Kahn and Plotkin's definition of sequential function. There are domains of basic values which are not concrete (any confused Petri net provides an example - see chapter 5).

Kahn and Plotkin first axiomatised the concrete domains and then discovered they could be represented by matrices (rather like Petri nets). Our presentation is the other way round. A matrix consists of places which can be occupied by at most one of a set of decisions or events. In general a place may not be occupied immediately but must wait until this is enabled by certain events. A place may be thus enabled by several different sets of events. (As an example the n th place of a list is enabled by the event of making the $(n-1)$ th entry.) We now give the formal definition of a matrix M and its configurations ordered by inclusion $\sqsupseteq(M)$. Note \perp in $\sqsupseteq(M)$ corresponds to nothing has happened.

Definition 3.2.3

A matrix M is a quadruple (P, E, l, \vdash) where:

1. P is a set of places
2. E is a countable set of events
3. l is a function from E onto P locating events at places.
4. \vdash is a subset of $\mathcal{F}(E) \times P$ called the enabling relation.

($\mathcal{F}(E)$ denotes the finite subsets of E .)

We say M is strongly-deterministic iff $A \vdash p \ \& \ A' \vdash p \Rightarrow A = A'$.

Let X be a subset of E .

Say X is consistent iff $\forall e, e' \in X \ l(e) = l(e') \Rightarrow e = e'$.

Suppose $e \in X$. Say e is secured in X iff $\exists e_0, \dots, e_n \in X$
 $e_n = e \ \& \ \forall i \leq n \ \exists A \subseteq \{e_0, \dots, e_{i-1}\} \ A \vdash l(e_i)$.

Say X is secured iff all elements of X are secured in X .

Say X is a configuration of M iff X is consistent and secured.

Denote the set of configurations ordered by inclusion by $\Gamma(M)$.
 Say M generates $\Gamma(M)$.

For a matrix M the partial ordering $\Gamma(M)$ will be an ω -algebraic domain satisfying certain axioms F,C,R and Q which determine the concrete domains. Conversely a concrete domain will be generated to within isomorphism by a matrix. (The representation theorem for concrete domains.)

The following definitions are important in defining sequential functions.

Definition 3.2.4

Let M be a matrix. Suppose $x \in \Gamma(M)$ and p is a place of M. Say x fills p iff $\exists e \in x \text{ l}(e) = p$.

Say p is accessible at x iff x does not fill p and $\exists e_0, \dots, e_n \in x$
 $\exists B \subseteq \{e_0, \dots, e_n\}. B \vdash p$ &

$\forall i \leq n \exists A \subseteq \{e_0, \dots, e_{i-1}\}. A \vdash \text{l}(e_i)$.

Write $p(x)$ for the set of places accessible at x.

For x, y in $\Gamma(M)$ write $x \underset{p}{\prec} y$ iff $x \subseteq y$ and p is accessible at x and y fills p.

Thus we can tentatively define a function $f: \Gamma(M) \rightarrow \Gamma(M')$ to be sequential if it is sequential at all x in $\Gamma(M)$ where this means

$\forall p' \in p(f(x)). (\exists z \supseteq x, f(x) \underset{p'}{\prec} f(z)) \Rightarrow \exists p \in p(x) \forall y \supseteq x, (f(x) \underset{p'}{\prec} f(y) \Rightarrow x \underset{p}{\prec} y)$

This says to fill p' accessible from $f(x)$ there is some p accessible from x which must be filled; it generalises V-sequential. Of course, it is not yet clear that this definition gives the same notion of sequential for different ways of generating isomorphic domains. This will fall out of the representation theorem. We give the main ideas in this section and the detailed proof in the next.

We give some examples of matrices (and thus concrete domains). The first example illustrates a convenient way of drawing matrices.

Example 3.2.5

Let M be the matrix given by:

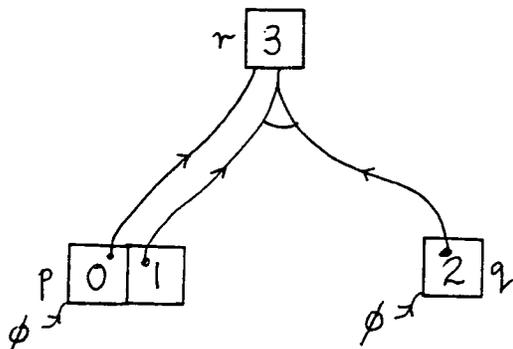
$$P = \{p, q, r\}$$

$$E = \{0, 1, 2, 3\}$$

$$\text{l}(0) = \text{l}(1) = p, \text{l}(2) = q, \text{l}(3) = r.$$

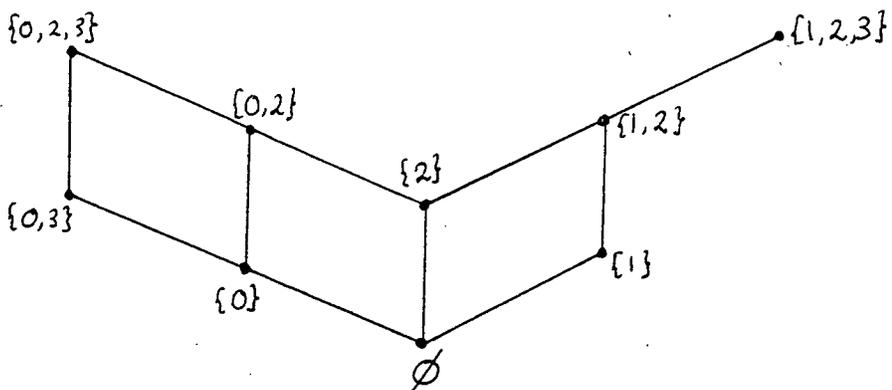
$\{0\} \vdash r, \{1,2\} \vdash r, \emptyset \vdash p, \emptyset \vdash q.$

We draw this as

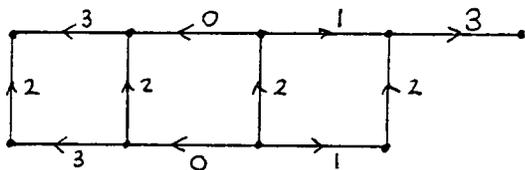


Boxes represent places, their contents the events which are located there, "fused" arrows  the enabling relation.

$\Gamma(M)$ has the form:



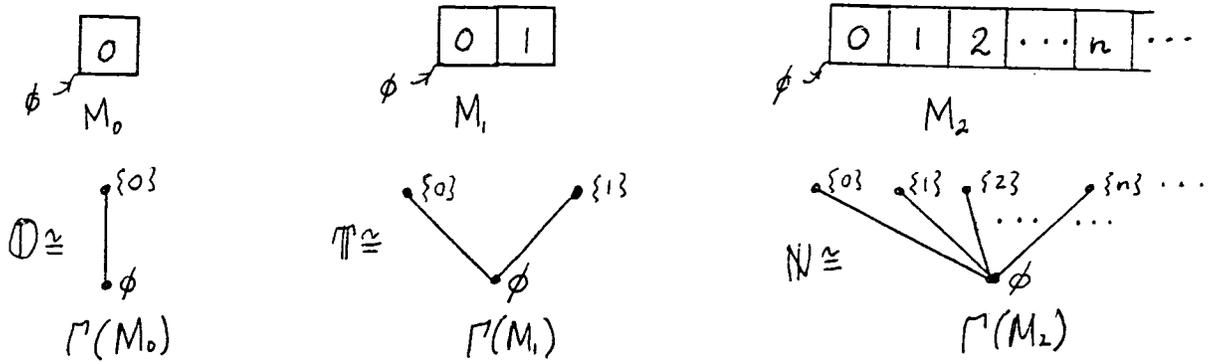
Represented by an aerial view labelling arcs by the additional events this is :



This is often a more convenient form.

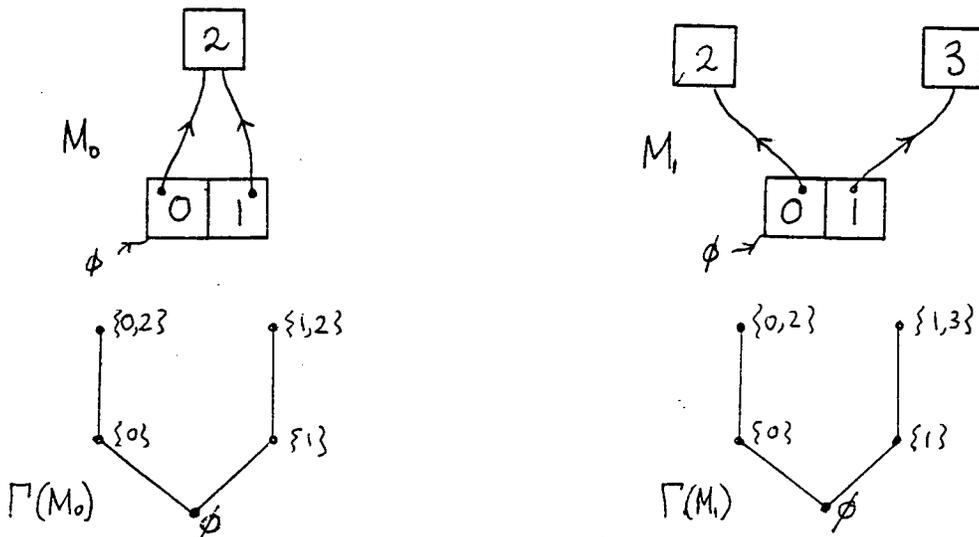
Flat domains are easily generated.

Example 3.2.6



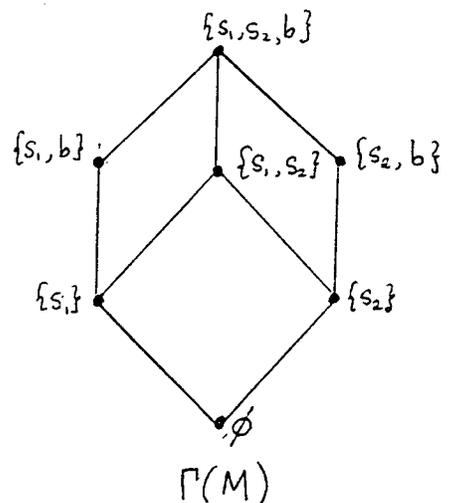
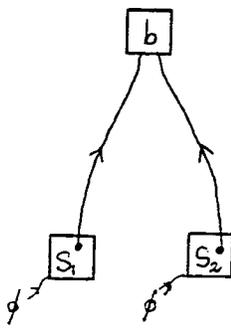
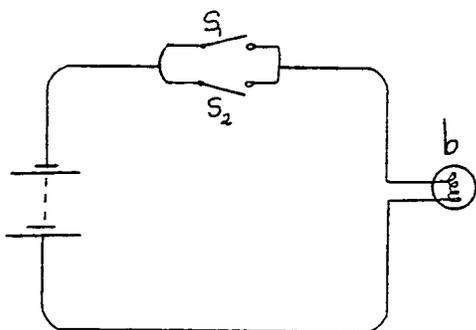
Sometimes two domains are isomorphic even though one is generated by a strongly-deterministic matrix and the other is not as here:

Example 3.2.7



Some matrices which are not strongly-deterministic represent physical things.

Example 3.2.8

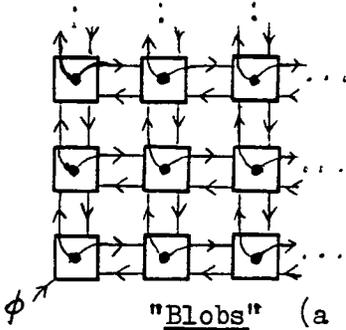


M

Γ(M)

The bulb b is turned on by either of the switches s_1 or s_2 which are not mutually exclusive. $\Gamma(M)$ is not generated by any strongly-deterministic matrix.

Example 3.2.9



Every place has one event.

A place is enabled by any adjacent event.

"Blobs" (a discrete approximation to the quarterplane)

A matrix is physically realisable in this sense: Interpret each place as a computer capable of not terminating or outputting a set in 1-1 correspondence with the events located at the place. Assume all computers are switched-off initially but are switched on according to the enabling relation.

From the definition of a matrix M and its configurations $\Gamma(M)$ the following properties are easily established.

Proposition 3.2.10

Let M be a matrix (P, E, l, \vdash) . Then:

1. Two configurations x, x' in $\Gamma(M)$ are compatible iff $\forall e \in x, e' \in x' \ l(e) = l(e') \Rightarrow e = e'$. If x and x' are compatible their supremum in $\Gamma(M)$ is $x \cup x'$.
2. The poset $\Gamma(M)$ is coherent. If X a subset of $\Gamma(M)$ is pairwise compatible then $\bigcup X$ is the supremum of X in $\Gamma(M)$.
3. The poset $\Gamma(M)$ forms an ω -algebraic domain. Its minimum element is \emptyset (so $\perp = \emptyset$). The isolated elements of $\Gamma(M)$ are precisely the finite configurations. An isolated element dominates only finitely many elements in $\Gamma(M)$.

Proof

1. and 2. follow obviously.

3. Clearly \emptyset is the minimum element of $\Gamma(M)$. From 2. $\Gamma(M)$ is a cpo. It is obvious that finite configurations are isolated in $\Gamma(M)$. To show the converse suppose X is isolated in $\Gamma(M)$. For each e in X choose $A_e = \{e_0, \dots, e_n\} \subseteq X$ so that

$e_n = e \ \& \ \emptyset \vdash 1(e_0) \ \& \ \forall i \leq n \exists B \subseteq \{e_0, \dots, e_{i-1}\} \ B \vdash 1(e_i)$ - clearly possible as X is secured. Take S to be the directed set consisting of all configurations $A_{e_1} \cup \dots \cup A_{e_n}$ for e_1, \dots, e_n in X . Then $X = \bigcup S$ so for some e_1, \dots, e_m in X , $X = A_{e_1} \cup \dots \cup A_{e_m}$. As each A_{e_i} is finite X is finite. As every configuration X is secured we have $X = \bigsqcup \{x \in \Gamma(M)^0 \mid x \subseteq X\}$. Thus $\Gamma(M)$ is algebraic. As E is countable $\Gamma(M)$ is ω -algebraic. As an isolated element is finite it can only dominate a finite number of elements. ■

Kahn and Plotkin [Kah and Plo] showed that a cpo is generated by some matrix iff it is ω -algebraic and satisfies four axioms F, C, R and Q. We now introduce the axioms and illustrate why they hold for domains of configurations.

Definition 3.2.11 (Axiom F)

Let D be an algebraic domain. Then D satisfies axiom F iff $\forall x \in D^0 \ |\{y \in D \mid y \subseteq x\}| < \infty$.

Of course we have already proved this for configurations in proposition 3.2.10 part 3.

Events of a matrix M show themselves in the domain $\Gamma(M)$ as coverings.

Definition 3.2.12

Let (D, \subseteq) be a partial order. Suppose $x, x' \in D$. Then x' is said to cover x , written $x \prec x'$, iff $x \subseteq x'$ & $x \neq x'$ &

$$\forall z \in D \ x \subseteq z \subseteq x' \Rightarrow (z = x \text{ or } z = x').$$

Let $x, y \in D$. Then a covering chain from x to y is a sequence $x = x_0, x_1, \dots, x_n = y$ where $x_i \prec x_{i+1}$ for $i < n$.

The next lemma follows easily.

Lemma 3.2.13

Let D be an algebraic domain which satisfies axiom F. Suppose $x \in D$ and $y \in D^0$ and $x \subseteq y$. Then $x \in D^0$ and there is a covering chain from x to y .

It is easy to characterise \prec in domains $\Gamma(M)$ for a matrix M .

Lemma 3.2.14

Let M be a matrix. For x, y in $\Gamma(M)$, $x \prec y$ iff

$\exists e \in E \ e \notin x \ \& \ y = x \cup \{e\}$.

Hence a covering in $\Gamma(M)$ corresponds to an occurrence of an event at a configuration. Also note that any covering (an occurrence of an event at a configuration) is reflected by a covering in $\Gamma(M)^0$:

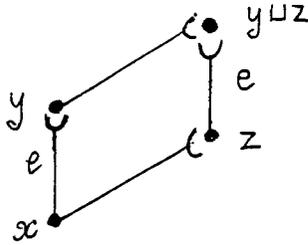
Lemma 3.2.15

Let M be a matrix. For x, y in $\Gamma(M)$, $x \prec y \Rightarrow \exists x', y' \in \Gamma(M)^0$
 $x', y' \sqsubseteq y \ \& \ x' \prec y' \ \& \ y' \setminus x' = y \setminus x$.

Proof

Take e as the unique element of $y \setminus x$ and use the ideas of proposition 3.2.10 (3). ■

Thus an event e of a matrix M manifests itself in $\Gamma(M)$, if at all, as a covering $x \prec y$ where $y \setminus x = \{e\}$ and x may be assumed isolated. Of course the same event may occur at some other configuration. For example we may have $x \prec y$, $x \prec z$, $y \uparrow z$ and $y \neq z$. This means $y = x \cup \{e\}$, $z = x \cup \{e'\}$ for two events e and e' such that $l(e) \neq l(e')$. Clearly $y \sqcup z$ exists and is $x \cup \{e, e'\}$ so $y \sqcup z \setminus z = \{e\}$.



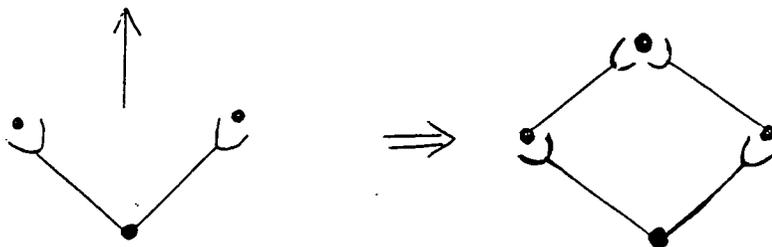
The covering $z \prec y \sqcup z$ represents the same event e as the covering $x \prec y$. (Also the coverings $x \prec z$ and $y \prec y \sqcup z$ represent the same event e' .) This suggests we can recover events from domains by a relation based on "little squares" like that above. Axiom C ensures there are enough "little squares".

Definition 3.2.16 (Axiom C)

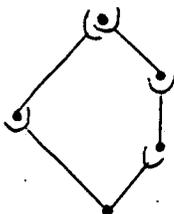
Let D be an algebraic domain. Then D satisfies axiom C iff for all x, y, z in D^0 $x \prec y \ \& \ x \prec z \ \& \ y \uparrow z \ \& \ y \neq z$ implies $y \sqcup z$ exists and $y \prec y \sqcup z \ \& \ z \prec y \sqcup z$.

We have seen above that $\Gamma(M)$ satisfies axiom C. It expresses a form of orthogonality between compatible coverings of an element.

In a picture it says



Axiom C typically forbids



(In fact in the presence of axiom F it gives upper semimodularity [Bir] which ensures all covering chains between comparable isolated elements have the same length. See lemma 3.3.4 in our proof of the representation theorem for this and a lot more.)

We now formalise how events are to be recovered from a domain.

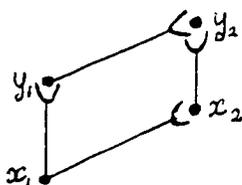
Definition 3.2.17

Let D be an algebraic domain satisfying F and C . A prime interval of D is a pair $[x, y]$ where $x \prec y$. If $[x_1, y_1]$ and $[x_2, y_2]$ are prime intervals with x_i, y_i in D^0 write

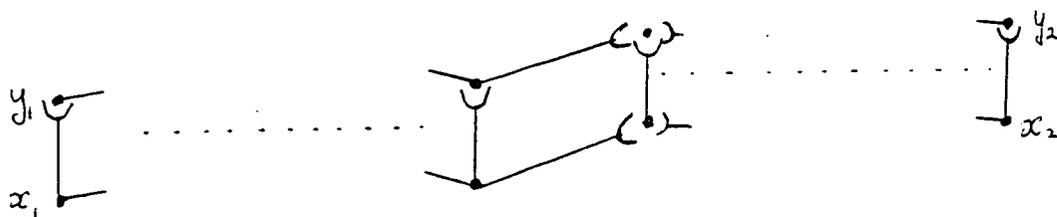
$$[x_1, y_1] \leq' [x_2, y_2] \text{ iff } x_1 \prec x_2 \text{ and } y_1 \prec y_2.$$

Define \sim to be the reflexive symmetric transitive closure of \leq' .

A prime interval is no more than a pair of elements in the covering relation. The relation $[x_1, y_1] \leq' [x_2, y_2]$ looks like



and the relation $[x_1, y_1] \sim [x_2, y_2]$ like



In a domain $\Gamma(M)$ a prime interval has the form $[x, x \cup \{e\}]$. When x and y are isolated it is easy to see that $[x, x \cup \{e\}] \sim [y, y \cup \{e'\}]$ implies $e = e'$ so that a \sim -equivalence class represents an occurrence of the same event at different isolated configurations. (It may not be all occurrences of this event because of examples like M_0 in 3.2.7.)

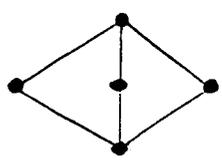
We extract events from domains by taking \sim -equivalence classes. For this to be done safely we must guarantee that an "event" has at most one occurrence at any isolated configuration; that is a \sim -equivalence class has at most one member $[x, y]$ for any fixed isolated x . This property is clearly true of $\Gamma(M)$. It is expressed by axiom R.

Definition 3.2.18 (Axiom R)

Let D be algebraic and satisfy F and C. Then D satisfies axiom R iff for x in D^0 and all prime intervals $[x, y], [x, z]$

$$[x, y] \sim [x, z] \Rightarrow y = z$$

Axiom R forbids domains like the following in which all prime intervals belong to the same \sim -equivalence class:



In a similar way we can extract places from domains. For this, notice if we consider a configuration x in $\Gamma(M)^0$ and two events e and e' such that $x \cup \{e\}$ and $x \cup \{e'\}$ are configurations we have $l(e) = l(e')$ iff either $x \cup \{e\} = x \cup \{e'\}$ or $x \cup \{e\} \uparrow x \cup \{e'\}$ in $\Gamma(M)$. This suggests the following definition:

Definition 3.2.19

Let D be an algebraic domain satisfying F and C. Let $[x, x_1]$ and $[x, x_2]$ be two prime intervals of D with x in D^0 . Define \approx' by $[x, x_1] \approx' [x, x_2]$ iff $x_1 = x_2$ or $x_1 \uparrow x_2$. Define \approx to be the symmetric transitive closure of $(\approx' \cup \sim)$. An equivalence class of \approx is called a direction of D .

Directions are to be the domain analogue of places. For this the further axiom Q is required.

Definition 3.2.20 (Axiom Q)

Let D be an algebraic domain. Then D satisfies axiom Q iff for all x, y, z in D^0

$$y \sqsupseteq x \subset z \ \& \ y \not\uparrow z \Rightarrow \exists ! t \sqsubseteq y \ x \subset t \uparrow z.$$

Axiom Q has two parts, an existence part (got by ignoring uniqueness) and a uniqueness part. These typically forbid these respective domains:



We look at Q in a domain $\sqsupseteq(M)$. Suppose $y \sqsupseteq x \subset z$ and $y \not\uparrow z$ in $\sqsupseteq(M)^0$. Then $z = x \cup \{e\}$ for some event e . As $y \not\uparrow z$ there is an event e' in y so that $l(e') = l(e)$ and $e' \neq e$. Then taking $t = x \cup \{e'\}$ shows the existence part of Q is satisfied. Suppose there were another $t' \sqsubseteq y$ so $x \subset t' \uparrow z$. Then $t' = x \cup \{e''\}$ with $l(e'') = l(e)$ and $e'' \neq e$. Then for events e', e'' in y we have $l(e') = l(e'')$. This must mean $e' = e''$, establishing uniqueness.

We can now define concrete domains and state Kahn and Plotkin's representation theorem.

Definition 3.2.21

A concrete domain is an ω -algebraic domain satisfying axioms F, C, R and Q.

Theorem 3.2.22

Any (strongly deterministic) matrix generates a (distributive) concrete domain. For any (distributive) concrete domain D there is a (strongly deterministic) matrix M such that $\sqsupseteq(M) \cong D$.

Basic construction:

We present a complete proof in section 3.3. Here we give the basic construction of a matrix from a concrete domain. Let D be a concrete domain. Define a matrix M in the following way:

P is the set of directions of M ($\{[x, x']_{\sim} \mid x \in D^0 \& x \prec x'\}$)

E is the set of \sim -equivalence classes ($\{[x, x']_{\sim} \mid x \in D^0 \& x \prec x'\}$)

l is the map $[x, x']_{\sim} \mapsto [x, x']_{\sim}$

$A \vdash p$ iff there are $[x, x'] \in p$ and a covering chain

$$\perp = x_0 \prec \dots \prec x_n = x \text{ and } A = \{[x, x_{i+1}]_{\sim} \mid 0 \leq i < n\}$$

(We show in § 3.3 that A is independent of the choice of covering chain.)

We show in 3.3 that $\Gamma(M) \cong D$ and that if D is distributive then

$A \vdash p \& A' \vdash p \Rightarrow A \cap A' \vdash p$. Thus then we may define an enabling

relation \vdash^* by taking $\bigcap \{A \mid A \vdash p\} \vdash^* p$. This gives a strongly-

deterministic matrix $M^* = (P, E, l, \vdash^*)$ s.t. $\Gamma(M^*) \cong D$.

Using the representation theorem it is easy to show that

concrete domains are closed under products. It is a consequence of the following observation.

Proposition 3.2.23

Let D_0 and D_1 be concrete domains. Then there are matrices,

$M_i = (P_i, E_i, l_i, \vdash_i)$ for $i=0,1$ with $P_0 \cap P_1 = E_0 \cap E_1 = \emptyset$ such that

$\Theta_i: \Gamma(M_i) \cong D_i$ for $i=0,1$. Define $M_0 \oplus M_1 = (P_0 \cup P_1,$

$E_0 \cup E_1, l_0 \cup l_1, \vdash_0 \cup \vdash_1)$. Then $M_0 \oplus M_1$ is a matrix with

$\Gamma(M_0 \oplus M_1) \cong D_0 \times D_1$ under $x \mapsto (\Theta_0(x \cap E_0), \Theta_1(x \cap E_1))$.

Similarly concrete domains may be shown closed under ω -products.

Early on in this section we indicated how sequentiality was to be defined. It was unclear whether or not the notion of sequential depended on the matrices generating the domains. We can follow the same idea on the canonical matrix produced by the representation theorem.

Definition 3.2.24

Let D be a concrete domain. Let d be a direction of D .

Suppose $x \in D$. Say x fills d iff $\exists [x_0, y_0] \in d \ y_0 \sqsubseteq x$. Say

d is accessible at x iff $\exists x_0, y_0 \in D^0 \ x_0 \sqsubseteq x \& [x_0, y_0] \in d \&$

$y_0 \uparrow x \& y_0 \not\sqsubseteq x$.

Write $d(x)$ for the set of directions accessible at x . For x, y in

D , write $x \prec_d y$ iff $x \sqsubseteq y$ and d is accessible at x and y fills d .

Fortunately a function being sequential with respect to the

definitions above is equivalent to it being sequential with respect to any other matrix generating an isomorphic domain. This is because of the following proposition.

Proposition 3.2.25

Let M be a matrix. Suppose $x \in \Gamma(M)$. Define $i_x: p(x) \rightarrow d(x)$ by $i_x(p) = [x, x \cup \{e\}]_{\sim}$ where e is any event s.t. $1(e) = p$. Then i_x is 1-1 and onto and is natural in the sense that if $x \sqsubseteq y$ and if $p \in p(x) \cap p(y)$ then $i_x(p) = i_y(p)$.

Definition 3.2.26

Let D, D' be concrete domains. Suppose $f \in [D \rightarrow D']$. Then f is sequential at x iff $\forall d' \in d(f(x)) (\exists z \sqsupseteq x \ f(x) \prec_{d'} f(z)) \Rightarrow \exists d \in d(x) \ \forall y \sqsupseteq x (f(x) \prec_{d'} f(y) \Rightarrow x \prec_d y)$.
Say f is sequential iff it is sequential at all x in D .

Such sequential functions in fact form a cpo (not generally concrete) when ordered pointwise. By virtue of proposition we have reassuringly that:

Proposition 3.2.27

Let M, M' be matrices. Suppose $x \in \Gamma(M)$ and $f \in [\Gamma(M) \rightarrow \Gamma(M')]$. Then f is sequential at x iff $\forall p' \in p(f(x)) (\exists z \sqsupseteq x \ f(x) \prec_{p'} f(z)) \Rightarrow \exists p \in p(x) \ \forall y \sqsupseteq x (f(x) \prec_{p'} f(y) \Rightarrow x \prec_p y)$.

Finally from the work of Curien and Berry ([Cur], [Ber and Cur]) the sequential functions between concrete domains are characterised as those functions which may be realized by a deterministic algorithm.

3.3 The representation theorem

Here we give a proof of the representation theorem for concrete domains. It improves the one in [Kah and Plo], mainly because of the early lemmas and because it also gives a more general result. At first we work with a new axiom, axiom V, which is weaker than axiom Q. We first prove a representation result between ω -algebraic domains satisfying F, C, R and V and event structures of the form (E, \vdash, \otimes) now defined.

Definition 3.3.1

An event structure consists of a triple (E, \vdash, \otimes) where E is a countable set of events E , $\vdash \subseteq \mathcal{P}(E) \times E$ is the enabling relation

and \times is a binary relation on E called the conflict relation.

Say E is strongly-deterministic ([Ber and Cur]) iff

$$A \vdash e \ \& \ A' \vdash e \Rightarrow A = A'.$$

Let X be a subset of E . Then X is consistent iff

$\forall e, e' \in X \neg (e \times e')$. Assume $e \in X$. Say e is secured in X iff

$$\exists e_0, \dots, e_n \in X \ e_n = e \ \& \ \forall i \leq n \ \exists A \subseteq \{e_0, \dots, e_{i-1}\} \ A \vdash e_i.$$

Then say X is secured iff all its elements are secured in X .

Define a configuration of E to be a consistent secured subset of E .

Let $\Gamma(E)$ denote the set of configurations ordered by inclusion.

Say E generates $\Gamma(E)$.

Clearly a matrix $M = (P, E, l, \vdash)$ produces such an event structure (E, \vdash, \times) by defining $e \times e'$ iff $l(e) = l(e')$ and $A \vdash e$ iff $A \vdash l(e)$. The structure $\Gamma(E)$ for an event structure E will be ω -algebraic and satisfy the axioms F, C, R and V. Here is the new axiom V:

Definition 3.3.2 (Axiom V)

Let D be an algebraic domain satisfying F and C. Then D satisfies axiom V iff for all x, x', y, y', x'', y'' in D^0

$$[x, x'] \sim [y, y'] \ \& \ [x, x''] \sim [y, y''] \ \& \ x' \uparrow x'' \Rightarrow y' \uparrow y''.$$

For the domain of configurations it expresses that the conflict of two events is independent of what other events have occurred. We outline a proof that the configurations of an event structure satisfy the axioms. In addition note that strongly-deterministic event structures generate distributive domains - we include a converse to this in the representation theorem.

Theorem 3.3.3

Let (E, \vdash, \times) be an event structure. Then $\Gamma(E)$ is an ω -algebraic domain satisfying F, C, R and V. If E is strongly-deterministic $\Gamma(E)$ is distributive.

Proof

Let (E, \vdash, \times) be such an event structure. First it is easily seen that for S a directed subset of $\Gamma(E)$ the supremum of S exists and is $\bigcup S$. Thus $\Gamma(E)$ is a cpo. As in proposition 3.2.10 the isolated elements of $\Gamma(E)$ can be characterised as precisely the finite configurations (the proof is virtually identical).

As every event is secured by some finite subset inside a configuration and E is countable we get $\prod(E)$ is ω -algebraic. The other axioms are easily shown because $X \prec X'$ for configurations X and X' means $X' = X \cup \{e\}$ for some e in E . To show axiom V for example: Suppose $[x, x'] \sim [y, y']$ & $[x, x''] \sim [y, y'']$ & $x' \uparrow x''$ in $\prod(E)^0$. Then $x' \setminus x = y' \setminus y = \{e\}$ say, and $x'' \setminus x = y'' \setminus y = \{e'\}$ say. As $x' \uparrow x''$ we have $\neg(e \times e')$. Thus $y' \cup y''$ is a configuration giving $y' \uparrow y''$ as required.

Now assume E is strongly-deterministic. Clearly now $\sqcap = \cap$ so the distributivity property $y \uparrow z \Rightarrow x \sqcap (y \sqcup z) = (x \sqcap y) \sqcup (x \sqcap z)$ obviously holds for $\prod(E)$. ■

We remark that algebraicity can fail when the enabling relation is allowed to range over arbitrary subsets of events.

We now begin a proof of the converse, that if D is an ω -algebraic domain satisfying axioms F, C, R and V then D is isomorphic to the configurations of some event structure. We initially work with ω -algebraic domains satisfying axioms F and C and impose R and V only when needed. Throughout we let D denote an ω -algebraic domain satisfying axioms F and C. Note because of axiom F there is always a (finite) covering chain of isolated elements between comparable isolated elements of D . We work almost solely with the isolated elements of D viz. D^0 . The first lemma extends the Jordan-Holder theorem a little bit [Bir].

Lemma 3.3.4

Suppose $y' \in D^0$ the isolated elements of D . If $y = x_0 \prec x_1 \prec \dots \prec x_n = y'$ and $y = z_0 \prec z_1 \prec \dots \prec z_m = y'$ are two covering chains from y to y' then $\{[x_i, x_{i+1}] \mid 0 \leq i < n\} \sim \{[z_i, z_{i+1}] \mid 0 \leq i < m\}$. Moreover the number of representatives of each \sim -equivalence class is the same in both chains i.e. for a \sim -equivalence class e

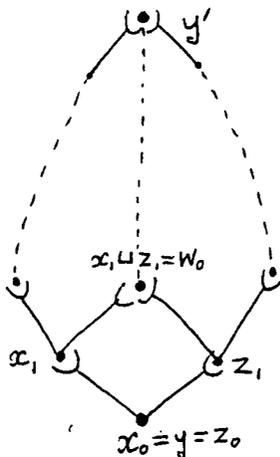
$$\left| \{[x_i, x_{i+1}] \mid [x_i, x_{i+1}] \in e\} \right| = \left| \{[z_i, z_{i+1}] \mid [z_i, z_{i+1}] \in e\} \right|$$

Proof

The proof is by induction on n taking as induction hypothesis the statement of the lemma. If $n=1$ then $m=1$ and $x_0=y=z_0$ and $x_1=y'=z_1$ by axiom C. Assume $n>1$ and the induction hypothesis for $n-1$. Suppose

$y = x_0 \prec \dots \prec x_n = y'$ and $y = z_0 \prec \dots \prec z_m = y'$. If $x_1 = z_1$,

we are home by induction so suppose $x_1 \neq z_1$. By axiom C, $x_1 \sqcup z_1$



exists and $x_1, z_1 \prec x \sqcup z_1$. By

axiom F we can find a covering

chain $x_1 \sqcup z_1 = w_0 \prec \dots \prec w_k = y'$.

By the induction hypothesis

$$\{[x_1, x_2]_{\sim}, \dots, [x_{n-1}, x_n]_{\sim}\} = \{[x_1, w_0]_{\sim}, [w_0, w_1]_{\sim}, \dots, [w_{k-1}, w_k]_{\sim}\}$$

where the number of representatives

of each event is the same in the

chains $x_1 \prec x_2 \prec \dots \prec x_n$ and

$x_1 \prec w_0 \prec \dots \prec w_k$. Consequently

$k = n-2$ so $z_1 \prec w_0 \prec \dots \prec w_k$ is of length $n-1$, so applying the

induction hypothesis again gives $\{[z_1, z_2]_{\sim}, \dots, [z_{m-1}, z_m]_{\sim}\} =$

$$\{[z_1, w_0]_{\sim}, [w_0, w_1]_{\sim}, \dots, [w_{k-1}, w_k]_{\sim}\}$$

where the number of representatives

of a particular event is the same in $z_1 \prec z_2 \prec \dots \prec z_m$ and

$z_1 \prec w_0 \prec \dots \prec w_k$. Combining these facts with $[x_1, w_0]_{\sim} [y_1, z_1]$

and $[z_1, w_0]_{\sim} [y_1, x_1]$ maintains the induction hypothesis. ■

The lemma above justifies the following definitions.

Definition 3.3.5

Define $E = \{[x, x']_{\sim} \mid x, x' \in D^0 \text{ \& } x \prec x'\}$. For x in D^0 define

$S(x) = \{[x_i, x_{i+1}]_{\sim} \mid 0 \leq i < n\}$ for some covering chain

$$x = x_0 \prec x_1 \prec \dots \prec x_n = x.$$

and $N(x, e)$ to be the number of representatives of e in such a covering chain.

Using N above we can count representatives along chains like $x_0 \prec x_1 \prec x_2 \prec x_3 \prec \dots$ where the covering relation may "switch direction". Such chains occur when considering \sim .

Lemma 3.3.6

Let x_0, x_1, \dots, x_n be a sequence in D^0 such that $x_i \prec x_{i+1}$ or $x_{i+1} \prec x_i$. Then $N(x_n, e) = N(x_0, e) + \left| \{ [x_i, x_{i+1}]_{\sim} \mid 0 \leq i < n \text{ \& } x_i \prec x_{i+1} \text{ \& } [x_i, x_{i+1}]_{\sim} \in e \} \right| - \left| \{ [x_{i+1}, x_i]_{\sim} \mid 0 \leq i < n \text{ \& } x_{i+1} \prec x_i \text{ \& } [x_{i+1}, x_i]_{\sim} \in e \} \right|$

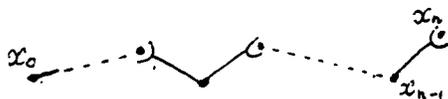
Proof

By induction on n .

If $n=0$ it is obvious.

Suppose $n>0$ and the result holds for $n-1$. First suppose $x_{n-1} \subset x_n$.

$$\begin{aligned} \text{Then } N(x_n, e) &= N(x_{n-1}, e) \text{ if } [x_{n-1}, x_n] \notin e \\ &= N(x_{n-1}, e) + 1 \text{ otherwise.} \end{aligned}$$



Now suppose $x_n \subset x_{n-1}$. This time

$$\begin{aligned} N(x_{n-1}, e) &= N(x_n, e) \text{ if } [x_n, x_{n-1}] \notin e \\ &= N(x_n, e) + 1 \text{ otherwise.} \end{aligned}$$



Equivalently $N(x_n, e) = N(x_{n-1}, e)$ if $[x_n, x_{n-1}] \notin e$
 $= N(x_{n-1}, e) - 1$ otherwise.

In either case the induction hypothesis is maintained. ■

Corollary 3.3.7

(i) Suppose $x \subset x'$ and $x = x_0, x_1, \dots, x_n$ is a sequence in D^0 such that $x_i \subset x_{i+1}$ or $x_{i+1} \subset x_i$ and $x' \subseteq x_n$. Then $x_i \subset x_{i+1}$ for some x_i, x_{i+1} so that $[x_i, x_{i+1}] \sim [x, x']$.

(ii) If D satisfies axiom R too and in D^0 $x \subset x' \subseteq y \subset y'$ then $[x, x'] \not\sim [y, y']$.

(iii) If D satisfies axiom R then for all x in D^0 , $N(x, e)$ equals 0 or 1.

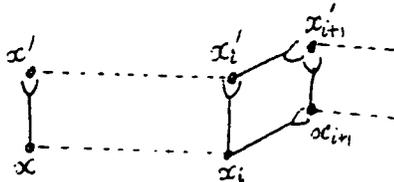
Proof

(i) Immediate by 3.3.6.

(ii) Suppose otherwise i.e. $x \subset x' \& y \subset y'$ and $[x, x'] \sim [y, y']$.

Then we would have $x_0, x'_0, x_1, x'_1, \dots, x_n, x'_n$ with $x_0 = x, x'_0 = x', x_n = y, x'_n = y'$ where $(x_i \subset x_{i+1} \text{ and } x'_i \subset x'_{i+1})$ or $(x_{i+1} \subset x_i \text{ and } x'_{i+1} \subset x'_i)$. By

(i) for some i we have $x_i \subset x_{i+1}$ and $[x_i, x_{i+1}] \sim [x, x']$. Considering the \sim -chain this would mean



for some x'_i, x'_{i+1} such that $[x'_i, x'_{i+1}] \sim [x, x']$.

But this contradicts axiom R.

(iii) Immediate by (ii). ■

We now look at how the map s behaves on supremum in D and characterise incompatibility.

Lemma 3.3.8

(i) Suppose $x, x', y \in D^0$ such that $x' \sqsupseteq x \prec y$ and $x' \uparrow y$. Then $x' \sqcup y$ exists, $x' \preceq x' \sqcup y$ and $s(x' \sqcup y) = s(x') \cup \{[x, y]_{\sim}\}$. Moreover if $x' \prec x' \sqcup y$ then $[x', x' \sqcup y]_{\sim} \sim [x, y]$.

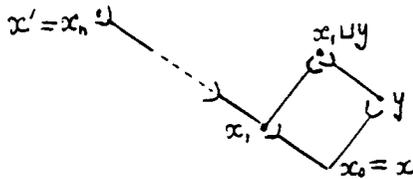
(ii) For x, y in D^0 , if $x \uparrow y$ then $x \sqcup y$ exists and $s(x \sqcup y) = s(x) \cup s(y)$.

(iii) For x, y in D^0 , if $x \not\uparrow y$ then $\exists z, z', z'' \in D^0 [z, z']_{\sim} \in s(x) \ \& \ [z, z'']_{\sim} \in s(y) \ \& \ z' \not\uparrow z''$.

Proof

(i) Take a covering chain $x = x_0 \prec \dots \prec x_n = x'$. We show

(i) by induction on n . For $n=0$ it is obvious. Suppose $n>0$ and that (i) holds for $n-1$. If $y=x_1$ it is obvious. Otherwise axiom C



gives $x_1 \sqcup y$ exists with $x_1 \prec x_1 \sqcup y$ and $y \prec x_1 \sqcup y$.

Clearly then $x' \sqsupseteq x_1 \prec x_1 \sqcup y$ with $x' \uparrow x_1 \sqcup y$ so we get $x' \preceq x' \sqcup y$ by induction. Also

$$s(x' \sqcup y) = s(x' \sqcup (x_1 \sqcup y)) = s(x') \cup \{[x_1, x_1 \sqcup y]_{\sim}\} \text{ by induction} \\ = s(x') \cup \{[x_0, y]_{\sim}\}.$$

(ii) Take a covering chain $\perp = y_0 \prec \dots \prec y_m = y$ and form $x \sqcup y_0, x \sqcup y_1 = (x \sqcup y_0) \sqcup y_1, \dots$ inductively showing $s(x \sqcup y_i) = s(x) \cup s(y_i)$.

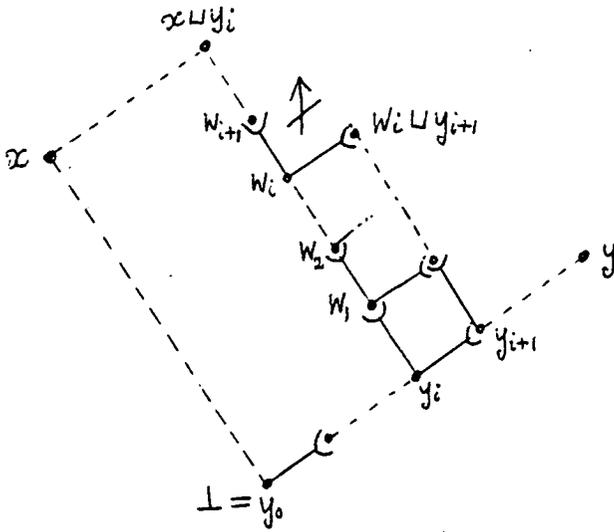
(iii) Take a covering chain up to y viz. $\perp = y_0 \prec \dots \prec y_n = y$.

As $x \not\uparrow y$ there is i s.t. $y_i \uparrow x$ and $y_{i+1} \not\uparrow x$. Form $x \sqcup y_i$. Then $y_{i+1} \not\uparrow x \sqcup y_i$. Take another covering chain from y_i to $x \sqcup y_i$ viz.

$y_i = w_0 \prec \dots \prec w_n = x \sqcup y_i$. (See the figure below.)

We have $[w_i, w_{i+1}]_{\sim} \in s(x)$ for all $i < n$. If $y_{i+1} \not\uparrow w_1$ we have the desired result. Otherwise, as $y_{i+1} \uparrow x \sqcup y_i$, repeated use of

axiom C must eventually give some j s.t. $w_j \uparrow y_{i+1} \ \& \ w_{j+1} \uparrow y_{i+1}$. But then $[w_j, w_j \sqcup y_{i+1}]_{\sim} \in s(y)$ and as $[w_j, w_{j+1}]_{\sim} \in s(x)$ we have the required result.



Illustrating the proof ■

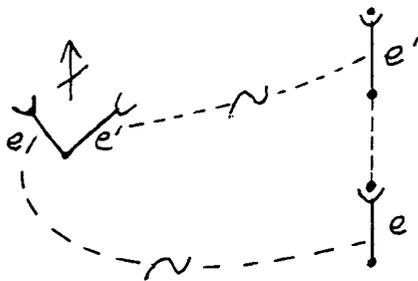
Corollary 3.3.9

The domain D is consistently complete.

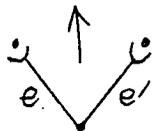
Proof

This follows directly from 3.3.8 (ii) using 3.1.9. ■

In constructing an event structure to represent the domain we take events to be the \sim -equivalence classes E with conflict relation given by: $e_0 \not\sim e_1$ iff $\exists [x, y_0] \in e_0, [x, y_1] \in e_1, y_0 \not\sim y_1$. Lemma 3.3.8 (iii) showed incompatibility could always be traced to such a situation. The next lemma is a key result. Axiom R is necessary. It says if we have this picture with the relations on prime intervals indicated



then somewhere we must also have



Lemma 3.3.10

Suppose D satisfies axiom R as well. Suppose for x, x', y, y', t, t' in D^0 that

(i) $x \prec x', x''$ and $x' \not\prec x''$

(ii) $y \prec y' \subseteq t \prec t'$

(iii) $[x, x'] \sim [y, y']$ and $[x, x''] \sim [t, t']$.

Then for some w, w', w'' in D^0 we have $w \prec w', w''$ and $[w, w'] \sim [x, x']$ and $[w, w''] \sim [x, x'']$ and $w' \uparrow w''$.

Proof

As $[y, y'] \sim [x, x']$ we get a sequence of prime intervals $[z_0, z'_0], \dots, [z_n, z'_n]$ where $z_0 = y$ and $z_n = x$ and for all i

$$[z_i, z'_i] \sim [x, x']$$

and ($z_i \prec z_{i+1}$ with $[z_i, z_{i+1}] \not\prec [x, x']$
or $z_{i+1} \prec z_i$ with $[z_{i+1}, z_i] \not\prec [x, x']$)

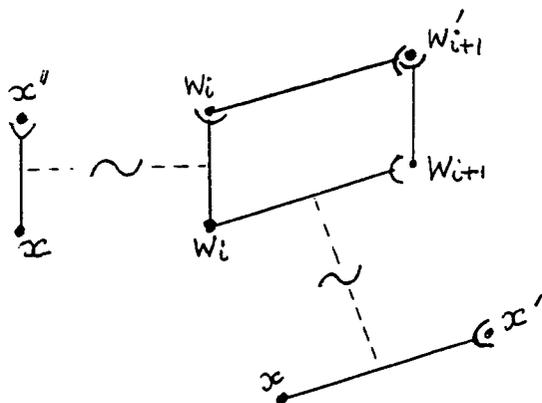
This uses axiom R.

As $[t, t'] \sim [x, x'']$, identically we get a sequence of prime intervals $[w_0, w'_0], \dots, [w_m, w'_m]$ with $x = w_0$, $t = w_m$ where for all i

$$[w, w'_i] \sim [x, x'']$$

and ($w_i \prec w_{i+1}$ with $[w_i, w_{i+1}] \not\prec [x, x'']$
or ($w_{i+1} \prec w_i$ with $[w_{i+1}, w_i] \not\prec [x, x'']$)

Now consider the sequence $y = z_0, z_1, \dots, z_n = x = w_0, w_1, \dots, w_m = t$.
By 3.3.7 (i) for some i , $w_i \prec w_{i+1}$ and $[w_i, w_{i+1}] \sim [x, x']$. Thus
somewhere along the chain giving $[x, x''] \sim [t, t']$ we have:



Taking $w = w_i$, $w' = w_{i+1}$, $w'' = w'_i$
gives the required result. ■

Unfortunately in the proof of the next lemma we need axiom V. If it could be avoided then we could immediately prove ω -algebraic domains satisfying F, C, R were represented by event structures where conflict $\not\prec$ was now a \subseteq -right closed predicate on events (or equivalently was replaced by the complement of such a predicate, a consistency relation). See example 3.3.17 and the remark which

follows it.

Lemma 3.3.11

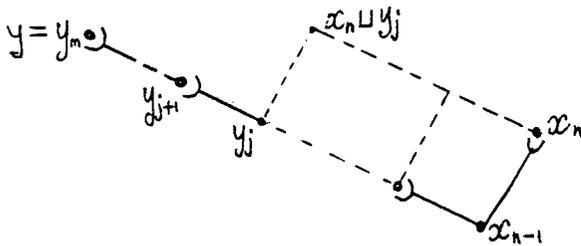
Suppose D in addition satisfies axioms R and V . Then for x, y in D^0

$$s(x) \subseteq s(y) \Rightarrow x \sqsubseteq y.$$

Proof

Suppose x, y are isolated elements of D with $s(x) \subseteq s(y)$.

Take a covering chain $\perp = x_0 \prec \dots \prec x_n = x$. We show by induction on n that $x_n \sqsubseteq y$. If $n=0$ it is obvious. Suppose $n>0$ and the result for $n-1$ i.e. that $x_{n-1} \sqsubseteq y$. Take a covering chain $x_{n-1} = y_0 \prec y_1 \dots \prec y_m = y$. For some i we have $[y_i, y_{i+1}] \sim$



$[x_{n-1}, x_n]$. We have $x_n \uparrow y_i$ as otherwise we would contradict axiom V by 3.3.9 (iii) and 3.3.10. By lemma 3.3.8 we get $y_j \prec x_n \sqcup y_j$ with $[y_j, x_n \sqcup y_j] \sim [x_{n-1}, x_n]$.

Thus $[y_j, x_n \sqcup y_j] \sim [y_j, y_{j+1}]$ so by axiom $R, y_{j+1} = x_n \sqcup y_j$.

Therefore certainly $x_n \sqsubseteq y$ as required to complete the induction step. ■

We now give the main theorem. We have seen how event structures E give domains $\Gamma(E)$ satisfying all our axioms. This theorem shows that if a domain D satisfies the axioms then there is an event structure E such that $\Gamma(E) \cong D$; the event structures of 3.3.1 represent domains satisfying the axioms. Moreover if D is distributive then there is a strongly-deterministic event structure E so that $\Gamma(E) = D$.

Theorem 3.3.12

Suppose D is an ω -algebraic domain satisfying axioms F, C, R and V . Then there is an event structure (E, \vdash, \otimes) as defined in 3.3.1 such that $\Gamma(E) \cong D$.

Also if D is distributive the event structure E may be taken to be strongly-deterministic.

Proof

Let D be such a domain. Define

$E = \{[x, x'] \sim \mid x, x' \in D^0 \ \& \ x \prec x'\}, A \vdash e \text{ iff } A \in \{s(x) \mid [x, x'] \in e\}$
 $e \not\bowtie e' \text{ iff } \exists x, x', x'' \in D^0 \ [x, x'] \in e \ \& \ [x, x''] \in e' \ \& \ x' \not\prec x''.$

(Note by axiom V, $e \not\bowtie e' \text{ iff } \forall x, x', x'' \in D^0 \ [x, x'] \in e \ \& \ [x, x''] \in e' \ \& \ x' \not\prec x''$)

To show $D \cong \prod(E)$ it is sufficient to show their isolated elements are order isomorphic (see 3.1.23).

Suppose $x \in D^0$. Clearly $s(x)$ is a finite configuration as otherwise by 3.3.11 axiom V is contradicted. The map $s: D^0 \rightarrow \prod(E)^0$ is monotonic by 3.3.4 and 1-1 by 3.3.11. Also by 3.3.11 $s^{-1}: sD^0 \rightarrow D^0$ is monotonic. Thus we only require that s is onto. To this end:

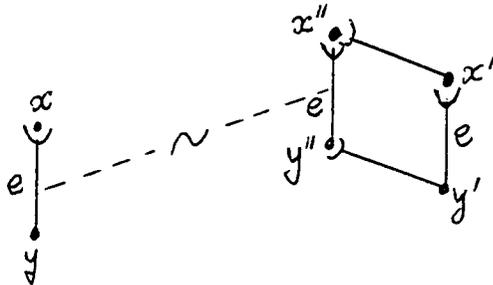
Suppose $A \in \prod(E)^0$. Then A is a finite configuration. Thus we have $A = \{a_1, \dots, a_n\}$ so that

$$\phi \vdash a_1 \text{ and } \forall i \exists B \subseteq \{a_1, \dots, a_{i-1}\} B \vdash a_i \text{ and } \forall i, j \neg (a_i \not\bowtie a_j).$$

For some x_1 we have $[1, x_1] \in a_1$. We inductively construct a covering chain $1 \prec x_1 \prec \dots \prec x_n$ s.t. $[x_{i-1}, x_i] \in a_i$. Then $s(x_n) = A$ as required. Suppose the chain has been constructed up to x_{i-1} for $i < n+1$. Then for some y, y' in D^0 we have $[y, y'] \in a_i$ with $s(y) \vdash a_i$ and $s(y) \subseteq s(x_{i-1})$. Thus $y \sqsubseteq x_{i-1}$ (by 3.3.11). By 3.3.9 (iii) we have $x_{i-1} \sqcup y'$ exists. As $[y, y'] \notin s(x_{i-1})$ we get $x_{i-1} \prec x_{i-1} \sqcup y'$. Take $x_i = x_{i-1} \sqcup y'$. Then $[x_{i-1}, x_i] \in a_i$, completing the induction step.

Now assume D is distributive. Taking E as defined above let $e \in E$. Choose x minimal so that $[y, x] \in e$. We show by induction on the length of the \sim -chain that if $[y', x'] \sim [y, x]$ then $x \sqsubseteq x'$.

Suppose $[y', x'] \sim [y, x]$ and the hypothesis is true for all \sim -chains of lesser length - it is clearly true for chains of length 0 and 1. The only difficulty occurs if we have



with $x \neq x''$ and $y \neq y''$ where by induction $x \sqsubseteq x''$. From distributivity as $y \uparrow x'$ we have

$$x \cap (y \sqcup x') = (x \cap y) \sqcup (x \cap x').$$

But $x \sqsubseteq y \sqcup x'$ as $s(x) \subseteq s(y) \cup s(x')$ and $x \cap y = y$ so the distributivity equation becomes

$$x = y \sqcup (x \cap x').$$

Because $e \in s(x) = s(y) \cup s(x \cap x')$ and $e \notin s(y)$ we have $e \in s(x \cap x')$.

As x is minimal $x \sqsubseteq x'$ as required.

Therefore if D is distributive we may define \vdash^* by

$$A \vdash^* e \text{ iff } A = \bigcap \{s(x) \mid [x, x'] \in e\}.$$

This gives a strongly-deterministic event structure (E, \vdash^*, \otimes) generating $\prod(E) \cong D$. ■

Of course now we may work either with domains satisfying the axioms or with their representation. As an illustration we show the domains are coherent and irreducible-algebraic, now defined.

Definition 3.3.13

Suppose (L, \sqsubseteq) is a partial order. Suppose $y \in L$. Then y is an (irreducible) complete irreducible iff for all (finite) subsets X of L with suprema $y = \bigsqcup X \Rightarrow \exists x \in X \ y = x$. If L is further an algebraic domain then L is irreducible-algebraic iff

$$\forall x \in L \ x = \bigsqcup \{y \sqsubseteq x \mid y \text{ is a complete irreducible}\}.$$

(Note for algebraic domains complete irreducibles are necessarily isolated; in general they need not be.)

Proposition 3.3.14

If D is an ω -algebraic domain satisfying axioms F, C, R and V then D is coherent and irreducible-algebraic.

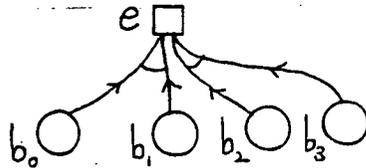
Proof

By the representation theorem we may work with $\prod(E) \cong D$ for some event structure E . Coherence is then obvious. For an event e in E a \sqsubseteq -minimal configuration containing e is a complete irreducible. Conversely any complete irreducible of $\prod(E)$ is such a configuration. Any configuration is clearly the union of these. ■

Later, from chapter 4 on, we shall make considerable use of a

particular kind of irreducible, the complete primes. For example in the case where D may be represented by a strongly-deterministic event structure the complete irreducibles coincide with the complete primes. We remark that one can by-pass the use of prime intervals to represent events and instead use complete irreducibles with equivalence relation based on one irreducible replacing another in an irredundant decomposition of an isolated element into irreducibles.

Structures of the form (E, \vdash, \ast) are interesting in themselves. They are a generalisation of the matrices of concrete domains. Later (from chapter 4 on) we shall consider a form of strongly-deterministic event structure; then \vdash can be replaced by a partial order \leq . Note we could relax the definition of securing \vdash so that an event could be enabled by an infinite set. Such structures would generalise matrices and the event structures of chapter 4. (Their configurations which were complete irreducibles need not be isolated and the configurations would no longer generally form an algebraic domain.) Structures like (E, \vdash, \ast) can be represented as Petri nets where an event may occur through several alternative sets of conditions holding; we can draw this as:



The event e can fire when b_0 and b_1 hold or when b_2 and b_3 hold. Such "disjunctive" causality relations occur naturally in physics (not just example 3.2.8! For example the post light-cone of a point p in space-time consists of all points at which events might occur to cause an event at p).

We have done most of the work necessary to get the representation theorem for concrete domains. These differ from the domains above in that axiom V is replaced by axiom Q . We use the following lemma to show axiom Q implies axiom V , in the presence of the other axioms, so then we can use the above representation result. Recall axiom Q :

$z \supseteq x \prec y$ & $z \not\prec y \Rightarrow \exists! t \in z$ $x \prec t \not\prec y$ where all elements can be assumed isolated.

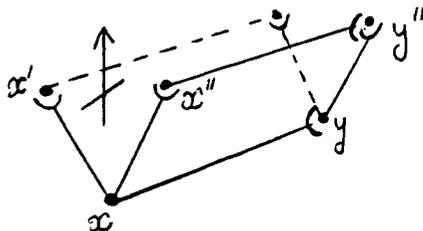
Lemma 3.3.15

Suppose D is ω -algebra and satisfies axioms F, C, R and Q.
Then for elements in D^0

- (i) If $x \prec x', x'' \& x' \uparrow x'' \& y \prec y'' \& [x, x''] \leq [t, t'']$ then
 $\exists y' y \prec y' \& y' \uparrow y'' \& [x, x'] \leq [y, y']$.
- (ii) If $x \prec x', x'' \& x' \uparrow x'' \& y \prec y'' \& [y, y''] \leq [x, x'']$ then
 $\exists y' y \prec y' \& y' \uparrow y'' \& [y, y'] \leq [x, x']$.
- (iii) If $x \prec x', x'' \& x' \uparrow x'' \& y \prec y'' \& [y, y''] \sim [x, x'']$ then
 $\exists y' y \prec y' \& y' \uparrow y'' \& [y, y'] \sim [x, x']$.

Proof

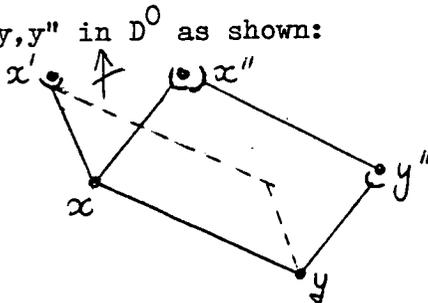
(i) Take x, x', x'', y, y'' in D^0 as shown:



From the uniqueness part of axiom Q $x' \uparrow y$. Then by axiom C
 $x' \sqcup y$ exists and $x', y \prec x' \sqcup y$.

Take $y' = x' \sqcup y$.

(ii) Take x, x', x'', y, y'' in D^0 as shown:



As $x' \uparrow x''$ we have $x' \uparrow y''$ because x'' is $x \sqcup y''$. Thus by the
existence part of axiom Q $\exists y' x y \prec y' \uparrow y''$. By axiom C
 $[y, y'] \leq [x, x']$.

(iii) This follows by repeated use of (i) and (ii) along a sequence
of \leq or \geq steps connecting $[x, x'']$ and $[y, y'']$ by the \sim -relation. ■

In the representation theorem for concrete domains we use the
above lemma to show concrete domains satisfy axiom V. Then we can
certainly represent the domain by an event structure of the form
 (E, \vdash, \times) where E , \vdash and \times were defined in the proof of 3.3.12.

The extra strength of axiom Q gives $\times \cup 1$ an equivalence relation (the equivalence classes are places) and that \vdash respects $\times \cup 1$ (it enables places).

Theorem 3.3.16

- (i) The configurations of a matrix M ordered by inclusion $\uparrow(M)$ form a concrete domain.
(ii) If D is a (distributive) concrete domain then there is a (strongly-deterministic) matrix M such that $\uparrow(M) \cong D$.

Proof

(i) As in 3.3.3.

(ii) Let D be a concrete domain. Thus it is ω -algebraic and satisfies axioms F,C,R and Q.

We first show Q implies V. Suppose in D^0 we have $x \subset x', x'' \not\prec x' \not\prec x'' \not\prec x''$ & $y \subset y', y'' \not\prec [x, x'] \sim [y, y']$ & $[x', x''] \sim [y', y'']$. By 3.3.15 (iii) above and axiom R we get $y' \not\prec y''$ as required.

Thus as in 3.3.12 we have $D \cong \uparrow(E)$ where $E = \{[x, x'] \mid x, x' \in D^0 \text{ \& } x \subset x'\}$

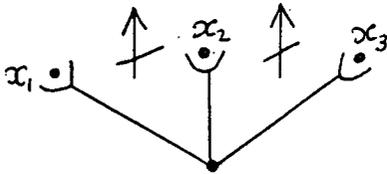
$A \vdash e$ iff $A \in \{s(x) \mid [x, x'] \in e\}$

$e \times e'$ iff $\exists x, x', x'' \in D^0 [x, x'] \in e \text{ \& } [x, x''] \in e' \text{ \& } x' \not\prec x''$.

However now because of axiom Q the relation $\times \cup 1$ is an equivalence relation: In showing this the only case of interest is when

$e_1 \times e_2 \times e_3$ and $e_1 \neq e_3$, where we require $e_1 \times e_3$. By 3.3.15

(iii) we obtain some x, x_1, x_2, x_3 so that



with $[x, x_1] \in e_1, [x, x_2] \in e,$
 $[x, x_3] \in e_3.$

By the uniqueness part of axiom Q, $x_1 \not\prec x_3$ thus $e_1 \times e_3$ as required.

Also by lemma 3.3.15 (iii) the relation \vdash respects $\times \cup 1$ -equivalence

classes: Suppose $e_1 \times e_2$ and $A \vdash e_1$. Then for some y, y' in D^0

$A = s(y), y \subset y'$ and $[y, y'] \in e_1$. Also for some x, x', x'' in D^0 we have $x \subset x', x''$ and $x' \not\prec x''$, $[x, x'] \in e_1$, and $[x, x''] \in e_2$. By

3.3.15 (iii) we get some y'' s.t. $y \subset y''$ and $y' \not\prec y''$ and $[y, y''] \in e_2$.

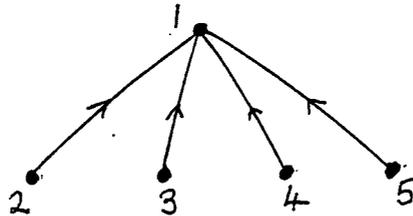
Thus $A \vdash e_2$.

Now we get a matrix by taking places as $\times \vee 1$ -equivalence classes and enabling relation from events to places induced by . If D is distributive a strongly-deterministic matrix can be made as in 3.3.12. ■

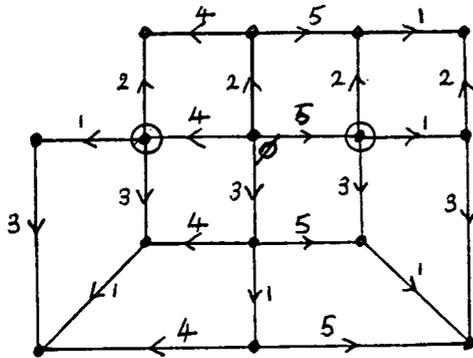
We conclude with a little example to show that axiom V is not implied by coherence in the presence of the other axioms ω -algebraicity and axioms F,C and R.

Example 3.3.17

We construct a domain which is finite, so certainly ω -algebraic, also satisfies F,C,R, is coherent but does not satisfy axiom V. It is best seen as the configuration of a new kind of event structure in which the binary conflict relation has been replaced by an inconsistency predicate. We have four events $E = \{1,2,3,4\}$. The enabling relation is $\emptyset \vdash 2,3,4,5$ and $\{2\} \vdash 1, \{3\} \vdash 1, \{4\} \vdash 1$ and $\{5\} \vdash 1$. Thus 1 is enabled in 4 different ways.



The inconsistency predicate \times contains $\{2,3\}$, $\{4,5\}$ and $\{1,2,4\}$. The configurations are then the secured subsets which do not include an element of \times . They give this domain pictured "aerially":



The points circled highlight where axiom V fails; the events 1 and 2 can occur compatibly at one configuration but not at the other. However the domain does satisfy C and R (consider its representation) and is coherent: Let A be a subset of configurations which is not compatible. This means $\cup A$ includes $\{2,3\}$, $\{4,5\}$ or $\{1,2,4\}$. If it includes $\{2,3\}$ or $\{4,5\}$ then there are a_1, a_2 in A such that either

$2 \in a_1$ & $3 \in a_2$ or $4 \in a_1$ & $5 \in a_2$; then in either case $a_1 \uparrow a_2$.
 Otherwise $\cup A$ includes $\{1,2,4\}$ but does not contain 3 or 5. Then
 there are a_1, a_2 in A with $(\{1,2\} \subseteq a_1$ & $4 \in a_2)$ or
 $(\{1,4\} \subseteq a_1$ & $2 \in a_2)$; in either case $a_1 \uparrow a_2$. Thus $A \uparrow$ implies
 there are a_1, a_2 in A with $a_1 \uparrow a_2$ i.e. the domain is coherent.

The form of event structure used in this example is a natural
 one. I conjecture that event structures of the form (E, \vdash, \mathbb{X}) as
 in 3.3.1 but where $\mathbb{X} \subseteq \mathcal{F}(E)$ (so configurations are secured and do
 not include an element of \mathbb{X}) represent domains which are
 ω -algebraic and satisfy axioms F, C, and R.

Chapter 4. Petri nets give Scott domains

In this chapter we shall establish some basic, and essentially formal, connections between Petri nets and domains using the intermediate notion of an event structure. Here we shall see an example of a (very simple) representation theorem in which a domain of state-like elements is represented by a partial order. Initially we shall work with causal nets later extending the results to occurrence nets (defined below) which are argued to be a possible semantics for contact-free transition nets with initial marking.

4.1 Causal nets, elementary event structures and lattices

Recall the definition of a causal net (definition 2.4.1) and that for them the conditions and events correspond to occurrences of holdings of conditions and occurrences of events. Further each event is "caused by" a unique subnet $(\{x \in B \cup E \mid x F^+ e\})$ and "causes" a unique subnet $(\{x \in B \cup E \mid e F^+ x\})$ a fact which may not be true for transition nets in general.

It is natural to focus on the pattern of occurrences of events of causal nets. The relation F specifies a certain dependency; if $e F^+ e'$ in the causal net then in the course of the computation described by the net e' cannot occur without e having occurred already. This leads to the following definition of a "causality" structure on events:

Definition 4.1.1

An elementary event structure is a partial order (E, \leq) where E is a set of events, and \leq is the partial order over E called the causality relation.

Thus here we choose to study the structure of events of a net rather than the structure of conditions. (One could explore the implications of dropping events) Our approach gives a neat translation of nets to domains but there are other reasons for focussing on events. Conditions can to some extent be recovered from the structure on events and, as will be seen in chapter 6, have a far more complicated structure. It is natural to consider the easier events first.

The relation between causal nets and elementary event structures is obvious.

Theorem 4.1.2

Let $N = (B, E, F)$ be a causal net. Then $\mathcal{E}(N) =_{\text{def}} (E, F^* \uparrow E^2)$ is an elementary event structure.

Pf. Only asymmetry in non-trivial and this follows from N6 of definition 2.4.1. ■

From an elementary event structure we can produce a causal net; in general there will be more than one.

Theorem 4.1.3

Let (E, \leq) be an elementary event structure. Then there is a causal net $\mathcal{N}(E)$ such that $E = \mathcal{E} \circ \mathcal{N}(E)$.

Pf. We take $\mathcal{N}(E)$ to be the net (B, E, F) formed from events E and

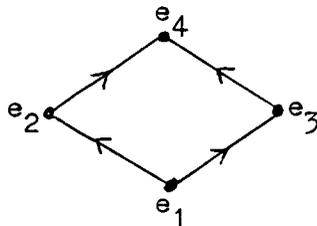
$$B = \{(e, e') \mid e, e' \in E, e \not\leq e'\} \cup \{(0, e) \mid e \in E\} \cup \{(e, 1) \mid e \in E\} \cup \{(0, 1)\}$$

and

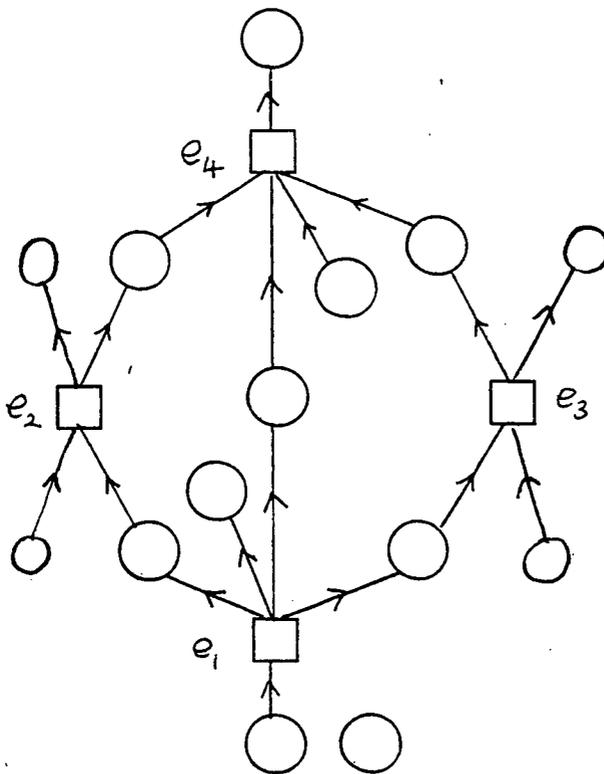
$$F = \{((e, e'), e') \mid e, e' \in E \ \& \ (e, e') \in B\} \cup \{(e, (e, e')) \mid e, e' \in E \ \& \ (e, e') \in B\} \cup \{((0, e), e) \mid e \in E\} \cup \{(e, (e, 1)) \mid e \in E\}.$$

Note if E is null the net $\mathcal{N}(E)$ consists of a single condition. The axioms on causal nets follow trivially as does the fact that $E = \mathcal{E} \circ \mathcal{N}(E)$. ■

Note that we have lost structure in passing from a causal net to its elementary event structure. Take the net N as example 2.4.2. Its associated elementary event structure $\mathcal{E}(N)$ is



and $\mathcal{N} \circ \mathcal{E}(N)$ is (notice the isolated condition $(0, 1)$)

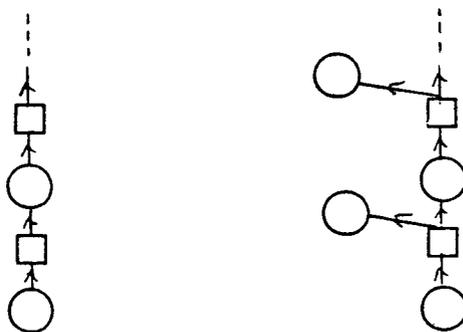


which contains more conditions. It is fairly clear that many definitions of \mathcal{N} would work in theorem 4.1.3. The one we have chosen is maximal once we accept an extensionality restriction on conditions (N2) which identifies conditions with the same pre and post events. This is why the isolated condition, $(0,1)$ in the construction, has been included.

From our point of view it is reasonable to accept the following equivalence relation on causal nets

$$N_1 \equiv N_2 \text{ iff } \mathcal{E}(N_1) = \mathcal{E}(N_2).$$

However it would seem undesirable from the view of traditional net theory; we lose track of too many conditions and the following K-dense and non-K-dense nets are identified.



However as mentioned before we disagree with K-density and we shall spell out our case in the next chapter.

We now use a little more computational intuition in answering: What is the natural domain of information points associated with an elementary event structure, and thus a causal net? In following a course of computation we are interested in what events have occurred and we also know that for one described by a causal net N , or its associated elementary event structure E , that an event having occurred implies its predecessors have occurred. Thus information points are certainly left-closed w.r.t. $F^*(E)$ or \leq .

Definition 4.1.4

Let (E, \leq) be an elementary event structure. Then $x \subseteq E$ is left-closed iff

$$e \leq e' \& e' \in x \Rightarrow e \in x.$$

We take $\mathcal{L}_c(E)$ to be the left-closed subsets of E ordered by inclusion.

Ordering $\mathcal{L}_c(E)$ by inclusion corresponds to the intuition that the more events that have occurred the more information we have. We can characterise the structures $\mathcal{L}_c(E)$ quite easily; we use the concept of a complete prime which will pop-up frequently.

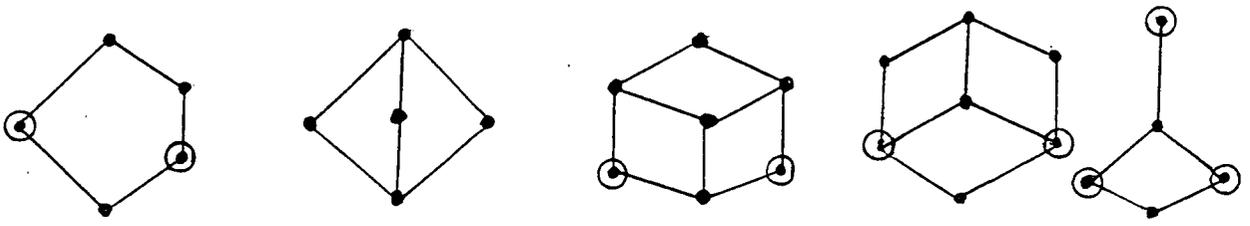
Definition 4.1.5

Let (D, \sqsubseteq) be a partial order. An element $p \in D$ is a complete prime (prime) iff for every $X \subseteq D$ (every finite $X \subseteq D$), if $\bigsqcup X$ exists and $p \sqsubseteq \bigsqcup X$, then there exists an $x \in X$ s.t. $p \sqsubseteq x$. The set of complete primes of D is denoted $\text{Pr}(D)$.

Definition 4.1.6

A partial order (D, \sqsubseteq) is prime algebraic iff for every element $d \in D$, $\bigsqcup P_d$ exists (where $P_d = \{p \sqsubseteq d \mid p \in \text{Pr}(D)\}$) and $d = \bigsqcup P_d$.

Example 4.1.7



In the above representation of partial orders the (complete) primes are circled, and it is easy to see that none but the last of these

partial orders are prime algebraic.

We relate the concept of prime algebraicity to more standard lattice-theoretic concepts in the next proposition.

Proposition 4.1.8

A complete lattice is prime algebraic iff it is algebraic and every finite (or isolated) element is a lub of complete primes. Further in such a lattice every complete prime is finite, an element is a complete prime iff it is completely irreducible and the distributivity property holds.

We now present results leading to the characterisation of the structures $\mathcal{L}(E)$.

Theorem 4.1.9

Let (E, \leq) be an elementary event structure. Then $\mathcal{L}(E)$ is a prime algebraic complete lattice. Its complete primes are those elements of the form $[e] \stackrel{\text{def}}{=} \{e' \in E \mid e' \leq e\}$ for $e \in E$.

Proof The structure $\mathcal{L}(E)$ is a complete lattice with $\bigsqcup X = \bigcup X$ (and $\bigsqcap X = \bigcap X$).

Each $[e]$ is clearly left-closed, and is a complete prime as if $[e] \subseteq \bigsqcup X = \bigcup X$, then $e \in [e] \subseteq X$ and so for some x in X , $e \in x$, and so $[e] \subseteq x$. As we have $x = \bigcup \{[e] \mid e \in x\}$, for any x in $\mathcal{L}(E)$, each element is a lub of the complete primes below it, and so $\mathcal{L}(E)$ is prime algebraic.

Finally, if x is a complete prime, then as we have $x = \bigcup \{[e] \mid e \in x\}$ we must have $x \subseteq [e]$ for some e in x . But then we must have $x = [e]$, which completes the proof. ■

This theorem indicates how to map our lattices to elementary event structures.

Definition 4.1.10

Let (D, \sqsubseteq) be a prime algebraic complete lattice. The elementary event structure $\mathcal{P}(D)$ is defined as $(\text{Pr}(D), \sqsubseteq \upharpoonright \text{Pr}(D)^2)$.

Before stating the characterisation of the structures $\mathcal{L}(E)$ we shall need the following general lemma.

Lemma 4.1.11

Let (D, \sqsubseteq) be a prime algebraic partial order. Then the map $\pi: D \rightarrow \mathcal{L}(\mathcal{P}(D))$ is defined by

$$\pi(d) =_{\text{def}} \{p \in \text{Pr}(D) \mid p \sqsubseteq d\}$$

is an order monic (i.e. $\pi(d) \sqsubseteq \pi(d')$ iff $d \sqsubseteq d'$), it preserves and reflects complete primes, and preserves those lubs that exist in D .

Proof Clearly π is monotonic. If, on the other hand, $\pi(d) \sqsubseteq \pi(d')$ then from prime algebraicity of P

$$d = \bigsqcup \{p \in \text{Pr}(D) \mid p \sqsubseteq d\} = \bigsqcup \pi(d) \sqsubseteq \bigsqcup \pi(d') = d'.$$

Let p be a complete prime of D , then $\pi(p)$ is a complete prime in $\mathcal{L}(\mathcal{P}(D))$ from Theorem 4.1.9. On the other hand, it also follows from the theorem that if $\pi(d)$ is a complete prime, then d is a complete prime, too. So, π preserves and reflects complete primes. Finally, if $\bigsqcup_D X$ exists then

$$\begin{aligned} \pi(\bigsqcup_p X) &= \{p \in \text{Pr}(D) \mid p \sqsubseteq \bigsqcup_p X\} \\ &= \bigcup_{x \in X} \{p \in \text{Pr}(D) \mid p \sqsubseteq x\} \quad (\text{by the definition of complete primeness}) \\ &= \bigcup_{x \in X} \pi(x) \end{aligned}$$

We shall often make use of the well-known fact that any mapping between partial orders which is onto and an order monic is an isomorphism. This happens in the proof of the next theorem, which states the very close relationship which exists between our lattices and event structures.

Theorem 4.1.12

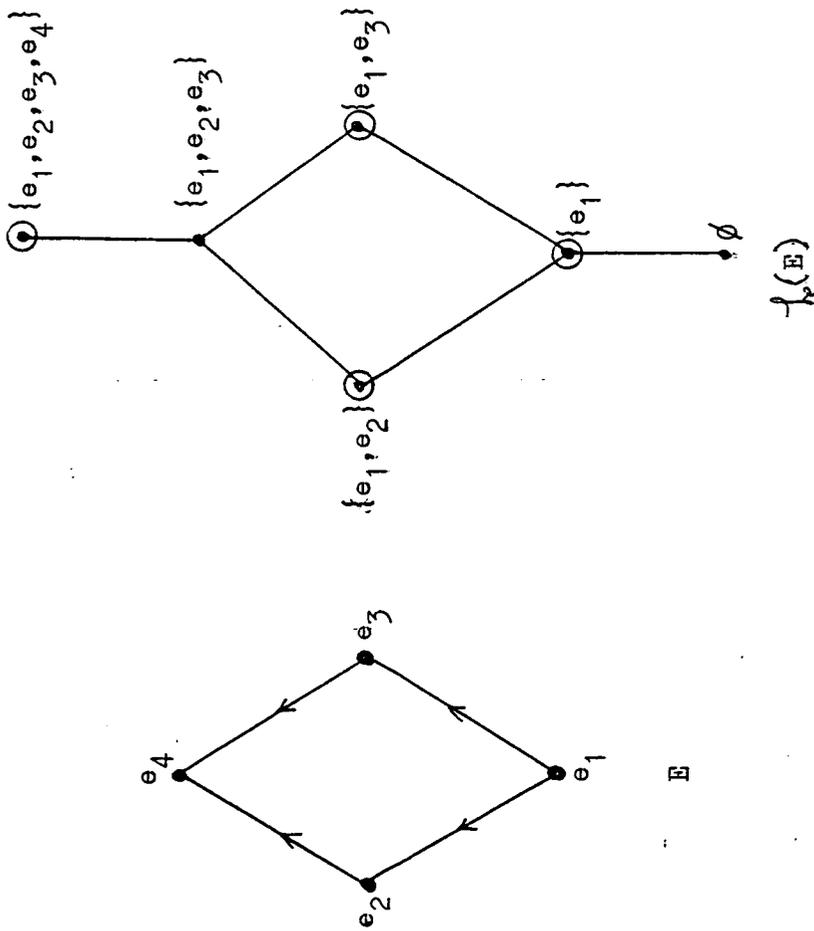
Let (E, \leq) be an elementary event structure; then $E \cong \mathcal{P}(\mathcal{L}(E))$. Similarly, let (D, \sqsubseteq) be a prime algebraic complete lattice; then $D \cong \mathcal{L}(\mathcal{P}(D))$.

Proof Define $\psi: E \rightarrow \mathcal{P}(\mathcal{L}(E))$ by $\psi(e) = [e]$. Then ψ is well-defined and onto from Theorem 4.1.9. Furthermore, ψ is easily proved to be an ordermonic, and hence it is an isomorphism, which proves the first part of the theorem. As for the second part - π is known from Lemma 4.1.11 to be an ordermonic; π is also onto, since for any element X of $\mathcal{L}(\mathcal{P}(D))$, $\bigsqcup_D X$ exists (D is a complete lattice) and

$$\begin{aligned} \pi(\bigsqcup_D X) &= \bigcup_{x \in X} \pi(x) && \text{(by Lemma 4.1.11)} \\ &= \bigcup \{[x] \mid x \in X\} && \text{(by the definition of } \pi \text{)} \\ &= X. \end{aligned}$$

So, π is indeed an isomorphism. ■

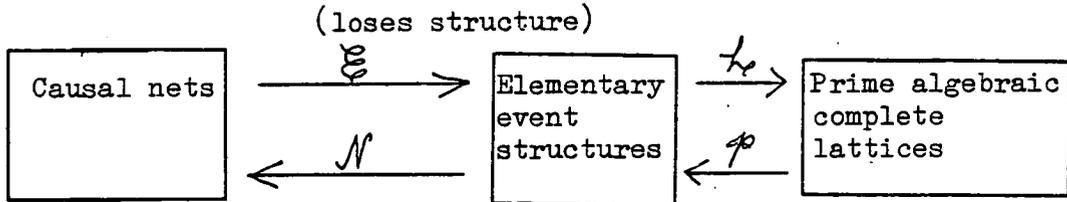
Example 4.1.13



Take E to be the elementary event structure associated with the causal net from Example 2.4.2. E and $\mathcal{L}(E)$ are pictured above. The primes of $\mathcal{L}(E)$ are circled, and it is easy to see that $E \cong \mathcal{P}(\mathcal{L}(E))$.

Theorem 4.1.12 shows that elementary event structures and prime algebraic complete lattices are equivalent structures, in the sense that one does not lose any structural information going from one to the other via the \mathcal{L} and \mathcal{P} mappings - in contrast to the earlier result about the relationship between causal nets and elementary event structures.

The framework we have set up so far can be pictured as



A lot of our work in the next few chapters will be in extending and consolidating this set-up.

In the last chapter on concrete domains we saw another representation theorem in which events were extracted from the domains by taking equivalence classes of prime intervals under \sim the reflexive, symmetric, transitive closure of \leq' given by $[x, x'] \leq' [y, y']$ iff $x \subset y$ & $x' \subset y'$. There the elements x, x', y, y' were assumed isolated. A more general relation, between arbitrary prime intervals, is the following:

Definition 4.1.14

Let D be a cpo. For $[x, x']$ and $[y, y']$ prime intervals of D define $[x, x'] \leq [y, y']$ iff $y' = y \sqcup x$ & $x = y \sqcap x'$. Define \sim' to be the symmetric transitive closure of \leq .

The relation \sim' extends the relation \sim of chapter 3. The \sim -equivalence classes are in 1-1 correspondence with the \sim' -equivalence classes for the domains of chapter 3; this follows from the representation theorem which shows that for such domains events are secured by a finite set of events.

In many ways prime intervals correspond more closely to our intuitions about events; a prime interval corresponds to a unit jump in information. How do these two notions of an event tie up? For a prime algebraic lattice there is a one-one correspondence between primes and \sim' -equivalence classes of prime intervals. This follows most easily using the above representation theorem.

Proposition 4.1.15

Let (D, \sqsubseteq) be a prime algebraic complete lattice. Then for any prime interval $[d, d']$, $\Pi(d') \setminus \Pi(d)$ is a singleton. Hence if we put

$$\text{pr}([d, d']) \in \Pi(d') \setminus \Pi(d)$$

then pr is a well-defined map from prime intervals of D to $\text{Pr}(D)$.

The following theorem states the relation between the equivalence \sim' and pr .

Theorem 4.1.16

Let (D, \sqsubseteq) be a prime algebraic complete lattice. Then the following are equivalent for prime intervals $[d_1, d_1']$ and $[d_2, d_2']$:

1. $[d_1, d_1'] \sim' [d_2, d_2']$
2. $\text{pr}([d_1, d_1']) = \text{pr}([d_2, d_2'])$
3. There exists a prime interval $[d_3, d_3']$ s.t.
 $[d_1, d_1'] \geq [d_3, d_3'] \leq [d_2, d_2']$.

Further, if p is a complete prime of D then
 $p = \text{pr}(\left[\bigsqcup \{p' \in \text{Pr}(D) \mid p' \not\sqsubseteq p\}, p \right])$.

Proof

1. \Rightarrow 2. It follows easily from the definition of \leq that

$$[d_1, d_1'] \leq [d_2, d_2'] \Rightarrow \text{pr}([d_1, d_1']) = \text{pr}([d_2, d_2']).$$

2. \Rightarrow 3. Define $d_3 = d_1 \sqcap d_2$ and $d_3' = d_1' \sqcap d_2'$.

3. \Rightarrow 1. Trivial.

The last part of the theorem is obvious. ■

This theorem is the lattice-theoretic statement of the fact that an event is enabled (or caused) in a unique way. It proves a one-to-one correspondence between the complete primes and the more intuitive equivalence classes of prime intervals. This justifies our translation of events into complete primes.

Now, it is easy to see that the events of a causal net N are in one-to-one correspondence with the events of $\mathcal{E}(N)$, and the events of an elementary event structure E are in one-to-one correspondence with those of $\mathcal{N}(E)$. On the other hand, the events of E are also

in one-to-one correspondence with those of $\mathcal{L}_c(E)$, and the events of a prime algebraic complete lattice are in one-to-one correspondence with those of $\mathcal{P}(D)$.

The situation for translation of conditions is a good deal less pleasant. Our main tool for handling conditions is the extensionality axiom N2 which allows us to identify any condition b with its pre- and postevent ($\cdot b$ and $b \cdot$). For simplicity, we shall only demonstrate how conditions translate into elementary event structures.

A condition of an elementary event structure E is taken to be any condition of $\mathcal{N}(E)$. By definition this gives a nice one-to-one relationship between conditions of E and $\mathcal{N}(E)$, but, obviously, it is more interesting to see how conditions of a causal net N correspond to certain conditions of $\mathcal{E}(N)$. Define the map, bed , between these two sets of conditions as follows:

$$\forall b \in B: \text{bed}(b) = \begin{cases} (0, e') & \text{if } \cdot b = \emptyset \text{ and } b \cdot = \{e'\} \\ (e, 1) & \text{if } \cdot b = \{e\} \text{ and } b \cdot = \emptyset \\ (0, 1) & \text{if } \cdot b = \emptyset \text{ and } b \cdot = \emptyset \\ (e, e') & \text{if } \cdot b = \{e\} \text{ and } b \cdot = \{e'\} \end{cases}$$

It follows from the axioms of causal nets that bed is well-defined, and that it is one-to-one. However, in general bed will not be onto, obviously because of our construction of $\mathcal{N}(E)$, which in general generates a lot of redundant conditions. One could try to remedy this by a characterisation of the "essential" conditions of E . The following lemma is such an attempt.

Lemma 4.1.17

Let (E, \triangleleft) be an elementary event structure, and b one of its conditions. Then the following two conditions are equivalent:

1. For every causal net $N = (B, E, F)$ for which $E = \mathcal{E}(N)$, $b \in \text{bed}(B)$.
2. $b = (e, e')$, where e' covers e (with respect to the relation \triangleleft).

Proof Assume b of the required form, then clearly for every causal net $N = (B, E, F)$ for which $E = \mathcal{E}(N)$, there must exist a condition $b' \in B$ such that $eFb'Fe'$, and hence $b = \text{bed}(b')$. On the other hand,

if b is not of this form, construct a slightly modified form, N , of $\mathcal{N}(E)$ leaving out the condition corresponding to b , such that $E = \mathcal{E}(N)$ and $b \notin \text{bed}(B)$. ■

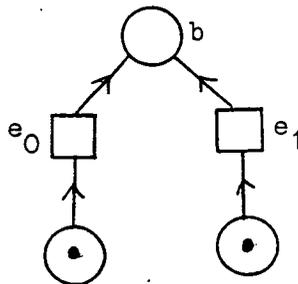
This lemma shows that the only essential conditions are the "points of non-density". However, the net consisting of the events of E and all essential conditions will not in general be mapped onto E by \mathcal{E} . Indeed, considering, for instance, the elementary event structure associated with the rationals shows that it is even possible for no condition to be essential.

In the next section we shall see how the causal dependency and the concurrency relation of causal nets translate nicely into the event and lattice structures.

4.2 Occurrence nets, event structures and domains

In chapter 2, introducing Petri nets, we often had to distinguish events (or transitions) from their occurrences and similarly conditions (or places) from their holdings (e.g. in the discussion of 2.2.10). Here we shall show^{how} an occurrence net, in which conditions and events stand for occurrences, can be associated with a contact-free transition net with initial marking. For one thing this will enable an especially simple definition of the concurrency relation. For another the associated occurrence net of a transition net seems a canonical representative of the computation described by the transition net at that level of description. We would like some category theoretic characterisation of the occurrence net of a transition net to clarify and support this view. At least it is an unfolding of the transition net (see section 2.5). Petri has said that the process level semantics of a transition net is the class of causal nets it unfolds into, where all the choices associated with such an unfolding are "made by the environment" [Pet2]. The occurrence net unfolding of a transition net represents such a class. Again we shall not worry too much about computational intuition here, sidestepping issues like what to take as states of the occurrence net (see chapter 5), how we play the token game on transition nets, whether or not we allow events to have concession forever etc. For the sake of definiteness however one can assume that no events are restless so that the transition nets here may be imagined to describe datatypes.

In general because of the presence of forwards and backward conflict the subnet "caused by" or "causing" an event or condition is not unique. In an occurrence net we wish the elements to represent occurrences as was the case with causal nets. From this point of view backwards conflict seems undesirable. For instance in



the condition b can be caused to hold in two ways, either through the occurrence of e_0 or e_1 . In occurrence nets we choose only to allow (formal) forwards conflict marked by events sharing a common precondition. (We say formal because for the moment we do not discuss whether or not there is a state at which this conflict really occurs.) In net theory this might seem undesirable as there one is sometimes concerned with "information leaving the system" which means getting to a state which could have arisen through different conflict resolutions. However our concerns are different. Firstly I am not clear what the semantics of a transition net with contact should be. Secondly we shall use occurrence nets to go from transition nets to domains of information. Here following Scott the level of information is determined by a partial order not, as would seem appropriate in net theory, by a digraph or category. This is because an information point in a domain "remembers" its past; it is like a partial history. On the other hand in net theory it is less standard to look at all the information potentially available to the environment as a system runs. There the information is stored by the system itself; because a system can loop there can be loops in the "can lead to" relation on information points.

As we have chosen to deal with forwards conflict only and we wish to stay close to causal nets it is natural to look for a replacement to axiom N4 in the definition of causal nets (2.4.1). Axioms N5 and N6 are maintained as, respectively, we still disallow backwards conflict and wish events and conditions to be occurrences.

Definition 4.2.1

Let $N = (B, E, F)$ be a Petri net satisfying N5 and N6 of definition 2.4.1 (that of a causal net). For any $a \in B \cup E$ let \bar{a} denote the subset of E defined by

$$\bar{a} = \{e \in E \mid e F^* a\}.$$

Two events e_1 and e_2 are said to be in (formal) direct conflict,

$$e_1 \#_{IN} e_2 \text{ iff } e_1 \neq e_2 \ \& \ \bullet e_1 \cap \bullet e_2 \neq \emptyset$$

Two elements of $B \cup E$, a_1 and a_2 , are said to be in (formal) conflict,

$$a_1 \#_N a_2 \text{ iff } \exists e_1, e_2 \in E \ e_1 \in \bar{a}_1 \ \& \ e_2 \in \bar{a}_2 \ \& \ e_1 \#_{IN} e_2.$$

We can now generalise the notion of a causal net.

Definition 4.2.2

A Petri net N is an occurrence net iff it satisfies N5 and N6 of definition 2.4.1 and further: N4' $\#_N$ is irreflexive.

We shall sometimes need to distinguish conflict as it arises in playing the token game (chapter 2) and what we call formal conflict which arises simply through F*-predecessors of two elements sharing a common precondition. This makes no mention of "reachable markings". Indeed here we have not discussed what a state of a causal net or occurrence net should be in our view. Until we do it cannot be clear how real formal conflict will be in general. Occurrence nets will be our new class of semantical nets. Elements of E and B still represent unique occurrences and holdings, respectively, and N4' guarantees that no event (or condition) is in conflict with itself (can occur on two different branches of the computation, so to speak). More importantly, the concept of concurrency carries over nicely:

Definition 4.2.3

For an occurrence net $N = (B, E, F)$, the concurrency relation $co_N \subseteq (B \cup E) \times (B \cup E)$ is defined by

$$co_N = ((B \cup E) \times (B \cup E)) \setminus (F^+ \cup (F^+)^{-1} \cup \#_N).$$

The following proposition is an immediate consequence of our definitions.

Proposition 4.2.4

Let $N = (B, E, F)$ be an occurrence net. Then co_N is symmetrical and reflexive. Furthermore, any two elements of $B \cup E$ are related in one of the three mutually exclusive ways: causally dependent, concurrent or in conflict.

Now we can generalise Petri's idea of case (though I do not regard it as the correct formulation of state - see next chapter). Recall the definition of ken (2.4.10).

Definition 4.2.4

For an occurrence net $N = (B, E, F)$ a case is defined to be a

For N_1 the set $\{b_i \mid i \in \omega\} \cup \{e_i \mid i \in \omega\} \cup \{s_i \mid i \in \omega\} \cup \{t_i \mid i \in \omega\}$ is a ken of $F^* \cup F^{*-1} \cup \#_{N_1}$. For N_2 the encircled events form a ken of $F^* \cup F^{*-1} \cup \#_{N_2}$.

We show how an occurrence net may be associated with a contact-free transition net with initial marking (N, M_0) . Recall that a net is contact-free iff for any reachable marking M and transition t , $\bullet t \subseteq M \Rightarrow \bullet t \cap M = \emptyset$. The idea behind our construction is that the behaviour of N will be described by an occurrence net with precisely one condition for each residence of a token on a place, and precisely one event for each firing possible for N . Roughly, in the construction the event and condition occurrences are taken to be transitions or places respectively together with the "minimal way" in which they are "caused" according to a local application of the token game. In more detail: The occurrence of a place is taken as the pair consisting of the place together with the transition occurrence which causes it to hold; the occurrence of a transition is taken as the pair consisting of the transition together with a set of concurrently holding occurrences of its preplaces from which it may occur. We grow the associated occurrence net inductively in stages starting from the initial marking as a set of occurrences.

Definition 4.2.7

Let $N = (P, T, \mathcal{F})$ be a contact-free transition net with initial marking M_0 . Define $\mathcal{O}((N, M_0))$ inductively as follows. (We use $(-)_0$ and $(-)_1$ to denote the first and second co-ordinate of a pair.)

$$\begin{aligned} \text{Initially define } B_0 &= \{0\} \times M_0 \\ E_0 &= \emptyset \end{aligned}$$

with $F_0 = \#_0 = \emptyset$ and $co_0 = B_0^2$.

Then inductively define

$$\begin{aligned} B_{n+1} &= B_n \cup \{(\{e\}, p) \mid p \in P \ \& \ e \in E_n \ \& \ p \in (e)_1\} \\ E_{n+1} &= E_n \cup \{(\beta, t) \mid t \in T \ \& \ \beta \subseteq B_n \ \& \ (\beta)_1 = \bullet t \ \& \ (\forall b, b' \in \beta \ b \ co_n b')\} \end{aligned}$$

with relations $F_{n+1}, \#_{n+1}, co_{n+1}$ on $(B_{n+1} \cup E_{n+1})^2$ given by

$$\begin{aligned} x \ F_{n+1} \ x' &\text{ iff } x \in (x')_0 \\ x \ \#_{n+1} \ x' &\text{ iff } \exists e, e' \in E_{n+1} \ e \neq e' \ \& \ e \ F_{n+1}^* \ x \ \& \ e' \ F_{n+1}^* \ x' \ \& \\ &\quad (e)_0 \cap (e')_0 \neq \emptyset \end{aligned}$$

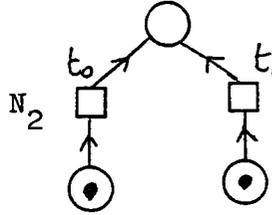
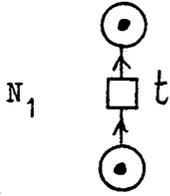
$$co_{n+1} = (B_{n+1} \cup E_{n+1})^2 \setminus (F_{n+1}^+ \cup (F_{n+1}^+)^{-1} \cup \#_{n+1}).$$

(NB. For A a set of pairs $(A)_0 = \{x \mid \exists y (x, y) \in A\}$ and similarly for $(-)_1$, i.e. $(-)_0$ and $(-)_1$ have been extended to sets).

Finally define $\mathcal{O}((N, M_0))$ to be the net (B, E, F) where $B = \bigcup_{n \in \omega} B_n$, $E = \bigcup_{n \in \omega} E_n$ and $F = \bigcup_{n \in \omega} F_n$.

We have used the contact-freeness of N where we assumed a transition could occur solely through its preconditions holding. The very simple transition nets below illustrate the point.

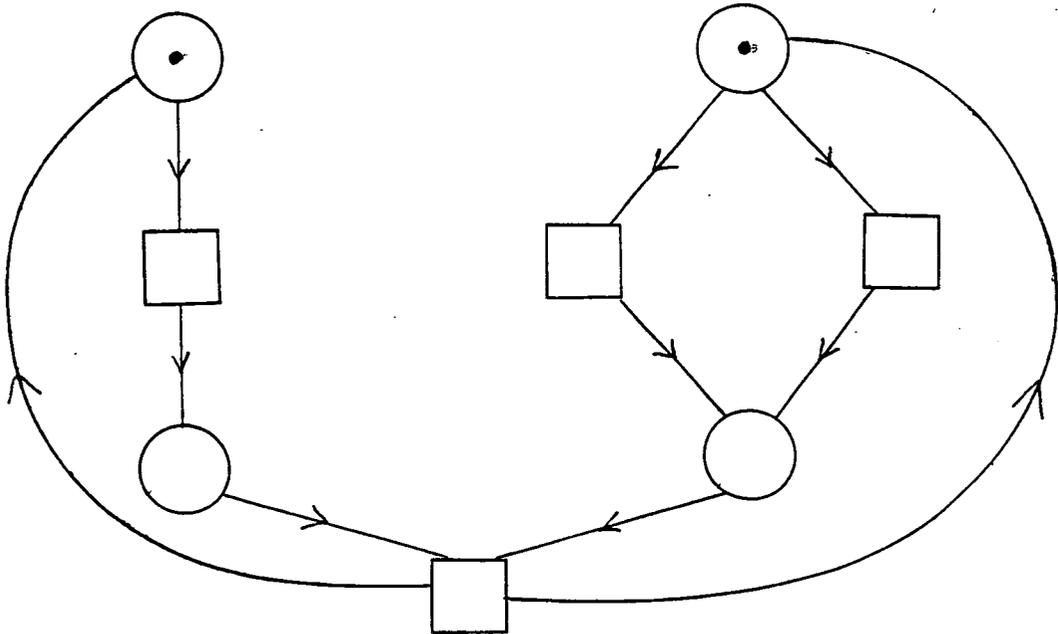
Example 4.2.8



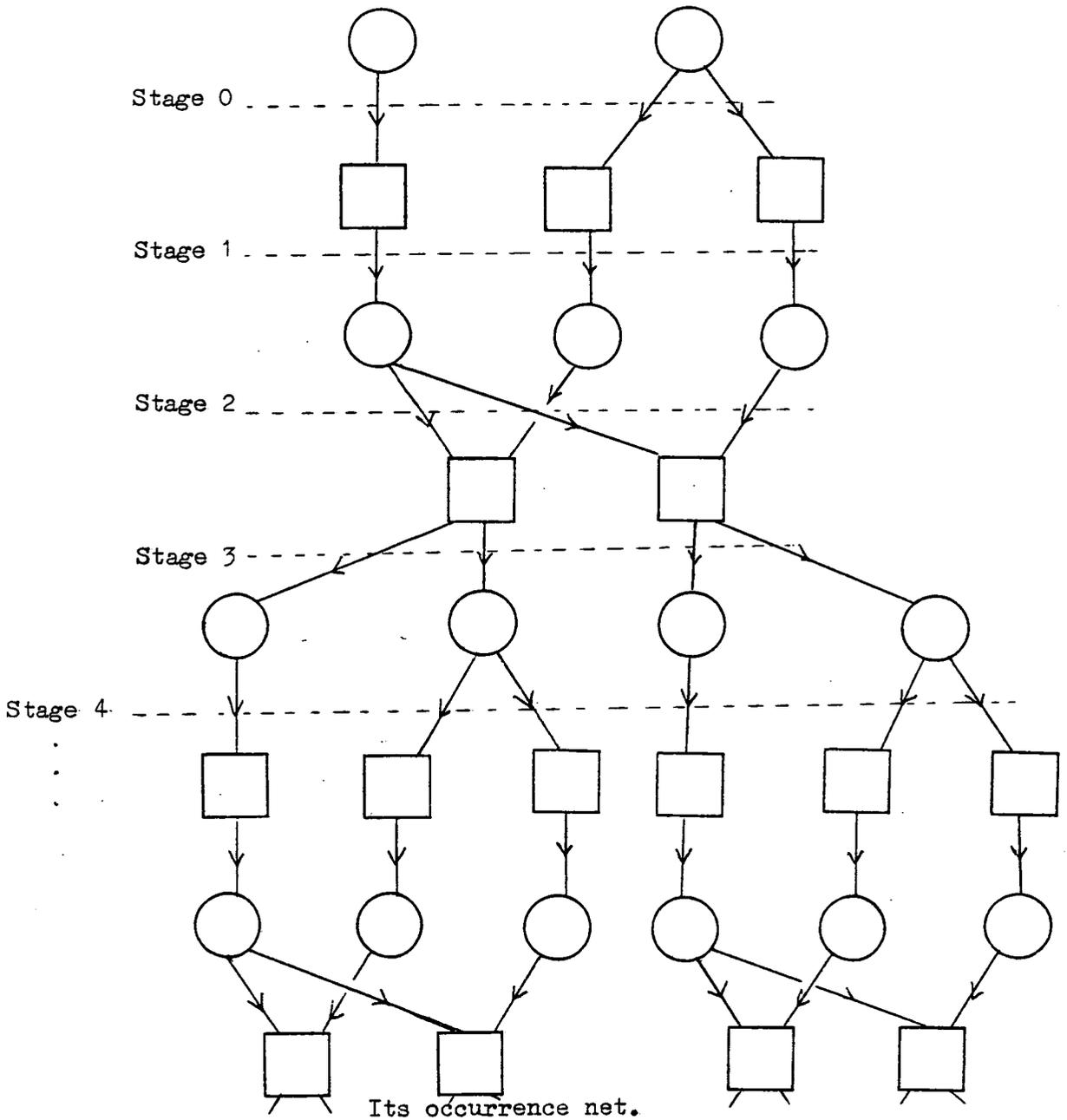
In N_1 there is contact immediately. It would be unreasonable to have an event occurrence for t firing. In N_2 contact can happen through backwards conflict; our construction would allow t_0 and t_1 to occur.

The next example illustrates a transition net with initial marking together with the occurrence net constructed as in 4.2.7. We have indicated what parts of the occurrence net have been grown by the n th stage of the inductive definition.

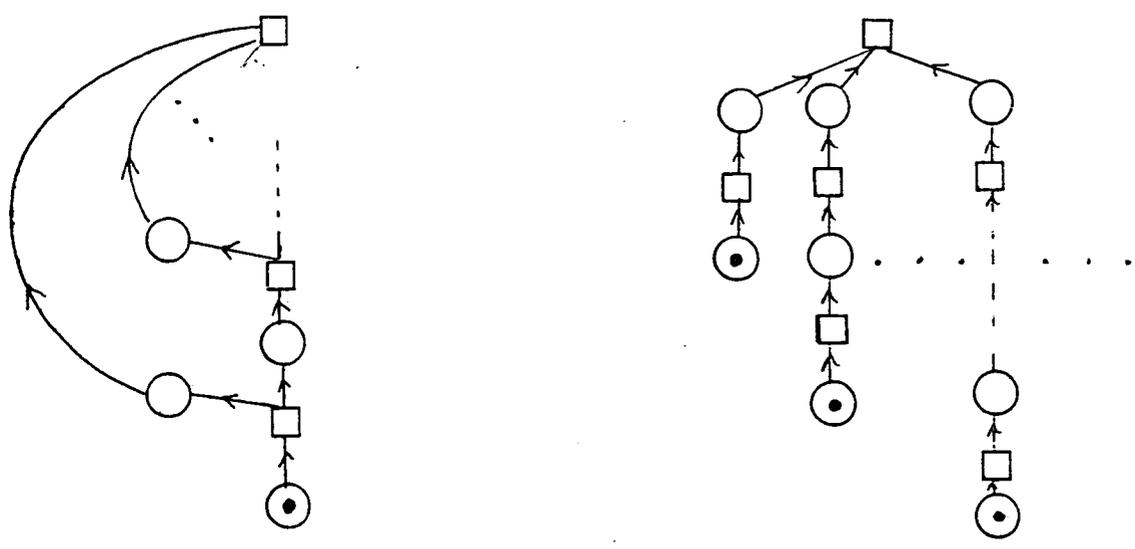
Example 4.2.9



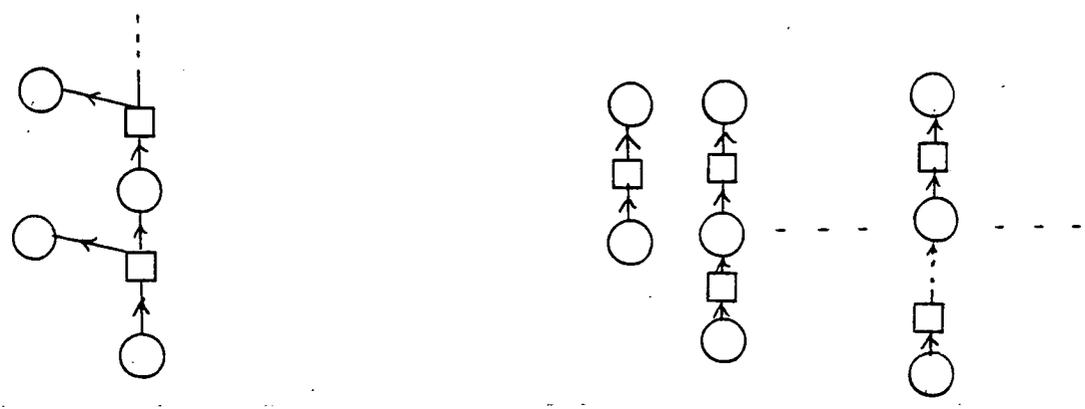
A net N with initial marking



In the inductive construction of the occurrence net associated with a transition net we have chosen to take the occurrence net as grown after ω iterations. It is noteworthy that the closure ordinal [Mos] associated with the inductive definition may well be greater than ω in general. For example the following transition nets with initial marking would give closure ordinal $\omega+1$.



According to definition 4.2.7 their occurrence nets would be



If one could play the token game very fast, so that the final events could occur, definition 4.2.7 would be inappropriate. (This kind of issue occurs in discussing the \aleph_0 -mind to lend intuition in recursion theory - see [Rog].) One could then accordingly continue the inductive construction up to the closure ordinal. Note this would require a more general definition of contact-free; ours is based on the reachable markings of chapter 2.

We remark that definition 4.2.7 is more general than that in [Nie] which was for finite transition nets; that approach would not produce a transition occurrence if it depended on an infinite set of transitions occurring concurrently. As in [Nie] the construction gives an occurrence net for which there is a natural folding to the original transition net. The proof of this proposition follows from the inductive construction.

Proposition 4.2.10

For any contact-free transition net N with initial marking M_0 , $\mathcal{O}((N, M_0))$ satisfies the axioms for occurrence nets. The map f ,

defined below, from $B \cup E$ to places and transitions of N is a folding:

$$f((0,p)) = f(\{\{e\},p\}) = p,$$

$$f((\beta,t)) = t.$$

Let us now see how conflict is handled in event structures and domains. Since elementary event structures were our "poorest" structures, it is not surprising that the only way of introducing conflict is by adding structure.

Definition 4.2.11

An event structure is a triple (E, \leq, \times) , where

E1. (E, \leq) is an elementary event structure,

E2. \times is a symmetrical and irreflexive relation in E , satisfying $\forall e_1, e_2, e_3 \in E: e_1 \geq e_2 \times e_3 \Rightarrow e_1 \times e_3$
 \times is called the conflict relation.

With these generalisations of causal nets and elementary event structures, the next two theorems provide straightforward generalisations of the mappings \mathcal{E} and \mathcal{N} the results of Theorems 4.1.2 and 4.1.3.

Theorem 4.2.12

Let $N = (B, E, F)$ be an occurrence net. Then

$$\mathcal{E}(N) =_{\text{def}} (E, F^* \upharpoonright E^2, \#_N \upharpoonright E^2) \text{ is an event structure.}$$

Proof The irreflexivity of $\#_N$ follows from N4'. Then E2 follows from the definition of $\#_N$. ■

Theorem 4.2.13

Let (E, \leq, \times) be an event structure. Then there is an occurrence net $\mathcal{N}(E)$ such that $E = \mathcal{E}(\mathcal{N}(E))$.

Proof Define the set $K(E)$ as follows:

$$K(E) =_{\text{def}} \{x \subseteq E \mid \forall e, e' \in x: e \neq e' \Rightarrow e \times e'\}.$$

The events of $\mathcal{N}(E)$ are obviously those of E , and the set of conditions is defined by

$$B = \{(e, x) \mid e \in E, x \in K(E) \text{ and } \forall e' \in x \ e < e'\} \cup \{(0, x) \mid x \in K(E)\}$$

Finally, the F relation is defined as

$$F = \{ \{(e, x), e'\} \mid (e, x) \in B, e' \in x \} \cup \\ \{ \{(o, x), e'\} \mid (o, x) \in B, e' \in x \} \cup \\ \{ \{(e, (e, x)) \mid (e, x) \in B \}.$$

It follows that $\mathcal{N}(E)$ is a well-defined occurrence net for which $\#_1 = \#$ and restricts to give \otimes on events, and hence $\mathcal{E}(\mathcal{N}(E)) = E$. ■

This construction of $\mathcal{N}(E)$ may seem more unnecessarily complicated than the one from the proof of Theorem 4.1.3. Obviously, many simpler ones would do; however, we have again chosen a "maximal" construction, in the sense that any condition in any occurrence net N for which conditions are extensional and for which $\mathcal{E}(N) = E$ has a representative in $\mathcal{N}(E)$ (which means that our treatment of conditions in elementary event structures discussed in the previous section carries over to event structures).

Things get a bit more interesting when we move on to our lattice structures and generalisations of the mappings \mathcal{L} and \mathcal{P} . Intuitively, an event structure represents a class of courses of computation (processes according to Petri) where $e \otimes e'$ means that e and e' never occur in the same course. So, not all left-closed subsets of an event structure make sense as information points. Only the conflict free left-closed subsets can be the sets of occurrences at some stage of an associated course of computation.

Definition 4.2.14

Let $E = (E, \leq, \otimes)$ be an event structure, and let x be a subset of E . Then x is conflict free iff

$$\forall e, e' \in x \quad \neg (e \otimes e').$$

Our idea about the ordering of information points is still the same, though.

Definition 4.2.15

Let $E = (E, \leq, \otimes)$ be an event structure. Then $\mathcal{L}(E)$ is the partial order of left-closed (w.r.t. \leq) and conflict free subsets of E , ordered by inclusion. We shall sometimes call x in $\mathcal{L}(E)$ a configuration of E .

What about our characterisation of the structures $\mathcal{L}_0(E)$? Obviously, we do not any longer get complete lattices. Two points will be incompatible (have no upper bound) iff their union (as sets of events) contain conflict. But any compatible set of points will have a lub (their union), so the structures will be consistently complete. For a characterisation we need the even stronger condition of coherence (see 3.1).

Theorem 4.2.16

Let (E, \leq, \times) be an event structure. Then $\mathcal{L}_0(E)$ is a prime algebraic coherent partial order. Its complete primes are those elements of the form $[e] = \{e' \in E \mid e' \leq e\}$.

Proof Let $X \subseteq \mathcal{L}_0(E)$ be pairwise consistent. Then $\bigcup X$ is conflict free, and so $\bigsqcup X = \bigcup X$, showing that $\mathcal{L}_0(E)$ is coherent.

The rest of the proof proceeds as in the proof of Theorem 4.1.9, noting that all elements of the form $[e]$ are conflict free from E^2 , and that for any x in $\mathcal{L}_0(E)$ the set $\{[e] \mid e \in x\}$ is pairwise compatible. ■

From this theorem we see how to generalise the mapping \mathcal{P} .

Definition 4.2.17

Let (D, \sqsubseteq) be a prime algebraic coherent partial order. Then $\mathcal{P}(D)$ is defined as the event structure $(\text{Pr}(D), \leq, \times)$, where \leq is \sqsubseteq restricted to $\text{Pr}(D)$, and for all $e, e' \in \text{Pr}(D)$: $e \times e'$ iff e and e' are incompatible in D .

It is easy to see that $\mathcal{P}(D)$ is indeed an event structure, and we are now ready to prove the equivalence between event structures and prime algebraic coherent partial orders corresponding to Theorem 4.1.12. An isomorphism between two event structures is naturally any one to one and onto mapping, which respects and reflects both causality and conflict.

Theorem 4.2.18

Let (E, \leq, \times) be an event structure, then $E \cong \mathcal{P}(\mathcal{L}_0(E))$. Similarly let (D, \sqsubseteq) be any prime algebraic coherent partial order, then $D \cong \mathcal{L}_0(\mathcal{P}(D))$.

Proof Define $\psi: E \rightarrow \mathcal{P}(\mathcal{L}_0(E))$ by $\psi(e) = [e]$. It follows

along the lines of the proof of Theorem 4.1.12 that ψ is an isomorphism with respect to \leq and the corresponding relation in $\mathcal{P}(\mathcal{L}(E))$. Furthermore, ψ is easily seen to respect and reflect the conflict relation.

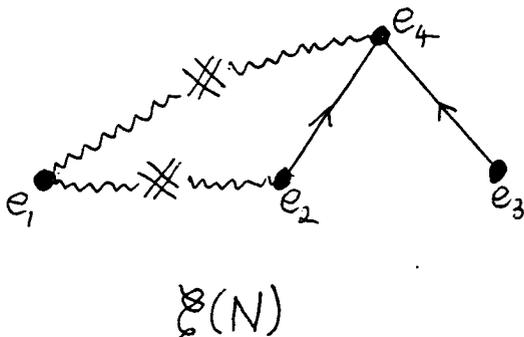
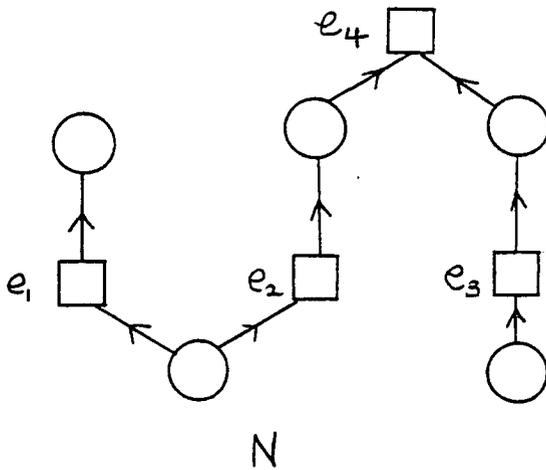
The mapping π as defined in Definition 4.1.11 is known to be an order monic from D to $\mathcal{L}((\text{Pr}(D), \subseteq \uparrow \text{Pr}(D), \emptyset))$ (from Lemma 4.1.11). From definition $\mathcal{L}(\mathcal{P}(D))$ is a subordering of $\mathcal{L}((\text{Pr}(D), \subseteq \uparrow \text{Pr}(D), \emptyset))$ so all we have to prove is that the range of π is equal to the set of elements of $\mathcal{L}(\mathcal{P}(D))$, i.e. for every left-closed set, X , of complete primes of D :

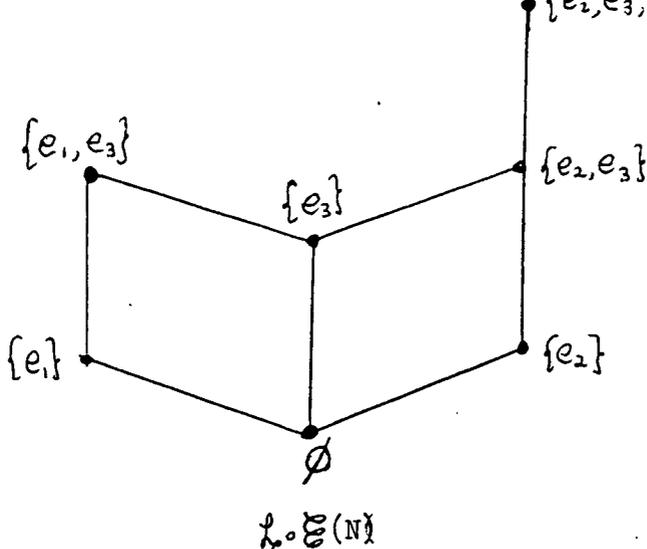
$$\exists d \in D \quad \pi(d) = X \text{ iff } \forall p, p' \in X \quad p \text{ and } p' \text{ are compatible.}$$

The "only if" part is trivial. Assume X satisfies the right hand side assumption. Coherence of D implies the existence of $\bigsqcup_D X$, and it follows that $\pi(\bigsqcup_D X) = X$ (just like in the proof of Theorem 4.1.12). ■

In Example 4.2.19 an occurrence net N is pictured with its associated event structure $\mathcal{E}(N)$ and the coherent prime algebraic partial order $\mathcal{L}(\mathcal{E}(N))$.

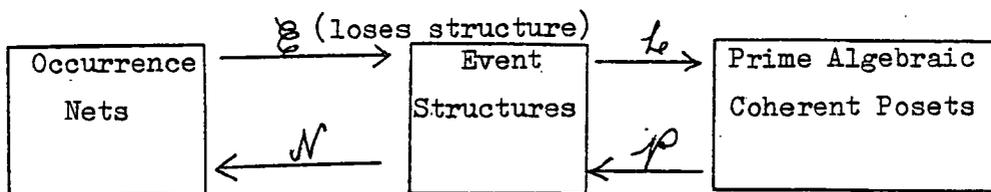
Example 4.2.19





Theorem 4.2.18 has an intuitive interpretation. For an event structure E the domain $\mathcal{L}(E)$ may be thought of as a set of possible courses of computation. The theorem says that two event structures are isomorphic iff the structure of the courses of computation they determine are isomorphic. Given an occurrence net N an element x of $\mathcal{L}(E(N))$ determines a causal subnet of N namely the net consisting of events x , conditions $\{b \mid \exists e \in x \ b \in \bullet e \cup e^\bullet\}$ with F -relation induced by N . Recall it is causal nets which Petri chooses to represent courses of computation. As a contact-free transition net with initial marking determines an occurrence net it also determines a class of causal nets.

So, we have now established a complete generalisation of the picture from the previous section:



All considerations about translation of events and conditions work as in there. Formally, Proposition 4.1.15 and Theorem 4.1.16 hold for prime algebraic coherent partial orders, and a straightforward version of Lemma 4.1.17 can be proved.

Restricting ourselves to these relations on events, the following should now be obvious to the reader.

	Occurrence Nets $N = (B, E, F)$	Event Structures $(E, \leq, \#)$	Prime Algebraic Coherent Posets (D, \sqsubseteq)
Causality	$F^+ \uparrow E^2$	$<$	$\sqsubseteq \uparrow \text{Pr}(D)^2$
Conflict	$\#_N \uparrow E^2$	$\#$	$\uparrow \uparrow \text{Pr}(D)^2$
Concurrency	$E^2 \setminus (F^+ \cup (F^+)^{-1} \cup \#_N)$	$E^2 \setminus (< \cup > \cup \#)$	$\text{Pr}(D)^2 \setminus (\sqsubseteq \cup \supseteq \cup \uparrow)$

Finally, let us see what these relations look like in terms of prime intervals of partial orders.

Definition 4.2.20

Let (D, \sqsubseteq) be a prime algebraic coherent partial order. The relation \rightarrow ("may occur before") on $\text{Pr}(D)$ is defined as follows: $p_1 \rightarrow p_2$ iff there exist prime intervals of P , $[x_1, x'_1], [x_2, x'_2]$, such that $\text{pr}([x_1, x'_1]) = p_1$, $\text{pr}([x_2, x'_2]) = p_2$ and $x'_1 \sqsubseteq x_2$. The complement of \rightarrow is denoted $\not\rightarrow$.

Proposition 4.2.21

Let (D, \sqsubseteq) be a prime algebraic coherent partial order, and let $p_1, p_2 \in \text{Pr}(D)$. Then

$$p_1 \not\sqsubseteq p_2 \text{ iff } (p_1 \rightarrow p_2) \& (p_2 \not\rightarrow p_1),$$

$$p_1 \uparrow p_2 \text{ iff } (p_1 \not\rightarrow p_2) \& (p_2 \not\rightarrow p_1)$$

and hence p_1 and p_2 are concurrent iff $(p_1 \rightarrow p_2) \& (p_2 \rightarrow p_1)$.

Chapter 5. States and observable states

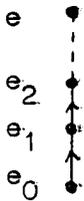
In this chapter we look at the key idea of states of an occurrence net in detail using event structures as an intermediate notion. We shall look at these initially later *relate them to* occurrence nets. We introduce two types of state of an event structure, observable states and states in general. Observable states correspond to states which may be observed in finite time whereas states may require unbounded time. Using the idea of an observer we arrive at definitions of these two notions of state consistent, it seems, with the net-theoretic intuitions. (Observable cases of an occurrence net will be determined by observable states of the associated event structure. The reachable markings of a transition net are the image of the observable cases of its occurrence net unfolding.) Throughout this chapter we shall assume the computations have a fixed initial state at which they start (see the initiality restriction). We shall relax this in chapter 7. We shall also assume that the extent of the holding of a condition lasts at least unit time (see the discreteness restriction). The technical machinery we develop on states leads to a batch of results. One is a more concrete appraisal of K-density. Unfortunately we shall disagree with it though give some results consistent with its spirit (as Petri himself has agreed in a letter). We shall also investigate the assumption of finite width which is appropriate to descriptions of computations involving only finitely many agents at any finite time. The property of finite width will depend on a finitely-branching property. However we shall reserve the term "finitely-branching" for event structures which possess only finite non-determinism in a sense to be made clear (5.3). In 5.5 we show how the notion of confusion translates over to event structures and domains, establishing a connection with concrete domains.

5.1 Observers, states and observable states

In chapter 2 we gave several examples of a transition net modelling a computation or datatype (itself an extreme form of computation in which no assumption is made about whether an event can have concession forever or not). In chapter 4 we showed how such a transition net could be unfolded into an occurrence net to which in turn we could associate an event structure. These then become

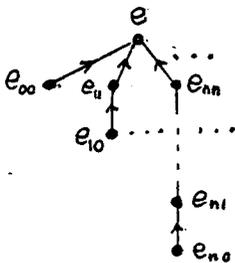
descriptions of computations. In more detail an event structure (E, \leq, \bowtie) is an abstract description of a computation which picks out certain event occurrences related to the computation and represents causality and conflict on E through the relations \leq and \bowtie . The concurrency relation and the relation $\bowtie \cup 1$ are not the identity in general; this reflects, respectively, the indeterminacy of the relative speeds in the various subprocesses and the choice of course that a run of the computation will follow. Having described a computation by an event structure, E , it is natural to associate information about a particular course of computation with an element of $\mathcal{L}(E)$. However it is not so clear whether every element of $\mathcal{L}(E)$ corresponds to a state that the computation may reach in finite or unbounded time. Informally, we take an observable state *to be* an element C of $\mathcal{L}(E)$ for which there is a finite time in the course of a computation for which events in C are precisely those observed by that time. A state is defined similarly but here the observation time is allowed to be unbounded. We give some examples to illustrate this.

Example 5.1.1



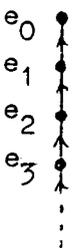
Here E_1 is the (elementary) event structure consisting of an unbounded chain $e_0 < e_1 < e_2 < \dots$ below an event e .

Example 5.1.2



Here E_2 is the (elementary) event structure consisting of e with chains $e_{n0} < e_{n1} < \dots < e_{nn}$ of unbounded length leading up to it.

Example 5.1.3



Here E_3 is the (elementary) event structure consisting of an infinite chain $e_0 > e_1 > e_2 > \dots$.

Consider computations described by E_1 , E_2 and E_3 . (Note that they are the event structures associated with the causal nets of examples 2.4.5, 2.4.8 and 2.4.6^{or 7} respectively.) First let us suppose, that there is a uniform lower bound on the extent of time which passes between the occurrences of e and e' if $e < e'$. Thinking of occurrence nets which induce E_1 , E_2 and E_3 , this is equivalent to assuming a uniform lower bound on the extend of the holdings of the conditions. Then as the events e in E_1 and E_2 and any event e_n of E_3 dominate chains of unbounded length, if the computations always start with no events having occurred $e \in E_1$, $e \in E_2$ and $e_n \in E_3$ can never occur. Thus for such computations $[e] \in \mathcal{L}(E_1)$, $[e] \in \mathcal{L}(E_2)$ and $[e_n] \in \mathcal{L}(E_3)$ are not states. If we keep the first assumption for computations but no longer insist that they start at some definite time the events e of E_2 and e_n of E_3 could now occur. (We shall look at this possibility in detail in a later chapter.) If we drop our first assumption as well then for instance example 5.1.1 is naturally associated with Zeno's paradox and the event e to Achilles catching up with the tortoise (a very peculiar computation). Thus depending on what assumptions we make on the computation and the event structure description of it the left-closed conflict-free subsets may or may not correspond to states. Also without extra assumptions the observable states are not derivable from the event structure alone.

In making the last statement we diverge from the approach of conventional net theory where we understand the observable states of a causal net are identified with its cases. (See section 2.4 in which it is shown that the K-density axiom is natural once this commitment is made.) With this interpretation of a case as an observable state, insisting on K-density for a causal net guarantees every observable state determines a unique point in every sequential process. We shall not feel bound by K-density but note we expect a revised version of it to hold in a causal net where we restrict cases to observable cases (viz. those determined by observable states of the associated event structure). We establish this in section 5.4.

Referring back to the examples and the ensuing discussion we shall make two restrictions on the nature of the computations and our event structure descriptions of them. With these restrictions we shall be able to identify states with left-closed conflict-free

subsets. We insist that if in an event structure E , for events e and e' , $e < e'$ then their occurrence must be separated by at least unit time. (We call this the discreteness restriction.) As pointed out above this is equivalent to assuming that the extents in time of the holdings of conditions in an occurrence net inducing the event structure have a uniform lower bound. Thus we avoid the problems of dense event structures such as the rationals and the reals. We will also assume there is a state of null information, when no events have occurred from which the computation starts (we call this the initiality restriction). In chapter 2 we defined one notion of what the "reachable markings" were in playing the token game. (The issue of how fast one could play it arose in defining the occurrence net unfolding of a transition net.) The initiality restriction accords with transition nets having initial markings and the discreteness restriction will imply a formulation of reachable which agrees with 2.2.8, probably the most intuitive.

We now formalise the intuitions above. We first define the concept of an observer which corresponds to a particular (complete) run or history of a computation where each event's occurrence is recorded together with the time at which it occurred. Time will be discrete starting at zero and we use the symbol " ∞ " to "record" events which never occur according to a particular observer. An event may never occur either through being in conflict with a preobserved event, through the computation diverging before the event, or simply through the event being "too far" from the starting state as in example 5.1.1. Time will be represented by $\omega \cup \{\infty\}$ ordered as usual.

Definition 5.1.4

Let E be the event structure (E, \leq, \times) . An observer for E is a map $O: E \rightarrow \omega \cup \{\infty\}$ such that

1. $e < e' \ \& \ O(e) < \infty \Rightarrow O(e) < O(e')$
2. $e < e' \ \& \ O(e) = \infty \Rightarrow O(e') = \infty$
3. $O(e) < \infty \ \& \ O(e') < \infty \Rightarrow \neg(e \times e')$

We denote the set of observers for E by $Ob(E)$.

The above paragraph explains clauses 2 and 3 in the definition and clause 1 formalises our first restriction on computations. Note that the above definition allows computations to diverge at

any stage; no events are obliged to lose concession eventually. ~~Such~~ extra assumptions would restrict the class of observers and the states though not the observable states. We have already motivated the following definition of the latter two notions.

Definition 5.1.5

Suppose (E, \leq, \otimes) is an event structure and $C \subseteq E$. Say C is an observable state of E iff

$$\exists 0 \in \text{Ob}(E) \exists t \in \omega \ C = \{e \in E \mid O(e) < t\}.$$

Also say C is a state of E iff

$$\exists 0 \in \text{Ob}(E) \exists t \in \omega \cup \{\infty\} \ C = \{e \in E \mid O(e) < t\}.$$

We write $\mathcal{O}(E)$ and $\mathcal{S}(E)$ for the observable states and states respectively, ordered by inclusion.

From these definitions it is obvious that

Lemma 5.1.6

For E an event structure,

$$\emptyset \in \mathcal{O}(E) \subseteq \mathcal{S}(E) \subseteq \mathcal{L}(E).$$

The next section provides a simple characterisation of $\mathcal{O}(E)$ and $\mathcal{S}(E)$.

5.2 Distance measures on events and states

In this section we define a distance measure on events and then use it to define an integer metric on left-closed conflict free subsets - strictly speaking it is not quite an integer metric as it is possible for two states to be infinitely far apart. The ideas are simple. The distance measure $\Delta(e, e')$ between two events e and e' of event structure E is the supremum of the lengths of chains between e and e' ; it represents the minimum time possible between the observation of e and e' . The distance $d(C_1, C_2)$ between two elements of $\mathcal{L}(E)$ is the supremum of $\Delta(e, e')$ for e and e' in $(C_1 + C_2)$ the symmetric difference of C_1 and C_2 . First we define the distance measure on events. The set $\omega \cup \{\infty\}$ is ordered as usual.

Definition 5.2.1

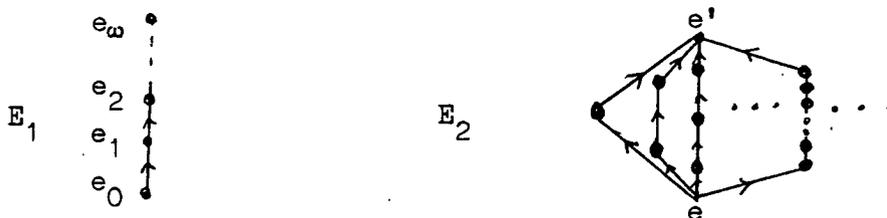
Let (E, \leq, \otimes) be an event structure. Define

$\Delta: E^2 \rightarrow \omega \cup \{\infty\}$ by

$$\Delta(e, e') = \text{Sup}\{n \mid \exists e_0, \dots, e_n \in E \ e_0 < \dots < e_n \ \& \ ((e_0 = e \ \& \ e_n = e') \text{ or } (e_0 = e' \ \& \ e_n = e))\}$$

Note Δ may be infinite as occurs in the next two examples.

Example 5.2.2



In E_1 there is an infinite chain between e_0 and e_ω so

$\Delta(e_0, e_\omega) = \infty$. In E_2 there are chains of unbounded length between e and e' so $\Delta(e, e') = \infty$.

We note that Δ is symmetric and that $\Delta(e, e') = 0$ iff $e = e'$ or e and e' are \leq -incomparable. Suppose we have three events $e \leq e' \leq e''$. Then in general there may be more chains from e to e'' than go through e' . These remarks account for the following lemma.

Lemma 5.2.3

For E and Δ as in definition 5.2.1 we have:

1. $\Delta(e, e') = \Delta(e', e)$
2. $\Delta(e, e') = 0$ iff $e = e'$ or e and e' are \leq -incomparable.
3. For $e \leq e' \leq e''$,

$$\Delta(e, e') + \Delta(e', e'') \leq \Delta(e, e'').$$

Notice that 3. is the "wrong way" triangle inequality. We remark that such measures occur in cosmology but there the analogue of \leq means "may be a cause of" (see exercises 8.3.1, 8.3.9 in [Sac]).

From Δ on E we obtain a metric on $\mathcal{L}(E)$ the left-closed consistent subsets. (Strictly speaking d is not quite a metric as it may be infinite.)

Definition 5.2.4

For E and Δ as in definition 5.2.1 we define

$d: \mathcal{L}(E)^2 \rightarrow \omega \cup \{\infty\}$ by

$$d(C_1, C_2) = \text{Sup}\{\Delta(e, e') + 1 \mid e, e' \in (C_1 + C_2)\}$$

We say for $C_1, C_2 \in \mathcal{L}(E)$ that they are reachable from each other iff $d(C_1, C_2) < \infty$.

The latter concept of reachability allows two incompatible conflict-free left-closed subsets to be reachable from each other. This may seem unusual. We shall relate it to the perhaps more standard idea of forwards reachability after the next lemma detailing the properties of d .

Lemma 5.2.5

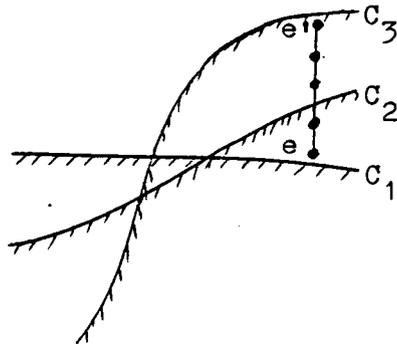
For d as defined in 5.2.4, if $C_1, C_2, C_3 \in \mathcal{L}(E)$:

1. $d(C_1, C_2) = 0 \iff C_1 = C_2$
2. $d(C_1, C_2) = d(C_2, C_1)$
3. $d(C_1, C_2) + d(C_2, C_3) \geq d(C_1, C_3)$
4. $d(C_1, C_2) = \text{Sup}\{d(C_1 \cap C_2, C_1), d(C_1 \cap C_2, C_2)\}$
5. $C_1 \subseteq C_2 \subseteq C_3 \implies d(C_1, C_2) \leq d(C_1, C_3)$.

Proof Use the fact that C_1, C_2, C_3 are left-closed.

For 3,

1. and 2. are obvious from the definition of d . If $C_1 = C_3$ the result is obvious so suppose w.l.o.g. there is chain between e and e' , with $e \leq e'$, in $C_3 \setminus C_1$. Then the chain splits into two chains one possibly null in $C_2 \setminus C_1$, the other possibly null in $C_3 \setminus C_2$. Pictorially we have:



The two parts make a contribution of at least the length of the chain to $d(C_1, C_2) + d(C_2, C_3)$.

4. Chains in $(C_1 \setminus C_2) \cup (C_2 \setminus C_1)$ are either in $C_1 \setminus (C_1 \cap C_2)$ or in $C_2 \setminus (C_1 \cap C_2)$.

5. Clear. ■

Now we can relate our relation of reachability given in 5.2.4

to forwards reachability. Note that the one-step-forward reachability relation below corresponds closely to the relations $[->$ and \rightarrow_1 of 2.2.

Proposition 5.2.6

Let E be an event structure. For C_1, C_2 in $\mathcal{L}(E)$ define one-step-forward reachability by

$$C_1 \vdash_1 C_2 \text{ iff } C_1 \subseteq C_2 \ \& \ \forall e \in C_2 \setminus C_1 \ \leftarrow^{-1} \{e\} \subseteq C_1.$$

Then define the forward reachability relation as the transitive closure of \vdash_1 .

Suppose C_1, C_2 are in $\mathcal{L}_e(E)$. We have

1. $C_1 \vdash_1^* C_2$ iff $C_1 \subseteq C_2$ & $d(C_1, C_2) < \infty$.
2. The reachability relation of 5.2.4 is the least equivalence relation extending \vdash_1 . In fact $d(C_1, C_2) < \infty$ iff $C_1 \wedge C_2 \vdash_1^* C_1$ & $C_1 \wedge C_2 \vdash_1^* C_2$.

Proof

1. Clear from the definitions.
2. This follows from property 4. in 5.2.5. ■

We use the following definition in characterising states.

Definition 5.2.7

For d and event structure (E, \leq, \times) as above and $e \in E$, say e has finite depth in E iff $d(\emptyset, [e]) < \infty$.

It is obvious that:

Lemma 5.2.8

If e has finite depth in event structure E and $e' \leq e$ then e' has finite depth in E .

We could have defined finite depth by introducing a fictitious event i below all events in the event structure E , defining Δ as above on the amended event structure, and then said an event e of E had finite depth iff $\Delta(i, e) < \infty$.

The characterisations of $\mathcal{S}(E)$ and $\mathcal{Q}(E)$ for event structure E now follow:

Theorem 5.2.9

Suppose E is an event structure with metric d on $\mathcal{L}(E)$ as defined

in 5.2.4. Then for $C \in \mathcal{L}(E)$

1. $C \in \mathcal{S}(E)$ iff $\forall e \in C$ e has finite depth.
2. $C \in \mathcal{O}(E)$ iff $d(\emptyset, C) < \infty$.

Proof

1. " \Rightarrow " Suppose $C \in \mathcal{S}(E)$. Then each event in C is observed in finite time and thus by the definition of an observer is of finite depth.

" \Leftarrow " Define the observer by $O(e) = d(\emptyset, [e])$ if $e \in C$, ∞ otherwise.

2. As for 1. but this time we have a uniform bound on the depths of the events. ■

Corollary 5.2.10

For an event structure E ,

1. $\mathcal{S}(E) = \mathcal{L}(E)$ iff for all events e are of finite depth.
2. $\mathcal{O}(E)$ is closed under intersections and finite consistent unions.

If an event is not of finite depth it can never be observed. Consequently the states only involve events of finite depth. Thus it is natural to restrict ourselves to event structures in which all events are of finite depth. For example this excludes the event structures E_1 , E_2 and E_3 of examples 5.1.1, 5.1.2 and 5.1.3 respectively, even though $\mathcal{N}(E_2)$ is K -dense.

Definition 5.2.11

An event structure E is of finite depth iff every event of E has finite depth.

Theorem 5.2.12

If (E, \leq, \otimes) is an event structure the following are equivalent:

1. E is of finite depth.
2. $\mathcal{S}(E) = \mathcal{L}(E)$
3. $\forall e \in E \exists O \in \text{Ob}(E) O(e) \in \omega$
4. $\forall A \subseteq E (\forall a_1, a_2 \in A \neg (a_1 \otimes a_2)) \Rightarrow \exists O \in \text{Ob}(E) A \subseteq O^{-1}\omega$.

Proof

Let $E = (E, \leq, \otimes)$ be an event structure.

1. \Rightarrow 2. by theorem 5.2.9 part 1. characterising states
 2. \Rightarrow 3. Assuming 2. we have $[e] \in \mathcal{S}(E)$ for any event e . Thus by the definition of state $\exists O \in \text{Ob}(E) O(e) \in \omega$.
 3. \Rightarrow 4. Supposing 3. gives that every event e has finite depth thus we may define the required observer O by $O(e) = d(\emptyset, [e])$ if $e \in [A]$.
 $= \infty$ otherwise.
 4. \Rightarrow 1. as $\{e\}$ is certainly a conflict-free subset of E so there is an observer seeing e , giving that e has finite depth. ■

Thus if an event structure E is of finite depth $\mathcal{S}(E) = \mathcal{L}(E)$ so, by the results of the last chapter, we can recover E , to within isomorphism from $\mathcal{S}(E)$. It can also be recovered directly from the observers for E . Precisely:

Theorem 5.2.13

If (E, \leq, \otimes) is an event structure of finite depth then:

$$\leq = \bigcap_{O \in \text{Ob}(E)} \leq_O \quad \text{and} \quad \otimes = \bigcap_{O \in \text{Ob}(E)} \otimes_O$$

where

$$e \leq_O e' \Leftrightarrow O(e') < \infty \Rightarrow O(e) \leq O(e')$$

$$e \otimes_O e' \Leftrightarrow (O(e) \neq \infty \Leftrightarrow O(e') = \infty).$$

Proof

Obviously by the definition of an observer $\leq \subseteq \bigcap_O \leq_O$ and $\otimes \subseteq \bigcap_O \otimes_O$ so we require

$$\neg(e \leq e') \Rightarrow \exists O \in \text{Ob}(E) \neg(e \leq_O e')$$

and $\neg(e \otimes e') \Rightarrow \exists O \in \text{Ob}(E) \neg(e \otimes_O e')$ respectively.

The latter follows from theorem 5.2.12 part 4. For the former, as E is of finite depth, take $O \in \text{Ob}(E)$ such that $O(e') \in \omega$. If $e \leq_O e'$ (i.e. O is unsuitable) take O' defined by

$$O'(e) = O(e) \text{ if } e \leq e'$$

$$O'(e) = O(e) + 1 \text{ otherwise.}$$

Then O' is the required observer. ■

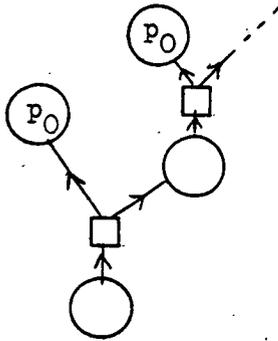
5.3 Event structures with finite width and finite branching

So far we still allow computations of a very general nature. For instance we allow an infinite number of concurrent events to form an occurrence net or event structure. For real computational

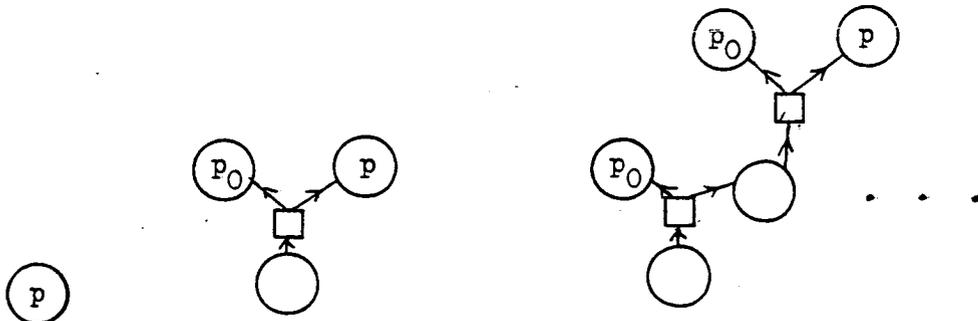
processes at normal levels of abstraction this seems unlikely. One would expect that an infinite Milner net for example would have to be grown, perhaps by a recursive definition, over an infinite stretch of time. In such a Milner net, in any finite time only a finite number of events (including communication and possibly "births" of agents) would occur. The next example shows this a little more formally.

Example 5.3.1

A Milner net might be defined recursively by $p = p_0 \parallel p$ the parallel combination of p_0 with p where p_0 is some fixed net. Imagine the behaviour of p_0 described by an occurrence net abbreviated as (p_0) and the behaviour of p by an occurrence net abbreviated as (p) . One implementation of the recursive definition of p would give rise to this occurrence net.



Here each event drawn represents the action of expanding the net further according to a single application of the recursive definition. We can draw successive expansions of the net like this:



The recursive definition preserves the fact that at any finite time only finitely many events can have occurred.

The above discussion motivates the next definition of finite width. However note that a more detailed analysis of what class of computations to allow would perhaps yield a more restrictive definition (see example 5.3.19).

Definition 5.3.2

Let $E = (E, \leq, \times)$ be an event structure of finite depth. Then E is of finite width iff all observable states of E are finite. Note that we presuppose E to be of finite depth. This is because such event structures are natural for our definition of observable state expressing those events which may occur in finite time. Such event structures will arise from the occurrence net unfolding of a finite transition net.

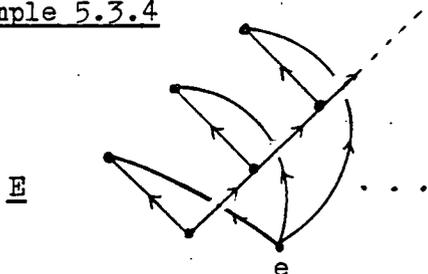
If E is an event structure of finite depth then for any event e in E we have $[e]$ is an observable state. Thus for finite width event structures $[e]$ must be finite. Also considering a total observer for an elementary finite width event structure E we have that E is a countable union of finite sets and is thus countable.

Lemma 5.3.3

Let E be an event structure of finite width. Then for all e in E we have $[e]$ is finite. If E is elementary too then E is *countable*.

Thus the left-closed consistent subsets of a finite width event structure satisfy axiom F of chapter 3. The converse does not hold however; the event structure consisting of an infinite set of \leq -incomparable events with null conflict relation is not of finite width and yet gives a domain satisfying axiom F.

Thinking of characterising finite width some finite-branching property springs to mind. Perhaps the most obvious one is that $\{e' \in E \mid e \prec e'\}$ is finite for all events e , where we have used \prec for the covering relation in E . This is incorrect however as the following example shows.

Example 5.3.4

The above example of an elementary event structure, E , is of finite width yet we do have $\{e' \in E \mid e \prec e'\}$ infinite. Thus imposing

$\forall e \in E \{e' \in E \mid e \prec e'\}$ is finite

is too strong even restricted to elementary event structures. The correct finite-branching property follows. First we have some notation generalising that in 4.1.9.

Definition 5.3.5

For E an event structure and $A \subseteq E$ define $[A]$ to be the left-closure of A i.e.

$$[A] = \{e \in E \mid \exists a \in A \ e \leq a\}.$$

Definition 5.3.6

For $E = (E, \leq, \times)$ an event structure and $A \subseteq E$ we define the concession of A by

$$\text{conc}(A) = \{e \in E \mid e \notin [A] \ \& \ \prec^{-1}\{e\} \subseteq [A]\}$$

and the immediate futures of A by

$$\text{IF}(A) = \{B \subseteq E \mid B \text{ is } \subseteq\text{-maximal s.t. } B \text{ is a conflict-free subset of } \text{conc}(A)\}.$$

Then E is said to be finitely-enabling iff

$$\forall A \subseteq E \mid A \mid \leq \infty \Rightarrow \forall B \in \text{IF}(A) \mid B \mid < \infty.$$

We avoid "finite-branching" which is more appropriate for finite non-determinism. We then have:

Theorem 5.3.7

For E an event structure (E, \leq, \times) of finite depth, E is of finite width iff E is finitely-enabling.

Proof

" \Rightarrow " Suppose E is of finite depth and finite width and that $A \subseteq E$ and $|A| < \infty$. Take $B \in \text{IF}(A)$. Define $C = A \cap [B]$. We have $B \in \text{IF}(C)$. As C is conflict-free and $|C| < \infty$ using finite depth and 5.2.9 part 2 we get $[C] \in \mathcal{O}_f(E)$. Now $d(\emptyset, [B]) \leq d(\emptyset, [C]) + 1 < \infty$. Thus by 5.2.9 again $[B] \in \mathcal{O}_f(E)$. As E has finite width this means $|B| < \infty$.

" \Leftarrow " Suppose E is finitely-enabling. Then one shows by induction on n that the following induction hypothesis holds:

$$\forall C \in \mathcal{O}_f(E) \ d(\emptyset, C) \leq n \Rightarrow |C| < \infty. \blacksquare$$

Corollary 5.3.8

For E an elementary event structure (E, \leq) of finite depth, E is of finite width iff $\forall A \subseteq E \ |A| < \infty \Rightarrow |\text{conc}(A)| < \infty$.

Proof

Simply note for elementary event structures we have

$$\text{IF}(A) = \{\text{conc}(A)\} \text{ for } A \subseteq E. \blacksquare$$

In general the observable states of an event structure E will not correspond to the isolated elements of $(\mathcal{S}(E), \subseteq)$ (written $\mathcal{S}(E)^0$). However:

Theorem 5.3.9

Let $E = (E, \leq, \otimes)$ be an event structure of finite depth. Then

$$\mathcal{S}(E)^0 = \mathcal{O}\mathcal{S}(E) \text{ iff } E \text{ is of finite width.}$$

Proof

Let $E = (E, \leq, \otimes)$ be an event structure of finite depth. First note that the isolated elements of $\mathcal{S}(E)$ take the form

$$x = \bigcup_{0 \leq i \leq n} [e_i] \text{ for } e_i \in x.$$

Obviously an element taking such a form is isolated. For the converse simply see that x is the supremum of the directed set $\{[e_0, \dots, e_n] \mid e_0, \dots, e_n \in x\}$ and use the fact that x is isolated.

Thus as E has finite depth $\mathcal{S}(E)^0 \subseteq \mathcal{O}\mathcal{S}(E)$.

" \Leftarrow " Suppose E has finite width, then observable states are finite so $\mathcal{O}\mathcal{S}(E) \subseteq \mathcal{S}(E)^0$ giving $\mathcal{S}(E)^0 = \mathcal{O}\mathcal{S}(E)$.

" \Rightarrow " Suppose $\mathcal{O}\mathcal{S}(E) = \mathcal{S}(E)^0$. We require $\forall x \in \mathcal{O}\mathcal{S}(E) \ |x| < \infty$. Suppose otherwise i.e. for some $x \in \mathcal{O}\mathcal{S}(E) \ |x| = \infty$.

Define $x_n = \{e \in x \mid d(\emptyset, [e]) = n\}$. As $x \in \mathcal{O}\mathcal{S}(E)$ by 5.2.9 with $m = d(\emptyset, x)$ we have $x_n = \emptyset$ for $n > m$. Thus for some i ($1 \leq i \leq m$) we

have x_i an infinite set of \leq -incomparable events. Thus $[x_i] \notin \mathcal{S}(E)^0$. Yet $[x_i] \in \mathcal{O}\mathcal{S}(E)$ by 5.2.9, a contradiction.

Therefore $\forall x \in \mathcal{O}\mathcal{S}(E) \ |x| < \infty$ and E has finite width as required. \blacksquare

Thus those event structures of finite depth and width are characterised by the observable states coinciding with the isolated elements in the domain of states.

Finite-branching ideas suggest ideas along the lines of König's

lemma. So it is with finite width. We shall use ~~the~~ result below later, in establishing an equivalent of the K-density axiom under some restrictions.

Theorem 5.3.10

Let $E = (E, \prec)$ be an elementary event structure of finite depth and finite width. Then if E is infinite there is an infinite chain in E .

Proof

Suppose E satisfies the hypotheses of the theorem and $|E| = \infty$. We divide E into sections according to depth by:

Define $E_n = \{e \in E \mid d(\emptyset, [e]) = n\}$ for $n = 1, 2, \dots$

We note: Every event belongs to a unique E_n ; no E_n is null; each event of depth $n+1$ has a \prec -predecessor of depth n .

We now define t , a (finitely branching) tree with all nodes but the root labelled by elements of E , as consisting of the least set satisfying

- (a) $0 \in t$
- (b) $e \in E_1 \Rightarrow (0, e) \in t$
- (c) $\alpha \in t \ \& \ \alpha \neq 0 \ \& \ (\alpha)_1 \in E_n \Rightarrow \{(\alpha, e) \mid (\alpha)_1 \prec e \ \& \ e \in E_{n+1}\} \subseteq t$.

ordered by the transitive reflexive closure of \prec where

$$\alpha \prec \alpha' \text{ iff } \alpha = (\alpha')_0 .$$

(We use $()_0, ()_1$ to denote the projection functions.)

Then (t, \prec^*) is a tree, a non-root node α being labelled by $(\alpha)_1 \in E$. It is finitely branching by the observations made of the E_n 's above. Moreover every event e of E labels some node of t . For suppose $e \in E$. Then we choose a chain $e_1 \prec e_2 \prec \dots \prec e_n = e$ where $e_i \in E_i$ and n is the depth of e . Induction on n shows that $\alpha = (\dots(((0, e_1), e_2), e_3), \dots, e_n) \in t$ as required.

Thus t is infinite and finitely branching so we may apply König's lemma to yield an infinite branch

$$0 \prec \alpha_1 \prec \alpha_2 \prec \dots \prec \alpha_n \prec \dots .$$

This gives an infinite chain in E i.e.

$$(\alpha_1)_1 < (\alpha_2)_1 < \dots < (\alpha_n)_1 < \dots$$

which proves the theorem. ■

Corollary 5.3.11

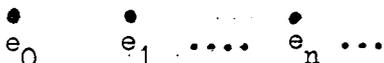
Let $E = (E, \leq, \otimes)$ be an event structure of finite depth and finite width. Then $x \in \mathcal{S}(E) \setminus \mathcal{O}(E) \setminus \mathcal{I}(E)$ includes an infinite chain.

Proof Let E satisfy the hypotheses of the theorem.

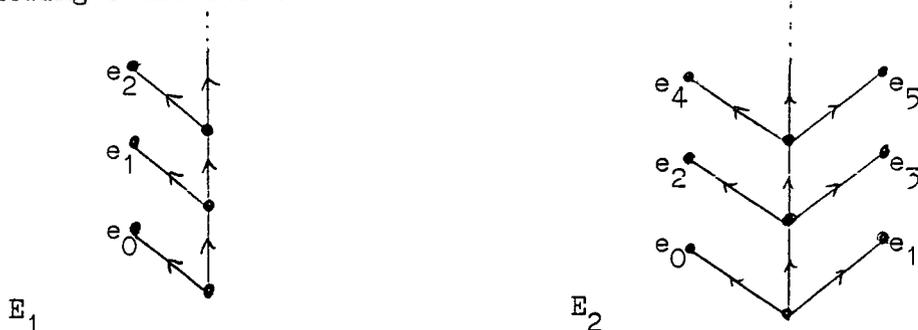
" \Leftarrow " obvious.

" \Rightarrow " Take $x \in \mathcal{S}(E) \setminus \mathcal{O}(E)$. Then define E_x to be the elementary event structure $(x, \leq \upharpoonright x)$. This is of finite depth and width. Moreover x is infinite. Therefore by 5.3.9 x has an infinite chain. ■

Consider the elementary event structure E_0 consisting simply of an infinite set of \leq -incomparable events. We can draw it as



Our definition of observer (5.1.4) allows all the events to occur within some bounded time. Of course the event structure is not of finite width. However we can regard it as derived from finite width event structures in which we ignore some events. For example the following event structures are of finite width:



Think of E_1 and E_2 as two possible finite width "implementations" of E_0 ; the event structure E_0 is obtained by ignoring the infinite branches of E_1 and E_2 . Think of E_0 as an abstraction from all possible implementations in the above sense. Then our definition of observer would be made less general so that any observer of E_0 is the restriction of the observer of a finite width implementation. In fact the observers of E_0 would then be all observers such that only finitely many events of E_0 occur by any finite time. We now spend a little time formalising these ideas but only for elementary event

structures.

Firstly we define two natural ideas of implementation.

Definition 5.3.12

Let (E, \leq) and (E', \leq') be elementary event structures. Define

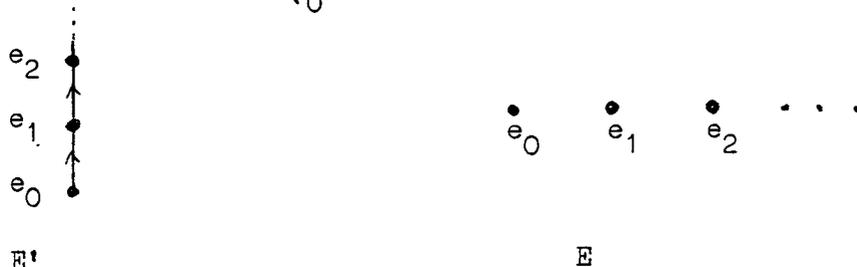
$E' \leq_0 E$ iff $E \subseteq E' \ \& \ \leq \subseteq \leq' \upharpoonright E$.

and $E' \leq E$ iff $E \subseteq E' \ \& \ \leq = \leq' \upharpoonright E$.

(Say E' \leq_0 -implements or \leq -implements E respectively.)

The relations $E' \leq_0 E$ and $E' \leq E$ give two ways that E' may implement E . The relation \leq corresponds to the idea above while for

\leq_0 we would have $E' \leq_0 E$ for the event structures:



Both relations are partial orders and \leq_0 has an easy characterisation. (We use $O \upharpoonright E$ to mean the observer O in $Ob(E')$ restricted to E a subset of E')

Lemma 5.3.13

Both the relations \leq_0 and \leq on elementary event structures are partial orders. We have $\leq \subseteq \leq_0$. Let E and E' be elementary event structures with E' of finite depth. Then $E' \leq_0 E$ is equivalent to either of

1. $E \subseteq E' \ \& \ \forall O' \in Ob(E') \ O' \upharpoonright E \in Ob(E)$
2. $E \subseteq E' \ \& \ \forall C' \in \mathcal{O}(E') \ C' \cap E \in \mathcal{O}(E)$.

Proof

Routine. ■

According to the views of this section "real computations" will give rise to event structures with finite width implementations. To characterising those event structures which have finite width implementations (in both the \leq_0 and \leq sense) the following lemma is useful. We give two proofs, one very simple, the other less so but more intuitive.

Lemma 5.3.14

Let (E, \leq) be a countable elementary event structure such that for all e in E we have $[e]$ is finite. Then there is an order-preserving countable enumeration of E i.e. there is a countable enumeration $e_0, e_1, \dots, e_n, \dots$ of E such that if $e \leq e'$ in E then $e = e_i$ and $e' = e_j$ with $i \leq j$ for some i, j in ω .

Proofs

Enumerate the countable elementary event structure E as $a_0, a_1, \dots, a_n, \dots$.

Easy proof: Let p_n be the n th prime. Represent e by $c(e) = \prod \{p_i \mid a_i \leq e\}$, the product of primes corresponding to those elements below or equal to e . The ordering \leq' given by $e \leq' e'$ iff $c(e) \leq c(e')$ is a total ordering of order type ω .

Intuitive proof: The idea is to regard the sequence a_0, a_1, \dots as assigning a priority to elements of E and then to serialise E by inductively "firing" the event with highest priority (earliest in the enumeration) amongst those with concession at any stage. Clearly \leq is well-founded. Take e_0 as the earliest \leq -minimal event in the enumeration. Inductively define e_n as the earliest \leq -minimal event of $E \setminus \{e_i \mid i < n\}$ in the enumeration. Thus we produce an enumeration $e_0, e_1, \dots, e_n, \dots$ of E . By its construction it is order-preserving. Also any element of E is in the enumeration by induction on \leq . Consider any element of E ; it will be a_n in the enumeration, for some n . Inductively assume $\{e \mid e < a_n\} \subseteq \{e_i \mid i \in \omega\}$. Then as $\{e \mid e < a_n\}$ is finite it is included in $\{e_0, e_1, \dots, e_m\}$ for some m . Also a_n is \leq -minimal in $E \setminus \{e_0, \dots, e_m\}$. As a_n is preceded by n elements in the enumeration, it will be contained in $\{e_0, \dots, e_{m+n}\}$. (Alternatively one can define the required enumeration ordering recursively from the original enumeration $a_0, a_1, \dots, a_n, \dots$ and work with that. Let the priority of e be written $p(e) = n$ if $e = a_n$ in the enumeration. Write $\circ e$ for the immediate \leq -predecessors of e . Then new enumeration \leq' is defined recursively by

$$e \leq' d \text{ iff } (\exists d_1 \in \circ d \ e \leq' d_1) \text{ or } (\circ d <' e \ \& \ \circ e <' d \ \& \ p(e) \leq p(d))$$

The recursive definition is justified by the well-foundedness of \leq . ■

Corollary 5.3.15

Let E be an elementary event structure. Then $E' \leq_0 E$ for some elementary event structure E' of finite width iff E is countable and for all e in E we have $[e]$ is finite.

Proof

Clearly if $E' \leq_0 E$ where E' is of finite width we have $E \subseteq E'$ with E' countable and $\leq \subseteq \leq' \upharpoonright E$ with $\leq'^{-1}\{e\}$ finite so E is countable with $[e]$ finite in E . The above lemma provides the converse; take E' to be the set E ordered as in the order-preserving enumeration it provides. ■

Event structures which may be \leq -implemented are characterised by the same properties. Lemma 5.3.14 simplifies the proof.

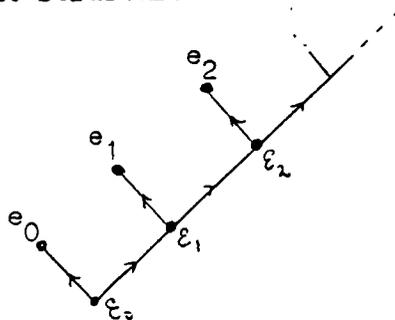
Theorem 5.3.16

Let E be an elementary event structure. Then $E' \leq E$ (or $E' \leq_0 E$) for some elementary event structure E' of finite width iff E is countable and for all e in E we have $[e]$ finite.

Proof

Suppose E is an elementary event structure. Suppose $E' \leq E$ with E' of finite width. Then clearly as $E \subseteq E'$ and E' is countable we have E countable. For e in E we have $[e]$ finite in E as $[e]$ is finite in E' .

Conversely suppose E is countable and for all e in E we have $[e]$ finite. If E is of finite width take $E' = E$. Otherwise countably enumerate E in an order-preserving way as $e_0, e_1, \dots, e_n, \dots$. Form E' by adjoining the event structure



More formally define $E' = E \cup \{\epsilon_i \mid i \in \omega\}$ where each $\epsilon_i \notin E$ with causality relation $\leq' = \leq \cup \{(\epsilon_i, \epsilon_j) \mid i, j \in \omega \& i \leq j\} \cup \{(\epsilon_i, e_j) \mid i, j \in \omega \& i \leq j\}$

As the enumeration e_0, e_1, \dots is order-preserving it follows that \leq' is a partial order. The event structure E' has finite width and $E' \preceq E$. ■

Thus domains of event structures which can be implemented by finite width event structures will satisfy axiom F of chapter 3.

Now we characterise those observers of an event structure which result by restricting the observers of its finite width implementations. Regarding an event structure as an abstraction from such implementations these observers are the only ones possible.

Theorem 5.3.17

Let E be a countable event structure such that for all events e we have $[e]$ finite. Suppose $O \in \text{Ob}(E)$. Then $\exists E' \preceq E$ E' has finite width & $O' \in \text{Ob}(E')$ & $O = O' \upharpoonright E$ iff

$$\forall t \in \omega \left\{ \left| \{ e \in E \mid O(e) < t \} \right| < \infty \right.$$

The observers formed by restricting observers of \preceq_0 -implementations are characterised identically.

Proof

" \Rightarrow " Clear.

" \Leftarrow " Suppose $O \in \text{Ob}(E)$ s.t. $\forall t \in \omega \left\{ \left| \{ e \in E \mid O(e) < t \} \right| < \infty \right.$

We extend E to a finite width event structure E' . However now we must take care that O extends to an observer of E' so the construction of E' is a little more complicated than that in 5.3.14. Let $e_0, e_1, \dots, e_n, \dots$ be a countable order-preserving enumeration of $E \setminus O^{-1}\omega$. Take $\{\mathcal{E}_i \mid i \in \omega\}$ disjoint from E . Define E' by:

$$\begin{aligned} E' &= E \cup \{\mathcal{E}_i \mid i \in \omega\} \\ \leq' &= \leq \cup \{(\mathcal{E}_i, \mathcal{E}_j) \mid i, j \in \omega \& i \leq j\} \cup \{(\mathcal{E}_i, e) \mid O(e) \in \omega \& O(e) > i\} \\ &\quad \cup \{(\mathcal{E}_i, e_j) \mid i, j \in \omega \& i \leq j\}. \end{aligned}$$

The idea: For the chain $\{\mathcal{E}_i \mid i \in \omega\}$ we ensure that \mathcal{E}_i is \leq' all events which are really observed by O after time i and also \leq' all events which are not observed (except at ∞) but at i or later in the enumeration. Because the enumeration was chosen to be order-preserving \leq' is a partial order. The event structure E' is of finite width with $E' \preceq E$, and has observer O' where

$$\begin{aligned}
O'(e) &= O(e) \text{ if } e \in E \\
&= i \quad \text{if } e = \mathcal{E}_i \\
&= \infty \text{ otherwise.}^1
\end{aligned}$$

Then $O = O' \upharpoonright E$ and $E' \preceq E$ as required.

The proof for \preceq_0 , rather than \preceq , is similar. ■

As a corollary we characterise the observable states of an event structure which result by restricting the observable states of a finite width implementation. Again regarding an event structure as an abstraction from all such implementations these observable states are the only ones possible. Recall that for event structures E of finite depth the isolated elements of $\mathcal{S}(E)$, written $\mathcal{S}(E)^0$, are precisely the finite sets in $\mathcal{S}(E)$.

Theorem 5.3.18

Let E be a countable event structure such that for all events e we have $[e]$ finite. Then $(\exists E' \preceq E, C' \in \mathcal{A}(E')) E'$ is of finite width & $C = C' \cap E$ iff $C \in \mathcal{S}(E)^0$.

An identical statement holds replacing \preceq by \preceq_0 .

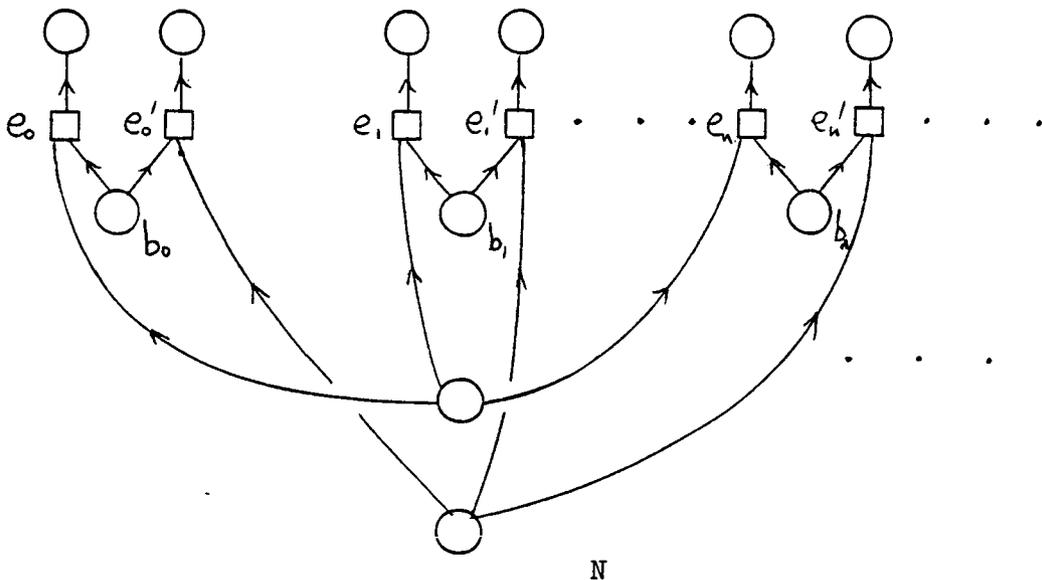
We summarise the last batch of results. (Here all event structures are elementary.) Assuming "real computations" determine finite width event structures we can still interpret event structures not of finite width; provided they are countable and any event has only finitely many pre-events, they can be regarded as an abstraction from all possible finite width implementations (5.3.15 and 5.3.16). The possible observers and possible observable states are restricted accordingly; in particular states which really can be observed at finite time are now exactly the isolated elements (5.3.18) in the domain of states.

We have argued that with respect to the definition of observer in this chapter "real" computations determine finite width event structures. The converse, that any finite width event structure is determined by a "real" computation is not so obvious. Clearly it would depend on precisely what class of computations we wished to represent. Reasonably it might be a class of Milner nets in which a single communication could be between a finite set of agents not

necessarily just two. Then as in chapter 2 communications could be represented as events and local states of agents as conditions in a transition net. A suitable class of Milner nets would give occurrence net unfoldings inducing event structures of finite width. Importantly one would expect only finitely many conditions to hold at any finite time corresponding to there only being finitely many agents at any finite time. However not all event structures of finite width are induced by such occurrence nets. The next example gives a finite width event structure such that any occurrence net inducing it must have infinitely many conditions holding initially.

Example 5.3.19

Consider the event structure E induced by this occurrence net N :



The event structure $E = \mathcal{E}(N)$ consists of an infinite set of pairs e_n, e'_n of conflicting events with $e_0, e_1, \dots, e_n, \dots$ pairwise in conflict and $e'_0, e'_1, \dots, e'_n, \dots$ pairwise in conflict. Formally

$$\mathcal{E} = \{(e_n, e'_n) \mid n \in \omega\} \cup \{(e_n, e_m) \mid n, m \in \omega \ \& \ n \neq m\} \\ \cup \{(e'_n, e'_m) \mid n, m \in \omega \ \& \ n \neq m\}.$$

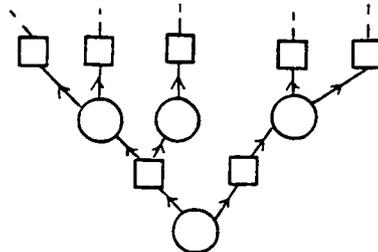
Suppose N' is an occurrence net s.t. $\mathcal{E}(N') = E$. Then N' must include the conditions b_n shown i.e. it must have an infinite set of initial conditions. However E is of finite width; at most two events can ever occur.

One can regard E as modelling the following computation: the

computation consists of two output places \square and \square' both of which may be filled by a single integer provided the integers in the two places differ.

The role of the finitely-enabling restriction (guaranteeing finite width) is to ensure that the number of conflict-free events can only grow finitely in finite time. It is natural to look at another finite-branching property namely one ensuring the computation possesses only finite non-determinism. We shall look briefly at ways to formalise this for event structures. The idea is well-known for purely non-deterministic processes which can then be modelled by finitely-branching trees. These computations are said to possess finite non-determinism, a property which has been useful in constructing powerdomains ([Plö], [Smy]). I believe that the assumption of finite non-determinism is more technical than that of finite width for example. With it one can give denotations to a wide class of non-deterministic programs. The assumption is made in constructing the possible denotations, the elements of a domain, and not about the structure of the domains themselves. The domain of integers does not present any technical problems even though it has infinite conflict (thinking of the associated event structure). In Petri nets and event structures there is no explicit distinction between datatype and denotation but still we press on with attempts to define finite branching in event structures so as to capture the intuition of finite non-determinism in a computation.

Any definition of finitely-branching event structure should generalise the finite-branching property of trees. One possible definition could express that the event structure is built from purely non-deterministic processes individually capable of at most one of a finite set of actions at any time. Such processes would generalise the sequential processes of chapter 2 and as nets look like



This gives a local idea of finite branching. The following seems the correct formal definition.

Definition 5.3.20

Let E be an event structure. Say E is locally finitely-branching iff there is an occurrence net N s.t. $\mathcal{C}(N) = E$ and for all conditions b of N we have b^\bullet finite.

But of course all event structures are locally finitely-branching in this sense.

Lemma 5.3.21

Any event structure is locally finitely-branching.

Proof

Let E be an event structure. Define N to consist of events E , conditions B defined by

$$B = \{(e, \{e'\}) \in E^2 \mid e < e'\} \cup \{(0, \{e, e'\}) \mid e \not\ll e'\}$$

with F -relation: bFe iff $e \in (b)_1$ and eFb iff $e = (b)_0$. Then

$\mathcal{C}(N) = E$ and for all b we have b^\bullet finite. ■

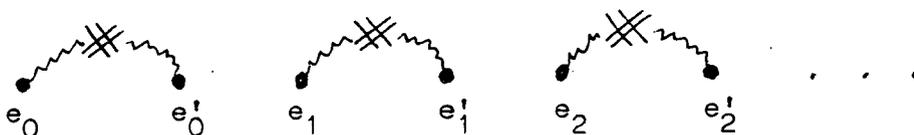
Thus we look for a more global definition of finitely-branching expressing that at any finite time the computation can only choose between finitely many courses. The idea of finite time is formalised by using observable states so we naturally take event structures to be of finite depth. The following is suggested:

Definition 5.3.22

Let E be an event structure of finite depth. Say E is finitely-branching iff $\forall C \in \mathcal{O}_f(E) \mid |IF(C)| < \infty$ (where IF was defined in 5.3.6).

The definition excludes the following example.

Example 5.3.23



Here the event structure consists of a countably infinite set of conflicting pairs. Thus in finite time the computation may choose between uncountably many courses.

I believe the definition of finitely-branching is equivalent to $\forall n \in \omega \{C \in \mathcal{O}_f(E) \mid C \text{ is } \subseteq\text{-maximal} \ \& \ d(\emptyset, C) \leq n\}$ is finite.

In the presence of finite width the following is equivalent:

$\forall C \in \mathcal{O}_s(E) \mid \text{conc}(C) \mid < \infty$ as is probably: $\forall C \in \mathcal{O}_s(E) \quad K$ a ken of \bigcup in $\text{conc}(C) \Rightarrow K$ finite. Of course this should not be the final word on finite-branching. One should seek an intuitive characterisation and if there are not any change the definition.

5.4 States of occurrence nets and K-density

So far we have worked with event structures. Here we translate our results to occurrence nets. Firstly we can extend the notions of finite depth and finite width to occurrence nets.

Definition 5.4.1

An occurrence net N is said to be of finite depth iff $\mathcal{E}(N)$ is of finite depth. Furthermore if N is of finite depth it is said to be of finite width iff $\mathcal{E}(N)$ is of finite width.

We wish to associate a case of an occurrence net $N = (B, E, F)$ consisting purely of conditions with an observable state of $\mathcal{E}(N)$. In order to do this we impose the axiom: $N3. \forall e \in E \cdot e \neq \emptyset \ \& \ e^* \neq \emptyset$.

We associate holdings of conditions in an occurrence net N with elements of $\mathcal{L}_o \mathcal{E}(N)$ by the following.

Definition 5.4.2

Let $N = (B, E, F)$ be an occurrence net. For $C \in \mathcal{L}_o \mathcal{E}(N)$ define the frontier of C in N , written $\text{Fr}_N(C)$, by
$$\text{Fr}_N(C) = (\bigcup \{e^* \mid e \in C\} \cup \{b \in B \mid \cdot b = \emptyset\}) \setminus \bigcup \{e \mid e \in C\}.$$

The idea: Given C a left-closed consistent subset of events of a net, the frontier of C is those conditions which hold because the events in C have occurred. The axiom $N3$ ensures that every event occurrence is reflected in a change in holding of the conditions.

In general such a frontier will not be a case. However

Proposition 5.4.3

Suppose $N = (B, E, F)$ is an occurrence net of finite depth satisfying $N3$. Then for $C \in \mathcal{L}_o \mathcal{E}(N)$, $\text{Fr}_N(C)$ will be a case. We call such frontiers observable cases of N and $\text{Fr}_N(\emptyset)$ the initial case. The map Fr_N is 1-1.

Proof

We sketch the proof that $\text{Fr}_N(C)$ is a case for observable states C .

From the fact that C is left-closed and consistent it follows that all conditions in $\text{Fr}_N(C)$ are co_N to each other. That it is a ken of co_N follows as C does not include any infinite F^* -ascending chains and its complement $E \setminus C$ does not include any infinite F^* -descending chains. ■

The definition of observable cases of an occurrence net allows us to extend proposition 4.2.9 a little.

Proposition 5.4.4

Let N be a contact-free transition net satisfying $N3$, with initial marking M_0 . Recall the occurrence net unfolding $\mathcal{O}((N, M_0))$ and the folding f from $\mathcal{O}((N, M_0))$ to N (see 4.2.9). Then f takes observable cases of $\mathcal{O}((N, M_0))$ to reachable markings of (N, M_0) . Conversely any reachable marking of (N, M_0) is the image of an observable case in $\mathcal{O}((N, M_0))$.

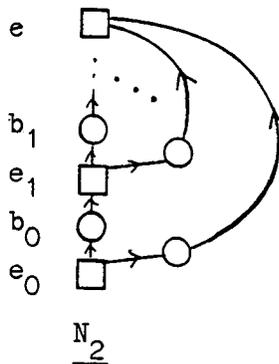
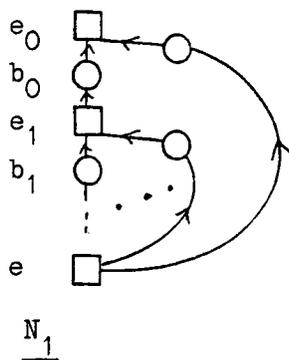
Proof

We give the idea. That observable cases $\text{Fr}(C)$ are mapped onto reachable markings is proved by induction on $d(\emptyset, C)$. To show the converse, take C to be those event occurrences giving $M_0 \rightarrow M$ for the reachable marking M . C is observable by induction on the length of \rightarrow and $M = f \text{Fr}(C)$. ■

We now move on to a discussion of K -density. First note that our assumptions of finite width, finite depth and axiom $N3$ are independent of K -density, either separately or in combination. The net $\mathcal{N}(E_2)$ for the event structure E_2 of example 5.1.2 is K -dense and satisfies $N3$ but is not of finite depth. Also note that the non K -dense net of example 2.4.4 satisfies $N3$ and is of finite depth and width.

It is useful to note that the restriction of finite depth forces sequential processes to take a particularly simple form. Without this restriction various order types are possible for sequential processes as the following causal nets illustrate.

Example 5.4.5



In both the causal nets N_1 and N_2 the set $\{e\} \cup \{e_n \mid n \in \omega\} \cup \{b_n \mid n \in \omega\}$ forms a sequential process. In N_1 it does not include any post-conditions of the event e while in N_2 it does not include any pre-conditions of the event e . For nets of finite depth this is impossible.

Theorem 5.4.6

Let N be a causal net of finite depth. Its sequential processes are precisely maximal sequences of the form $x_0 Fx_1 Fx_2 \dots$ where x_0 is an F^* -minimal element of N .

Proof

Let N be a causal net of finite depth. Using finite depth and proposition 2.4.10, maximal sequences of the form above are sequential processes. Conversely suppose S is a sequential process. Then inductively produce a maximal subsequence $x_0 Fx_1 F \dots Fx_n \dots$ of S using proposition 2.4.10; while $S \setminus \{x_i \mid 0 \leq i \leq n\} \neq \emptyset$ inductively take x_{n+1} as the F^* -minimum element of $S \setminus \{x_i \mid 0 \leq i \leq n\}$. This process either yields a maximal finite chain whose elements are S or an infinite chain. In the latter case finite depth guarantees the chain includes all elements of S . ■

We now prove a restricted form of K-density.

Theorem 5.4.7

Let N be a causal net of finite depth satisfying N3. Then every observable case meets every sequential process.

Proof

Let $N = (B, E, F)$ be a causal net of finite depth satisfying N3.

Suppose C is a case not meeting some sequential process S so $S \cap C = \emptyset$. We show that C is above S (i.e. $\forall s \in S \exists c \in C s \leq F^+ c$) and that S is infinite. From this it follows that C cannot be observable.

By theorem 5.4.6 we know S has the form $b_0 F e_0 F b_1 F e_1 \dots b_n F e_n \dots$ where b_0 is an F^* -minimal condition in N . As $b_0 \notin C$ and C is a ken of c_0 where $c_0 = (B \cup E) \times (B \cup E) \setminus (F^+ \cup (F^+)^{-1})$, we have $b_0 F^+ c_0$ for some $c_0 \in X$. As $b_0 = \{e_0\}$, for some $e_0 \in E$, we have $e_0 \in S$. Thus $e_0 \notin C$ giving $e_0 F^+ c_0$. Then for some $b_1 \in B$, $\{b_1\} = S \cap e_0$. Therefore as $b_1 \notin C$, a ken of c_0 , we have either $b_1 F^+ c_1$ or $c_1 F^+ b_1$ for some $c_1 \in C$. The latter yields $c_1 F^+ e_0$ which with $e_0 F^+ c_0$ gives $c_1 F^+ c_0$ contradicting $c_0 \leq c_1$. Thus $b_1 F^+ c_1$. This process may be continued inductively to show that S is an infinite sequential process below C as required. Thus an observable case meets every sequential process.

The proof indicates how essential conditions are for K-density of a restricted form of it to be true. See 7.4.3 for a generalisation of the above theorem.

This follows as observable states do not include infinite ascending chains. For both the above proposition and theorem we note that a weaker notion of observable case and finite depth would suffice. Taking N as (B, E, F) , the restriction of finite depth could be replaced by:

For $x \in B \cup E$, any ken of $F^* \cup F^{*-1}$ in $\{x' \mid x' F x\}$ is finite. This says no (sub) sequential processes below an element are infinite. Of course the element x may be restricted to range over events. The new observable cases could be taken as the frontiers of left-closed conflict-free subsets C in which any ken of $(\leq \cup \succeq)$ is finite. Presumably one could parallel the results of this chapter for these different notions and a generalised idea of observer. For finite width structures, new and old definitions and results should coincide in the main as by Corollary 5.3.10 the two ideas of observable state do.

We conclude our discussion of K-density here with a result which illuminates and reinforces our net-theoretic argument for K-density in chapter 2. With suitable restrictions on a causal net we can give an equivalent of the K-density axioms; then a causal net is K-dense iff all cases (in Petri's sense) consisting solely of conditions are

observable cases (in our sense).

Theorem 5.4.8

Let $N = (B, E, F)$ be a causal net of finite depth, finite width and satisfying axiom $N3$. Then, taking Fr_N as defined in 5.4.2: N is K -dense iff the map Fr_N from $\mathcal{O}(\mathcal{E}(N))$ is onto the cases of N consisting purely of conditions.

Proof

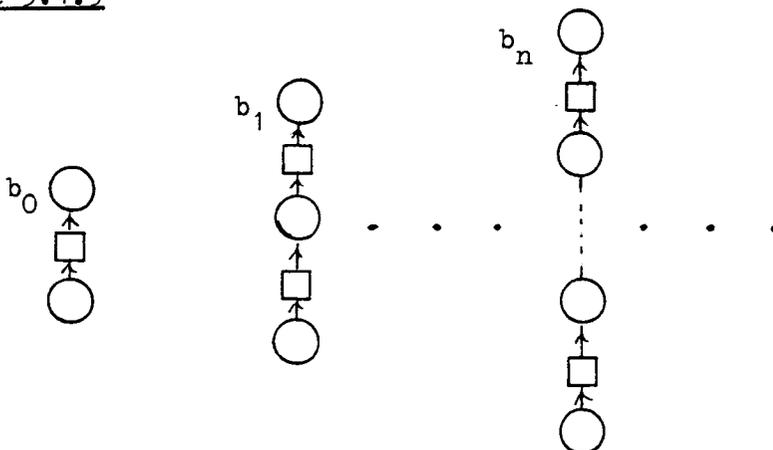
Let $N = (B, E, F)$ be a net satisfying the above conditions.

" \Leftarrow " Suppose Fr_N is onto the cases consisting purely of conditions i.e. all such cases are observable cases. Moreover assume N is not K -dense i.e. $C \cap S = \emptyset$ for some sequential process S and case $C \subseteq E \cup B$. Defining $C' = (C \cup (C \cap E)^c) \setminus C \cap E$ gives C' a case with $C' \subseteq B$ and $C' \cap S = \emptyset$. But then C' is an observable case, as Fr_N is onto, not meeting S - a contradiction by theorem 5.4.7.

" \Rightarrow " Suppose N is K -dense and that C is a case of N with $C \subseteq B$. Define $x = \{e \in E \mid \exists b \in C \ e F^* b\}$. We require $x \in \mathcal{O}_0 \mathcal{E}(N)$. Suppose otherwise i.e. there are chains of unbounded length in x . By the assumption of finite width this implies there is an infinite chain in x (theorem 5.3.10). The infinite chain will determine a sequential process S in N such that $S \cap C = \emptyset$ - a contradiction as we assume N is K -dense. ■

The role of finite width in the above proof is to convert there being chains of unbounded length in x to there being an infinite chain in x . A revised version of this theorem would hold in which we merely required that observable states included no infinite chains; then we could omit the requirement of finite width. The next example shows why finite width is necessary for the above theorem with our definition of observable case.

Example 5.4.9



The causal net consists of an infinite set of sequential processes each of finite length - the n th process has length n - but overall of unbounded length. The net is not of finite width. The net is K -dense but clearly the case $\{b_n \mid n \in \omega\}$ is not observable. This shows that finite width is necessary for the equivalence of theorem 5.4.8.

Reasonably assume a course of computation is represented by a causal net of finite depth and width. By theorem 5.4.8 the assumption of K -density is then equivalent to assuming all cases are observable cases. But why should all cases be observable? Assuming so bans the innocent net of example 2.4.4. According to our view K -density is too restrictive an axiom. However the intuition motivating it remains: An observable case does meet any sequential process (theorem 5.4.7).

5.5 Confusion and concrete domains

K -density proved to be a concept which did not translate very cleanly into the framework of event structures and domains. Fortunately confusion does translate well; indeed confusion-freeness was discovered independently by Gilles Kahn and Gordon Plotkin in their work on concrete domains.

Recall our discussion of confusion in chapter 2. It arose because of two violating situations called symmetric and asymmetric confusion. In net theory these are introduced formally at the level of transition nets. The following are the obvious corresponding definitions for an occurrence net.

Definition 5.5.1

Let $N = (B, E, F)$ be an occurrence net of finite depth satisfying $N3$.

We say N is symmetrically confused iff there are an observable case C and events e, e', e'' such that

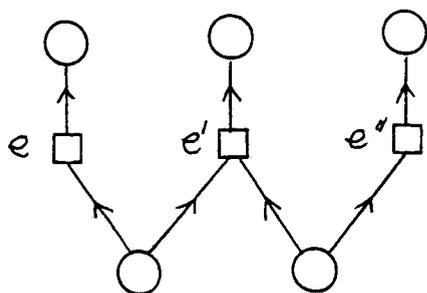
$$(\cdot e, \cdot e', \cdot e'' \subseteq C) \& (\cdot e \cap \cdot e' \neq \emptyset) \& (\cdot e' \cap \cdot e'' \neq \emptyset) \& (\cdot e \cap \cdot e'' = \emptyset).$$

We say N is asymmetrically confused iff there are an observable case C and events e, e', e'' such that

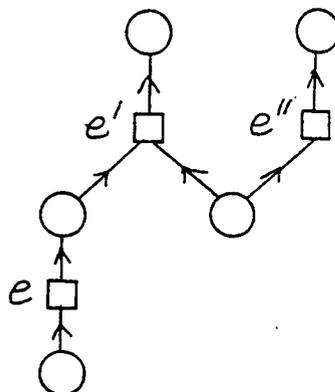
$$(\cdot e, \cdot e'' \subseteq C) \& (\cdot e' \not\subseteq C) \& (\cdot e' \subseteq (C \setminus \cdot e) \cup \cdot e) \& (\cdot e \cap \cdot e'' = \emptyset) \\ \& (\cdot e' \cap \cdot e'' \neq \emptyset).$$

Finally we say N is confused iff N is symmetrically or asymmetrically confused; otherwise N is confusion-free.

Example 5.5.2



N_1 - Symmetric confusion



N_2 - Asymmetric confusion

In the special case where the occurrence net is the unfolding of a transition net definition 5.5.1 reflects the situation in the transition net; observable cases of the unfolding determine the reachable markings under the folding map and firings from a reachable marking are images of occurrences from an associated observable case.

Proposition 5.5.3

Let (N, M_0) be a contact-free transition net with initial marking M_0 , satisfying N3. Then (N, M_0) is symmetrically (respectively asymmetrically) confused iff the occurrence net unfolding $\mathcal{O}((N, M_0))$ is symmetrically (respectively asymmetrically) confused.

In order to see how confusion manifests itself in event structures and domains we define the relation \times_{μ} over an event structure, representing immediate conflict.

Definition 5.5.4

Let $E = (E, \leq, \#)$ be an event structure. Define \times_{μ} by putting for e, e' in E :

$$e \times_{\mu} e' \text{ iff } e \# e' \ \& \ \exists C \in \mathcal{O}(E) \ C \cup \{e\}, C \cup \{e'\} \in \mathcal{O}(E).$$

We then say e and e' are in immediate conflict.

The relation of immediate conflict between events e and e' represents the possibility of a stage in the computation at which either of e and e' (but not both) may occur. Its properties are summarised in the lemma below.

Lemma 5.5.5

Let $E = (E, \leq, \#)$ be an event structure of finite depth and suppose $\#_{\mu}$ is as defined in 5.5.4. Then

1. $\#_{\mu}$ is a symmetric relation.
2. $e \#_{\mu} e'$ iff $e \# e'$ & $(\forall \varepsilon < e \neg(\varepsilon \# e')) \& (\forall \varepsilon' < e' \neg(\varepsilon' \# e))$.
3. $e \# e'$ iff $\exists \varepsilon, \varepsilon' \in E \ \varepsilon \leq e \& \varepsilon' \leq e' \& \varepsilon \#_{\mu} \varepsilon'$.

Proof

1. Obvious.
2. " \Leftarrow " This follows by taking $C = [e] \cup [e'] \setminus \{e, e'\}$.
" \Rightarrow " Suppose $e \#_{\mu} e'$ i.e. $e \# e'$ and $C \cup \{e\}, C \cup \{e'\} \in \mathcal{O}(E)$ for some $C \in \mathcal{O}(E)$. Merely note $\langle^{-1}\{e\}, \langle^{-1}\{e'\} \subseteq C$.

3. Suppose $e \# e'$. By the well-foundedness of $<$ that finite depth provides we may find a minimal pair in $\{(\varepsilon, \varepsilon') \mid \varepsilon \leq e \& \varepsilon' \leq e' \& \varepsilon \# \varepsilon'\}$ w.r.t. to the ordering on pairs defined componentwise. Such a pair will be $\#_{\mu}$ related. ■

We can now transfer the notion of confusion to event structures using $\#_{\mu}$ and its properties.

Theorem 5.5.6

Let $N = (B, E, F)$ be an occurrence net of finite depth satisfying N3 and define $\#_{\mu}$ as in 5.5.4. Write $\mathcal{E}(N)$ as E . Then

1. N is symmetrically confused iff $\exists e, e', e'' \in E \ e \#_{\mu} e' \#_{\mu} e'' \& \neg (e \# \cup \mid e'')$.
2. N is asymmetrically confused iff $\exists e, e', e'' \in E \ e' \#_{\mu} e'' \& e < e' \& \neg (e < e'')$.

Proof

" \Rightarrow " for 1. and 2. follows by "unwrapping" definitions.

1. " \Leftarrow " Take C , the required observable case, to be

$$\text{Fr}_N([e] \cup [e'] \cup [e''] \setminus \{e, e', e''\})$$

2. " \Rightarrow " Without loss of generality suppose e is a \leq -maximal element below e' with $\neg (e < e'')$, so $e \prec e'$. Take C , the required observable case, to be $\text{Fr}_N([e'] \cup [e''] \setminus \{e, e', e''\})$. ■

Note the occurrence of " $\#$ " and not " $\#_{\mu}$ " in part 1 of the above theorem. In our next theorem we shall show, in the course of the proof, that $\#_{\mu}$ may replace $\#$ in part 1: once N is known to not be

asymmetrically confused. With our next theorem in mind as justification we give a definition of confusion-freeness for event structures. Clause 2 below can be interpreted as requiring enablings to respect the $\bowtie_{\mu} \cup |$ -equivalence classes provided by clause 1.

Definition 5.5.7

Let $E = (E, \leq, \bowtie)$ be an event structure of finite depth and take \bowtie_{μ} as defined in 5.5.4. Then

E is confusion-free iff

1. $\bowtie_{\mu} \cup |$ is an equivalence relation
2. $e < e' \bowtie_{\mu} e'' \Rightarrow e < e''$.

Now we look at the domain version.

Theorem 5.5.8

Let N be an occurrence net of finite depth satisfying N3. The following are equivalent.

1. N is not confused.
2. $\mathcal{E}(N)$ is confusion-free.
3. $\mathcal{L}_0 \mathcal{E}(N)$ satisfies axiom Q of concrete domains.

Proof

"1 \Leftrightarrow 2" By theorem 5.5.6 N not being asymmetrically confused is directly equivalent to 5.5.7 part 2 holding for $\mathcal{E}(N)$. From this it follows that if N is not asymmetrically confused then for $e, e', e'' \in E$ $e \bowtie_{\mu} e' \bowtie_{\mu} e''$ & $\neg(e \bowtie \cup | e'')$ \Leftrightarrow $e \bowtie_{\mu} e' \bowtie_{\mu} e''$ & $\neg(e \bowtie_{\mu} \cup | e'')$ using the fact that then the enabling \leq respects \bowtie_{μ} . Thus given N is not asymmetrically confused, N is not symmetrically confused iff part 1 of 5.5.7 holds. (This justifies part 1 of definition 5.5.7.) Therefore 1 \Leftrightarrow 2.

"2 \Rightarrow 3" Suppose $\mathcal{E}(N)$ is confusion-free. We wish to prove axiom Q which we remind the reader takes the form

$$z \succ x \sqsubseteq y \text{ \& } z \not\prec y \Rightarrow \exists ! t \ x \prec t \sqsubseteq y \text{ \& } z \not\prec t.$$

Thus suppose $z \succ x \sqsubseteq y$ & $z \not\prec y$ in $\mathcal{S}_0 \mathcal{E}(N)$. Then $z = x \cup \{e\}$, $e \bowtie_{\mu} e'$ and $e' \in y \setminus x$ for some events e and e' of $\mathcal{E}(N)$. Then by part 2 of definition 5.5.7 of a confusion-free event structure, $t = x \cup \{e'\}$ is also in $\mathcal{S}_0 \mathcal{E}(N)$. Thus using

part 2 of definition 5.5.7 we have the existence part of axiom Q. The uniqueness follows from part 1 of definition 5.5.7.

"3 \Rightarrow 2" The existence part of axiom Q yields part 2 of 5.5.7 and then the uniqueness, part 1. To show part 2 of 5.5.7 suppose it were false i.e. that we have $e < e' \not\ll_{\mu} e''$ and $e \not\prec e''$. We may assume e is \leq -maximal so that $e < e' \not\prec e''$ - then e' covers e in the event structure. Take $x = ([e'] \setminus \{e, e'\}) \cup ([e''] \setminus \{e''\})$. Take $z = ([e'] \setminus \{e, e'\}) \cup [e'']$. Take $y = [e'] \cup [e'']$. Then $z \supseteq x \sqsubseteq y$. However by the choice of x, y, z we have $x \prec t \sqsubseteq y$ implies $t \setminus x = \{e\}$ so $t \uparrow z$ contradicting the existence part of Q. To show part 1 of 5.5.7 assume $e \not\ll_{\mu} e' \not\ll_{\mu} e''$ and $e \neq e''$. By the above the existence part of Q gives $\langle^{-1}\{e\} = \langle^{-1}\{e'\} = \langle^{-1}\{e''\}$. Suppose $\neg (e \ll_{\mu} e'')$. Then take $x = \langle^{-1}\{e\}$, $y = [e] \cup [e'']$ and $z = [e']$. This choice contradicts the uniqueness part of axiom Q so we have $e \ll_{\mu} e''$ as required. ■

Corollary 5.5.9

Let N be an occurrence net satisfying N3. Then

- (i) E is countable
 - (ii) $F^{*-1}\{e\}$ is finite for all events e , and
 - (iii) N is confusion-free
- iff $\mathcal{L} \circ \mathcal{E}(N)$ is a distributive concrete domain..

Proof

The domain $\mathcal{L} \circ \mathcal{E}(N)$ is prime algebraic so distributive and satisfies axioms C and R by the work of chapter 4. It being ω -algebraic and satisfying axiom F correspond to (i) and (ii) respectively. Axiom Q corresponds to (iii) by the above theorem. ■

Recall the intuition in net theory that confusion leads to conflict-resolution not being objective; whether or not conflict appeared to be resolved between events depended on the observer. Confusion-free nets can be represented by the matrices of Kahn and Plotkin. Then conflict between events is localised in that two immediately conflicting events will always be enabled at the same time and be competing for the same place. All observers will agree whether or not conflict has been resolved and at which place the resolution occurred.

5.6 Alternative axioms on event structures and other ideas of observable state

In this section we remark on other ways of formalising the intuition behind observable states. We have worked largely with $\mathcal{O}(E)$ for an event structure E . The elements of $\mathcal{O}(E)$ are consistent left-closed subsets of E uniformly reachable from the initial null-state. The restriction to event structures of finite depth is then natural; no event not of finite depth can ever occur. We mentioned the weaker definition taking consistent left-closed subsets which do not include infinite chains. Then the finite depth restriction is replaced by:

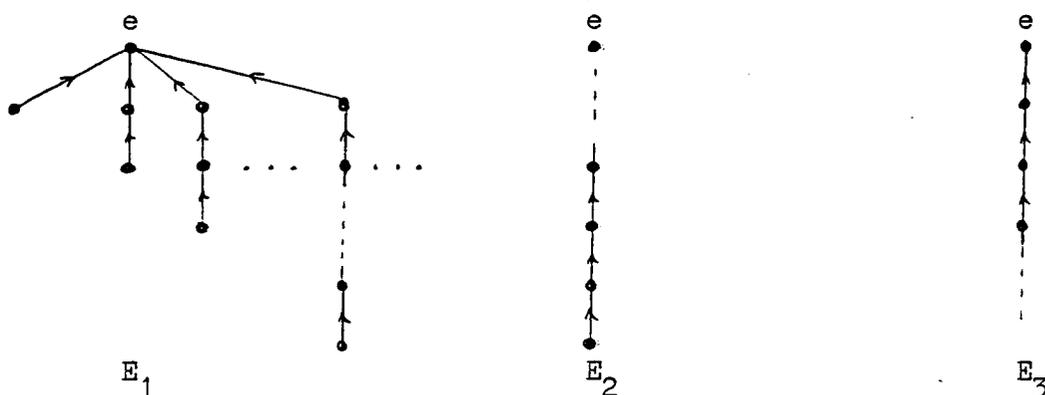
Definition 5.6.1

Say an event structure E is well-based iff for all events e any total order below e is finite. \int (These definitions were sufficient to prove the results on K-density in 5.4.) We prove further restrictions (implying axiom F) follow from Dana Scott's thesis that computable functions are continuous. All the definitions express a finiteness constraint on event structures and on those states which can be observed in finite time. For event structures of finite width they agree. All these restrictions on event structures imply a form of discreteness. As yet it is unclear how to represent non-discrete or "continuous" processes by event structures.

Recall the idea of observable state. An observable state is a subset of events consisting of all those events which may be observed in finite time in a history of observation. In this chapter we have taken an observer to be intuitively a run or history of computation. This form of observer is passive, playing no computational role.

We take another look at $\mathcal{O}(E)$. Apparently this definition rather than the weaker one is more appropriate to net theory. (In a letter Petri said he wished to ban nets associated with the event structure of example 5.1.2). This definition is also appropriate to the ideas of local time introduced in [Lam]. In [Lam] an elementary event structure is built up from chains of events representing processes in which some events represent the sending or receipt of messages between processes. A ("logical") clock is associated with each process so that the time ascribed to an event is greater than the time ascribed to all events on which it is causally dependent.

The weaker definition, taking observable states to not include infinite chains is implied by Hewitt's axioms [Hew] on the event structures associated with actors. Hewitt imposes the axiom, called E-discreteness in [Bes], that there are no infinite chains between events. Then saying there are no infinite chains between an initial fictitious starting event and any other event (i.e. the event structure with initial event in E-dense) is equivalent to the well-based restriction 5.6.1. According to this restriction starting from the initial null state the event e may occur in E_1 but not in E_2 or E_3 below:



Only infinite chains of events are obliged to take infinite time. Regarding the event structure as modelling a set-up as in [Lam] no restriction is made on the relative rates of clocks ascribed to process beyond that they all agree that only finite time has passed at events corresponding to communications.

In chapter 4 we took $\mathcal{L}_0(E)$ as the natural Scott domain of information to associate with an event structure E . Let us explore a little further how the ideas of Scott [Sco] translate to event structures. Scott proposed the thesis that all computable functions are continuous (see 3.1). In more detail, datatypes are represented as complete partial orders of information (cpos) and computations from one datatype to another as functions between the associated cpos; Scott's thesis says computable functions are continuous in this framework. The thesis has an intuitive justification (see 3.1, or [Wad] for more detail). We give an argument which characterises those elementary event structures which agree (in a formally defined way) with Scott's thesis.

In examples 2.3.7 and 2.3.9 we showed nets associated with computations between datatypes. The datatypes were subnets of the net of the computation with less causal structure than the computation as a whole. Recall the relation \leq_0 on event structures introduced in 5.3. For elementary event structures E and E' we have $E \leq_0 E'$ iff $E' \subseteq E$ and $\leq' \subseteq \leq \upharpoonright E'$. We shall regard E' as a datatype involved in the computation described by E . Suppose $E \leq_0 E_0$ and $E \leq_0 E_1$. Regard E_0 as representing an input datatype, E_1 as representing an output datatype and E as the computation between them. Take $\mathcal{L}(E_0)$ and $\mathcal{L}(E_1)$ as the associated domains of information. The event structure E determines a function between $\mathcal{L}(E_0)$ and $\mathcal{L}(E_1)$ in this way:

Definition 5.6.2

Let E be an elementary event structure. Suppose $E \leq_0 E_0$ and $E \leq_0 E_1$. Then define

$$f_{E_0, E_1}: \mathcal{L}(E_0) \rightarrow \mathcal{L}(E_1) \text{ by}$$

$$f_{E_0, E_1}(x) = \{e \in E_1 \mid [e] \cap E_0 \subseteq x\}.$$

To intuitively justify the function f_{E_0, E_1} suppose an event of E occurs once the appropriate "reading" events in E_0 can occur through input having been supplied. It is clear that:

Lemma 5.6.3

The function f_{E_0, E_1} defined above is monotonic.

However in general the function will not be continuous. We give examples below. According to Scott's thesis it should be; furthermore it should be for any choice of E_0 and E_1 with $E \leq_0 E_0$ and $E \leq_0 E_1$. Intuitively such event structures are those consistent with Scott's thesis, they respect continuity.

Definition 5.6.4

Let E be an elementary event structure. Say E is continuity-respecting iff

$$\forall E_0, E_1 (E \leq_0 E_0 \ \& \ E \leq_0 E_1 \Rightarrow f_{E_0, E_1} \text{ is continuous}).$$

Such event structures have a familiar characterisation.

Theorem 5.6.5

Let E be an elementary event structure. Then E is continuity-

respecting iff $\forall e \in E \ | [e] | < \infty$.

Proof

Let (E, \leq) be an elementary event structure.

" \Rightarrow " Suppose E is continuity-respecting i.e.

$\forall E_0, E_1 (E \leq_0 E_0 \ \& \ E \leq_0 E_1 \Rightarrow f_{E_0, E_1}$ is continuous). Suppose for some e in E we had $[e]$ infinite.

Take $E_0 = \{e' \in E \mid e' < e\}$ and \leq_0 the identity relation on E_0 . Take $E_1 = \{e\}$. Define S to be all finite subsets of E_0 . Then S is a directed set in $\mathcal{L}(E_0)$. Moreover no element of S is E_0 as E_0 is infinite. However then $f_{E_0, E_1}(\bigcup S) = \{e\}$ while

$\bigcup f_{E_0, E_1} S = \emptyset$ in $\mathcal{L}(E_1)$. Thus f_{E_0, E_1} is not continuous, contradicting the fact that E is continuity-respecting. Thus $[e]$ is finite for all e in E .

" \Leftarrow " Suppose $[e]$ is finite for all e in E . Assume $E \leq_0 E_0$ and $E \leq_0 E_1$. Let S be a directed set of $\mathcal{L}(E_0)$. Abbreviate f_{E_0, E_1} to f . As f is always monotonic we have $\bigcup f S \subseteq f(\bigcup S)$. Suppose $e \in f(\bigcup S)$. Then $[e] \cap E_0 \subseteq \bigcup S$. As $[e]$ is finite so is $[e] \cap E_0$. Thus because S is directed $[e] \cap E_0 \subseteq S$ for some s in S . Then $e \in f(s)$. This gives $f(\bigcup S) \subseteq \bigcup f S$ so $f(\bigcup S) = \bigcup f S$. Therefore f is continuous and E is continuity-respecting as required. ■

If the notion \leq were used instead of \leq_0 in the definition of continuity-respecting the corresponding weaker characterisation would be that the event structure E satisfies:

$$(1) \ \forall e, e' \in E (e' < e \Rightarrow \exists e'' \in E \ e' \leq e'' \prec e)$$

$$(2) \ \text{For } e \text{ in } E \text{ if } A \text{ is a pairwise incomparable subset of } [e]$$

then A is finite.

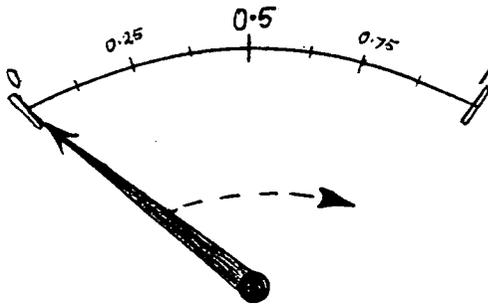
(We use \prec to mean the covering relation in E i.e. $e \prec e'$ iff $e < e' \ \& \ \forall e'' (e \leq e'' \leq e' \Rightarrow e'' = e \text{ or } e'' = e')$.)

In this context axiom F on domains is a consequence of Scott's thesis. Of course we do not expect axiom F to apply to domains in general, such as function spaces; our argument depended on the domains being of basic input or output values where increased information corresponded to later behaviour in time.

The theorem is a little surprising - continuity-respecting event

structures are discrete! How is it that non-discrete event structures, (e.g. the reals) have been ruled out? It might be thought due to taking $\mathcal{L}_0(E)$ as the domain of information even when the event structure represents a "continuous" computation. The following example suggests not and that in order to extend the notion of continuity-respecting to "continuous" event structures the relation \leq should be restricted in accord with some topological structure. (The causal order should follow or at least be closely related to the topological structure.) Appropriate mathematics might be [Nac] and [Cts. lat.].

Example 5.6.6



We consider two very simple analogue computations based on a meter which may indicate any real value in $[0,1]$. We assume the indicator is initially at zero and that the value indicated can only increase in time. It is natural to associate the meter with the event structure $E = [0,1]$ ordered by \leq on reals. The event e in $[0,1]$ stands for "the value e is indicated".

For the first computation suppose we know nothing further about the meter; regard it as a datatype. Then two kinds of deflection of the indicator are possible; it may deflect to some real value e in $[0,1]$ and stay there or it may deflect so as to approach closer and closer to some real value e in $[0,1]$ but never reach it. The two kinds of deflection give information $[0,e]$ and $[0,e)$ respectively. Thus in this situation $\mathcal{L}_0([0,1])$ is appropriate as the domain of information.

For the second computation the indicator makes a maximum deflection to value 1. (By the way is $[0,1]$ now more appropriate than $\mathcal{L}_0([0,1])$ as the domain of information of E ?) For some r in $(0,1]$ take $E_0 = [0,r)$ ordered by \leq and $E_1 = \{1\}$. The f_{E_0, E_1} is not continuous. However choosing E'_0 of the form $([0,r], \leq)$ and $E_1 = \{1\}$

does give $f_{E'_0, E_1}$ continuous. The set E'_0 is closed while E_0 is not. Thus it is hoped that by restricting \leq_0 according to topological structure the functions $f_{E'_0, E_1}$ will be continuous.

So Scott's ideas imply axioms on event structures. Can we interpret isolated elements as some form of observable state? Yes, by the results of 5.3, but only if we accept that the event structure is an abstraction from one of finite width. Note that Scott's thesis does not seem to tell us, for example, how to interpret an event structure consisting of ω incomparable events, if it should be regarded as an abstraction from a finite width event structure or whether all the events can occur in finite time. However by theorem 5.6.5 it does imply that no event can occur if it depends on an infinite set of events occurring. In this sense a computation cannot recognise or observe in finite time that the infinite set of events has occurred; only the isolated elements can be so observed as is formalised in the next lemma.

Lemma 5.6.7

Let E be an elementary event structure such that $[e]$ is finite for all events e (i.e. E is continuity-respecting). Then for $x \in \mathcal{L}(E)$, $x \in \mathcal{L}(E)^0$ iff $\exists E' \leq_0 E$ ($\forall e' \in E' \mid [e'] \mid < \infty$) & ($\exists e' \in E'$
 $x = \{e \in E' \mid e < e'\}$)

Proof

" \Rightarrow " To get E' adjoin an event e' above the finite set of events x .

" \Leftarrow " Given the r.h.s. x is finite so isolated. ■

For a very simple situation, it says isolated elements correspond precisely to information which can cause an event to occur, that is can be "observed" by a computation. This intuition is held for isolated elements of domains of a far more general nature - isolated elements are regarded as finite information. Appropriately there will be more general results (with more difficult proofs).

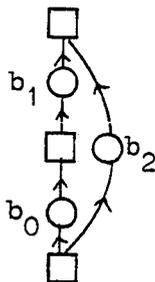
As a final remark it should be possible to cast Scott's thesis in the form: Behaviour over infinite time is the "limit" of the behaviours over finite times. As such it would be seen to express a physical principle.

Chapter 6. Conditions

In the previous chapters we have dealt only a little with conditions. In net theory they have three main uses: To mark conflict; as part of the modelling process where they stand for physical or abstract states; to define a case, a notion of state. In this chapter we interpret conditions having extents in time. In the first section we show how to associate conditions with an event structure and study an intuitive relation on conditions. It yields a new construction of a net from an event structure. In the second section we introduce the idea of an expressiveness relation on nets; roughly one net is more expressive than another if it supports more interpretations. Expressiveness provides a characterisation of the new net-construction from an event structure - the third section. Finally we look briefly at the extra structure on an event structure which distinguishes certain events as being "restless" (recall such events cannot have concession forever). This seems to involve a kind of generalised condition.

6.1 Conditions of an event structure

We illustrate some basic ideas by examining conditions of a causal net. Consider this simple causal net:



A condition is associated with its pre and post events. In fact if the net is condition-extensional (i.e. $\cdot b = \cdot b' \ \& \ b = b' \Rightarrow b = b'$), as this one is, the association is a 1-1 correspondence. The pre-event of a condition marks the beginning of the condition holding. The post-event marks the end of the condition holding. Regard a condition's holding as having an extent in time. Then clearly whenever b_0 or b_1 holds so too does b_2 . Of course for causal nets this is easy to formalise in terms of the pre and post events of conditions.

Definition 6.1.1

Let $N = (B, E, F)$ be a condition-extensional causal net. Define $b \triangleleft b'$ iff $\cdot b' \leq \cdot b$ & $b' \leq \cdot b'$

Recall the idea of essential conditions of a causal net in 4.1. A condition was said to be essential iff it occurred (to within condition-extensionality) in every net inducing the elementary event structure. In 4.1.17 these were characterised as those conditions b such that $\cdot b'$ covered $\cdot b$ in the associated event structure i.e. b is \triangleleft -minimal.

Lemma 6.1.2

Let E be an elementary event structure of finite depth (or well-based). Let b be a condition of a net inducing E . Then b is \triangleleft -minimal iff every causal net N inducing E has a condition b' s.t. $\cdot b' = \cdot b$ & $b' = \cdot b'$. Also for any causal net N such that $\mathcal{E}(N) = E$ the subnet determined by its \triangleleft -minimal conditions induces E .

Thus the \triangleleft -relation enables us to construct the minimum condition-extensional causal net inducing an elementary event structure of finite depth. We look for occurrence-net analogues of these ideas.

In 4.2 we showed how to produce a net $\mathcal{N}(E)$ from an event structure E . The net was the maximum condition-extensional net preserving the underlying event structure E . We pick out part of its construction as a definition.

Definition 6.1.3

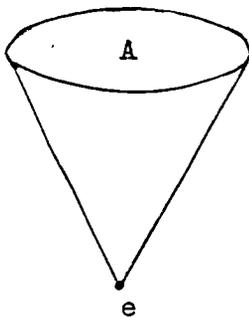
Let $E = (E, \leq, \times)$ be an event structure. Define $\mathcal{K}(E) = \{A \subseteq E \mid \forall a_1, a_2 \in A \ a_1 \times a_2 \cup \{a_2\}\}$. Then define the conditions of E by

$$\mathcal{B}(E) = \{(e, A) \mid e \in E \ \& \ A \in \mathcal{K}(E) \ \& \ e < A\} \cup \{(0, A) \mid A \in \mathcal{K}(E)\}$$

(We use $e < A$ to abbreviate $\forall a \in A \ e < a$. It is convenient to regard the symbol 0 as a fictitious starting event below all other events and by convention we shall regard it as a member of every left-closed subset of E .)

Recall from chapter 4 that the conditions of a condition-

extensional net inducing E can be regarded as a subset of $\mathcal{B}(E)$.
We shall sometimes draw a condition (e,A) as a "cone", like:



A condition holding is associated with the condition beginning and not having ended. It is easy to formalise the idea. (Recall the conventions concerning the fictitious starting event 0 .)

Definition 6.1.4

Let E be an event structure. Suppose $b \in \mathcal{B}(E)$ of the form $b = (e,A)$ and $C \in \mathcal{L}_0(E)$. Then define

$\text{beg}(b,C)$ iff $e \in C$

$\text{end}(b,C)$ iff $A \cap C \neq \emptyset$

$\text{on}(b,C)$ iff $e \in C$ & $A \cap C = \emptyset$

For b a condition and C a member of $\mathcal{L}_0(E)$ the predicate $\text{beg}(b,C)$ means b has begun to hold for C , $\text{end}(b,C)$ that has begun and ended holding while $\text{on}(b,C)$ means that b holds at C , it has both begun and not yet ended.

From these basic predicates we can construct relations between conditions. For example here are some familiar ones:

Lemma 6.1.5

Let E be an event structure. Suppose $b = (e,A)$ and $b' = (e',A')$ are conditions of E and so conditions of $\mathcal{N}(E)$.

Then

1. $b \text{cob} b'$ iff $\exists C \in \mathcal{L}_0(E)$ $\text{on}(b,C)$ & $\text{on}(b',C)$
iff $\neg(e \times e')$ & $(A \cup A') \cap ([e] \cup [e']) = \emptyset$
2. $b \# b'$ iff $\forall C \in \mathcal{L}_0(E)$ $\text{beg}(b,C) \Rightarrow \neg \text{beg}(b',C)$
iff $e \times e'$
3. $b \text{F}^* b'$ iff $\forall C \in \mathcal{L}_0(E)$ $\text{beg}(b',C) \Rightarrow \text{end}(b,C)$
iff $\exists a \in A$ $a \leq e'$.

Proof

Trivial consequence of the definitions. ■

There is a natural partial order on conditions, called \rightarrow , which has this intuitive interpretation: For conditions b and b' of an event structure, $b \rightarrow b'$ iff whenever b holds b' holds too.

Definition 6.1.6

Let E be an event structure. Define the relation \rightarrow on conditions of E by: For b and b' conditions of E ,
 $b \rightarrow b'$ iff $\forall C \in \mathcal{L}(E) (on(b,C) \Rightarrow on(b',C))$.

In the next lemma we characterise \rightarrow and as a corollary show it is a partial order. We also show that for event structures of finite depth the relation \rightarrow could have been defined equivalently by restricting *quantification* to the observable states. This means $b \rightarrow b'$ iff whenever b is observed to hold b' is observed to hold. (One could formalise this further by extending our definition of observer to conditions of the event structure - a condition would be observed after the occurrence of its pre-event and before the occurrence of any of its post-events.)

Lemma 6.1.7

Let E be an event structure. Let $b = (e, A)$ and $b' = (e', A')$ be conditions of E . Then

(1) $b \rightarrow b'$ iff $e' \leq e \wedge \forall a' \in A' (a' \not\leq e \text{ or } \exists a \in A a \leq a')$.

(2) If E is of finite depth then

$b \rightarrow b'$ iff $\forall C \in \mathcal{O}(E) on(b,C) \Rightarrow on(b',C)$.

Proof

Suppose $b = (e, A)$ and $b' = (e', A')$ are conditions of the event structure E .

(1) " \Rightarrow " Assume $b \rightarrow b'$. Take C in $\mathcal{L}(E)$ to be $[e]$. Then $on(b, [e])$ so $on(b', [e])$. Thus $e' \leq e$. Take a' in A' . Assume $\neg (a' \leq e)$. Then $C \stackrel{\text{def}}{=} [a'] \cup [e] \in \mathcal{L}(E)$. As $\neg on(b', C)$ we also have $\neg on(b, C)$. This means either $e \notin [a'] \cup [e]$, clearly impossible, or $A \cap ([a'] \cup [e]) \neq \emptyset$. Thus $\exists a \in A a \leq a'$.

(1) " \Leftarrow " Assume the r.h.s. of (1) above. Suppose $on(b, C)$ for some C in $\mathcal{L}(E)$. Then $e \in C$ and $A \cap C = \emptyset$. Thus $e' \in C$. If $a' \in C$ for some a' in A' then by the r.h.s. either $a' \leq e$

contradicting the consistency of C or $A \wedge C \neq \emptyset$ a contradiction. Thus $A' \wedge C = \emptyset$. Therefore $\text{on}(b', C)$.

(2) Suppose E is of finite depth. Now (2) is clear as all the elements of $\mathcal{L}_o(E)$ used in the above proof are then observable. ■

Corollary 6.1.8

The relation \rightarrow is a partial order.

Proof

Reflexivity and transitivity were already clear. To show antisymmetry suppose we have $(e, A) \rightarrow (e', A') \rightarrow (e, A)$ for conditions (e, A) and (e', A') of an event structure. By the above $e = e'$ immediately. Take $a' \in A'$. As $\neg(a' \times e')$ for some a in A we have $a \leq a'$. Similarly for some a'' in A' we get $a'' \leq a$. Therefore $a'' \leq a \leq a'$ with $a' \times a''$. Thus $a = a'$. This shows $A' \subseteq A$ and the converse $A \subseteq A'$ follows the same way giving $A = A'$. Therefore $(e, A) = (e', A')$ as required. ■

Concurrency propagates upwards under \rightarrow . Formally:

Lemma 6.1.9

Let E be an event structure. Let co be the concurrency relation on $\mathcal{N}(E)$. Then for b, b', b'' in $\mathcal{S}(E)$ we have

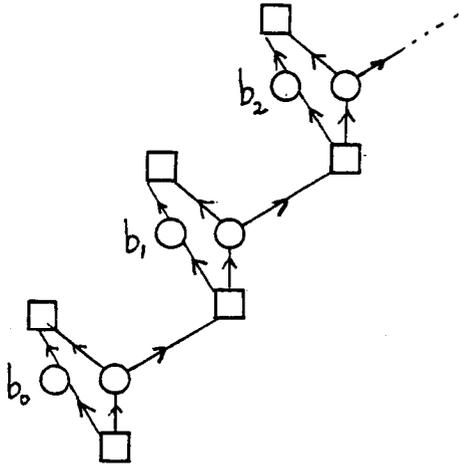
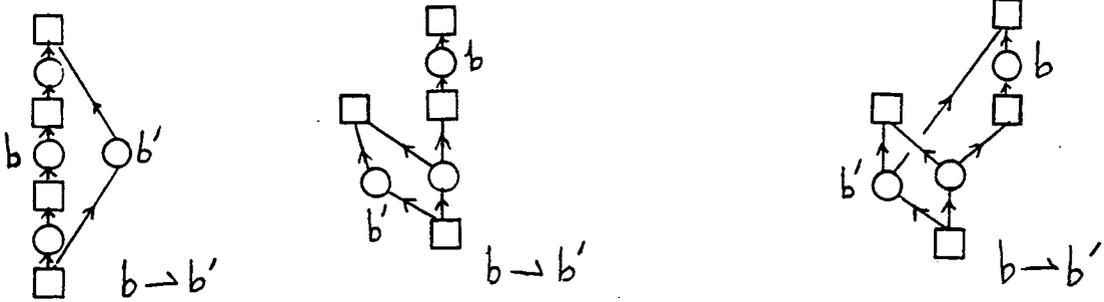
$$b \text{ co } b' \rightarrow b'' \Rightarrow b \text{ co } b''$$

Proof

Clear as the concurrency relation may be equivalently expressed by $b \text{ co } b'$ iff $\exists C \in \mathcal{L}_o(E)$ $\text{on}(b, C) \& \text{on}(b', C)$. ■

We illustrate the relation \rightarrow with some examples.

Example 6.1.10



Here... $b_n \dots \rightarrow b_2 \rightarrow b_1 \rightarrow b_0$. (Thus the relation \rightarrow may "propagate high up" the net.)

The last example shows how "non-local" is the relation \rightarrow . We now define a "local" subrelation of \rightarrow called \triangleleft - soon we shall justify extending the notation of 6.1.1. We use \triangleleft to construct a net $\mathcal{N}(E)$ from an event structure E ; the net $\mathcal{N}(E)$ will express conflict in an economical way. In fact we shall show its conditions are essential in some generalised sense over an important subclass of occurrence nets, those which are maximally expressive. Clearly from example 6.1.10 if $b \rightarrow b'$ then it is possible for b' to end holding without b ever having held. By restricting \rightarrow to \triangleleft this is forbidden: if $b \triangleleft b'$ and b' ends holding then b must have held for a subinterval of the time that b' held.

Definition 6.1.11

Let E be an event structure. For subsets A, A' of E define

$$A \sqsubseteq_0 A' \text{ iff } \forall a' \in A' \exists a \in A a \leq a'.$$

Then for conditions $b = (e, A)$ and $b' = (e', A')$ of E define

$$b \triangleleft b' \text{ iff } e' \leq e \ \& \ A \sqsubseteq_0 A'$$

(Recall the convention for 0.) The definition of $b \triangleleft b'$ has two parts; the first says if b has started holding then so has b' ; the second that if b' has ended holding then so has b (started and) ended holding. The relation \triangleleft is a partial ordering. (In fact so is \subseteq_0 when restricted to $\mathcal{K}(E)$.)

Lemma 6.1.12

Let E be an event structure. The relation \triangleleft is a subpartial order of \rightarrow . Suppose $b = (e, A)$ and $b' = (e', A')$ are conditions of E . Then

1. $A \subseteq_0 A'$ iff $\forall C \in \mathcal{L}_0(E) \text{ end}(b', C) \Rightarrow \text{end}(b, C)$
2. $b \triangleleft b'$ iff $b \rightarrow b'$ & $\forall C \in \mathcal{L}_0(E) (\text{end}(b', C) \Rightarrow \text{end}(b, C))$
3. $b \rightarrow b'$ & $\bullet b = \bullet b' \Rightarrow b \triangleleft b'$

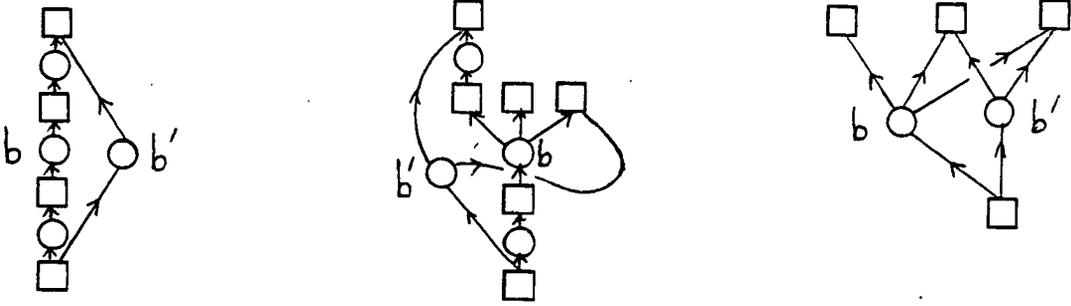
Finally for E an elementary event structure $\triangleleft = \rightarrow$ and \triangleleft coincides with the relation in 6.1.1 for $\mathcal{W}(E)$.

Proof

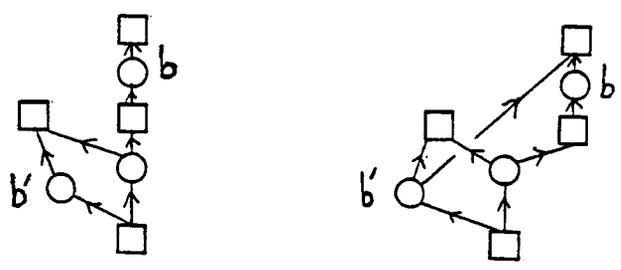
By the characterisation of \rightarrow we have \triangleleft is a subpartial order of \rightarrow . Properties 1., 2. and 3. follow in an obvious way from the definitions. The conditions of an elementary event structure are always of the form (e, A) where A is null or a singleton. This gives the final remark. ■

We illustrate \triangleleft with some examples.

Example 6.1.13



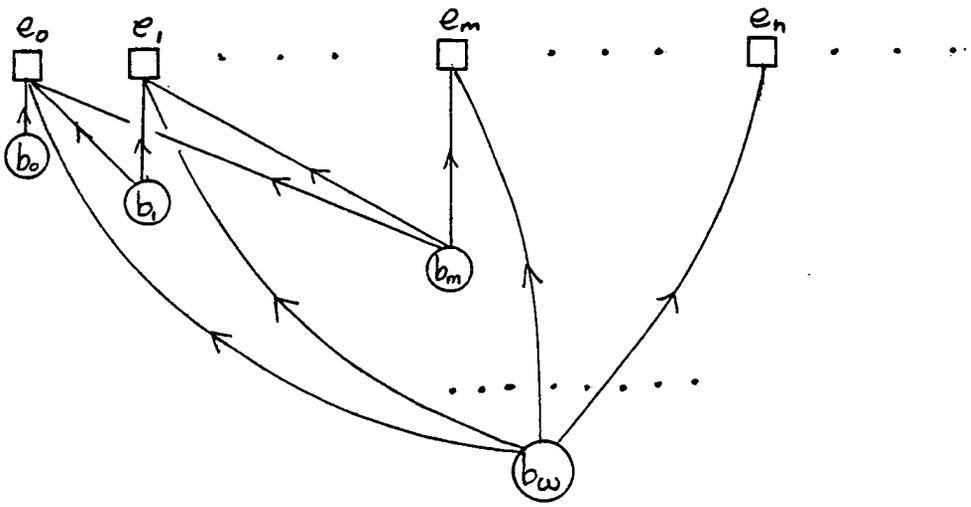
$b \leq b'$



$b \rightarrow b' \& b \not\leq b'$

The following example shows \leq is not well-founded in general, even for event structures of finite depth.

Example 6.1.14

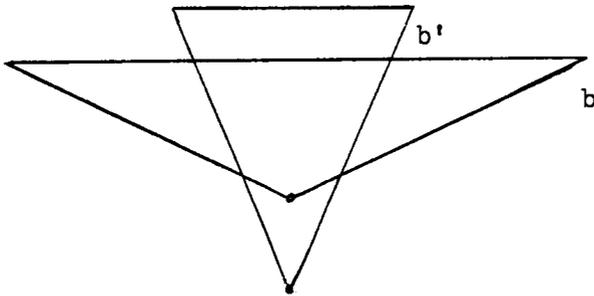


The event structure consists of an infinite set $\{e_i \mid i \in \omega\}$ of pairwise conflicting events. Clearly $b_m = (0, \{e_i \mid i \leq m\})$, for $m \in \omega$, is a condition as is $b_\omega = (0, \{e_i \mid i \in \omega\})$. Obviously $b_\omega \trianglelefteq \dots \trianglelefteq b_n \trianglelefteq \dots \trianglelefteq b_1 \trianglelefteq b_0$.

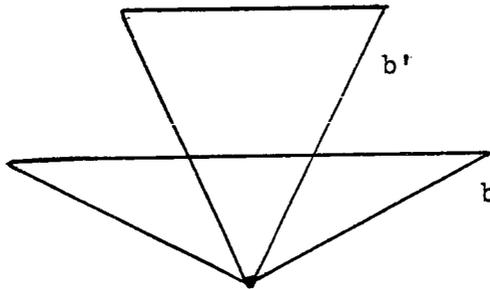
So we see the ordering \trianglelefteq is not well-founded in general. Assume E is an event structure which is well-based (5.6.1), implied, of course, if E is of finite depth. Then there are sufficient \trianglelefteq -minimal conditions to determine the event structure. In fact then \trianglelefteq will be atomic in the following sense:

$$\forall b \exists b' \trianglelefteq b \text{ } b' \text{ is } \trianglelefteq\text{-minimal.}$$

The relation $b \trianglelefteq b'$ on two conditions b, b' of E may be pictured as:



In subsequent work we shall use a particular form of \trianglelefteq -minimal condition below b' . Suppose b' is (e, A') . Then there is a \trianglelefteq -minimal condition $b = (e, A)$ with $b \trianglelefteq b'$. Pictorially it looks like:



The condition b begins to hold when b' does but may end before. We show the existence of such a condition b as a corollary to the following.

Lemma 6.1.15

Let E be an event structure so \leq is well-founded. Suppose $(e, A) \in \mathcal{D}(E)$. Then the set $\{A' \in \mathcal{K}(E) \mid e < A' \sqsubseteq_0 A\}$ has a \sqsubseteq_0 -minimal element.

Proof

Let E be an event structure so \leq is well-founded. Suppose $(e, A) \in \mathcal{B}(E)$. We show \sqsubseteq_0 -descending chains in $\{A' \in \mathcal{K}(E) \mid e < A' \sqsubseteq_0 A\}$ have a lower bound in the set. The result then follows by Zorn's lemma.

Let $\{A_\gamma \mid \gamma \in \Gamma\}$ be such a chain indexed by a total order Γ . Define A^* to be the \leq -minimal elements of $\bigcup_{\gamma \in \Gamma} A_\gamma$. By the well-foundedness of \leq we have $A^* \sqsubseteq_0 \bigcup_{\gamma \in \Gamma} A$.

In fact $A^* \in \mathcal{K}(E)$: For suppose $e, e' \in A^*$. Then $e' \in A_{\gamma_1}$ and $e \in A_{\gamma_2}$ where w.l.o.g. $A_{\gamma_1} \sqsubseteq_0 A_{\gamma_2}$. But then $e \in A_{\gamma_1}$ by the definitions of A^* and \sqsubseteq_0 . Thus as $A_{\gamma_1} \in \mathcal{K}(E)$ we have $e \not\leq e'$ so $A^* \in \mathcal{K}(E)$.

Obviously $e < A^*$. Thus we have the desired lower bound. ■

Corollary 6.1.16

Let E be an event structure so \leq is well-founded. Suppose $b = (e, A)$ is a condition of E . Then there is a \triangleleft -minimal element b^* of the form $b^* = (e, A^*)$ with $b^* \triangleleft b$.

Proof

Suppose $b = (e, A)$ is in $\mathcal{B}(E)$. Take A^* to be a \sqsubseteq_0 -minimal element of $\{A' \in \mathcal{K}(E) \mid e < A' \sqsubseteq_0 A\}$. Define $b^* = (e, A^*)$. If $b' = (e', a') \triangleleft (e, A^*)$ we have $e \leq e'$ and $A' \sqsubseteq_0 A^*$ with $e' < A'$. Thus $A' = A^*$. Supposing $e < e'$ then implies $e < \{e'\} \not\sqsubseteq_0 A^*$ contradicting the definition of A^* . Thus b^* is \triangleleft -minimal as required. ■

In example 6.1.14 b corresponds to any b_n and b^* to b_ω . The condition b_ω was formed from a ken of $\times \cup 1$ above the "event" 0. This is true in general.

Lemma 6.1.17

Let E be an event structure. Suppose $e \in E \cup \{0\}$ and $A \in \mathcal{K}(E)$. Then any \sqsubseteq_0 -minimal element of $\{A' \in \mathcal{K}(E) \mid e < A' \sqsubseteq_0 A\}$ in $\mathcal{K}(E)$ is a ken of $\times \cup 1$ in $\{e' \in E \mid e < e'\}$.

Proof

Suppose A^* in $\mathcal{K}(E)$ is a \sqsubseteq_0 -minimal element described above. Certainly $\forall e, e' \in A^* e \not\leq e'$. Suppose A^* were not a ken. Then A^* may be strictly extended to a ken B . But then $B \not\sqsubseteq_0 A$, a

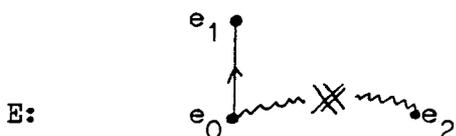
contradiction. ■

Corollary 6.1.18

Let E be an event structure. Suppose b is a \triangleleft -minimal condition of E . Then for some event e we have $b = (e, A)$ where A is a ken of $\times \cup 1$ in $\{e' \in E \mid e < e'\}$.

Note it is not true that any ken A of $\times \cup 1$ in $\{e' \mid e < e'\}$ for some event e always arises from such \triangleleft -minimal condition. This is shown by the next simple example:

Example 6.1.19



The ken of $\times \cup 1, \{e_1, e_2\}$ can never appear as a \triangleleft -minimal condition. Such a condition must be of the form $(0, \{e_1, e_2\})$. However clearly $(0, \{e_0, e_2\}) \triangleleft \neq (0, \{e_1, e_2\})$.

We can now show that the net formed from an event structure by taking the \triangleleft -minimal conditions induces the original event structure provided it is well-based. First we formally define the net construction. Note $\mathcal{N}(E)$ does not have the isolated condition $(0, \emptyset)$ possessed by $\mathcal{N}(E)$ unless E is null.

Definition 6.1.20

Let E be an event structure. Define $\mathcal{b}(E)$ to be the \triangleleft -minimal conditions of $\mathcal{B}(E)$. Define $\mathcal{N}(E)$ to be the occurrence net with events E , conditions $\mathcal{b}(E)$ and causal dependency relation F given by

$$eFb \text{ iff } e = (b)_0$$

$$\text{and } bFe \text{ iff } e \in (b)_1$$

for e in E and b in $\mathcal{b}(E)$.

Theorem 6.1.21

Suppose E is an event structure which is well-based. The net $\mathcal{N}(E)$ is a condition extensional occurrence net satisfying $N3$ and $\mathcal{C}(\mathcal{N}(E)) = E$.

Proof

Let E be a well-based event structure. It is obvious that \mathcal{N}

yields a condition extensional occurrence net. We show

$$\mathcal{E}(\mathcal{N}(E)) = E \text{ and } \mathcal{N}(E) \text{ satisfies N3.}$$

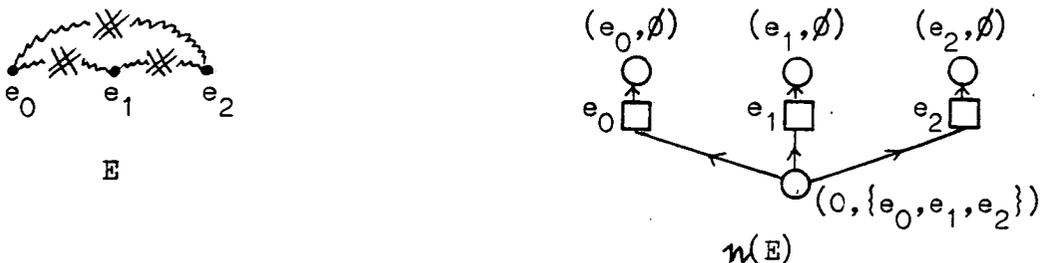
Obviously eF^*e' in $\mathcal{N}(E)$ implies $e \leq e'$. The converse follows by induction on the length of chain using corollary 6.1.16. If for some b in $\mathcal{C}(E)$, $b \in F^{*-1}\{e\} \cap F^{*-1}\{e'\}$ in $\mathcal{N}(E)$ then $e \not\ll e'$. Conversely supposing $e \not\ll e'$, take $e'' \leq$ -maximal in $\{e \in E \mid e < e, e'\}$. Using corollary 6.1.16 there is a condition $b^* = (e, A) \trianglelefteq (e'', \{e', e''\})$. Then by the choice of e'' as $e \not\ll e'$ we have e, e' in A so $b^* \in \cdot e \cap \cdot e'$ in $\mathcal{N}(E)$. Thus $\mathcal{E} \circ \mathcal{N}(E) = E$.

For an event e there is a condition (e, \emptyset) . Then using corollary 6.1.16 there is b in $\mathcal{C}(E)$ with $b = (e, A) \trianglelefteq (e, \emptyset)$. Thus $\cdot e \neq \emptyset$ in $\mathcal{N}(E)$. To show $\cdot e \neq \emptyset$ let e' be \leq -maximal in $\{e \in E \cup \{0\} \mid e < e\}$. Then $(e', \{e\})$ is a condition. Using 6.1.16 we produce b in $\cdot e$. Therefore $\mathcal{N}(E)$ satisfies N3. ■

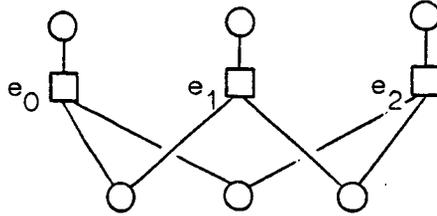
The construction of $\mathcal{N}(E)$ is natural, at least mathematically. We shall characterise it later in section 6.3. For the time being we point out why a few obvious conjectures fail.

As earlier when we looked at causal nets we may define a condition to be essential iff it belongs to every net inducing the event structure. Because there are so many different ways to express the same conflict by conditions rarely are sufficient conditions essential to recover the underlying event structure from them. For instance any pairwise conflict between three events can be expressed at least two ways by conditions as is shown in the next example.

Example 6.1.22



As the same event structure is induced by

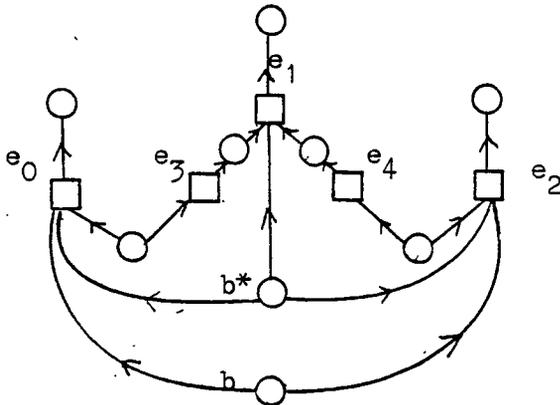


the condition $(0, \{e_1, e_2, e_2\})$ is not essential.

In section 6.3 however we shall show that \triangleleft -minimal conditions are essential for a suitable subclass of nets namely those which are "maximally expressive".

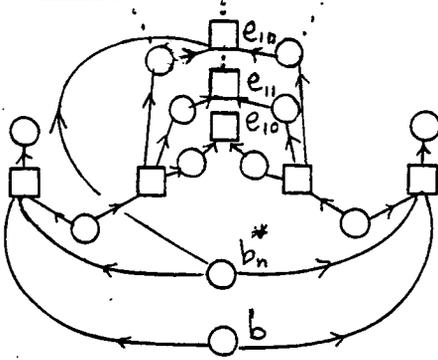
Note that \triangleleft -minimal conditions do not always express immediate conflict (denoted \otimes_μ) between events. Here is an example showing this.

Example 6.1.23



The induced event structure of this occurrence net is clear. The conditions b^* and b are identified as $(0, \{e_0, e_1, e_3\})$ and $(0, \{e_0, e_2\})$. The condition b^* is \triangleleft -minimal (and $b^* \triangleleft b$) yet, while $e_0 \otimes_\mu e_2$, we do not have $e_0 \otimes_\mu e_1$ or $e_1 \otimes_\mu e_2$. (Note the above net is symmetrically confused - consider e_0, e_2, e_4 .)

This example serves as a basis for the next example in which e_1 above has been replaced by an infinite conflict-free set of events. This means there will be an infinite number of copies of b^* each a \triangleleft -minimal condition.

Example 6.1.24

The event e_1 of 6.1.23 has been replaced by $\{e_{1n} \mid n \in \omega\}$. Correspondingly there are an infinite number of copies of b^* written $b_n^* (n \in \omega)$. Here $\forall n \ b_n^* \leq b$.

Thus in general there are far more (possibly infinitely more) \leq -minimal conditions than are needed to express the underlying event structure. This example also shows that the net $\mathfrak{N}(E)$ may be such that $\bullet e$ is infinite for an event e even though there exists a net N , such that $\mathfrak{S}(N) = E$, with a finite number of preconditions for each event.

Definition 6.1.25

Say a Petri net $N = (B, E, F)$ has finite-preconditions iff for all events e we have $\bullet e$ finite.

Say an event structure E satisfies the finite-preconditions property iff there is an occurrence net N having finite-preconditions such that $\mathfrak{S}(N) = E$.

The following gives a characterisation of the finite-preconditions property for event structures. It refers to the immediate conflict relation \times_{μ} of 5.5.

Lemma 6.1.26

Let E be a countable event structure of finite depth. Then E satisfies the finite pre-conditions property iff (i) $\forall e \in E \ | [e] | < \infty$ and (ii) $\exists A_1, \dots, A_n \in \mathcal{K}(E) \times_{\mu} \cup \{e\} = \bigcup_{i \leq n} A_i$.

Proof

Let E be a countable event structure of finite depth.

Assume E satisfies the finite preconditions property. Assume $[e]$ is infinite for some event e . Without loss of generality suppose e is of minimal depth so that $[e]$ is infinite. Then e covers an infinite number of events in the ordering \leq . Thus any net inducing E must have $\bullet e$ infinite, a contradiction. Therefore $[e]$ is finite for all events e . To show (ii) consider any event e . In some net

inducing E we have $e = \{b_1, \dots, b_n\}$. If $e \not\ll \mu e'$ we have $b_i \in e$ and $b_i \notin e'$ for some i . Thus taking $A_i = b_i$ gives property (ii).

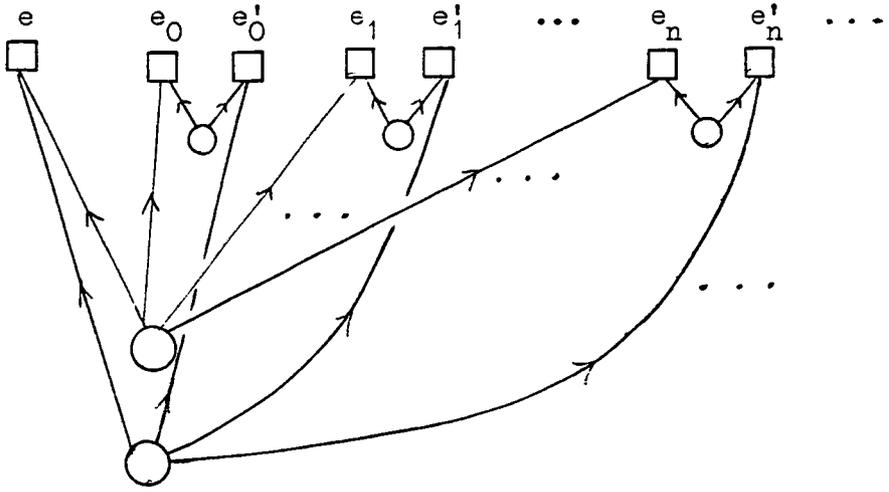
Conversely assume properties (i) and (ii) above hold. We give a very crude construction of a net having finite preconditions and inducing E. We determine it by determining its conditions. First we include all conditions of the form $(e, \{e'\})$ where e' covers e for the \leq -ordering - this ensures the net induces the partial order \leq . So that it induces the conflict relation $\not\ll$ while maintaining finite preconditions first enumerate E as $e_0, e_1, \dots, e_n, \dots$. By (ii) we have for any m that there are A_1^m, \dots, A_n^m with $\not\ll \mu \{e_m\} = \bigcup_{i \leq n_m} A_i^m$. Clearly we may assume $e_m \in A_i^m$.

Inductively add these conditions: Initially add the finite set $\{(O, A_1^0), \dots, (O, A_n^0)\}$ as preconditions of e_0 ; subsequently add the finite set $\{(O, A_i^m \setminus \{e_0, \dots, e_{m-1}\}) \mid 0 < i < n_m\}$ as preconditions of e_m . By the construction, for a particular event, no extra preconditions are added after a finite stage in the induction. Thus the net determined has finite preconditions. ■

The above proof is a bit unsatisfying. The net constructed depends on the countable enumeration of E. It is hard to see a more canonical definition or construction (on the lines of the definition of $\mu(E)$) for the general class of countable event structures with the finite preconditions property. The following example illustrates the difficulty.

Example 6.1.27

The net below has finite preconditions.



The net consists of an infinite set of pairs of conflicting events e_i, e'_i with $\{e\} \cup \{e_i \mid i \in \omega\}$ and $\{e\} \cup \{e'_i \mid i \in \omega\}$ pairwise conflicting. Note that the sets $\{e, e_i, e'_i\}$ are kens of $\mathbb{X} \cup \{ \}$ and there are associated conditions. If included, e would have an infinite set of preconditions and the associated net would not have finite preconditions. Yet, it is hard to see any significant difference in kind between conditions of the form $(0, \{e, e_i, e'_i\})$ and those of the form $(0, \{e_i, e'_i\})$. Certainly the net construction n would include conditions of the former sort too.

When event structures with the finite preconditions property satisfy restrictions there may be a canonical net which has finite preconditions. Confusion-freeness is one such restriction (the next lemma) while finite width does not appear to be - the net of example 6.1.27 above is of finite width.

Lemma 6.1.28

Let E be a confusion-free event structure such that $[e]$ is finite for all events e . Then E satisfies the finite-preconditions property. In fact $n(E)$ has finite-preconditions.

Proof

Let E be a confusion-free event structure s.t. $[e]$ is finite for all events e . We show $n(E)$ has finite-preconditions. By the definition of confusion-free, the \triangleleft -minimal pre-condition of an event e will be of the form $(e', \mathbb{X}_n \cup \{e\})$ where e covers e' in the event structure with the fictitious starting event 0 adjoined. There are only finitely such conditions. ■

Of course one would prefer a similar result based on a less powerful restriction than confusion-freeness. This would further justify the net construction n .

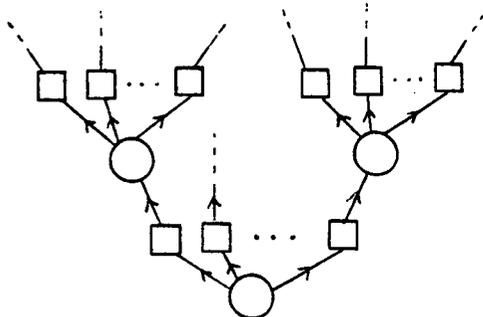
In section 4.2 we showed there were peculiarities in generalising Petri's notion of sequential process of a causal net to occurrence nets. The obvious definition, taking a sequential process of an occurrence net to be a ken of the complement of the concurrency relation, gave odd-looking subnets which did not meet every case. This was so even for finite occurrence nets! Fortunately if E is an event structure of finite depth, kens of the complement of co have a

simple form in the nets $\mathcal{N}(E)$ and $\mathfrak{n}(E)$. Then in $\mathcal{N}(E)$ and $\mathfrak{n}(E)$ a "sequential process" looks like a tree and a revised-K-density result can be proved once cases are restricted to being observable.

Definition 6.1.29

Let $N = (B, E, F)$ be an occurrence net. Say N is tree-like iff $(B, F^* \uparrow B)$ is a tree.

Note the tree may be infinite. A tree-like net has the form:



Thus tree-like nets are a generalisation of sequential processes of causal nets of finite depth (see 5.4.6). Clearly no two distinct elements of a tree-like net can be in the concurrency relation which is the complement of $(F^* \cup F^{*-1} \cup \#)$. Thus:

Lemma 6.1.30

Let $N = (B, E, F)$ be a tree-like occurrence net. Then for all x, x' in $B \cup E$ we have $x(F^* \cup F^{*-1} \cup \#)x'$ that is $x \text{ co } x' \Rightarrow x = x'$.

Now we characterise "sequential processes", regarded as the kens of the complement of co , in the nets $\mathcal{N}(E)$ and $\mathfrak{n}(E)$ for E of finite depth. They are tree-like and satisfy further conditions (a), (b) and (c) to ensure their maximality.

Proposition 6.1.31

Let E be an event structure of finite depth.

1. Let S be a subnet of $\mathcal{N}(E)$. Then S is a ken of $(F^* \cup F^{*-1} \cup \#)$ iff S is tree-like and
 - (a) For some condition b in S we have $(b)_0 = 0$.
 - (b) For all conditions b in S we have $b^* \subseteq S$ & $b^* \subseteq S \& b$ is a ken of $\mathcal{N} \cup \uparrow$ in $\{e \in E \mid (b)_0 < e\}$.
 - (c) For all events e in S we have $e^* \cap S \neq \emptyset$.
2. Let S be a subnet of $\mathfrak{n}(E)$. Then S is a ken of $(F^* \cup F^{*-1} \cup \#)$ iff S is tree-like and
 - (a) For some condition b in S we have $(b)_0 = 0$.

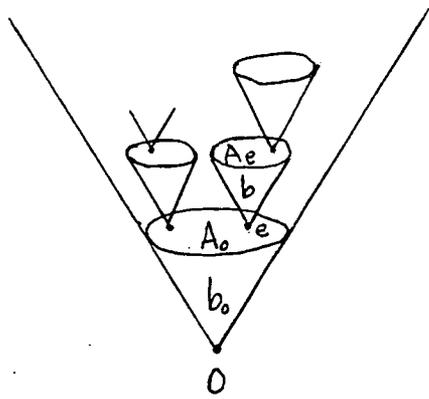
- (b) For all conditions b in S we have $\cdot b \subseteq S \& b^\circ \subseteq S$.
 (c) For all events e in S we have $e^\circ \neq \emptyset \Rightarrow e^\circ \cap S \neq \emptyset$.

Proof

Let E be an event structure of finite depth. Recall $F^* \cup F^{*-1} \cup \#$ is the complement of co in $\mathcal{N}(E)$.

1. " \Leftarrow " Assume S is a subnet of $\mathcal{N}(E)$ which is tree-like and satisfies (a), (b) and (c). As S is tree-like we clearly have $x(F^* \cup F^{*-1} \cup \#)x'$ for all x, x' in S . For S to be a ken we further require $x(F^* \cup F^{*-1} \cup \#)S$ to imply $x \in S$. Assume x is an event e and $e(F^* \cup F^{*-1} \cup \#)S$. Let b_0 be the condition of S with $(b_0)_0 = 0$. As $\neg(e \text{ co } b_0)$ we must have $e_0 \leq e$ for some e_0 in b_0° . Take b to be the F^* -maximal condition in S so that $e' \leq e$ for some e' in b° - such a b exists as e has finite depth. It follows that $e = e'$ and so $e \in S$: Suppose otherwise, that $e' < e$; then $e'^\circ \neq \emptyset$ so there is a condition b' in S with $\cdot b' = e'$; as $\neg(e \text{ co } b')$ we get $e \geq e''$ where $e'' \in b'^\circ$ contradicting the maximality of b . If x happened to be a condition b then the above argument shows $e = \cdot b \in S$. The condition in S with pre-event e is concurrent to b and so is identical with b , giving $b \in S$.

" \Rightarrow " Assume S is a ken of $(F^* \cup F^{*-1} \cup \#)$. It is inductively shown that S has a subnet S' which is tree-like and satisfies (a), (b) and (c). By the above S' is a ken so $S = S'$. As S is a ken for any b in S we have $\cdot b, b^\circ \subseteq S$. We define the subnet S' by inductively picking its conditions. Initially, let A_0 be the \leq -minimal events of $S \cap E$. Then as S is a ken of $(F^* \cup F^{*-1} \cup \#)$ we have $A_0 \in \mathcal{K}(E)$ so we may define b_0 to be the condition $(0, A_0)$. Then $b_0 \in S$ and A_0 is a ken of $\times \cup |$. We initially pick b_0 as a condition of S' . For each event e in $A_0 (= b_0^\circ)$ define A_e to be the set of \leq -minimal events in $S \cap \{e' \in E \mid e < e'\}$; then (e, A_e) is a condition in S which we include in S' . Continuing we define a tree-like subnet S' satisfying (a), (b) and (c).



2. " \leq " This follows from "1. \leq " as for a condition b of $\mathfrak{n}(E)$ we have b^* is a ken of $\mathfrak{X} \cup \{ \mid \text{in } \{e \mid (b)_0 < e\} \}$.
 " \Rightarrow " Following the induction in "1. \Rightarrow " each condition chosen will now be \leq -minimal. ■

For the special nets $\mathfrak{N}(E)$ and $\mathfrak{n}(E)$ of a finite depth event structure E we show a restricted form of \mathfrak{K} -density holds.

Proposition 6.1.32 (Restricted \mathfrak{K} -density)

Let E be an event structure of finite depth. Then for the nets $\mathfrak{N}(E)$ and $\mathfrak{n}(E)$ every ken of the complement of c_0 meets every observable case.

Proof

Let E be an event structure of finite depth. The same proof works for $N = \mathfrak{n}(E)$ or $N = \mathfrak{N}(E)$. Let S be a ken of $(F^* \cup F^{*-1} \cup \#)$ in N . Suppose $C \in \mathcal{O}_Q(E)$. By finite depth we take e to be the \leq -maximal event of S in C if such exists; otherwise take $e = 0$. Let b be the unique condition in S s.t. $(b)_0 = e$. If $\text{end}(b, C)$ then $(b)_1 \cap C \neq \emptyset$. However $(b)_1 \subseteq S$ so supposing $\text{end}(b, C)$ contradicts the maximality of e . Thus $b \in \text{Fr}(C) \cap S$ as required. ■

Note the above proof would work taking S to be a ken of $(F^* \cup F^{*-1})$; the proof depends only on S being an \leq -maximal tree-like subnet - the simplest example of such a net would be a chain $b_0 F e_0 F \dots b_n F e_n \dots$ of maximal length where $(b_0)_0 = 0$. Presumably the last two propositions also hold when finite-depth is replaced by well-based and the definition of observable state weakened appropriately.

6.2 Expressiveness

In this section we present a formal way of interpreting an occurrence net. Each condition is interpreted as asserting a conjunction of propositions. This induces an expressiveness relation between nets associated with the same event structure. Roughly one net is more expressive than another if it supports more interpretations. In the next section we shall use the ideas to characterise the construction $\mathfrak{n}(E)$ from an event structure E .

In the main our formal development is rather brutal. Many of the ideas should work to produce expressiveness relations between the more general class of transition nets with initial marking. This may open a Pandora's box of possibilities. In the final part of this section we shall sketch some of them.

Throughout we shall assume a fixed (sufficiently large) set of propositions P . We shall also assume all nets are condition - extensional and satisfy axiom $N3$ (i.e. all events have at least one pre-condition and post-condition).

Definition 6.2.1

Let N be a net (B, E, F) . An interpretation of N is a map $I: B \rightarrow \mathcal{P}(P)$. We denote the set of interpretations by $\mathcal{J}(N)$.

With respect to an interpretation I a condition b asserts all propositions $I(b)$ are true.

In general one works with interpretations satisfying restrictions (there will be examples later). Restrictions determine an interpretation class.

Definition 6.2.2

An interpretation class is a map \mathcal{J}' from nets such that for all nets

$$\mathcal{J}'(N) \subseteq \mathcal{J}(N).$$

We denote the interpretation class of all interpretations by \mathcal{J} .

An interpretation extends to markings in the obvious way.

Definition 6.2.3

Let N be a net (B, E, F) and I an interpretation of N . For

$M \subseteq B$ define

$$I(M) = \bigcup_{b \in M} I(b).$$

We summarise the idea of expressiveness (with respect to an interpretation class) in the following proposition. Here it is defined only between occurrence nets inducing the same event structure. We shall outline extensions of the idea later.

Proposition 6.2.4

Let E be an event structure of finite depth. Let \mathcal{J}' be an interpretation class. We define an expressiveness relation between nets $\{N \mid N \text{ is an occurrence net and } \mathcal{E}(N) = E\}$ by

$$N \leq_{\mathcal{J}'} N' \text{ iff } \forall I \in \mathcal{J}'(N) \exists I' \in \mathcal{J}'(N') \forall C \in \mathcal{O}(E) \\ I \circ \text{Fr}_N(C) = I' \circ \text{Fr}_{N'}(C).$$

Then $\leq_{\mathcal{J}'}$ is a preorder. Thus the relation $\approx_{\mathcal{J}'}$, defined by

$$N \approx_{\mathcal{J}'} N' \text{ iff } N \leq_{\mathcal{J}'} N' \ \& \ N' \leq_{\mathcal{J}'} N$$

is an equivalence relation.

The definition of expressiveness depends on what we take to be "essential structure" of an interpreted net. In the above definition of expressiveness we have taken it to be the interpreted observable states defined using the map Fr .

Definition 6.2.5

Let N be an occurrence net of finite depth. Let I be an interpretation of N . Then define $\mathcal{O}^I(N)$ to be the set

$$\{(C, I \circ \text{Fr}_N(C)) \mid C \in \mathcal{O} \circ \mathcal{E}(N)\}$$

with relation \rightarrow_I given by

$$(C, I \circ \text{Fr}_N(C)) \rightarrow_I (C', I \circ \text{Fr}_N(C')) \text{ iff } C \subseteq C' \ \& \ d(C, C') = 1$$

The structures $\mathcal{O}^I(N)$ are useful in establishing the relation $\leq_{\mathcal{J}'}$ between nets (see the examples below). More importantly they draw attention to a "parameter" in the definition of expressiveness pointed out in the following obvious lemma.

Lemma 6.2.6

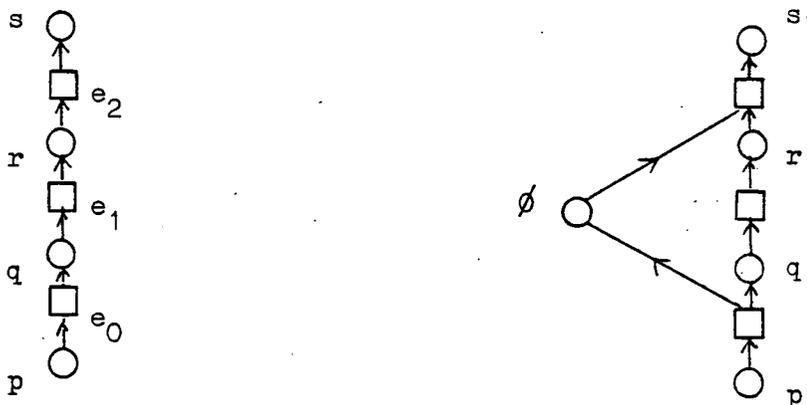
Suppose the event structure E , \mathcal{J}' , and nets N and N' are as in the definition of expressiveness (6.2.4). Then

$N \preceq_{\mathcal{J}} N'$ iff $\forall I \in \mathcal{J}(N) \exists I' \in \mathcal{J}(N') \alpha^I(N) = \alpha^{I'}(N')$

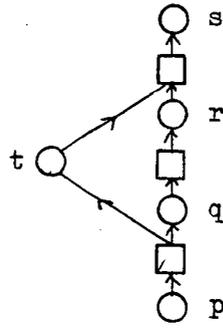
The lemma can be regarded as saying $N \preceq_{\mathcal{J}} N'$ iff for any interpretation of N there is an interpretation of N' such that the interpreted nets are equivalent or have essentially the same structure. Here that structure is taken as $\alpha^I(N)$ for an occurrence net N with interpretation I . One would get different expressiveness relations by replacing the $\alpha^I(N)$'s by different formalisations of essential structure.

We now look at some examples illustrating the expressiveness relation $\preceq_{\mathcal{J}}$ where \mathcal{J} is the interpretation class of all interpretations. Clearly for this interpretation class in establishing $N_1 \preceq_{\mathcal{J}} N_2$ we may assume the conditions of N_1 are interpreted as singletons. (This will also be the case for the other interpretation classes we deal with.)

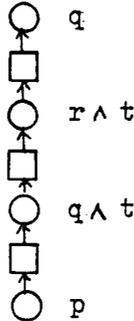
Example 6.2.7



In this example we have $N_1 \approx_{\mathcal{J}} N_2$ where \mathcal{J} is the interpretation class of all interpretations. To establish $N_1 \preceq_{\mathcal{J}} N_2$ it is sufficient to consider only those interpretations I_1 such that $I_1(b)$ is always a singleton. Above we have marked such an interpretation I_1 and an appropriate I_2 showing $N_1 \preceq_{\mathcal{J}} N_2$. To show the converse that $N_2 \preceq_{\mathcal{J}} N_1$, again a singleton interpretation I_2 of N_2 suffices. Suppose it is given as:

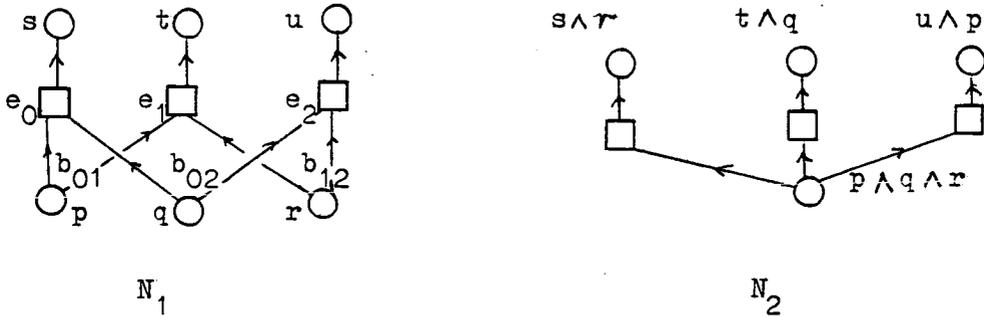


Then an appropriate I_1 establishing $N_2 \preceq_f N_1$ is:

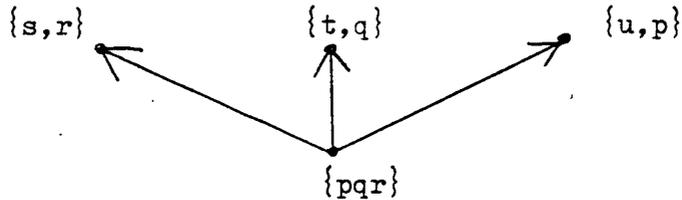


Importantly not all nets of an event structure are equally expressive as the following example shows.

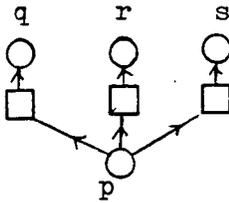
Example 6.2.8



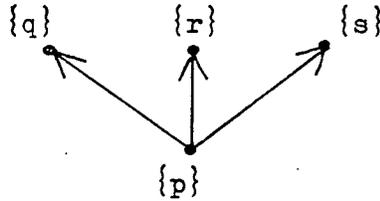
Certainly $N_1 \preceq_f N_2$: For the typical singleton interpretation of N_1 shown above the interpretation I_2 of N_2 suffices; both $\mathcal{O}_S^{I_1}(N_1)$ and $\mathcal{O}_S^{I_2}(N_2)$ take the form:



However we do not have $N_2 \preceq_f N_1$. Interpret N_2 by I_2 marked by

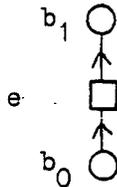


Then $\mathcal{O}_\alpha^{I_2}(N_2)$ has the form

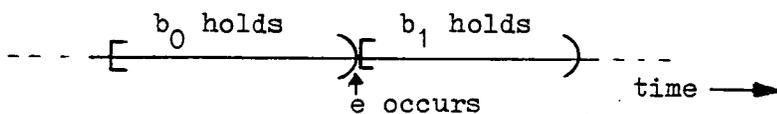


Suppose there was an interpretation I_1 of N_1 such that $\mathcal{O}_\alpha^{I_1}(N_1)$ had this form. Then without loss of generality $I_1(b_{01})$ may be supposed to contain p. But then p would hold after the occurrence of e_2 , a contradiction. Thus $N_2 \not\sim N_1$.

Consider the equivalently expressive nets of example 6.2.7. Their equivalence can be made more intuitive by assuming that event occurrences do not occupy extents in time but that they are instantaneous changes in the holdings of conditions. Consider a typical event occurrence. For simplicity assume e has only one precondition b_0 and only one postcondition b_1 so it looks like



Regard the event e as marking the end of the holding of b_0 and simultaneously the beginning of the holding of b_1 without any gap in time in between. Thus the extents in time (represented by \mathbb{R}) of the holdings of b_0 and b_1 might be represented by the following intervals



(This suggests a formal definition of an observer for interpretations according to which an observer allocates abutting semiclosed intervals of \mathbb{R} to holdings of propositions of F^2 -related conditions.

However we do not follow-up this.)

We now focus on some particular interpretation classes.

We might assume that no single proposition can be concurrently true through the concurrent holding of two distinct conditions. This means that holdings of the same proposition must be causally related. This would occur for example in modelling a Milner net by an interpreted occurrence net so that each proposition referred to strictly one agent. This restriction attempts to capture an idea that propositions refer to local states of affairs. Formally:

Definition 6.2.9

Let \mathcal{J}_1 be the interpretation class on occurrence nets given by:
For N an occurrence net (B, E, F)

$I \in \mathcal{J}_1(N)$ iff $\forall b, b' \in B \ b \text{ co } b' \ \& \ I(b) \cap I(b') \neq \emptyset \Rightarrow b = b'$

In other words for such local interpretations two assertions of the same proposition must be causally related.

We have mentioned that intuitively event occurrences may be taken to be instantaneous changes in holdings of conditions. Accordingly propositions interpreting the pre and post conditions of an event will hold before during or after the event's occurrence. We may wish to identify an event with the change in proposition holdings its occurrence sometimes or always incurs. To guarantee such "event extensionality" we can restrict interpretations. The stronger restriction is:

$\forall C, C' \in \mathcal{O}_f \circ \mathcal{E}(N) \ C' = C \cup \{e\} \Rightarrow I \circ Fr_N(C) \neq I \circ Fr_N(C')$

(An event e must always incur a change in proposition holdings.)

The weaker restriction is:

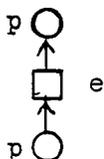
$\exists C, C' \in \mathcal{O}_f \circ \mathcal{E}(N) \ C' = C \cup \{e\} \ \& \ I \circ Fr_N(C) \neq I \circ Fr_N(C')$.

(An event e sometimes incurs a change in proposition holdings.)

Consider the following examples. Example 6.2.10 fails both restrictions while example 6.2.12 fails only the stronger.

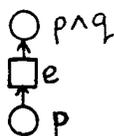
Example 6.2.11 satisfies both.

Example 6.2.10

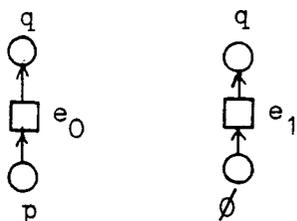


For this net with the interpretation shown the instantaneous occurrence of e involves no change in those propositions which hold.

Example 6.2.11



Example 6.2.12



For this net and interpretation (not in the interpretations class \mathcal{J}_1) the occurrence of e_1 is sometimes associated with a change in the holding of propositions and sometimes not.

For the interpretation class \mathcal{J}_1 both restrictions are equivalent to the extra restriction in the following definition.

Definition 6.2.13

Let \mathcal{J}_{1e} be the interpretation class consisting of interpretations I in \mathcal{J}_1 which in addition satisfy: For all events e

$$I(\cdot e) \neq I(e \cdot).$$

(Then say I is event extensional.)

It is natural to ask how the expressiveness relation changes for different interpretation classes. In the next section we consider \preceq , $\preceq_{\mathcal{J}_1}$ and $\preceq_{\mathcal{J}_{1e}}$ for occurrence nets associated with the same event structure.

Of course one may restrict the interpretation class further basically transferring more of the computational structure to the interpretation. For example one might like an interpretation class consisting of interpretations, I , for which the structure consisting of interpreted markings of the form $I \circ \text{Fr}_N(C)$ with induced reachability relation determined the event structure.

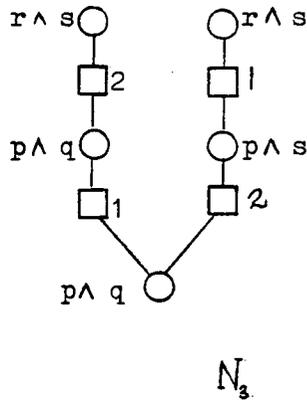
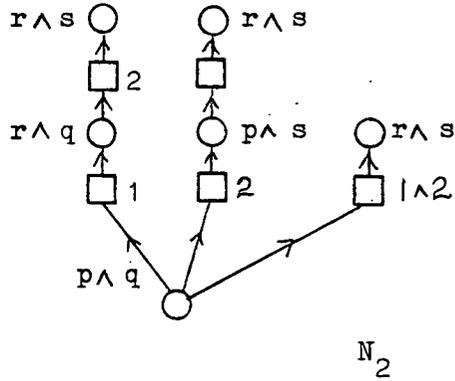
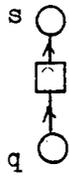
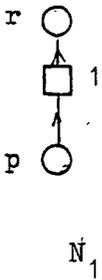
We now examine some issues involved in extending the idea of expressiveness to more general classes of nets such as all occurrence nets or initially-marked transition nets. Such a relation will depend on what we choose as the essential structure of an interpreted net. Let us suppose a net N (perhaps with initial marking) with interpretation I in interpretation class \mathcal{J}' has essential structure $\mathcal{M}^I(N)$. Then the expressiveness relation over an interpretation class \mathcal{J}' will have a definition of the following form:

$$N_1 \preceq_{\mathcal{J}'} N_2 \text{ iff } \forall I_1 \in \mathcal{J}'(N_1) \exists I_2 \in \mathcal{J}'(N_2) \mathcal{M}^{I_1}(N_1) = \mathcal{M}^{I_2}(N_2).$$

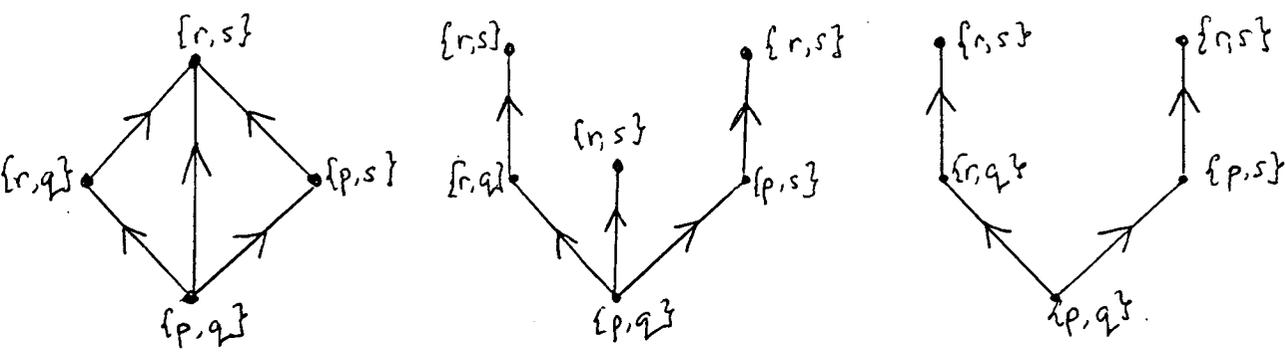
The problem is thus to find intuitively acceptable \mathcal{M} and \mathcal{J}' .

Consider first defining an expressiveness relation between occurrence nets not necessarily associated with the same event structure. Certainly taking the $\mathcal{M}^I(N)$ above as $\mathcal{O}^I(N)$ makes nets with different event structures incomparable under an expressiveness relation. The following example suggests more general choices of $\mathcal{M}^I(N)$.

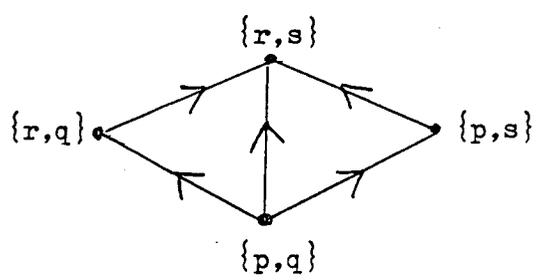
Example 6.2.14



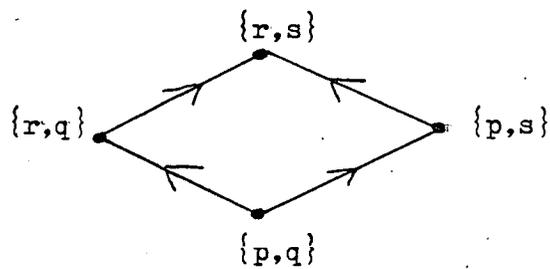
In this example the nets N_2 and N_3 with the interpretations shown are "interleaved simulations" of the net N_1 with interpretation shown. We have indexed the interleaved events of N_1 and N_2 by the events of N_1 they correspond to. The net N_2 has an additional event $1 \wedge 2$ denoting the simultaneous occurrence of events 1 and 2. If we draw the observable states together with the one-step-forward reachability relation we get for N_1, N_2 and N_3 respectively:



where we have marked-in $I \circ Fr_N(C)$ for the observable states C .
 If we identify states when the same propositions hold there we get



for both N_1 and N_2 . This reflects the fact that the possible extents of time of the holdings of propositions for the interpreted nets N_1 and N_2 are the same. For N_3 , however, we get



Taking such diagrams as the essential structure thus gives $N_1 \preceq N_2$. In fact also $N_2 \not\preceq N_1$ and $N_1 \not\preceq N_3$. The diagrams are based on one-step-forward reachability. If instead we based essential structure on forwards reachability (its transitive reflexive closure) we would then have $N_1 \preceq N_3$ as well as $N_1 \preceq N_2$.

The above example suggests that given an occurrence net N and interpretation I we take as its essential structure the set

$$\{I \circ Fr_N(C) \mid C \text{ an observable state}\}$$

together with some reachability relation \rightarrow_I induced by the reachability relation on observable states. Such a definition requires care. For definiteness take \rightarrow the 1-step forward

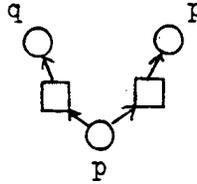
reachability relation on observable states. An obvious definition of \rightarrow_I is

$$I \circ Fr_N(C) \rightarrow_I I \circ Fr_N(C') \text{ iff } C \rightarrow C'.$$

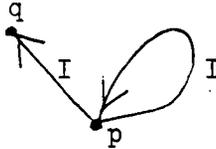
In general this will lead to loops in \rightarrow_I^* or even \rightarrow_I which are not intuitively reasonable as the following example shows.

Example 6.2.15

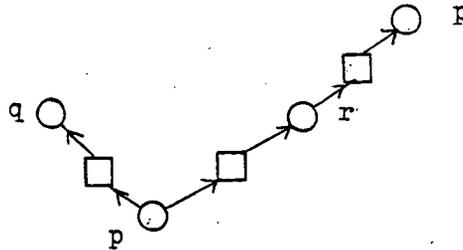
For the interpreted net



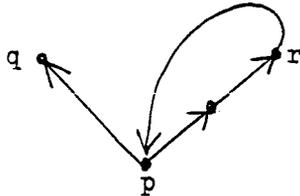
we get, according the above definitions,



For the interpreted net



we get



In both cases the initial condition interpreted by p can end so q holds while the terminal conditions interpreted by p cannot. Thus states have been identified which have different future behaviours.

One could avoid such problems by restricting interpretations, for instance so loops were banned, while keeping the above definition of \rightarrow_I . This would not generalise to transition nets. Alternatively one could seek a more refined definition of equivalence of interpreted nets including transition nets. It is suggested that a definition of observational equivalence of interpreted nets along the lines of that used by Hennessy and Milner in [Hen] for defining

equivalence of synchronisation trees is appropriate. Roughly this would say two interpreted nets are equivalent (have essentially the same structure) iff whatever "interpreted state" can be reached in one can be reached in the other with the same subsequent behaviour under the interpretations. Perhaps category theory is the appropriate framework; take objects to be (interpreted) states and morphisms to be events.

6.3 The constructions \mathcal{N} and \mathfrak{n} give maximally expressive nets

Here we shall look at the constructions of occurrence nets $\mathcal{N}(E)$ and $\mathfrak{n}(E)$ for an event structure E from the point of view of expressiveness. Our main result is to characterise the construction $\mathfrak{n}(E)$. For the three interpretation classes \mathcal{J} , \mathcal{J}_1 , \mathcal{J}_{1e} of the last section the net $\mathfrak{n}(E)$ will be maximally expressive in the set of nets associated with E . In addition the net $\mathfrak{n}(E)$ will be included in all such maximally expressive nets. We work with the expressiveness relation defined in proposition 6.2.4 and chiefly with the interpretation class \mathcal{J}_1 .

Throughout this section we assume nets are of finite depth condition-extensional and satisfy axiom $N\exists$ i.e. for all events e we have $\cdot e$ and $e \cdot$ non-null. Note the results go through for a weaker notion of event structure and observable state; we shall only use the fact that observable states do not include infinite chains of events.

Notation 6.3.1

We write \preceq , \preceq_1 and \preceq_{1e} for the expressiveness relations associated with the interpretation classes \mathcal{J} , \mathcal{J}_1 and \mathcal{J}_{1e} respectively.

Amongst the set of occurrence nets inducing the same event structure it is obvious the maximal net $\mathcal{N}(E)$ consisting of all possible conditions of an event structure E is maximal with respect to the expressiveness relations \preceq , \preceq_1 or \preceq_{1e} .

Theorem 6.3.2

Suppose $E = (E, \preceq, \times)$ is an event structure of finite depth. Let $\mathcal{N}(E)$ be the occurrence net defined in 4.2.13. Then for all nets N

$$\mathcal{E}(N) = E \Rightarrow N \preceq \mathcal{N}(E).$$

where \leq ' is any of the expressiveness relations \leq , \leq_1 or \leq_{le} .

Proof

As we assume all nets are condition-extensional all conditions of the net N above are "included" in the conditions of $\mathcal{N}(E)$. Interpret such conditions in $\mathcal{N}(E)$ identically and others as \emptyset . ■

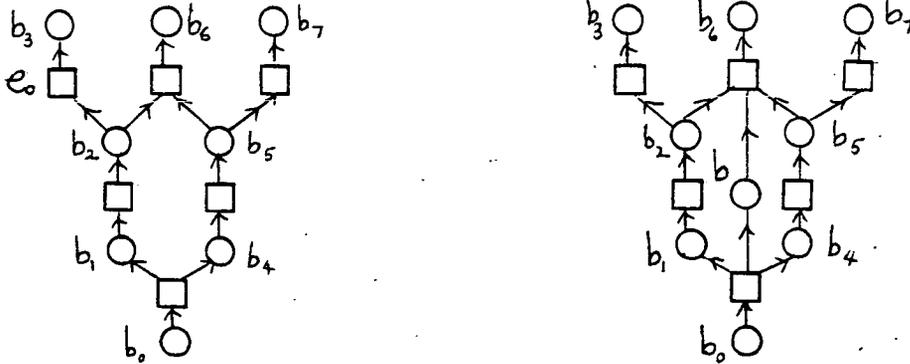
It is no surprise that the maximum net associated with an event structure is maximally expressive. That net includes all conditions possible under condition-extensionality. We now show that the net $\mathfrak{n}(E)$ of 6.1 constructed by taking conditions to be \leq -minimal is also maximally expressive. In addition every maximally expressive net will include $\mathfrak{n}(E)$. This means every condition of $\mathfrak{n}(E)$ will be included in every maximally expressive net i.e. the \leq -minimal conditions of an event structure are precisely the "essential" conditions of the maximally expressive nets. (Compare 4.1.17 characterising essential conditions of a causal net.)

Suppose N is an occurrence net such that $\mathcal{E}(N)$ is the event structure E . For any \mathcal{J}_1 -interpretation I of N we require an \mathcal{J}_1 interpretation I' of $\mathfrak{n}(E)$ such that

$$\forall C \in \mathcal{C}(E) \quad I \circ Fr_N(C) = I' \circ Fr_{\mathfrak{n}(E)}(C).$$

We illustrate how I' is determined by I through an example.

Example 6.3.3

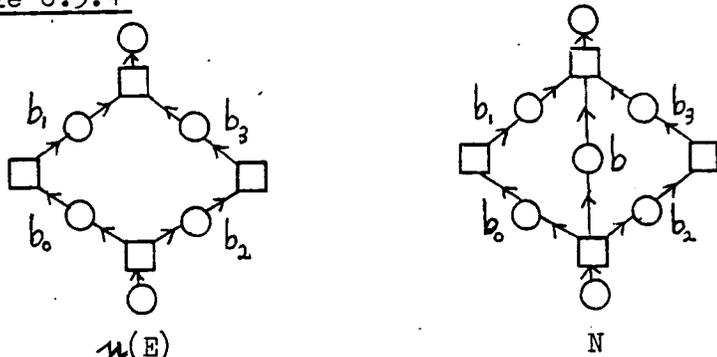


Above we have drawn $\mathfrak{n}(E)$ and a net N with $\mathcal{E}(N) = E$ for an event structure E . Suppose $p \in I(b)$. What conditions of $\mathfrak{n}(E)$ are to be labelled by p ? We have a choice. We could label b_1 and b_2 by p . However then e_0 might occur so b still holds while b_1 and b_2 do not. Thus we must also label b_3 by p . Alternatively we could label b_4, b_5 and b_7 by p . As the interpretation of $\mathfrak{n}(E)$ is to be

in \mathcal{J}_1 we cannot label all $b_1, b_2, b_3, b_4, b_5, b_7$ by p . Note that in, for example, the first choice although in a sense the subnet determined by b_1, b_2, b_3 simulates b we do not have $b_3 \triangleleft b$.

It might be thought that the ambiguity in the labelling is due to confusion. The following example gets rid of that idea.

Example 6.3.4



The condition b may be "simulated" by either $\{b_0, b_1\}$ or $\{b_2, b_3\}$.

We accent the choice of conditions of $\mathcal{u}(E)$ used to simulate a condition by means of a choice function. Given a condition (e, A) this simply chooses a unique \triangleleft -minimal condition (e, A') with $(e, A') \triangleleft (e, A)$ (such exist by lemma 6.1.16).

Definition 6.3.5

Suppose E is an event structure. A choice function for E is a map $\chi: \mathcal{B}(E) \rightarrow \mathcal{b}(E)$ s.t.

$$\chi((e, A)) = (e, A') \triangleleft (e, A) \text{ for some } A'.$$

Thus in example 6.3.3 we might have $\chi(b) = b_1$ and $\chi(b_n) = b_n$ for $n = 0, \dots, 7$.

Henceforth in this section we work with a fixed event structure E of finite depth together with a fixed choice function χ . For a condition b of $\mathcal{B}(E)$ we now define a set $S_\chi(b)$ of conditions in $\mathcal{b}(E)$ which simulate b in this sense:

$$\forall c \in \mathcal{O}(E) \text{ (on}(b, c) \Leftrightarrow \exists ! b' \in S_\chi(b) \text{ on}(b', c)).$$

The idea is to use χ to divide up the extent of b into a set of \triangleleft -minimal conditions which determine a tree-like subnet of $\mathcal{u}(E)$.

(For the obvious χ this would yield $S_\chi(b) = \{b_1, b_2, b_3\}$ in example 6.3.3.)

Definition 6.3.6

For A a subset of E define

$$p(A, e) = \{a \in A \mid e < a\}.$$

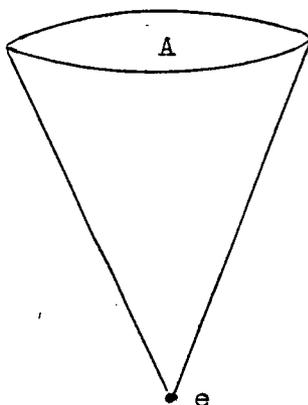
Definition 6.3.7

Let $b = (e, A)$ be in $\mathcal{B}(E)$. Define $S_{\chi}(b) = \bigcup_{n \in \omega} S_{\chi}^{(n)}(b)$ where $S_{\chi}^{(n)}(b)$ is defined inductively by:

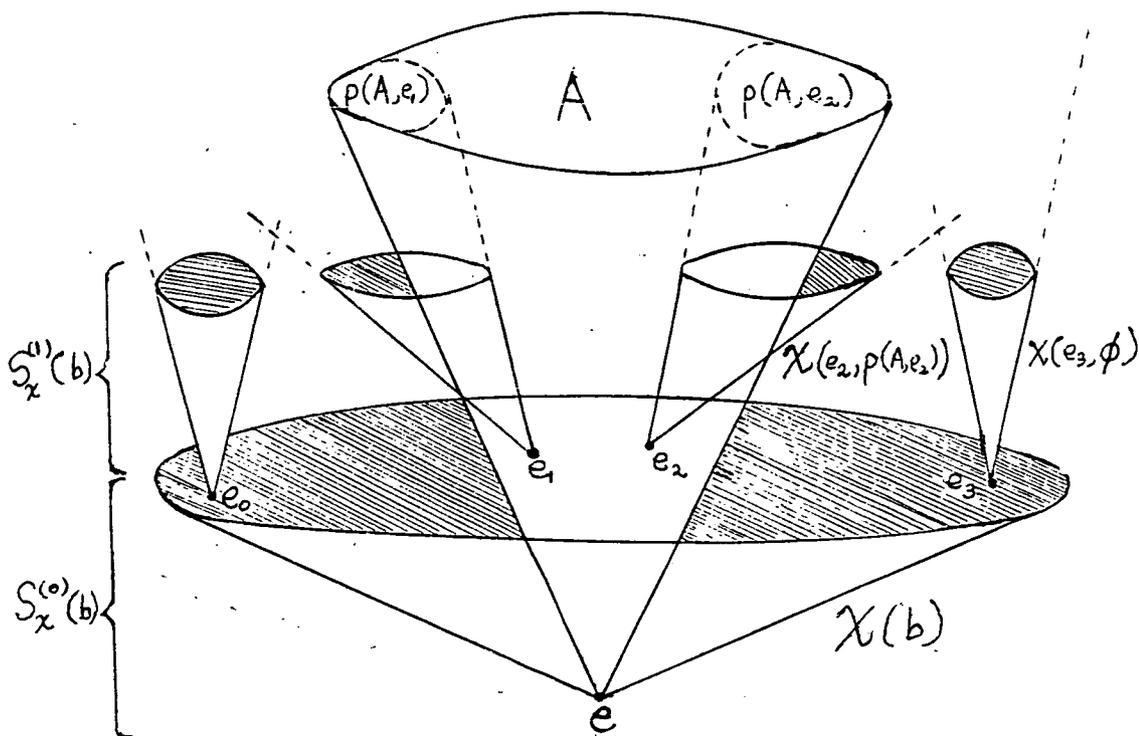
$$S_{\chi}^{(0)}(b) = \{\chi(b)\}$$

$$S_{\chi}^{(n+1)}(b) = \{\chi((e', p(A, e'))) \mid \exists b' \in S_{\chi}^{(n)}(b) \ e' \in b' \setminus A\}.$$

Picture $b = (e, A)$ as



Then the second stage of the construction of $S_{\chi}(b)$ may be pictured as



The events e_0, e_1, e_2, e_3 are taken to be in $(\mathcal{X}(b))_1$. The shaded regions denote events not below A so $p(e_0, A)$ and $p(e_3, A)$ are null. In the drawing $\mathcal{X}(e_2, p(A, e_2))$ is a condition with $p(A, e_2)$ non-null. There are extra conditions in $S_{\mathcal{X}}(b)$, corresponding to b_3 of example 6.3.3, of which one holds whenever b can no longer end holding. In the drawing $\mathcal{X}(e_0, \emptyset)$ and $\mathcal{X}(e_3, \emptyset)$ represent such conditions. The set $S_{\mathcal{X}}(b)$ has been constructed so that b holds iff one and only one condition in $S_{\mathcal{X}}(b)$ holds. We now prove this. Firstly $S_{\mathcal{X}}(b)$ determines a tree-like subnet of $\mathfrak{n}(E)$ called $\sum_{\mathcal{X}}(b)$.

Definition 6.3.8

For b in $\mathcal{B}(E)$ define the net $\sum_{\mathcal{X}}(b)$ to consist of conditions $S_{\mathcal{X}}(b)$ and events $\{b' \mid b' \in S_{\mathcal{X}}(b)\} \cup \{b' \bullet \mid b' \in S_{\mathcal{X}}(b)\}$ with F -relation F_b induced by $\mathcal{W}(E)$.

Lemma 6.3.9

For b in $\mathcal{B}(E)$ the set $\sum_{\mathcal{X}}(b)$ is a tree-like subnet of $\mathfrak{n}(E)$. Further if b is of the form (e, A) then A equals the set of F_b^* -maximal elements in the net $\sum_{\mathcal{X}}(b)$ which are events.

Proof

Suppose b in $\mathcal{B}(E)$ has the form (e, A) . From its inductive construction it follows that $\sum_{\mathcal{X}}(b)$ is a tree-like subnet of $\mathfrak{n}(E)$.

We show for all a in A there is a chain $e_0 \overset{F_b}{\rightarrow} e_1 \overset{F_b}{\rightarrow} \dots \overset{F_b}{\rightarrow} e_k$ in $\sum_{\mathcal{X}}(b)$ with $e_0 = e$ and $e_k = a$. The chain is constructed inductively. Initially put $e_0 = e$ and $b_1 = \mathcal{X}(b)$. Suppose we have defined $e_0 \overset{F_b}{\rightarrow} e_1 \overset{F_b}{\rightarrow} \dots \overset{F_b}{\rightarrow} e_n$ a chain in $\sum_{\mathcal{X}}(b)$ with $e_n \leq a$. If $e_n = a$ we have produced the desired chain. Otherwise extend the chain by putting $b_{n+1} = \mathcal{X}(e_n, p(A, e_n))$ and e_{n+1} as the unique event in b_{n+1} below a . As there are no infinite chains below a we eventually construct the required chain.

Thus by the definition of $S_{\mathcal{X}}(b)$ no condition of $S_{\mathcal{X}}(b)$ has pre-event a in A so each a in A is a maximal event in $\sum_{\mathcal{X}}(b)$. The set A is precisely all such events as by the construction of $S_{\mathcal{X}}(b)$ any event in $\sum_{\mathcal{X}}(b) \setminus A$ has a postcondition in $S_{\mathcal{X}}(b)$. ■

In theorem 6.3.11 we use the above lemma to show that if a condition b holds for an observable state then a unique condition in $S_{\mathcal{X}}(b)$ holds. The converse, that a condition of $S_{\mathcal{X}}(b)$ holding for an observable state implies b holds too, is ensured by the next lemma.

Lemma 6.3.10

Suppose $b \in \mathcal{B}(E)$. Then
 $\forall b' \in S(b) \ b' \rightarrow b$.

Proof

Suppose b has the form (e, A) . Assume $b' \in S_\chi(b)$ and $b' = (e', A')$. Clearly $e \leq e'$. Suppose $a \in A$. By the characterisation of \rightarrow we require $a \not\leq e'$ or $\exists a' \in A' \ a' \leq a$. From the construction of $S_\chi(b)$ we have A' is \subseteq_0 -minimal s.t. $e' < A' \subseteq_0 p(A, e')$. If $a \in p(A, e')$ then $\exists a' \in A' \ a' \leq a$ as required so assume $a \notin p(A, e')$. Then $e' \not\leq a$. By the above lemma $a, e' \in \sum_\chi(b)$ and a and e' are F^* -incomparable in $\sum_\chi(b)$. Thus as $\sum_\chi(b)$ is tree-like there is an F^* -maximum condition b_0 in $S_\chi(b)$ so that $b_0 F^* e'$ and $b_0 F^* a$ in the net $\sum_\chi(b)$. This gives $a \not\leq e'$ as required. ■

Now we can prove the precise sense in which $S_\chi(b)$ simulates b .

Theorem 6.3.11

For $b \in \mathcal{B}(E)$,
 $\forall C \in \mathcal{O}_d(E) \ (\text{on}(b, C) \Leftrightarrow \exists ! b' \in S_\chi(b) \ \text{on}(b', C))$.

Proof

Let $b = (e, A) \in \mathcal{B}(E)$ and assume $C \in \mathcal{O}_d(E)$.

If $\text{on}(b, C)$ then $e \in C$ and $A \cap C = \emptyset$. Let e' be the \leq -maximum event in $\sum_\chi(b) \cap C$ - as C does not include infinite chains e' exists. Take b' to be the condition in $S_\chi(b)$ with $(b')_0 = e'$. Such a b' exists as $e' \notin A$ as A is the set of F^* -maximal events in $\sum_\chi(b)$. Then $\text{on}(b', C)$ and b' is unique as $\sum_\chi(b)$ is tree-like.

If $\text{on}(b', C)$ for some b' , necessarily unique, in $S_\chi(b)$ then, as $b' \rightarrow b$, we have $\text{on}(b, C)$. ■

It is now simple to show that $\mathfrak{n}(E)$ is maximally expressive amongst the nets inducing E .

Theorem 6.3.12

Suppose N is an occurrence net such that $\mathcal{E}(N) = E$. Then
 $N \preceq_1 \mathfrak{n}(E)$. Also $N \preceq \mathfrak{n}(E)$ and $N \preceq_{1e} \mathfrak{n}(E)$.

Proof

Suppose an occurrence net N is such that $\mathcal{E}(N) = E$. For I an \mathcal{J}_1 -interpretation of N , define the \mathcal{J}_1 -interpretation of $\mathfrak{n}(E)$

by $I'(b) = \bigcup \{I(b') \mid b \in S_{\chi}(b')\}$. For all observable states we have

$$\text{Fr}_N \circ I(C) = \text{Fr}_{\mathfrak{n}(E)} \circ I'(C).$$

Thus $N \leq_1 \mathfrak{n}(E)$ as required.

In addition for I in either of the interpretation classes \mathcal{J} or \mathcal{J}_{1e} taking $I'(b) = \bigcup \{I(b') \mid b \in S_{\chi}(b')\}$ gives I' also in the interpretation class \mathcal{J} or \mathcal{J}_{1e} respectively. (From the properties of $S_{\chi}(b)$ it is easy to show $I(\bullet e) = I'(\bullet e)$ and $I(e \bullet) = I'(e \bullet)$, so that I in \mathcal{J}_{1e} implies I' in \mathcal{J}_{1e}). As above this choice of I' from I gives $N \leq \mathfrak{n}(E)$ and $N \leq_{1e} \mathfrak{n}(E)$. ■

The following is an occurrence-net analogue of 4.1.17. It means \triangleleft -minimal conditions are essential for the subclass of maximally expressive nets (w.r.t. to any interpretation class \mathcal{J} , \mathcal{J}_1 or \mathcal{J}_{1e}).

Theorem 6.3.13

Suppose $N = (B, E, F)$ is a maximally expressive net (w.r.t. \mathcal{J} , \mathcal{J}_1 or \mathcal{J}_{1e}) and $\mathfrak{E}(N) = E$. Then $\mathfrak{b}(E) \subseteq B$.

Proof

Let $N = (B, E, F)$ be such a maximally expressive net. We know $N \approx_1 \mathfrak{n}(E)$. Take I_1 to be the interpretation of $\mathfrak{n}(E)$ which to condition b associates the singleton $\{p_b\}$ so that $p_b = p_{b'} \Rightarrow b = b'$. As $\mathfrak{n}(E) \leq_1 N$ there is an interpretation of N , call it I_2 , such that $I_1 \circ \text{Fr}_{\mathfrak{n}(E)}(C) = I_2 \circ \text{Fr}_N(C)$ for all observable states C . Assume $b \in \mathfrak{b}(E)$ is of the form (e, A) . Taking $C = [e]$ gives some b' in B s.t. $p_b \in I_2(b')$. Obviously b' has form (e, A') for some A' in $K(E)$ - consider the beginning of the assertion of p_b . Consider endings of the assertion p_b , formally: Take $C = [a]$ for a in A ; then as b' has ended for some a' in A' , $a' \in [a]$; thus $A' \sqsubseteq_0 A_0$. This gives $b' \triangleleft b$. But b is \triangleleft -minimal so $b = b'$. Therefore $\mathfrak{b}(E) \subseteq B$ as required. ■

Thus the conditions $\mathfrak{b}(E)$ are essential within the class of maximally expressive nets; any \triangleleft -minimal condition is contained in the conditions of any maximally expressive net. The net $\mathfrak{n}(E)$ is a subnet of every maximally expressive net.

The demonstration that $\mathfrak{n}(E)$ is maximally expressive suggests the following characterisation of the expressiveness relations on nets

inducing E. The expressiveness relation with respect to an interpretation class merely expresses that in some sense each condition of one net may be simulated by a subset of conditions of the other - the manner of simulation is restricted in accord with restrictions on the interpretation class.

Proposition 6.3.14

Let N_0 and N_1 be condition-extensional occurrence nets satisfying N3, inducing E, with conditions B_0 and B_1 respectively. Then

1. $N_0 \leq N_1$ iff $\exists f: B_0 \rightarrow \mathcal{P}(B_1)$ $f(b) \rightarrow b$ &
 $\forall C \in \mathcal{Q}(E) (\text{on}(b, C) \Rightarrow \exists b' \in f(b) \text{ on}(b', C))$
2. $N_0 \leq_1 N_1$ (or $N_0 \leq_{1e} N_1$) iff $\exists f: B_0 \rightarrow \mathcal{P}(B_1)$ $f(b) \rightarrow b$ &
 $\forall C \in \mathcal{Q}(E) (\text{on}(b, C) \Rightarrow \exists! b' \in f(b) \text{ on}(b', C)).$

Proof

"1. and 2. \Rightarrow " Interpret N_0 by I_0 which associates B_0 with distinct singletons of propositions. As $N_0 \leq N_1$ (or $N_0 \leq_1 N_1$) there is a corresponding interpretation I_1 of N_1 . Define $f(b)$ to be the subset of conditions of B_1 whose interpretations contain $I_0(b)$. (For $N_0 \leq_1 N_1$ the nature of I_1 gives the uniqueness in 2).

"1. and 2. \Leftarrow " For an interpretation I_0 of N_0 define I_1 by $I_1(b_1) = \bigcup \{I_0(b) \mid b_1 \in f(b)\}$. ■

Consider a subset of conditions X satisfying the conditions of $f(b)$ in 2. i.e. suppose for a condition b

$$X \rightarrow b \text{ \& } \forall C \in \mathcal{Q}(E) (\text{on}(b, C) \Rightarrow \exists! b' \in X \text{ on}(b', C))$$

One expects such X to determine a tree-like subnet satisfying some further restrictions dependent on b . It may be that any set Y such that

$$Y \rightarrow b \text{ \& } \forall C \in \mathcal{Q}(E) (\text{on}(b, C) \Rightarrow \exists b' \in Y \text{ on}(b', C))$$

always includes such a set (I expect so). If so the above proposition gives $\leq = \leq_1 = \leq_{1e}$ on occurrence nets inducing E.

6.4 Restless events

It is time we dealt with restless events. Mathematically they seem to involve constructions similar to those of the previous section. How similar is not clear from this section's incomplete

development. They may be important to a study of fairness.

Certainly whether or not the framework suggested in this section is appropriate in detail some extra structure must be imposed on nets and event structures in order to model situations in which something will inevitably occur sometime. That something might be an event or some more general property such as an event losing its concession. Recall the situations which involved some idea of inevitability: A \mathcal{T} -communication in a Milner net was not supposed to be able to occur, and not occur, forever (see 2.3A); in a computation determining a function from one datatype to another, events other than input events occurred eventually if they could (see 2.3C, and 5.6 where we discuss continuity-respecting event structures); the events of causal nets representing Petri's real processes are thought of as having occurred or inevitably occurring (see 2.4 and chapter 7). Of course the idea also arises, but implicitly, in deterministic computations; it is assumed that having finished one task, flow-of-control will move on to the next.

Recall the idea of restless events. An event is said to be "restless" if it is not possible for it to have concession forever; of course it may lose concession through occurring itself or if another event in conflict with it occurs. We wish to place extra structure on event structures to express this idea of inevitability for a subset of events; the extra structure will be a distinguished subset of events, those to be regarded as restless.

Now we look at the formal implications. Firstly we can define when an event has concession.

Definition 6.4.1

Let E be an event structure. Suppose $e \in E$ and $C \in \mathcal{L}(E)$. Then e has concession at C ,

$$\text{con}(e, C) \Leftrightarrow \langle^{-1}\{e\} \subseteq C \ \& \ (\mathbb{X} \cup \{e\}) \cap C = \emptyset.$$

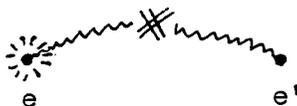
Note this is reminiscent of the on-predicate formalising when conditions hold for consistent left-closed subsets. We could invent a new form of condition which for each event e would consist of a pair $(\langle^{-1}\{e\}, \mathbb{X} \cup \{e\})$ (or perhaps $(\langle^{-1}\{e\}, \mathbb{X}_\mu \cup \{e\})$ if E were of finite depth for example). Then 6.4.1 simply expresses that this generalised condition holds whenever $\langle^{-1}\{e\}$ have all occurred and none

of $\times \cup \{e\}$ have occurred. Note in general that even for "conditions" of the form $\langle^{-1}\{e\}, \times \cup \{e\}\rangle$ we might not have $\times \cup \{e\}$ in $\mathcal{K}(E)$ - the event structure need not be confusion-free; however if E is confusion-free they correspond to places in a matrix.

If an event e is specified as restless any observer who sees $\langle^{-1}\{e\}$ at some finite time must eventually see at least one of $\times \cup \{e\}$. Similarly if a subset of events A is specified as restless then this is the case for every event e in A . It is obvious how to code in mathematical notation the restriction on observers $\text{Ob}(E)$ that results when a subset of E is distinguished as restless. It is neater however to work with $\mathcal{L}(E)$ rather than $\text{Ob}(E)$. To justify this we require that for a restless event e we have that $\langle^{-1}\{e\}$ if observed is observed in finite time. Otherwise the event may get concession only after an infinite time; clearly then we would not expect it to occur. For this reason, in this section we shall henceforth assume that event structures satisfy:

For all events e , the set $[e]$ is finite.

Distinguishing certain events as restless disallows particular states at infinite time. For example suppose e is restless in the simple event structure consisting of a pair e and e' of conflicting events.



Then over infinite time we would get states $\{e\}$ or $\{e'\}$; the null-state after infinite time would not be consistent with the restlessness of e . More generally suppose E is an event structure with a set of restless events R . Those states which are allowed at infinite time (call them eventual states) are those $C \in \mathcal{L}(E)$ such that

$$\forall e \in R \neg \text{con}(e, C)$$

i.e. $\forall e \in R (\langle^{-1}\{e\} \subseteq C \Rightarrow \times \cup \{e\} \cap C \neq \emptyset)$.

In this sense all eventual states are closed under R .

In the simple example above, consisting of a pair of conflicting events e and e' with e restless, for no eventual state does e' have concession. In this sense e' is also restless.

Assuming e is restless implies e' is restless. In general suppose R is a set of restless events of an event structure E . It determines eventual states C where $\forall e \in R \neg \text{con}(e, C)$. Often there will be an event $e' \notin R$ such that

$$\forall C \in \mathcal{L}(E) ((\forall e \in R \neg \text{con}(e, C)) \Rightarrow \neg \text{con}(e', C))$$

i.e. $\forall C \in \mathcal{L}(E) (\text{con}(e', C) \Rightarrow \exists e \in R \text{con}(e, C))$

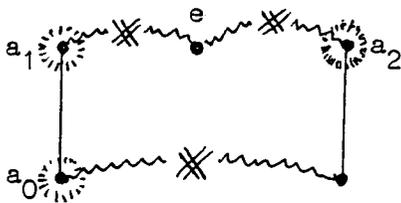
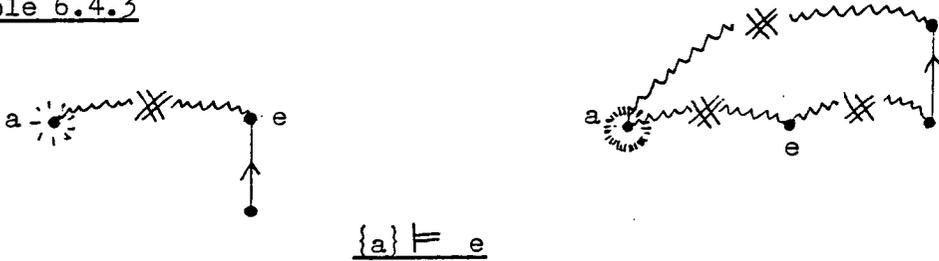
which says that if events R are restless then so is e' . We turn this into a definition.

Definition 6.4.2

Let E be an event structure such that $[e]$ is finite for all e in E . Suppose $A \subseteq E$ and $e \in E$. Define

$$A \models e \text{ iff } \forall C \in \mathcal{L}(E) (\text{con}(e, C) \Rightarrow \exists a \in A \text{con}(a, C))$$

Example 6.4.3



$$\underline{\{a_0, a_1, a_2\} \models e} \text{ (and yet } \{a_i\} \not\models e \text{ for } i = 0, 1, 2)$$

As the extra structure on events it would be natural to take subsets R which are closed under \models in the sense that

$$R \models e \Rightarrow e \in R$$

Unfortunately I cannot yet characterise such R and the relation \models . Any nice characterisation seems to involve a generalisation of Petri's conditions. The next lemma characterises $A \models e$ in the simple case where A is a singleton.

Lemma 6.4.4

For the relation \models defined in 6.4.2 we have $\{e\} \models e'$ iff

$$\langle^{-1}\{e\} \subseteq \langle^{-1}\{e'\} \& \text{X}\cup\{e\} \subseteq \text{X}\cup\{e'\}.$$

Proof

" \Leftarrow " obvious.

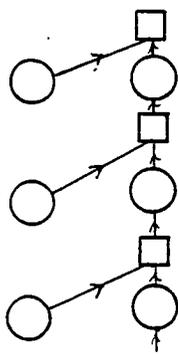
" \Rightarrow " Suppose $\{e\} \models e'$. Then $\forall C \in \mathcal{L}(E) (\text{con}(e', C) \Rightarrow \text{con}(e, C))$.
 Take $C = \langle^{-1}\{e'\}$. Then $\text{con}(e', C)$ so $\text{con}(e, C)$. Thus $\langle^{-1}\{e\} \subseteq \langle^{-1}\{e'\}$.
 Assume $e'' \text{X}\cup\{e\}$. We require $e'' \text{X}\cup\{e'\}$. If $e'' \text{X}\{e\}$ this is
 obvious so assume $\neg e'' \text{X}\{e\}$. Take $C = [e''] \cup \langle^{-1}\{e'\}$.
 Then $\neg \text{con}(e, C)$ so $\neg \text{con}(e', C)$. Thus $\langle^{-1}\{e'\} \subseteq C$.
 $\Rightarrow (\text{X}\cup\{e'\}) \cap C \neq \emptyset$. As $\langle^{-1}\{e'\} \subseteq C$ we have $\text{X}\cup\{e'\} \cap$
 $([e''] \cup \langle^{-1}\{e'\}) \neq \emptyset$. But then $\text{X}\cup\{e'\} \cap [e''] \neq \emptyset$. Thus
 $e'' \text{X}\cup\{e'\}$. ■

Of course distinguished subsets of restless events may not be the appropriate extra structure in general. Perhaps labelled event structures on the lines of 2.3A would be a more suitable framework; there would be two kinds of events, "complete" events labelled by γ which would eventually occur or lose concession, and "incomplete" events which could only occur through communication with the environment.

Chapter 7. Event structures with infinite pasts

In this chapter we present some mathematical results on modelling courses of computation with possibly infinite pasts. More precisely we examine the implications of removing the initiality restriction of chapter 5, while keeping the discreteness restriction and imposing the restriction that all events must occur sometime.

From the point of view of denotational semantics this is a little off-beat and maybe it is. However in net theory causal nets the net-theoretic analogue of history are certainly allowed to have infinite pasts. For instance the discussion of K-density in [Bes] explicitly refers to the following net:



It is not disallowed because it has an infinite descending chain of events but because it is not K-dense.

The definition of causal nets and the axiom of K-density in [Pet 1] is an attempt to define a net-theoretic analogue of history possibly with an infinite past. In this chapter we have a similar goal for event structures. Again we shall make use of a notion of observer. These determine observable states. In defining observers we make restrictions on event structure descriptions of computations considered. For instance they will be discrete as in chapter 5 and similarly they induce a reachability relation on observable states. The results on observable states of an elementary event structure in chapter 5. There will be a special case - simply append a fictitious starting event and apply the results here.

If there is more than one reachability class one can argue that the event structure alone does not represent a course of computation. The main result of this chapter is to characterise those event structures with one and only one reachability class. They are

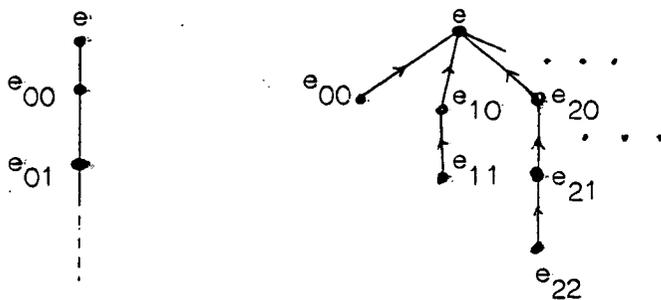
called adequate. This involves some cute mathematics. By allowing extra structure on event structures a broader class of courses can be represented.

In chapter 5 we have argued that K-density is too restrictive an axiom. In view of this the results of this chapter should be significant in defining the class of causal nets corresponding to courses of computation. It is suggested that a causal net alone represents a course of computation iff its associated event structure is adequate. It certainly seems that one would wish two cases of a causal net to be reachable from each other (something like this is stated in [Pet 2] to motivate K-density). As in chapter 5 a restricted form of K-density will hold for a suitable class of nets when cases are restricted to being observable.

7.1 Observers and observable states

Throughout this chapter event structures will be elementary i.e. of the simple form (E, \leq) .

Example 7.1.1



These drawings represent event structures consisting of an event e causally dependent on chains of unbounded lengths.

Here an event structure models a course which may have an infinite past. As in section 5.1 an observer is a record of when events occur. It is assumed that according to an observer every event occurs sometime and also that the occurrences of two causally related events are separated by unit time (the discreteness restriction of section 5.1). Unlike definition 5.1.4 events may occur unboundedly far back in the past. Accordingly time is represented by \mathbb{Z} , the positive and negative integers, ordered as usual.

Definition 7.1.2

An observer for an event structure E is a map $O : E \rightarrow \mathbb{Z}$ such

that

$$e < e' \Rightarrow O(e) < O(e').$$

We denote the set of observers by $Ob(E)$.

Note the event structures of example 7.1.1 have observers. (In either case define an observer O by $O(e) = 1$ and $O(e_{ij}) = -j$.)

Using this idea of observers we can define a notion of state.

Definition 7.1.3

For an observer O of an event structure E we define the state observed by O at time t to be

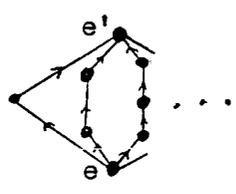
$$os(O, t) = \{e \in E \mid O(e) \leq t\}$$

and further define the observable states of E to be

$$\mathcal{O}(E) = \{os(O, t) \mid O \in Ob(E) \ \& \ t \in \mathbb{Z}\}.$$

Of course not all event structures E have observers so $Ob(E)$ and $\mathcal{O}(E)$ may be null. The restriction on observers is ^adiscreteness restriction; it is clear, for example, that the event structure formed by the reals does not have an observer in the above sense. Neither does the following example.

Example 7.1.4



This event structure consists of events e and e' with chains of unbounded length between them.

For the distance-measure on events Δ of §5.2, $\Delta(e, e')$ is infinite in the above example. Obviously when $\Delta(e, e')$ is infinite for any events e and e' of an event structure the event structure cannot have an observer. When the event structure is countable the converse also holds. The proof uses convex subsets of the event structure.

Definition 7.1.5

Suppose E is an event structure and A is a subset of E . Then the convex closure of A is defined by

$$con(A) = \{e \in E \mid \exists a_1, a_2 \in A \ a_1 \leq e \leq a_2\}$$

Also A is said to be convex iff $A = \text{con}(A)$.

It is clear that the convex closure of a set A includes A .

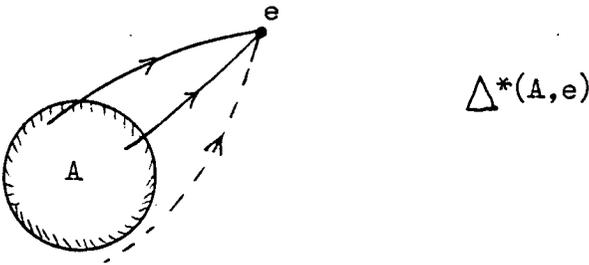
It is convenient to generalise Δ of section 5.2 to convex subsets.

Definition 7.1.6

Let E be an event structure. For e in E and A a non-null convex subset of E define

$$\Delta^*(A, e) = \text{Sup}\{n \mid \exists e_0, \dots, e_n \ e_0 < \dots < e_n \ \& \ ((e_0 \in A \ \& \ e_1 \notin A \ \& \ e_n = e) \ \text{or} \ (e_0 = e \ \& \ e_n \in A \ \& \ e_{n-1} \notin A))\}$$

We can picture $\Delta^*(A, e)$ - the solid lines denote chains which count:



The distance $\Delta^*(A, e)$ is the supremum of chains between the convex subset A and event e . As A is convex the direction of the chains between A and e will always be the same; if there are any chains between A and e they must either all go from inside A to e or all go from e to inside A . As for Δ the distance measure Δ^* may be infinite.

We use the new distance measure in the proof of the theorem below. Note the event structure is assumed countable.

Theorem 7.1.7

Suppose E is a countable event structure. Then

$$\text{Ob}(E) \neq \emptyset \text{ iff } \forall e, e' \in E \ \Delta(e, e') < \infty .$$

Proof

" \Rightarrow " obvious.

" \Leftarrow " Enumerate E as $e_0, e_1, \dots, e_i, \dots$ and define $E_i = \text{con}(\{e_0, \dots, e_i\})$. Construct an observer O inductively. Suppose O is defined for E_i and $O E_i \subseteq [-k_i, k_i]$ for some k_i in ω . Extend O to E_{i+1} by putting, for $e \in E_{i+1} \setminus E_i$,

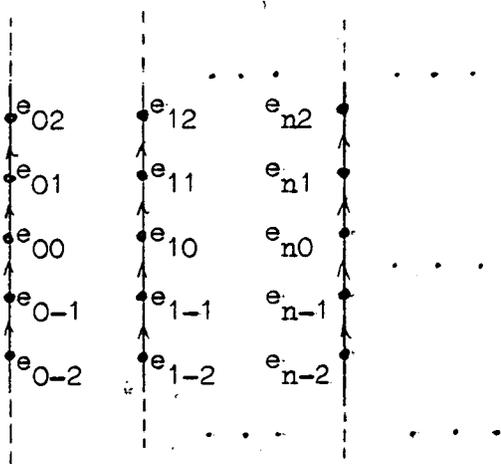
$$\begin{aligned}
 O(e) &= k_i + \Delta^*(E_i, e) \text{ if } \exists e' \in E_i \ e' \leq e_{i+1} \\
 &= -k_i - \Delta^*(E_i, e) \text{ if } \exists e' \in E_i \ e_{i+1} \leq e' \\
 &= 0 \qquad \qquad \qquad \text{otherwise.} \blacksquare
 \end{aligned}$$

The following example shows that the countability assumption is necessary in theorem 7.1.7 above.

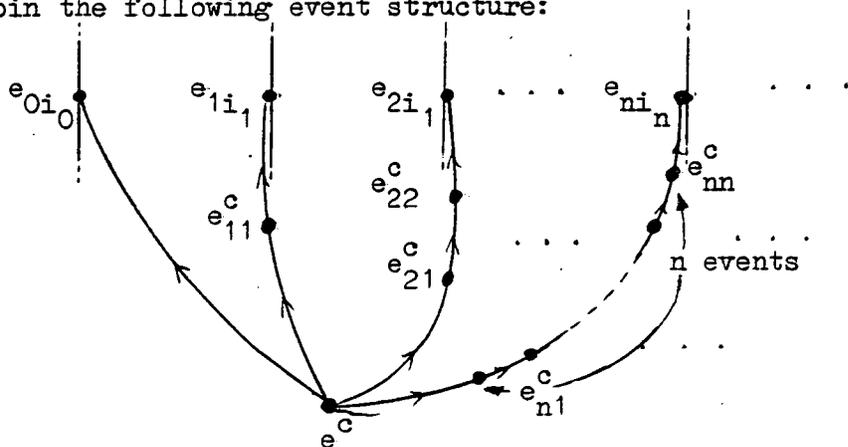
Example 7.1.8

We construct an event structure E (not countable) such that $Ob(E) = \emptyset$ and yet $\forall e, e' \in E \ \Delta(e, e') < \infty$.

The construction starts with E_0 a countable infinity of infinite chains unbounded above and below:



This clearly has an observer as it stands. By adjoining further events we make the existence of an observer impossible. By a cut of E_0 we mean a subset of E_0 containing a unique event from every chain. To each such cut C written as $e_{0i_0}, e_{1i_1}, \dots, e_{ni_n}, \dots$ we join the following event structure:



Thus in E each cut of E_0 is above chains of unbounded length from some event. Note that Δ is still always finite. (The event structure E is uncountable as the set of cuts is uncountable.)

The event structure E does not have an observer. Suppose $0 \in \text{Ob}(E)$. Let C be the cut consisting of \leq -maximal elements in $\text{os}(0,1)$. Then as all events C are observed before time 1 the event e^C cannot be observed, a contradiction.

Henceforth we shall chiefly be interested in countable event structures with observers. Theorem 7.1.7 justifies the following.

Definition 7.1.9

Say an event structure E is countable-observable iff E is countable and $\forall e, e' \in E \Delta(e, e') < \infty$.

Formally at least convex subsets may be regarded as events. Convex subsets of an event structure when "collapsed" to a point yield a new event structure.

Definition 7.1.10

Let E be an event structure with convex subset A . By E/A is meant the event structure consisting of events

$$\{\{e\} \mid e \in E \setminus A\} \cup \{A\}$$

ordered by

$$e \leq e' \text{ iff } \exists e, e' \in E \ e \in \mathcal{E} \ \& \ e' \in \mathcal{E}' \ \& \ e \leq e'.$$

(It is convenient to allow A to be null in the above definition.)

The following define bounded subsets of an event structure and time respectively.

Definition 7.1.11

Let E be an event structure. Suppose A is a subset of E and $k \in \omega$. Say A is k -bounded iff $\forall a_1, a_2 \in A \Delta(a_1, a_2) < k$. Say A is bounded iff A is k -bounded for some k in ω .

Definition 7.1.12

For $k_1, k_2 \in \mathbb{Z}$ with $k_1 \leq k_2$, define the bounded interval $[k_1, k_2]$ to be $\{n \in \mathbb{Z} \mid k_1 \leq n \leq k_2\}$. Define the length of such an interval to be $k_2 - k_1$.

Recall the metric d defined from Δ in section 5.2. Its use abbreviates the following proof.

Lemma 7.1.13

Let E be an event structure. Then E is bounded iff there is a bounded interval $[k_1, k_2]$ and observer O in $Ob(E)$ such that $OE \subseteq [k_1, k_2]$.

Proof

" \Leftarrow " is obvious.

" \Rightarrow " Define the observer O by $O(e) = d(\emptyset, [e])$. It is clear that as E is bounded $d(\emptyset, E)$ is finite and that the range of O is the bounded interval $[0, d(\emptyset, E)]$. ■

The construction of definition 7.1.10 is used in proving the following lemma. Under certain conditions, it says for a k -bounded convex subset there is an observer recording precisely the events A within an interval of time of length k .

Lemma 7.1.14

Let E be a countable-observable event structure. Suppose A is a k -bounded subset of E . Then:

$(\exists k_1, k_2 \in \mathbb{Z} \ k_2 - k_1 = k \ \& \ \exists O \in Ob(E) \ A = O^{-1}[k_1, k_2])$
iff $\forall e \in E \ \Delta^*(A, e) < \infty$.

Proof

" \Rightarrow " is obvious.

" \Leftarrow " Supposing $\forall e \in E \ \Delta^*(A, e) < \infty$ together with the hypothesis on E give Δ always finite on E/A . Thus there is an observer O^* for E/A . Without loss of generality suppose $O^*(A) = 0$. Considered as an event structure A has an observer O_A such that $O_A A \subseteq [0, k]$ by lemma 7.1.13. Then define the required observer O by

$$\begin{aligned} O(e) &= O_A(e) \text{ if } e \in A \\ &= k + O^*({e}) \text{ if } e \notin A \ \& \ O^*({e}) \geq 0 \\ &= O^*({e}) \quad \text{otherwise.} \quad \blacksquare \end{aligned}$$

Corollary 7.1.15

Let E be a countable-observable event structure. Suppose A is a pairwise incomparable subset of E . Then

$\exists 0 \in \text{Ob}(E) \exists t \in \mathbb{Z} \forall a \in A O(a) = t \text{ iff } \forall e \in E \Delta^*(A, e) < \infty.$

Proof

The set A is pairwise incomparable. Thus A is 0-bounded. It is obviously convex. The result then follows trivially from lemma 7.1.14. ■

Now we characterise observable states. Unfortunately this involves the definition of yet another distance measure δ defined from the metric d of 5.2.

Definition 7.1.16

Let E be an event structure. Suppose C is a left-closed subset of E and e an event. Then define

$$\delta(C, e) = \text{Sup}\{d(C, C \cup \{e' \mid e' \leq e\}), d(C, C \setminus \{e' \mid e' > e\})\}$$

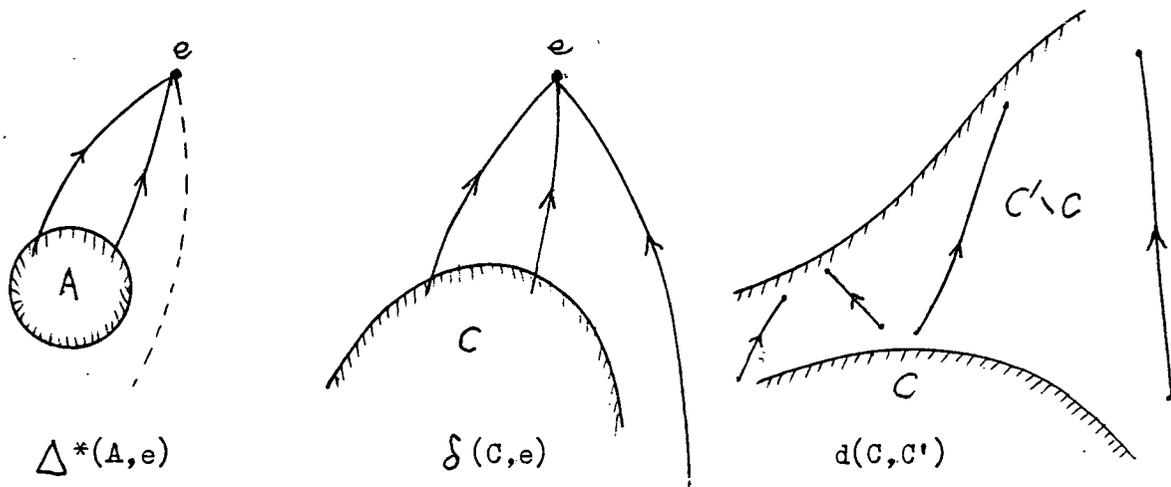
This may be thought of as giving the distance from e to the "cut" of \leq -maximal events of C; unlike Δ^* however the distance is the supremum of lengths of chains which need not "end up at" the cut.

(With a trick we can define δ from a Δ^* -measure; adjoin $\pm \infty$ elements to the event structure and then take

$$\delta(C, e) = \text{Sup}\{\Delta^*(\text{con}(C' \cup \{+\infty\}), e), \Delta^*(\text{con}(C' \cup \{-\infty\}), e)\}$$

where C' is the set of \leq -maximal events of C.)

We summarise the three distance measures Δ^* , δ and d together pictorially - the solid lines denote chains which make a contribution to the value:



The next theorem characterises observable states using δ .

Theorem 7.1.17

Suppose E is an event structure and C a left-closed subset of E.

Then

$$C \in \mathcal{O}(E) \text{ iff } \forall e \in E \delta(C, e) < \infty.$$

Proof

" \Rightarrow " If $C \in \mathcal{O}(E)$ we have $C = \text{os}(O, t)$ for some observer O and time t . For e in E we have $\delta(C, e) \leq |t - O(e)| < \infty$.

" \Leftarrow " If $\delta(C, e) < \infty$ for all e then define

$$\begin{aligned} O(e) &= \delta(C, e) \text{ if } e \in C \\ &= -\delta(C, e) \text{ otherwise.} \end{aligned}$$

Then $C = \text{os}(O, 0)$. ■

7.2 Reachability classes

We first note that there is a natural equivalence relation on observers which induces a reachability relation on observable states. (Throughout this section event structures will be countable observable.)

Definition 7.2.1

Suppose E is a countable observable event structure. For O, O' in $\text{Ob}(E)$ define

$$O \sim_1 O' \text{ iff } \exists t, t' \text{ os}(O, t) = \text{os}(O', t').$$

Then define \sim as the transitive closure of \sim_1 . Further, for C, C' in $\mathcal{O}(E)$, define

$$C \approx C' \text{ iff } \exists O, O' \in \text{Ob}(E) \exists t, t' O \sim O' \ \& \ \text{os}(O, t) = C \ \& \ \text{os}(O', t') = C'.$$

A major point is that there may be more than one \approx -equivalence class. (Certainly there is at least one as the event structures are assumed countable observable.) This is best seen through a characterisation of \approx using the metric d .

Theorem 7.2.2

Let E be a countable observable event structure. Suppose C, C' are observable states. Then

$$C \approx C' \text{ iff } d(C, C') < \infty.$$

Proof

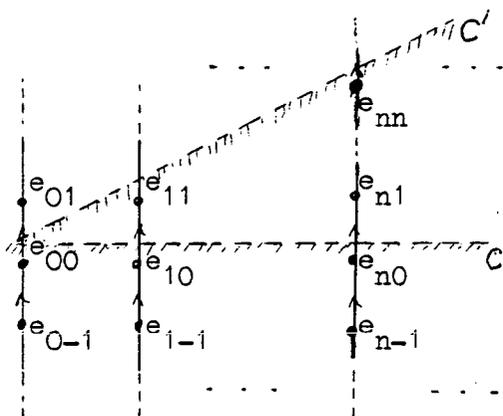
" \Leftarrow " Suppose C, C' are observable states such that $d(C, C') < \infty$.

Then by the properties of the metric d (see 5.2.5) we have $d(C \cap C', C) < \infty$ and $d(C \cap C', C') < \infty$. The convex subset $C \setminus C'$ is thus bounded. Also $\Delta^*(C \setminus C', e) < \infty$ for all e (otherwise $\delta(C, e) = \infty$ or $\delta(C', e) = \infty$ for some e). Thus application of lemma 7.1.14 yields an observer O and times k_1 and k_2 such that $os(O, k_1) = C \cap C'$ and $os(O, k_2) = C$. Similarly there is an observer O' and times k'_1 and k'_2 such that $os(O', k'_1) = C \cap C'$ and $os(O', k'_2) = C'$. Thus $C \approx C'$.

" \Rightarrow " Suppose $C \approx C'$ for observable states C and C' . Then for some observers O and O' and times t and t' we have $O \sim O'$ and $C = os(O, t)$ and $C' = os(O', t')$. Induction on the number of \sim_1 steps in $O \sim O'$, using the triangle inequality for d , gives $d(C, C') < \infty$.

The event structure in the following example is now easily seen to possess more than one \approx -equivalence class and correspondingly more than one \sim -equivalence class of observers.

Example 7.2.3



This event structure consists of a countable infinity of unbounded chains of events. The observable states $C = [\{e_{i0} | i \in \omega\}]$ and $C' = [\{e_{ii} | i \in \omega\}]$ ("diagonal to" C) have been indicated. Obviously $d(C, C') = \infty$.

We note a countable-observable event structure may be recovered from a \sim -equivalence class of observers.

Theorem 7.2.4

Suppose E a countable-observable event structure. For each observer O define:

$$e \leq_0 e' \text{ iff } O(e) \leq O(e').$$

Then $\leq = \bigcap_{O \sim O'} \leq_0$.

Proof Suppose O is an observer of the event structure E .

Obviously $\leq \subseteq \bigcap_{O \sim O'} \leq_0$. Conversely suppose $e \leq e'$. If

$O(e) > O(e')$ then $(e, e') \notin \leq_0$ so $(e, e') \notin \overset{\curvearrowright}{O \sim_0} \leq_0$, as required.
 Otherwise define an observer O' for which $O'(e) > O'(e')$ and
 $O' \sim O$ by $O'(\varepsilon) = O(\varepsilon)$ if $e \not\leq \varepsilon$
 $= O(\varepsilon) + O(e') - O(e) + 1$ otherwise. ■

It seems a course of computation should be associated with a unique \approx -equivalence of observable states and accordingly with one and only one \sim -equivalence class of observers. Certainly in [Pet 2], where the axioms for "ropes" are presented, Petri motivates the K-density axiom by saying that "otherwise, there would exist cases c_1, c_2 such that c_2 can be reached from c_1 only by an infinite number of steps, by performing a "super task".

So, cases are to be reachable from each other in some sense.

(Interestingly K-density does not do this for the reachability relation \approx induces on cases. There is an obvious K-dense net associated with the event structure of example 7.2.3.) The main result of this section is to characterise event structures with a unique \approx -equivalence class. Alone, without extra structure, they are adequate to represent a course of computation.

Definition 7.2.5

Suppose E is a countable-observable event structure. Then E is said to be adequate

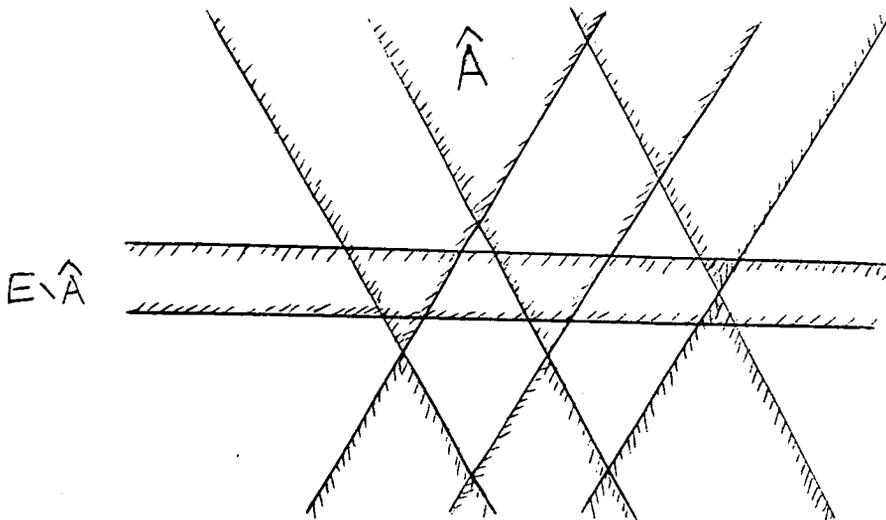
$$\text{iff } \forall C, C' \in \mathcal{OS}(E) \ d(C, C') < \infty.$$

We define the property characterising adequate event structures.

Definition 7.2.6

Let E be an event structure. For A a subset of E we define $\hat{A} = \{e \in E \mid \exists a \in A \ a \leq e \text{ or } e \leq a\}$. We say E is almost bounded iff for some finite subset A of E , $E \setminus \hat{A}$ is bounded.

If E is almost bounded then it consists of a "tall thin bit" (\hat{A}) and a "short broad bit" $(E \setminus \hat{A})$. So pictorially it looks like:



Theorem 7.2.7

Let E be a countable observable event structure. Then E is adequate iff E is almost bounded.

Proof Let E be an event structure. We are assuming that E is countable and $\forall e, e' \in E \Delta(e, e') < \infty$.

" \Leftarrow " If E is almost bounded then for a finite subset A we have $E \setminus \hat{A}$ is bounded by k , say. Suppose $C, C' \in \mathcal{O}(E)$. We have $d(C, C') \leq \text{Sup}(\{k\} \cup \{\delta(C, a) \mid a \in A\} \cup \{\delta(C', a) \mid a \in A\})$ by the definitions of d and δ . As A is finite theorem 7.1.17 ensures $d(C, C') < \infty$ as required.

" \Rightarrow " Suppose E is adequate. We assume E is not almost bounded to obtain a contradiction.

Enumerate E as $\varepsilon_0, \varepsilon_1, \dots, \varepsilon_i, \dots$ and define $E_i = \{\varepsilon_0, \dots, \varepsilon_i\}$. As E is assumed not almost bounded we can inductively define pairs e_i, e'_i where $e_i \leq e'_i$ with $e_i = \varepsilon_{k_i}$ and $e'_i = \varepsilon_{l_i}$ in the enumeration such that

- (a) $\Delta(e_i, e'_i) > i$
- (b) $e_i, e'_i \notin \hat{E}_{\max\{k_{i-1}, l_{i-1}\}}$

Now define $C = \{e_i \mid i \in \omega\}$ and $C' = \{e'_i \mid i \in \omega\}$. Both C and C' are pairwise incomparable. In order to apply corollary 7.1.15 we establish $\Delta^*(C, e) < \infty$ and $\Delta^*(C', e) < \infty$ for all e . To show $\Delta^*(C, e) < \infty$ suppose $e = \varepsilon_k$ in the enumeration. We have $k \leq \max\{k_n, l_n\}$ for some n . Thus by the definition of the pairs e_i, e'_i for $i > n$ we have e incomparable with e_i . Therefore

$$\Delta^*(C, e) = \Delta^*({e_0, \dots, e_n}, e) < \infty.$$

Similarly one may show $\Delta^*(C', e) < \infty$ for all e .

By application of corollary 7.1.15 there are observers O and O' and times t and t' such that $\forall c \in C \ O(c) = t$ and $\forall c' \in C' \ O'(c') = t'$. Defining $D = \text{os}(O, t)$ and $D' = \text{os}(O', t')$ gives two observable states D and D' with $d(D, D') = \infty$ i.e. the event structure is not adequate.

This is a contradiction so E must be almost bounded. ■

7.3 An axiomatisation of the reachability class

We have defined the reachability classes of an event structure. The elements of a reachability class are ordered naturally by inclusion. We can axiomatise those structures and mention how to prove the axiomatisation is complete by establishing a representation theorem. This provides a reachability class of an event structure from a partial order satisfying the axioms. In stating them we first introduce some new definitions.

Definition 7.3.1

Let $L = (L, \subseteq)$ be a poset. Say L is non-null consistently complete iff for every non-null subset $A \ \exists x \in L \ A \subseteq x$ implies $\sqcup A$ exists in L .

The consistent-completeness property is commonly used. Here as we do not necessarily have an initial state we have weakened it a little to only cover non-null subsets.

In our previous work on event structures in chapters 4 to 6 the concept of complete primes was the domain analogue of event; in the representation theorems of chapter 4 a prime corresponded to $[e]$ where e was an event. Here such left-closures may not be observable states. For this reason the more general concept of "relatively (complete) prime" is introduced.

Definition 7.3.2

Let $L = (L, \subseteq)$ be a partial order with elements x and p . Then we say p is completely prime relative to x , and write this as $x \dashv\vdash p$, iff for all non-null subsets A of L for which $\sqcup A$ exists we have:

$$x \subseteq A \ \& \ p \in \sqcup A \Rightarrow \exists a \in A \ p \subseteq a.$$

We write $x \twoheadrightarrow p$ iff $x \rightarrow p$ or $x = p$.

Note that \rightarrow need not be transitive. (Consider the observable states of the event structure consisting of two \leq -incomparable events e and e' . Then $\emptyset \rightarrow \{e\} \rightarrow \{e, e'\}$ but $\emptyset \not\rightarrow \{e, e'\}$.)

Unfortunately I cannot see how to avoid almost explicitly introducing the idea of reachability into the axiomatisation. To do this we make the following definition of a domain analogue of the metric d .

Definition 7.3.3

Let $L = (L, \sqsubseteq)$ be a partial order. For $x, y \in L$ s.t. $x \sqsubseteq y$ define

$$\text{depth}(x, y) = \text{Sup}\{n \mid \exists p_1, \dots, p_n (\forall i x \rightarrow p_i) \wedge p_1 \sqsubset p_2 \sqsubset \dots \sqsubset p_n \sqsubseteq y\}$$

(If the supremum is infinite we denote its value by ∞ .)

We can now state the axioms which will characterise the reachability classes.

Definition 7.3.4 (Axioms for reachability classes)

Let $L = (L, \sqsubseteq)$ be a partial order. Referring to the above definitions we are interested in the following set of axioms.

1. L is a lattice.
2. L satisfies non-null bounded-joins.
3. If $x \sqsubseteq y$ then $\bigsqcup \{p \sqsubseteq y \mid x \twoheadrightarrow p\}$ exists in L and equals y .
4. $\bigsqcup \{y \mid x \prec y\}$ and $\bigsqcap \{y \mid y \prec x\}$ exist in L .
5. If $x \sqsubseteq y$ then $\text{depth}(x, y) < \infty$.

A few comments on the axioms: Axioms 1 and 2 are clear; axiom 3 replaces that of prime algebraicity in the absence of an initial null state; axiom 4 is a completeness axiom mirroring the fact that we allow an arbitrary set of events to fire concurrently; as mentioned above the intention of axiom 5 is to restrict us to a reachable class.

It can be shown that the reachability class obtained from an event structure (of this chapter) satisfy the above axioms. Far more tediously, from such a structure L one can obtain an event structure with reachability classes (ordered by inclusion) naturally

isomorphic to L . The basic idea is simple. From such a partial order L define events to be equivalence classes of pairs $[x, y]$ where $x \rightarrow y$. The equivalence relation is the transitive symmetric closure \equiv of \leq_1 where

$$[x, y] \leq_1 [x', y'] \text{ iff } x \sqsubseteq x' \ \& \ y' = x' \sqcup y.$$

The required partial ordering on such events is

$$e \leq e' \text{ iff } \exists x, x', x'' \ x \rightarrow x' \ \& \ x \rightarrow x'' \ \& \ [x, x'] \in e \ \& \\ [x, x''] \in e' \ \& \ x' \sqsubseteq x''$$

(It requires a fair bit of tedium to show it is a partial order.)

7.4 Causal nets representing processes with infinite pasts and K-density

As in chapter 5 the results on event structures may be transferred to nets so that a restricted form of K-density holds.

Definition 7.4.1

Let $N = (B, E, F)$ be a causal net. As in chapter 5 define $\mathcal{E}(N) = (E, F^* \uparrow E)$. Say N is countable-observable iff $\mathcal{E}(N)$ is countable-observable.

Say N is adequate iff $\mathcal{E}(N)$ is adequate.

Again as in chapter 5 observable states of the event structure induced by a net N determine observable cases of the net via the Fr_N map introduced in chapter 5; we require the net to satisfy axiom $N3$ in order to get real cases.

Definition 7.4.2

Let N be a countable-observable causal net satisfying $N3$. Define the observable cases of N to be those subsets of conditions of the form $\text{Fr}_N(C)$ where $C \in \mathcal{O}(\mathcal{E}(N))$.

Proposition 7.4.3 (Restricted K-density)

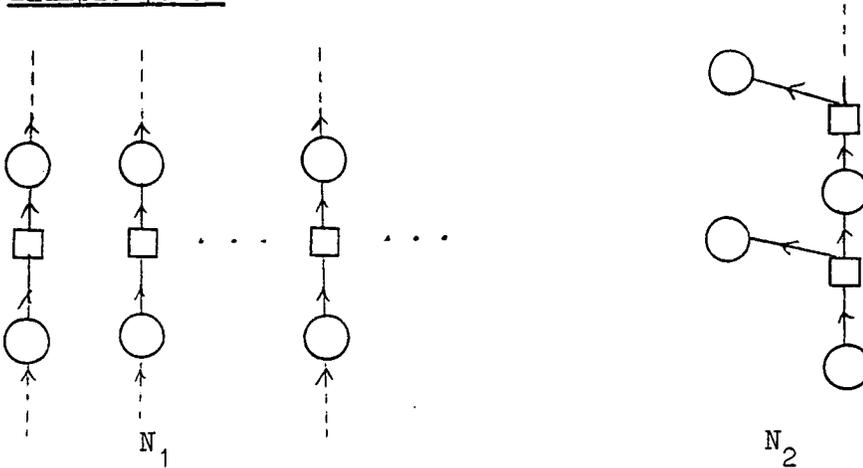
Let N be a countable-observable causal net satisfying $N3$. Then any observable case is a Petri case. Also any observable case meets any sequential process of N .

Proof We sketch the proof that a restricted form of K-density holds: Clearly any kens of \leq in the induced event structure must have order type n , \mathbb{Z}^+ , \mathbb{Z}^- or \mathbb{Z} . Let C be the observable case observed by

observer O at time t in \mathbb{Z} . Thus observation of a kens of F^* must "straddle" O , have finished or not yet stated at time t . In all cases a condition holds at time t which is in the corresponding Petri-case. ■

Finally we note from the following example that neither does K-density imply adequacy nor adequacy imply K-density.

Example 7.4.4



The net N_1 is K-dense but not adequate. The net N_2 is adequate but not K-dense.

Chapter 8. The full-abstractness problem for PCF - an introduction

We introduce an open problem in denotational semantics. It concerns the language PCF (programming language for computable functionals) a kind of typed lambda calculus. Terms of ground type called programs are evaluated deterministically by rules including the lambda calculus conversion rules. This gives a natural criterion for determining the operational equivalence of terms of PCF. The problem is to construct a denotational model which exactly reflects this equivalence in a way which does not refer directly to the operational behaviour. Only then can we rely on abstract semantic properties of the model to prove such things as the operational equivalence or non-equivalence of terms. Although the language PCF is superficially unlike many programming languages essentially the same phenomenon can be found in "real" languages such as Algol, Pascal and Iswim whose programs are generally evaluated deterministically on a machine.

In this chapter we outline the existing work. Gordon Plotkin introduced the problem [Plo1], Robin Milner showed the denotational semantics was unique [Mil2] and Gérard Berry made significant steps in characterising the model for the denotational semantics [Ber]. We summarise Plotkin's and Milner's work in the first section and Berry's in the second. We give sufficient details of Berry's work to support our use of event structures to duplicate a bit of his work. We shall not discuss the important work of Curien [Cur], [Ber and Cur] in much detail because we do not refer to it in chapter 9.

If this chapter contains anything original it is probably a mistake in copying out, translating or understanding. We refer the reader to [Mac] or [Arb] for the relevant category theory.

8.1 The problem

PCF is a programming language based on LCF, Scott's logic of computable functions, ([Plo1],[Mil2]). It is a form of typed lambda calculus in which certain terms are singled out as programs.

The set of types is the least set containing \top (for Booleans), $\mathbb{1}$ (for integers) and $(\sigma \rightarrow \tau)$ whenever it contains σ and τ . We use $(\sigma_1, \dots, \sigma_n; \tau)$ to abbreviate $(\sigma_1 \rightarrow (\sigma_2 \rightarrow \dots (\sigma_n \rightarrow \tau) \dots))$. The types \top and $\mathbb{1}$ are called ground types.

Terms are produced from the following collection of constant functions with the indicated types:

$0, 1, \dots, n, \dots$: type ι	(numerals)
tt, ff	: type δ	(truth values)
$+1, -1$: type $\iota \rightarrow \iota$	(increment and decrement by 1)
Z	: type $\iota \rightarrow \delta$	(test for zero)
\mathcal{I}	: type $(\delta, \iota, \iota; \iota)$	(conditional giving integer result)
\mathcal{B}	: type $(\delta, \delta, \delta; \delta)$	(conditional giving boolean result)
\mathcal{Y}_σ	: type $((\sigma \rightarrow \sigma) \rightarrow \sigma)$	(least fixed point operator)

Starting with the above collection of constants and countably many variables x_i ($i \in \omega$) for each type the terms are given by the formation rules:

1. Every variable x_i^σ is a term of type σ .
2. Every constant of type σ is a term of type σ .
3. If M and N are terms of type $\sigma \rightarrow \tau$ and σ respectively then (MN) is a term of type τ .
4. If M is a term of type τ then $(\lambda x_i^\sigma M)$ is one of type $\sigma \rightarrow \tau$.

In the standard way one defines the free variables of a term, the closed terms and contexts which are terms with "holes" to be filled by terms of the appropriate type; we write $C[\dots,]$ for a context which when filled looks like $C[M_1, \dots, M_n]$. By $[M/x_i]N$ is meant the result of substituting the term M for all free occurrences of x_i in N , making appropriate changes in the bound variables of N so that no free variables of M become bound.

The programs are closed terms of ground type. Intuitively they yield concrete output; other terms are significant only as subterms of programs.

An operational semantics is given to the language by defining eval a partial function from programs to constants. It is defined using an immediate reduction relation \rightarrow between terms:
 $\text{eval}(M) = c$ iff $M \rightarrow^* c$, for any program M and constant c .

The immediate reduction relation is given by:

1. $+1 \underline{n} \rightarrow \underline{n+1}$
2. $-1 \underline{n+1} \rightarrow \underline{n}$
3. $Z \underline{0} \rightarrow \underline{tt}$
 $Z \underline{n+1} \rightarrow \underline{ff}$

4. $\begin{cases} \mathcal{D}_\sigma \text{tt } MN \rightarrow M \\ \mathcal{D}_\sigma \text{ff } MN \rightarrow N \end{cases} \} \sigma = \bar{0}, \bar{1}$
5. If $M \rightarrow M'$ then $\mathcal{D}_\sigma M \rightarrow \mathcal{D}_\sigma M'$ for M, M' of type $\bar{0}$ and σ a type $\bar{0}$ or $\bar{1}$
6. If $M \rightarrow M'$ then $(MN) \rightarrow (M'N)$
7. If M is $\underline{+1}$ or $\underline{-1}$ or Z and $N \rightarrow N'$ then $(MN) \rightarrow (MN')$
8. $Y_\sigma M \rightarrow M(Y_\sigma M)$
9. $((\lambda x.M)N) \rightarrow [N/x]M$

The relation \rightarrow is a partial function so eval is well-defined above.

We base the notion of a standard model for PCF on type structures. A (standard) type structure consists of

1. A cpo D_σ for each type σ with $D_{\bar{1}} = \mathbb{N}$ and $D_{\bar{0}} = \mathbb{T}$.
2. For all types σ and τ a two place application operation $\bullet : D_{\sigma \rightarrow \tau} \times D_\sigma \rightarrow D_\tau$ which is continuous and order extensional i.e. $x \sqsubseteq x'$ iff $\forall y x.y \sqsubseteq x'.y$.

Condition 2. ensures that the elements of $D_{\sigma \rightarrow \tau}$ are in 1-1 correspondence with a subset of the continuous functions $[D_\sigma \rightarrow D_\tau]$ so that the ordering on $D_{\sigma \rightarrow \tau}$ is the restriction of the pointwise ordering on functions.

With respect to a type structure the environment Env consists of all type-respecting functions ρ from variables into $\bigcup_\sigma D_\sigma$.

A standard model for PCF consists of a type structure D and a semantics \mathcal{M} a type-respecting map giving values in D to terms in an environment ρ . They are required to satisfy the following conditions:

1. The terms \underline{n} , $\underline{+1}$, $\underline{-1}$, Z , \mathcal{D}_σ , \mathcal{D}_τ and Y get their usual interpretation. Thus

$$\mathcal{M}[\underline{Y}] \rho . \alpha = \bigsqcup_{n \in \omega} \alpha^n . \perp_\sigma \quad \text{where } \alpha^n \text{ abbreviates } \underbrace{\alpha . (\alpha . (\alpha . \dots))}_n$$

n α 's.

2. $\mathcal{M}[\underline{x}] \rho = \rho(x)$
 $\mathcal{M}[\underline{MN}] = \mathcal{M}[\underline{M}] \rho . \mathcal{M}[\underline{N}] \rho$
 $\mathcal{M}[\underline{\lambda x.M}] \rho . \alpha = \mathcal{M}[\underline{M}] \rho [x/\alpha]$
 $(\rho [x/\alpha])$ is the environment obtained from ρ by changing it so the

variable x is associated with α).

Not all type structures determine models; there may be simply not enough functions in the domains to support the semantics. An obvious standard model is obtained by taking the type structure so that $D_{\sigma \rightarrow \tau} = [D_{\sigma} \rightarrow D_{\tau}]$, all continuous functions from D_{σ} to D_{τ} , with the application operator just the ordinary application of functions. Many other models are possible and according to criteria derived from the operational semantics the obvious model is not the best.

The denotational semantics should "match" the operational semantics. Plotkin defined two natural operational relations. Terms are of interest only insofar as they are part of programs. For this reason it is natural to regard two terms as operationally equivalent if they can be freely substituted for each other in a program without affecting its behaviour. Formally define the equivalence relation by: $M_{\sigma} \approx N_{\sigma}$ iff whenever $C[M_{\sigma}]$ and $C[N_{\sigma}]$ are programs $eval(C[M_{\sigma}])$ and $eval(C[N_{\sigma}])$ are both undefined or otherwise defined and equal. More generally an operational preorder can be defined by: $M_{\sigma} \preceq N_{\sigma}$ iff whenever $C[M_{\sigma}]$ and $C[N_{\sigma}]$ are programs then whenever $eval(C[M_{\sigma}])$ is c then so is $eval(C[N_{\sigma}])$.

Clearly $M \approx N$ iff $M \preceq N$ and $N \preceq M$. For a semantics \mathcal{M} the expected semantic counterparts of these two relations are the relations on terms given by $M_{\sigma} \preceq_m N_{\sigma}$ iff $\mathcal{M}[[M]]\rho \preceq \mathcal{M}[[N]]\rho$ for all ρ , $M_{\sigma} \equiv_m N_{\sigma}$ iff $M_{\sigma} \preceq_m N_{\sigma}$ and $N_{\sigma} \preceq_m M_{\sigma}$. In the circumstance when the relations \preceq and \preceq_m coincide the semantics \mathcal{M} is said to be fully abstract.

For a standard semantics \mathcal{M} the denotational relations will be included in the corresponding operational ones. However the converse will not generally hold. In particular Plotkin showed the obvious semantics based on taking $D_{\sigma \rightarrow \tau}$ as all continuous functions $[D_{\sigma} \rightarrow D_{\tau}]$ is not fully abstract. The counterexample depended on producing two terms which were operationally equivalent but denotationally distinct through acting differently on parallel or. Parallel or, (call it por) is of type $(\mathcal{D}, \mathcal{D}; \mathcal{D})$ and has this truth table.

por	⊥	tt	ff
⊥	⊥	tt	⊥
tt	tt	tt	tt
ff	⊥	tt	ff

It examines two arguments in parallel and if either is tt it yields tt. Compare it with sequential versions of or (called lor and ror) which are obliged to look at one argument first (the left argument or the right argument).

lor	⊥	tt	ff
⊥	⊥	⊥	⊥
tt	tt	tt	tt
ff	⊥	tt	ff

ror	⊥	tt	ff
⊥	⊥	tt	⊥
tt	⊥	tt	tt
ff	⊥	tt	ff

$$\text{lor} = \lambda xy. x \supset x, y$$

$$\text{ror} = \lambda xy. y \supset y, x.$$

Unlike lor and ror parallel or turns out not to be definable in PCF and because of this no program context can discriminate between the two terms. Plotkin produced. He showed how by extending the language PCF to allow limited parallelism the obvious model became fully abstract.

Rather than extend the language PCF Milner showed how by restricting the model the semantics would be fully abstract. As a corollary of more general results he showed there was a unique fully abstract model for PCF (to within isomorphism) which he characterised as being that model in which all isolated elements of the domain were definable in PCF. (An element is definable if there is a closed term which denotes it.)

In fact in establishing the model's existence, Milner essentially constructed it from equivalence classes of terms determined by the operational relations. This method failed to specify directly, without reference to terms, precisely those functions which were allowed in the model. From the results of Plotkin and Milner it was clear that they had to be sequential in some sense but no existing definition of sequential cuts down the functions appropriately. The Kahn-Plotkin definition although precisely right

for low types of the form $(\sigma_1, \dots, \sigma_n; \tau)$ where σ_i and τ are ground types does not extend up the types as the concreteness axioms fail there. The Milner and Vuillemin definitions, though satisfied by the functions are not restrictive enough. The problem remains of giving a purely semantic characterisation of the fully abstract model.

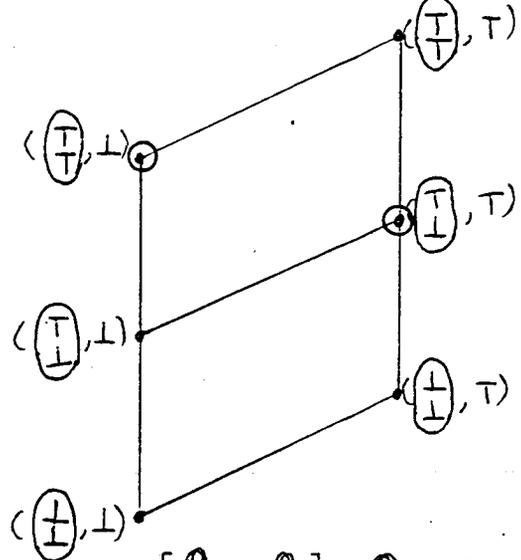
8.2 The work of Gérard Berry

In the last section domains possessed only one ordering. Call it the extensional ordering as it reflects the extensional behaviour of the elements. On functions it was determined pointwise and it relates functions according to what values they give on arguments. With respect to this order the functions defined in PCF were continuous. If further operational behaviour of terms is to be reflected semantically so as to cut down the functions in a model of PCF one expects that domains should carry extra structure. For instance any notion of sequential function between domains should account for the nature of the objects represented in the domain. A function being sequential between concrete domains representing concrete input and output should not mean the function is sequential when the same domains stand for functions ordered extensionally. Nor is the converse expected - see examples 8.2.1 and 8.2.2. Once the extra structure has been introduced to restrict the functions of the model one hopes that by then dropping it Milner's fully abstract model will be obtained. These are the ideas of Gérard Berry who introduced the stable ordering as new structure ($[\text{Ber}]$, $[\text{Ber and Cur}]$).

The following two examples illustrate the need for extra structure which must at least distinguish functions from basic values.

Example 8.2.1

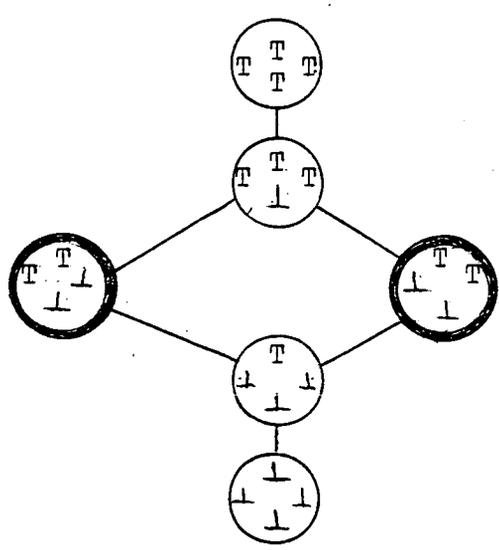
The application map $\text{ap}: [\mathbb{O} \rightarrow \mathbb{O}] \times \mathbb{O} \rightarrow \mathbb{O}$, acting as $\text{ap}(f, x) = f(x)$, is intuitively sequential. Encircling the least values of $[\mathbb{O} \rightarrow \mathbb{O}] \times \mathbb{O}$ which yield \top under ap we get:



Clearly the domain $[D \rightarrow D] \times D$ ordered extensionally satisfies all the axioms of concrete domains and ap is not Kahn-Plotkin sequential.

Example 8.2.2

The function $f: [D^2 \rightarrow D] \rightarrow D$ defined by $\lambda g.g(g(T, \perp), g(\perp, T))$ gives T for the following least values. Again it is not Kahn-Plotkin sequential.



We trace how the stable ordering arose. One line of motivation is from the construction of syntactic models of the lambda calculus. The idea is to capture syntactic properties in a semantic way and so restrict the functions present in a model. For example Berry has shown that the operation of enclosing terms in a context induces a Kahn-Plotkin sequential function between domains of the syntactic model. The syntactic ordering in the syntactic model is the prefix ordering on Boehm trees, a kind of normal form ([Ber]). He conjectures that for the fully abstract model of PCF the stable ordering is the image of this syntactic order.

In defining syntactic models of the typed lambda calculus it was natural to abandon the extensional ordering and even forget that terms defined functions. This led to a more general definition of model without the order extensional condition of the last section. For Berry a model of a typed lambda calculus is composed of the following ([Ber]):

1. A set of cpos E_σ one for each type σ . (A term in an environment denotes an element in one of these.)
2. A set of cpos D_σ one for each type σ . These are the domains of values which variables may be associated with. The environment Env consists of all type respecting functions ρ from variables into $\bigcup_{\sigma} D_\sigma$.

3. Two continuous application functions:

$$\bullet : D_{\sigma \rightarrow \tau} \times D_\sigma \rightarrow D_\tau$$

$$\bullet : E_\sigma \times \text{Env} \rightarrow D_\sigma$$

4. A semantics \mathcal{M} which is a type-respecting map from terms into $\bigcup_{\sigma} E_\sigma$ so that:

$$\mathcal{M}[\mathbf{x}].\rho = \rho(\mathbf{x})$$

$$\mathcal{M}[\mathbf{M}\mathbf{N}].\rho = (\mathcal{M}[\mathbf{M}].\rho) \cdot (\mathcal{M}[\mathbf{N}].\rho)$$

$$(\mathcal{M}[\lambda x.M].\rho)\alpha = \mathcal{M}[\mathbf{M}].\rho[x/\alpha] \text{ for all } \alpha \text{ in } D.$$

Such a model is said to be extensional when for α, α' in either $D_{\sigma \rightarrow \tau}$ or E_σ we have $\alpha = \alpha'$ iff $\alpha \cdot \beta = \alpha' \cdot \beta$ for all β . It is said to be order extensional when for α, α' in either $D_{\sigma \rightarrow \tau}$ or E_σ we have $\alpha \sqsubseteq \alpha'$ iff $\forall \beta \alpha \cdot \beta \sqsubseteq \alpha' \cdot \beta$.

In this definition of model the cpos E can be thought of as functions from Env to values; the use of E leaves open precisely what functions to allow and what order to put on them. The definition ignores the constant functions of the language. Note however that fixed point operators Y_σ can be given a denotation exactly as for the standard models of the previous section because \exists gives the required monotonicity ($\beta \sqsubseteq \beta' \Rightarrow \alpha \cdot \beta \sqsubseteq \alpha \cdot \beta'$). The models we shall discuss will always be extensional though not necessarily order extensional. In the work of Pierre-Louis Curien the model of algorithms is not even extensional ([Cur], [Ber and Cur]). Note that the standard models of the last section are order-

extensional models according to the above definition.

Berry and Curien together found a means of constructing models from suitable order-enriched categories called \bigwedge -categories. An order enriched category [Wan] is a category in which morphisms are ordered so that the hom-sets form cpos making composition continuous. A \bigwedge -category is an order-enriched category which is cartesian-closed so that category-theoretic constructions satisfy sufficient strictness and continuity restrictions. We refer the reader to the definition of \bigwedge -categories in [Ber] or [Cur] for the exact details. We give the general idea precisely enough to support our exposition of Berry's work.

Suppose we wish to constrain the model by imposing a condition P on domains and a condition Q on continuous functions. We shall do this soon when functions will have to be stable (Q) and cpos distributive with continuous meet (P). To obtain a model it is sufficient to verify the following conditions (which determine a \bigwedge -category):

1. Closure under composition: If D,E,F satisfy P and if $h: D \rightarrow E$ and $h': E \rightarrow F$ satisfy Q then $h' \circ h: D \rightarrow F$ satisfies Q. The identity 1_D for all D satisfying P satisfies Q.
2. Closure under products: If D,E satisfy P then $D \times E$ satisfies P. The projections from $D \times E$ onto D and E satisfy Q. For all F satisfying P and all $h: F \rightarrow D$ and $h': F \rightarrow E$ satisfying Q, the function $[h,h']: F \rightarrow D \times E$ defined by $[h,h'](\alpha) = (h(\alpha), h'(\alpha))$ satisfies Q. Also the same for countable products.
3. Closure under exponentiation: If D,E satisfy P then the set of functions $[D \rightarrow_Q E]$ which satisfy Q are ordered by \sqsubseteq_Q such that:
 - 3.1 $([D \rightarrow_Q E], \sqsubseteq_Q)$ is a cpo satisfying P.
 - 3.2 Application app: $[D \rightarrow_Q E] \times D \rightarrow E$ defined by $\text{app}(h, \alpha) = h(\alpha)$ satisfies Q.
 - 3.3 If D,E,F satisfy P and if $h: D \times E \rightarrow F$ satisfies Q then the map $\text{curry}(h): D \rightarrow [E \rightarrow_Q F]$ defined by $\text{curry}(h)(\alpha)(\beta) = h(\alpha, \beta)$ satisfies Q.
4. Continuity properties: The maps determined by composition \circ , the operation $[\cdot, \cdot]$ and "curryfication" are continuous (w.r.t. \sqsubseteq_Q).

Within the above set-up it is easy to construct a model from the morphisms:

Choose D_σ so $D_\sigma \xrightarrow{\gamma} = ([D_\sigma \rightarrow_Q D_\tau], \subseteq_Q)$ and $E_\sigma = ([Env \rightarrow_Q D_\sigma], \subseteq_Q)$. The environment satisfies P by closure under products. Put $\Pi_x(\rho) = \rho(x)$ and $S_x(\rho, \alpha) = \rho[x/\alpha]$ - again by closure under products Π_x and S_x satisfy Q.

Define the semantic function $\mathcal{M}[\cdot]$ by

$$\mathcal{M}[x] = \Pi_x$$

$$\mathcal{M}[MN] = \text{app} \circ [\mathcal{M}[M], \mathcal{M}[N]]$$

$$\mathcal{M}[\lambda x.M] = \text{curry}(\mathcal{M}[M] \circ S_x)$$

This determines a model. The above three definitions are abstract formulations of condition 4 in the definition of a model:

$$\begin{aligned} \mathcal{M}[x].\rho &= \Pi_x.\rho = \rho(x) \\ \mathcal{M}[MN].\rho &= \text{app} \circ [\mathcal{M}[M].\rho, \mathcal{M}[N].\rho] \\ &= (\mathcal{M}[M].\rho) \cdot (\mathcal{M}[N].\rho) \\ \mathcal{M}[\lambda x.M].\rho.\alpha &= \text{curry}(\mathcal{M}[M] \circ S_x).\rho.\alpha \\ &= (\mathcal{M}[M] \circ S_x)(\rho, \alpha) \\ &= \mathcal{M}[M](S_x(\rho, \alpha)) \\ &= \mathcal{M}[M](\rho[x/\alpha]). \end{aligned}$$

The category of cpos with morphisms the continuous functions ordered pointwise (extensionally) forms a \downarrow -category. The category of concrete domains with morphisms the sequential functions ordered extensionally does not; this is because it is not closed under exponentiation (see [Ber and Cur]).

Because of major difficulties in constructing a sequential model Berry initially narrowed his ambitions to forming one from an approximate notion of sequential function. He called such functions stable functions. Stability is a property in between sequentiality and continuity.

Definition 8.2.3

Suppose f is a continuous function from cpo D to cpo E .

Then f is stable iff it satisfies

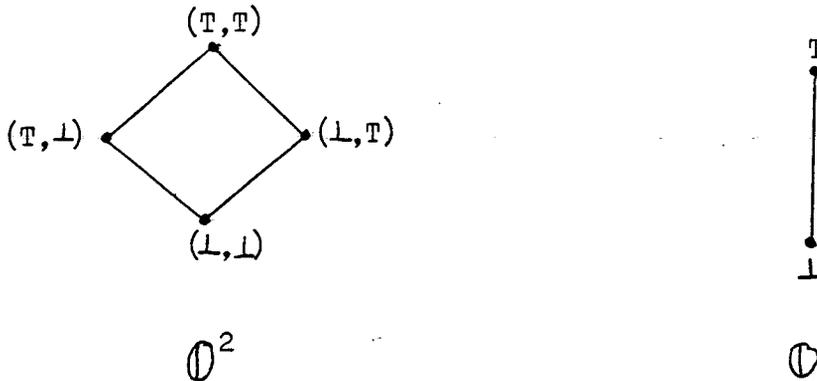
$$\forall x \in D \forall y \sqsubseteq f(x) \exists m(f, x, y) \in D, y \sqsubseteq f(x) \Leftrightarrow m(f, x, y) \sqsubseteq z.$$

The set of stable functions D to E is written as $[D \rightarrow_S E]$.

A function is stable if for all arguments x and all approximations y

of the result $f(x)$ there is a minimum approximation $m(f,x,y)$ which produces y under f . Thus the following functions are not stable. (Note parallel or is not.)

Example 8.2.4 (Non-stable functions)



The function $f: \mathbb{O}^2 \rightarrow \mathbb{O}$ defined by $f(\perp, \perp) = \perp$, $f(T, \perp) = f(\perp, T) = T$ is not stable as there are two minimal values (T, \perp) and (\perp, T) which produce T under f .

Parallel or: Importantly the function parallel or is not stable. It has two minimal values (tt, \perp) and (\perp, tt) which produce tt .

All Kahn-Plotkin sequential functions are stable. However the converse is false as is now shown.

Example 8.2.5 (A stable, non-sequential function)

Define $f: \mathbb{T}^3 \rightarrow \mathbb{O}$ to be the least monotonic function such that $f(\perp, tt, ff) = f(ff, \perp, tt) = f(tt, ff, \perp) = T$. Then f is stable; if $f(x) = T$ then x dominates one and only one of the points (\perp, tt, ff) etc. However f is not sequential; the directions from (\perp, \perp, \perp) correspond to argument places and no one is crucial to producing T .

Often it is convenient to work with a more general definition than that for stable functions. This definition determines the class of functions called conditionally multiplicative (mc). Often they are precisely the stable functions.

Definition 8.2.6

Suppose D, E are two cpos with meets denoted by \sqcap . Then a continuous function $f: D \rightarrow E$ is conditionally multiplicative (or mc) iff

$$\forall x, x' \in D \quad x \uparrow x' \Rightarrow f(x \sqcap x') = f(x) \sqcap f(x')$$

Call the set of such functions $[D \rightarrow_{mc} E]$.

Stable functions are always mc between domains with meets. The converse holds whenever the domains are algebraic, consistently complete and the restriction of the domain's orders to isolated elements is well-founded. In general neither the stable or mc functions form a cpo under the extensional or pointwise ordering. When the domains are consistently complete and algebraic the mc functions do form a cpo when ordered extensionally.

In order to form models from stable or mc functions they are required to form \bigwedge -categories. In this construction there is one major obstacle; the application function is not generally stable or mc with respect to the extensional ordering. For this reason Berry introduced another ordering, called the stable ordering \leq , on functions from D to E. Let D and E be two domains both with meets. To guarantee the application map app, defined $app(h, \alpha) = h(\alpha)$, is mc it is required that

$$h \uparrow_{\leq} h' \ \& \ \alpha \uparrow \alpha' \Rightarrow h \wedge h'(\alpha \sqcap \alpha') = h(\alpha) \sqcap h'(\alpha')$$

where " \wedge " denotes the meet of the stable ordering \leq . The stable ordering is chosen to ensure precisely this.

Definition 8.2.7

Let D,E be domains with meets. The stable ordering \leq on $[D \rightarrow_{mc} E]$ is defined by

$$h \leq h' \text{ iff } h \sqsubseteq h' \ \& \ \forall \alpha, \alpha' \in D \ \alpha \uparrow \alpha' \Rightarrow h(\alpha) \sqcap h'(\alpha') = h(\alpha') \sqcap h'(\alpha)$$

(Here $h \sqsubseteq h'$ means h is extensionally less than h')

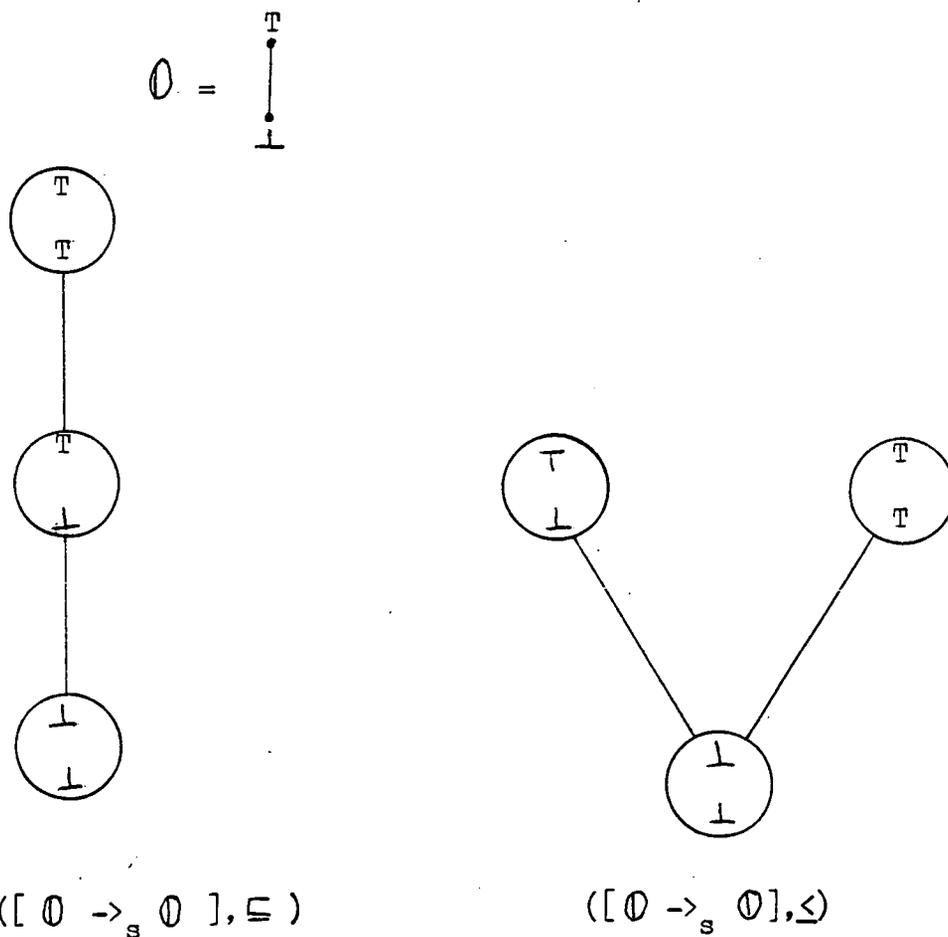
Intuitively the stable ordering orders functions according to the fashion in which they calculate values from arguments. For stable functions h and h' the function h being less than h' for the stable ordering means: whenever h gives an approximation to its final value for an argument then h' gives that approximation to its final value for the argument and moreover the minimal argument determining that approximation is the same for h and h'. The stable ordering is an ordering on the "behaviours" of functions. We make this more precise.

Proposition 8.2.8

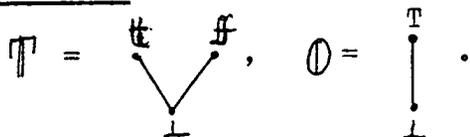
Let h and h' be stable functions from domains D to E which have meets and whose isolated elements are well-founded. Then $h \leq h'$ iff $h \sqsubseteq h'$ and $\forall x \in D \forall y \sqsubseteq h(x) m(h,x,y) = m(h',x,y)$ where $m(h,x,y)$ and $m(h',x,y)$ are the minimal arguments given by the definition 8.2.1 of stable functions.

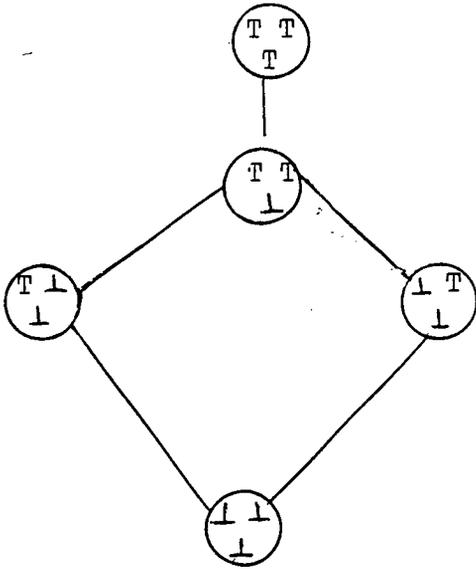
We omit the proof (which is not hard) but give some examples. We denote the extensional or pointwise ordering on functions by \sqsubseteq and the stable ordering by \leq . For these examples stable functions equal mc functions.

Example 8.2.9

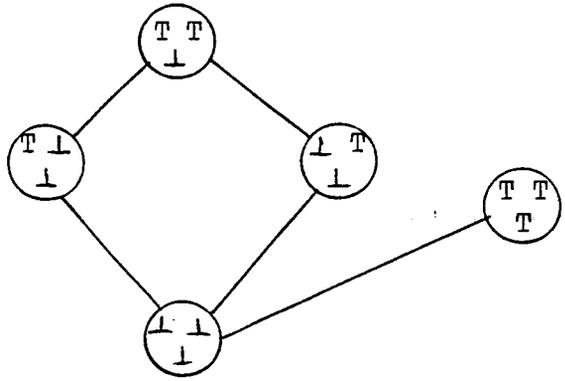


Example 8.2.10



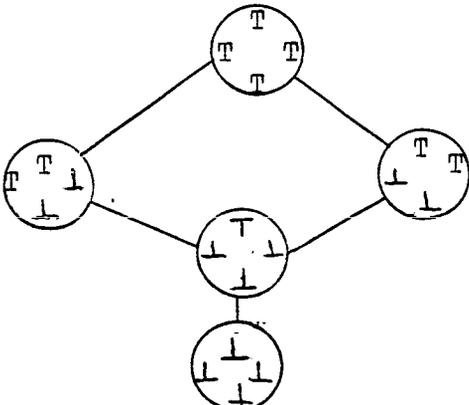
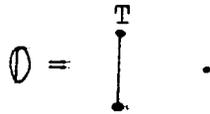
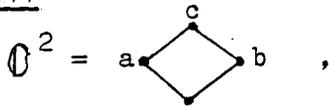


$([T \rightarrow_s \emptyset], \sqsubseteq)$

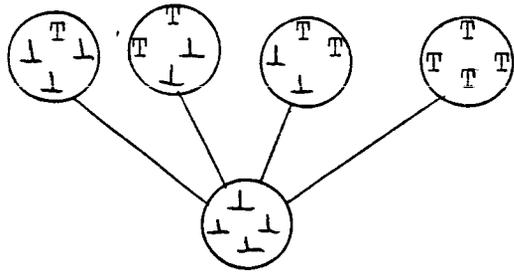


$([T \rightarrow_s \emptyset], \leq)$

Example 8.2.11



$([O^2 \rightarrow_s \emptyset], \sqsubseteq)$



$([O^2 \rightarrow_s \emptyset], \leq)$

Having quit the extensional order in favour of the stable one some further properties must be imposed on domains to get exponentiation. As yet we do not even know stable functions and mc functions from a cpo under the stable ordering. However the exponentiation of two domains will exist when they have continuous meets. This assumption is preserved by stable exponentiation when the domains are distributive, a property which is easily inherited

by products and exponentiations. The end result:

The category of distributive cpos with continuous meet having morphisms the mc functions ordered by the stable ordering is a \bigwedge -category. (And analogously when the morphisms are stable functions.)

Berry distinguishes a full subcategory of both the above categories. It is the category of dI-domains with objects those cpos which are in addition consistently complete, ω -algebraic and satisfy axiom F. In this category the notions of mc and stable coincide.

From the above \bigwedge -categories a model for PCF can be constructed. The "parasite" parallel or has been eliminated. However a new kind of "parasite" has been introduced namely functions which are not monotonic with respect to the extensional ordering. Such models cannot be fully abstract; they are not even order extensional with respect to the "hidden" extensional ordering. Fortunately this can be remedied. The trick is to order the domains in two ways, both extensionally and stably. Then in forming the exponentiation functions must be continuous with respect to the extensional ordering and mc or stable with respect to the stable ordering. Then dropping the stable ordering on morphisms gives a \bigwedge -category ordered extensionally. This produces an order extensional model (a standard model of the previous section); ground types are chosen so that the two orderings coincide.

The most general bi-ordered domains Berry considers form the category of BIOPCDs.

Definition 8.2.12

A biopcd is a structure $(D, \sqsubseteq, \leq, \perp)$ such that

- (i) The structure (D, \sqsubseteq, \perp) is a cpo with continuous meet .
- (ii) The structure (D, \leq, \perp) is a cpo. The identity $1_D: (D, \leq, \perp) \rightarrow (D, \sqsubseteq, \perp)$ is continuous.
- (iii) The function \sqcap is \leq -continuous.
- (iv) The following property holds

$\forall S, S' \subseteq D$ S, S' \sqsubseteq -directed

$(\forall s \in S \forall s' \in S' \exists t \in S, t' \in S' s \sqsubseteq t, s' \sqsubseteq t', t \leq t') \Rightarrow \sqcup S \leq \sqcup S'$.

Definition 8.2.13

The category BIOPCD is defined to consist of biopcds as objects with morphisms functions which are continuous w.r.t. the extensional ordering and mc w.r.t. the stable ordering.

The category BIOPCD is cartesian closed and "forgetting" one or other of the orders on morphisms yields two \bigwedge -categories. One is ordered extensionally and produces order extensional models.

An important cartesian closed full subcategory of BIOPCD is DBIOPCD which has distributive biopcds as objects.

Definition 8.2.14

A biopcd (D, \sqsubseteq, \leq) is distributive iff (D, \leq) is distributive and $x \uparrow^{\leq} y$ implies the stable supremum $x \vee y$ exists and equals the extensional supremum $x \sqcup y$.

The category DBIOPCD consists of objects the distributive biopcds with morphisms the mc functions.

The smallest category Berry introduces is the category of bidomain BIDOM. The extra restriction defining them ensures that w.r.t. the stable ordering they are dI-domains. Thus considered as a full subcategory of BIOPCD the mc restriction on functions in 8.2.13 is equivalent to insisting they are stable w.r.t. \leq .

Definition 8.2.15

A biopcd D is said to be a bidomain iff D is distributive and there is a \leq -increasing sequence $\{\psi_n \mid n \in \omega\}$ in $[D \rightarrow_{mc} D]$ so that the ψ_n are (\leq) -isolated and \leq -projections with limit 1_D .

The category BIDOM is defined to consist of objects the bidomains with morphisms functions which are continuous w.r.t. \sqsubseteq and stable w.r.t. \leq .

BIDOM is a cartesian closed full subcategory of BIOPCD (and DBIOPCD). Forgetting about one or other of the orders \sqsubseteq or \leq it produces two \bigwedge -categories; the extensional one gives an order extensional (standard) model of PCF - the domains at ground type are chosen to be $D_\nu = (\mathbb{N}, \sqsubseteq, \sqsubseteq)$ and $D_\circ = (\mathbb{N}, \sqsubseteq, \sqsubseteq)$. The model cuts out such functions as parallel or. However it is still not fully-

abstract because functions like that of example 8.2.5 which are not sequential but still included.

By induction on types Berry shows that the stable ordering is "hidden" in the fully abstract model of PCF and that the functions in it are stable with respect to it. As remarked above the fully-abstract model cannot contain all such functions. For first order types (of the form $(\sigma_1, \dots, \sigma_n; \tau)$ where σ_i and τ are ground types) he shows that the stable order is the image of the syntactic order and that the extensional order is the image of Plotkin's operational preorder \sqsubseteq on terms. He conjectures that this state of affairs holds at all types in the fully-abstract model.

The work of Berry and Curien ([Ber and Cur], [Cur]) on models of algorithms shows the stable ordering will be very important for a semantic construction of the fully-abstract model. Some obvious approaches do not work however. The stable ordering alone does not support sequential functions; both parts of axioms Q for \leq can fail (see 8.2.10) and even coherence of \leq goes (consider \leq for example 8.2.5). This is why they have produced models of algorithms which are not extensional but do preserve the concreteness axioms up the types. Crudely put, an algorithm is built up from "events" which may be decisions to output or decisions to test input.

Chapter 9. Higher type event structures

In this chapter we show how event structures may be used to represent exponentiations and products of domains. In particular we produce a category of stable event structures which represent a cartesian closed full subcategory of Berry's bidomains. We construct the category independently of Berry's results though, of course, the basic intuitions come from Berry's work. Finally we link up configurations of the event structures with bidomains. In fact this is how it was done based on a few heuristic guidelines which we present in the first section. There are many gaps in our understanding. In particular we introduce a new ordering \sqsubseteq^L , a sort of dual to Berry's stable ordering; how is it to be interpreted and is there a natural operational characterisation like the one Berry conjectures for the stable ordering? In the final section we indicate how the techniques might be refined to construct a fully-abstract model of PCF which depends on capturing its sequential evaluation. There are many issues raised and left open by this chapter; in this sense it is an introduction albeit a rather lengthy one. We refer to [Mac] for the basic category theory used.

9.1 Introducing higher type event structures

We start with a simple example of a higher type event structure which illustrates what we mean by them and how they are to be used. Let us look at event structures of the form (E, \leq, \otimes) satisfying the single axiom $e \geq e' \otimes e'' \Rightarrow e \otimes e''$. These were introduced in chapter 4 where we showed how such event structures represented coherent prime algebraic domains. We showed that such an event structure determined and was essentially determined by a coherent prime algebraic domain; the left closed consistent subsets of an event structure E ordered by inclusion formed the coherent prime algebraic domain $\mathcal{L}(E)$ and conversely such a domain D determined an event structure E , with events the complete primes, so that $\mathcal{L}(E) \cong D$.

Suppose (E_i, \leq_i, \otimes_i) for $i = 0, 1$ are two such event structures. Can we also represent the function space $[\mathcal{L}(E_0) \rightarrow \mathcal{L}(E_1)]$ of all continuous functions ordered pointwise? After Scott [Sco] we know the step functions form a basis of isolated elements. A little work characterises the complete primes of $[\mathcal{L}(E_0) \rightarrow \mathcal{L}(E_1)]$ as precisely those step functions of the form $\lambda y.y \ni x \rightarrow [e], \perp$, abbreviated

as $e[x,e]$, where x is an isolated element of $\mathcal{L}(E_0)$ and e an event of E_1 . In fact $[\mathcal{L}(E_0) \rightarrow \mathcal{L}(E_1)]$ is coherent and prime algebraic. Define the event structure $E_0 \rightarrow E_1$ to consist of events (x,e) (standing for $e[x,y]$) ordered by $(x,e) \leq (x',e')$ iff $x' \sqsubseteq_0 x$ & $e \leq_1 e'$ with conflict relation $(x,e) \not\ll (x',e')$ iff $x \uparrow x'$ & $e \not\ll_1 e'$ where we have simply expressed the ordering and incompatibility in the functions space. Then by the representation result of chapter 4 we have $\mathcal{L}_e(E_0 \rightarrow E_1) \cong [\mathcal{L}(E_0) \rightarrow \mathcal{L}(E_1)]$; the isomorphism simply expresses a continuous function f as the configuration $\{(x,e) \mid e \in f(x)\}$. We have represented the function space as an event structure.

Even more simply, we can represent products of coherent prime algebraic domains. Let $(E_i, \leq_i, \not\ll_i)$ for $i = 0,1$ be two event structures as above. Take $E_0 \oplus E_1$ to be their disjoint juxtaposition defined by the disjoint union ($\tilde{\cup}$) of their sets and relations:

$$E_0 \oplus E_1 = (E_0 \tilde{\cup} E_1, \leq_0 \tilde{\cup} \leq_1, \not\ll_0 \tilde{\cup} \not\ll_1).$$

Then $\mathcal{L}_e(E_0 \oplus E_1) \cong \mathcal{L}_e(E_0) \times \mathcal{L}_e(E_1)$; the isomorphism expresses a pair as the configuration which is a disjoint union of the pair's arguments.

Of course we have ignored intuition about what the causality relation \leq on event structures means. In the above constructions it can no longer generally mean "must occur before in time". Accordingly a finiteness restriction on the relation such as an event dominates only finitely many events will not generally hold in representing a function space. (This occurs for the construction $E_0 \rightarrow E_1$ in the innocent circumstances of E_0 including an infinite conflict-free subset and E_1 being non-null.) A chief virtue of event structures is supposed to be their operational nature; they have previously prescribed possible behaviours in time. Can event structures like $E_0 \rightarrow E_1$ representing a function space be made to reflect behaviour in time? What finiteness restrictions can be imposed which reflect this? We expect some extra structure is involved in order to distinguish the behaviour of the functional events (in $E_0 \rightarrow E_1$ say) from say basic input events.

Suppose $(E_i, \leq_i, \not\ll_i)$ for $i = 0,1$ are event structures representing input and output domains. To reflect this, on both we impose the additional axiom

$$|\leq_i^{-1}\{e\}| < \infty \text{ for events } e.$$

The domain of continuous functions between the input and output domains is represented by $E_0 \rightarrow E_1$. It is the ordering given by $(x, e) \leq (x', e')$ iff $x' \sqsubseteq x$ and $e \leq e'$ which forces the finiteness restriction to go. However it naturally factors into two parts $(x, e) \leq^L (x', e) \leq^R (x', e')$ where:
 $(x, e) \leq^L (x', e)$ iff $x' \sqsubseteq_0 x$ & $e = e'$
 $(x, e) \leq^R (x', e')$ iff $x = x'$ & $e \leq e'$.

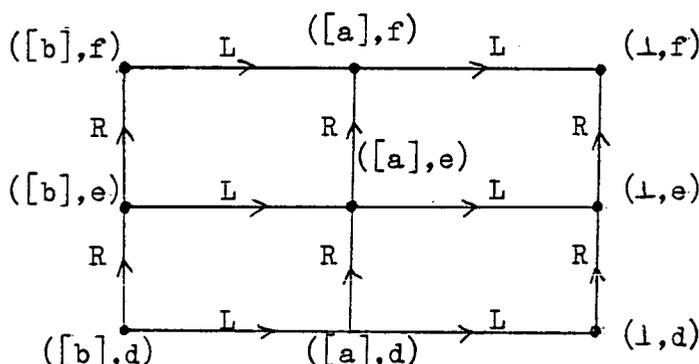
Then we have the two finiteness properties:

$$|(\leq^R)^{-1} \{e\}| < \infty \text{ and } |\leq^L \{e\}| < \infty.$$

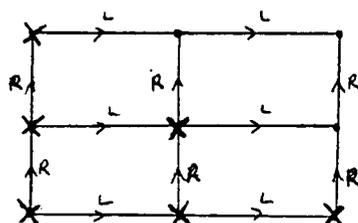
The original order \leq can be recovered as $(\leq^L \cup \leq^R)^*$ with \leq factoring as $\leq^L \circ \leq^R$. (Clearly the factorisation is unique too.) We can draw pictures of event structures using the orders \leq^L and \leq^R .

Example 9.1.1

Let E_0 be the event structure consisting of two events a and b with $a < b$. Let E_1 be the event structure consisting of three events d, e, f with $d < e < f$. The continuous functions, $[L(E_0) \rightarrow L(E_1)]$, can be represented by \leq -left closed subsets of $E_0 \rightarrow E_1$. Draw $E_0 \rightarrow E_1$ with the \leq^L and \leq^R orderings between events:

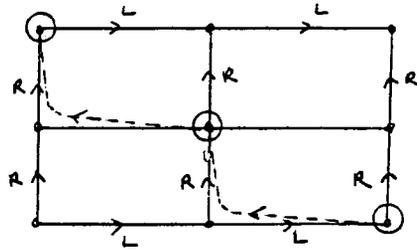


The function ϕ_1 is determined by the following \leq -left closed subset of $E_0 \rightarrow E_1$:



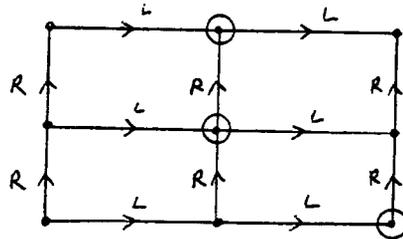
The function ϕ_1 can be viewed as having this behaviour: output event

d regardless of input; thereupon inspect the input for [a]; whereupon output e; thereupon inspect the input for [b]; whereupon output f. This behaviour traces out a "path":



Notice that the behaviour is determined by the \leq^L -maximal events of ϕ_1 , marked by "O"s in the above diagram.

Consider another function ϕ_2 determined by the following \leq^L -maximal events:



The function ϕ_2 is certainly extensionally greater than ϕ_1 . However neither has a behaviour which is part of the other's. They do however share a common subbehaviour, namely: regardless of input, output d. Call the third function this induces ϕ_3 . The extensional ordering between functions ϕ_1, ϕ_2, ϕ_3 corresponds to inclusion of their configurations whereas the ordering on behaviours ("is a sub-behaviour of") corresponds to inclusion of their \leq^L -maximal event-sets.

This is no more than a suggestive example, of course. However note that for a configuration x of $E_0 \rightarrow E_1$, corresponding to a function, we can define $M(x)$ to be its \leq^L -maximal events so that every event of x is \leq^L -below an event of $M(x)$. This is because $\mathcal{L}(E_0)$ satisfies axiom F'. Then M is a 1-1 correspondence from configurations x to their \leq^L -maximal elements $M(x)$. The above example suggests this ordering as one on the behaviours of functions:

$$x \sqsubseteq^R x' \text{ iff } M(x) \subseteq M(x').$$

The stable functions can be characterised easily using \leq^L ; they correspond to configurations x such that

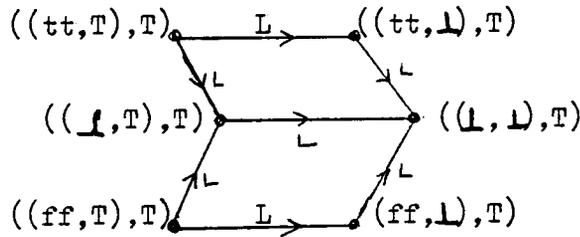
$$\forall e \in x \exists e' \in M(x) e \leq^L e'.$$

Call these stable configurations.

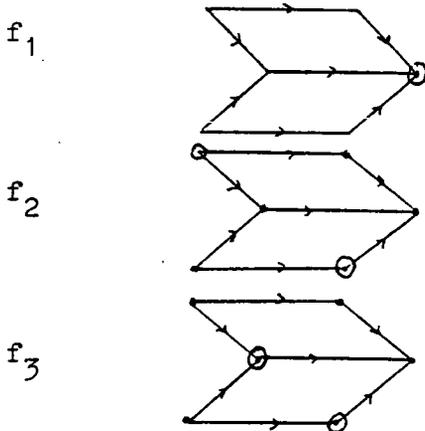
A pay-off: The ordering \sqsubseteq^R is the image of Berry's stable ordering on stable functions. These facts follow from the definition of stable function (8.2.3) and the characterisation of the stable ordering (8.2.8). It also turns out that there is an ordering \sqsubseteq^L on stable configurations so that \sqsubseteq factors uniquely as $\sqsubseteq^L \circ \sqsubseteq^R$. (This fails if we take all configurations however; factorisation exists but is not generally unique.) Both \sqsubseteq^L and \sqsubseteq^R "extend" the corresponding relations \leq^L and \leq^R of the event structure.

Example 9.1.2

The continuous functions from $\mathbb{T} \times \mathcal{O}$ to \mathcal{O} , $[\mathbb{T} \times \mathcal{O} \rightarrow \mathcal{O}]$, are built up from these events.



We use T to denote both the maximum element of \mathcal{O} and the corresponding event. A function in $[\mathbb{T} \times \mathcal{O} \rightarrow \mathcal{O}]$ is represented on this diagram by marking its \leq^L -maximal events its M-image. We define the functions f_1, f_2 and f_3 in this way.



The function f_1 disregards its inputs and outputs T. The function f_2 inspects its first argument giving T if this is ff otherwise it inspects the second argument until T appears whereupon it gives T as output. The function f_3 has an intrinsic parallelism in

that if the first argument turns out to be ff or if the second argument gives out T it yields output T. Functions f_1 and f_2 are stable whereas f_3 is not. Using 8.2.3 the functions f_1 and f_2 are easily checked to be stable. Function f_3 is not because it outputs T for minimal inputs $(1, T)$ and (ff, \perp) which have the (least) upper bound (ff, T) . This means that the event $((ff, T), T)$ is \leq^L -below two elements of $M(f_3)$ the \leq^L -maximal events of f_3 .

We extend these results beyond first order functions. Event structures have the general form $(E, \leq^L, \leq^R, \#)$ where the extensional order \leq is recovered as $(\leq^L \cup \leq^R)^*$. For an event structure representing basic input or output $\leq^L = 1$ and $\leq^R = \leq$. The precise nature of the axioms they satisfy depends on the definition of configuration used.

In this chapter we are chiefly interested in stable configurations - the definition mimics that of the first order. The associated event structures are called stable. They satisfy axioms which are preserved by a stable exponentiation \rightarrow_s . They possess a unique factorisation property: If \leq is defined from \leq^L and \leq^R as $(\leq^L \cup \leq^R)^*$ then \leq factors uniquely as $\leq^L \circ \leq^R$. A stable event structure E has configurations $R(E)$ ordered in three ways, by inclusion \sqsubseteq , by \sqsubseteq^R and by \sqsubseteq^L so that \sqsubseteq factors uniquely as $\sqsubseteq^L \circ \sqsubseteq^R$; in fact the structure $(R(E), \sqsubseteq, \sqsubseteq^R)$ is a bidomain. Given two stable event structures $(E_i, \leq_i^L, \leq_i^R, \#_i)$ for $i = 0, 1$ we define the orderings \leq^L and \leq^R by:

$$\begin{aligned} (x, e) \leq^L (x', e') & \text{ iff } x' \sqsubseteq_o^R x \text{ and } e \leq_i^L e' \\ (x, e) \leq^R (x', e') & \text{ iff } x' \sqsubseteq_o^L x \text{ and } e \leq_i^R e'. \end{aligned}$$

This generalises the first and zeroth orders dealt with, has an elegant symmetry, clearly preserves unique factorisation and the finiteness properties of \leq^L and \leq^R and provides a representation of Berry's exponentiation on bidomains. In other words it works. Surely there must be a more direct justification. (I have in mind some argument based on intuitive interpretations of \leq^L and \leq^R or some formal argument forcing this definition as that which gives cartesian closedness of events structures under some general assumptions sifted from the work of 9.8 demonstrating cartesian-closedness.) The conflict relation on $E_0 \rightarrow_s E_1$ is defined by:

$(x, e) \not\ll (x', e')$ iff $x \uparrow^E x'$ & $e \not\ll_1 e'$.

Configurations will be \leq -left-closed and satisfy two constraints, one ensuring consistency with respect to conflict relation $\not\ll$ and the other stability. In fact $\not\ll$ will only impose a weak constraint in forming configurations, expressing the fact that configurations do not determine many valued functions. If one wished to represent domains of ground-type which were not coherent the conflict relation would have to be abandoned. Instead we could work with an inconsistency predicate (as in 3.3.17) or a consistency relation on events. Virtually all results of this chapter (not necessarily those stating coherence) go through if either of these is used instead. A consistency relation con on events E is a subset of the subsets of E such that:

$\text{con } A \text{ \& } B \subseteq A \Rightarrow \text{con } B$

$\text{con } A \Leftrightarrow \forall B \in \mathcal{Y}(E). B \subseteq A \Rightarrow \text{con } B$
($\mathcal{Y}(E)$ is the set of finite subsets of E .)

If E_0 and E_1 are event structures with consistency relations con_0 , con_1 respectively the consistency relation con of $E_0 \rightarrow_s E_1$ would be given by

$\text{con}\{(x_\alpha, e_\alpha) \mid \alpha \in A\}$ iff $\forall B \subseteq A \{x_\beta \mid \beta \in B\} \uparrow^E \Rightarrow \text{con}_1\{e_\beta \mid \beta \in B\}$.

Because the assumption $\neg \text{con } A \Rightarrow \exists e_1, e_2 \in A \neg \text{con}\{e_1, e_2\}$ (for A finite) is preserved by \rightarrow_s we can get by with a simple conflict relation. (In section 9.10 the sets of \leq^L -maximal events associated with sequential functions of order 1 will be characterised as themselves being configurations with respect to some enabling and consistency relations. A conflict relation alone would not be adequate.)

A word on the examples: We shall draw event structures to illustrate properties or failure of properties. Event structures will represent bidomains and often those examples will correspond to fairly simple bidomains constructed from Π and \mathcal{O} by exponentiation and product. Where this is so we shall indicate the corresponding bidomain and sometimes one which has essentially the same features. The manner of the correspondence is not strictly justified until later so we enclose these indications in brackets.

9.2 Stable event structures

We begin the formal development motivated by the last section. The following axioms arose to support the definitions of stable configuration and exponentiation given there. "Arose" is a euphemism because other axioms true up to first order seemed natural too but were not preserved by exponentiation so had to be dropped.

Definition 9.2.1

A stable event structure consists of a quadruple $(E, \leq^L, \leq^R, \bowtie)$

where

1. E is a countable set of events
2. The relations \leq^L and \leq^R are partial orders on E .
3. Define $\leq = (\leq^L \cup \leq^R)^*$. Then
 $e \leq e' \Rightarrow \exists ! e'' \in E \ e \leq^L e'' \leq^R e'$.
4. Define $\preceq = (\geq^L \cup \leq^R)^*$. Then
 - (i) The set $\{e' \mid e' \preceq e\}$ is finite for all events e .
 - (ii) The relation \preceq is a partial order.
5. If two events e and e' are \leq^L -compatible then they have a \leq^L -supremum in E .
6. The conflict relation \bowtie is a binary irreflexive, symmetric relation on E such that for the \leq defined in 3. we have $e \geq e' \& e' \bowtie e'' \Rightarrow e \bowtie e''$.

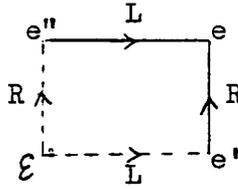
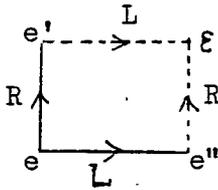
The key axioms are 2., 3. and 4 (i). The relation \leq defined in 3. represents the extensional ordering - we shall show it is a partial order. Axiom 3 expresses that \leq factors uniquely as $\leq^L \circ \leq^R$. Axiom 4(i) certainly implies the finiteness properties of \leq^L and \leq^R we introduced in the last section (viz. $\leq^{R-1} \{e\}$ and $\leq^L \{e\}$ are finite); its extra strength is needed so that \rightarrow_s preserves them. Orderings based on \preceq have operational significance as we shall see and has been suggested in the introductory example 9.1.1. While not strictly necessary 4(ii) facilitates showing this. Axioms 1. and 5. mean we get a bidomain from configurations while axiom 6. means \bowtie expresses an extensional conflict relation; it imposes a weak constraint in forming configurations. Later we shall see some further assumptions which can be imposed on event structures so that \rightarrow_s preserves them. In an informal sense the axioms given are minimal with respect to the proofs. We give an example of one

natural choice of axiom true at order 1 and suggested by example

9.1.1 but unfortunately false. It might seem that

$$(e \leq^R e' \ \& \ e \leq^L e'') \Rightarrow \exists \mathcal{E} \in E \ e'' \leq^R \mathcal{E} \ \& \ e' \leq^L \mathcal{E}$$

$$\text{or that } (e' \leq^R e \ \& \ e'' \leq^L e) \Rightarrow \exists \mathcal{E} \in E \ \mathcal{E} \leq^R e'' \ \& \ \mathcal{E} \leq^L e'.$$



However neither is preserved by \rightarrow_s (see ex. 9.7.8).

Throughout this section we shall work with a fixed stable event structure E referring to orderings as they are defined in 9.2.1.

The unique factorisation property expressed by axiom 3. is very powerful. It enables a style of "picture proof" using arrows " \xrightarrow{L} " and " \xrightarrow{R} " for \leq^L and \leq^R . This is illustrated in the following lemma.

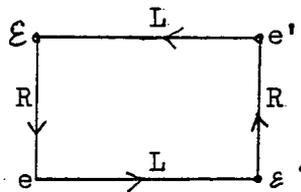
Lemma 9.2.2

The relation \leq defined in 9.2.1 is a partial order.

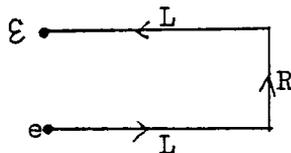
Proof

The relation \leq is certainly reflexive and transitive. To prove antisymmetry we use a picture proof.

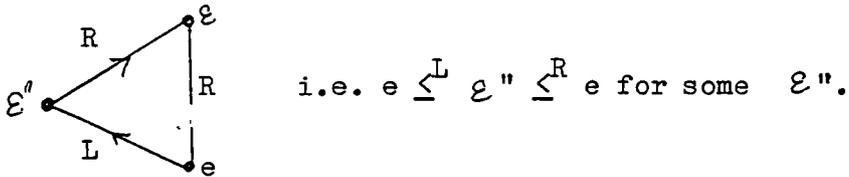
Suppose $e \leq e'$ and $e' \leq e$. Then pictorially by factorising \leq for some events \mathcal{E} and \mathcal{E}' we have:



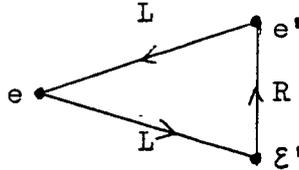
From



we know $e \leq \mathcal{E}$. Thus by factorising $e \leq \mathcal{E}$ we get:



But $e \leq^L e'$ and $e' \leq^R e$ so the uniqueness of factorisation gives $e = e''$.
Then as \leq^R is a po $e = e'$. Therefore the first picture collapses to



The uniqueness of the factorisation of $e' \leq e'$ gives $e = e'$ as required. ■

The following notation is useful.

Notation 9.2.3

For events e and e' write

$$e \uparrow^L e' \text{ iff } \exists e'' \in E \ e \leq^L e'' \ \& \ e' \leq^L e''$$

$$e \downarrow^L e' \text{ iff } \exists e'' \in E \ e'' \leq^L e \ \& \ e'' \leq^L e'$$

and when the \leq^L -join and \leq^L -meet exist write them as $e \vee^L e'$ and $e \wedge^L e'$ respectively. Define $\uparrow^R, \downarrow^R, \vee^R, \wedge^R$ similarly. For the ordering \leq we use $\uparrow, \downarrow, \vee, \wedge$. Thus for example axiom 5. may be expressed as:

If $e \uparrow^L e'$ then $e \vee^L e'$ exists in E .

We also write $\overset{L}{\leftarrow}$, $\overset{R}{\leftarrow}$ and \leftarrow for the covering relations of \leq^L, \leq^R and \leq respectively.

Thus $e \overset{R}{\leftarrow} e'$ means $e \not\leq^R e'$ and $\forall e'' \in E \ e \leq^R e'' \leq^R e' \Rightarrow e = e''$ or $e' = e''$.

9.3 Stable configurations

Suppose E is a stable event structure. In this section we define its stable configurations, characterise them in terms of their \leq^L -maximal events (given by M) and examine the extensional order (\sqsubseteq) given by inclusion.

Definition 9.3.1

Let x be a subset of E . Say x is \times -consistent iff
 $\forall e, e' \in x \neg (e \times e')$.

Now we define the (stable) configurations of E .

Definition 9.3.2

Define the stable configurations of E to be subsets x of E such that

- (i) x is \leq -left closed and \times -consistent.
- (ii) $e, e' \in x \ \& \ e \downarrow^L e' \Rightarrow \exists e'' \in x \ e, e' \leq^L e''$.

Define $(R(E), \subseteq)$ to be the stable configurations $R(E)$ ordered by inclusion. (Thus $\subseteq = \subseteq \uparrow R(E)$.) We write \sqcup, \sqcap and \bigsqcup, \bigsqcap for suprema and infima of $(R(E), \subseteq)$ where they exist.

The definition imitates the first order one in § 9.1. The condition (ii) restricts configurations to be stable. The ordering of inclusion on stable configurations corresponds to the extensional ordering on functions.

As in section 9.1 the stable ordering will correspond to inclusion of the \leq^L -maximal events of stable configurations. Such sets of \leq^L -maximal events of configurations also provide another way of looking at stable configurations and in particular a characterisation of them (9.3.8).

Definition 9.3.3

For x in $R(E)$ define $M(x)$ to be the \leq^L -maximal events in x .

We can establish the existence of sufficiently many \leq^L -maximal events of stable configurations for the map M to be a 1-1 correspondence.

Lemma 9.3.4

$$\forall x \in R(E) \ \forall e \in x \ \exists ! e' \in M(x) \ e \leq^L e'.$$

Proof

Suppose $e \in x \in R(E)$. From 4(i) of definition 9.2.1 we have $\{e' \mid e <^L e'\}$ finite. Thus $\exists e' \in M(x) \ e \leq^L e'$. To establish uniqueness suppose $e \leq^L e'$ and $e \leq^L e''$ for $e', e'' \in M(x)$. Then $e' \downarrow^L e''$ so using condition (ii) of 9.3.2 defining stable

configurations we have $e' = e''$. ■

Definition 9.3.5

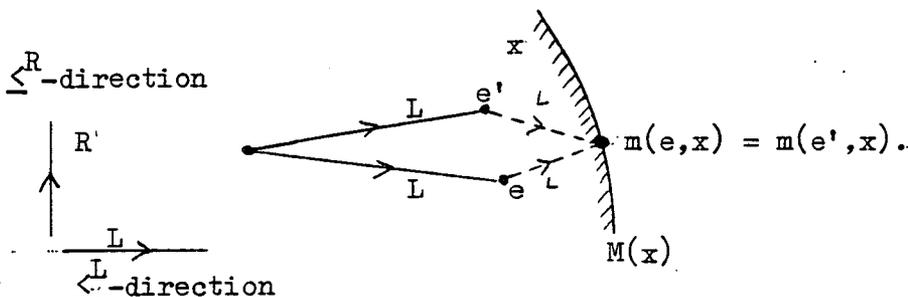
For x in $R(E)$ and e an event in x define $m(e, x)$ to be the unique event e' provided by lemma 9.3.4.

We can now use the following obvious fact in our picture-proofs.

Lemma 9.3.6

Suppose $x \in R(E)$. Then
 $e \in x \ \& \ e' \in x \ \& \ e \downarrow^L e' \Rightarrow m(e, x) = m(e', x)$.

In the main we shall draw \leq^L (or \rightarrow^L) across the page and \leq^R (or \rightarrow^R) up the page. Then lemma 9.3.6 can be pictured as



It is now obvious that M is 1-1.

Lemma 9.3.7

The map M defined in 9.3.3 is 1-1.

Proof

Suppose $x, x' \in R(E)$ and that $M(x) = M(x')$. Take e in x . Then $m(e, x) \in M(x')$. As $M(x') \subseteq x'$ and x' is \leq^L -left closed we have $e \in x'$. Thus $x \subseteq x'$ and similarly $x' \subseteq x$ so $x = x'$. ■

We can characterise sets of the form $M(x)$ for x in $R(E)$.

Theorem 9.3.8 (Characterisation of the range of M)

$$\exists x \in R(E) \ y = M(x)$$

iff

- (i) y is \times -consistent
- (ii) $\forall e, e' \in y \ e \downarrow^L e' \Rightarrow e = e'$
- (iii) $\forall e \in y \ \forall e' \leq^R e \ \exists e'' \in y \ e' \leq^L e''$.

Proof

" \Rightarrow " Suppose $y = M(x)$ for some x in $R(E)$. Then (i) is obvious

and (ii) is clear by 9.3.4. To show (iii) suppose $e \in y$ and $e' \leq^R e$. Then $e' \in x$ so $e' \leq^L m(e', x) \in y$.

" \Leftarrow " Suppose $y \subseteq E$ and y satisfies (i), (ii) and (iii). Define $x = \{e \in E \mid \exists e' \in y e \leq^L e'\}$. We show $x \in R(E)$ and $y = M(x)$.

First note $x = \{e \in E \mid \exists e' \in y e \leq e'\}$. For suppose $e \leq e' \in y$. Then $e \leq^L e'' \leq^R e' \in y$ so by (iii) above $\exists \varepsilon \in y e'' \leq^L \varepsilon$ giving $e \leq^L \varepsilon$.

Thus x is \leq -left closed. Also x is consistent as y is. Suppose $e, e' \in x$ and $e \downarrow^L e'$. Then $e \leq^L \varepsilon$ and $e' \leq^L \varepsilon'$ for some $\varepsilon, \varepsilon' \in y$. But $\varepsilon \downarrow^L \varepsilon'$ so by (ii) above $\varepsilon = \varepsilon'$. Thus $e, e' \leq^L \varepsilon \in x$. Therefore $x \in R(E)$. Obviously $M(x) \subseteq y$ and from (ii) the converse inclusion is clear giving $y = M(x)$. ■

This theorem is very important technically. It also is very suggestive. Conditions (i) and (ii) can be regarded as together being a consistency requirement while (iii) indicates a kind of securing. We explore this later in section 9.4.

We now examine the structure $(R(E), \sqsubseteq)$ - the domain ordered extensionally.

First some notation.

Definition 9.3.9

For A a subset of E we define $[A]$ to be the \leq -left closure of A i.e.

$$[A] = \{e \in E \mid \exists a \in A e \leq a\}.$$

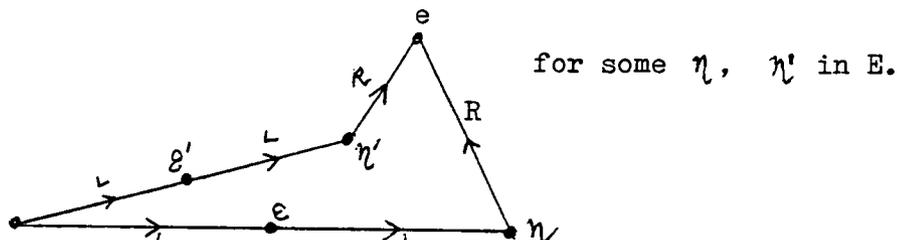
We shall write $[e]$ for $[\{e\}]$.

Theorem 9.3.10 (Properties of $(R(E), \sqsubseteq)$)

- (i) $\forall e \in E [e] \in R(E)$ and $\forall e, e' \in E (e \leq e' \Leftrightarrow [e] \sqsubseteq [e'])$.
- (ii) $(R(E), \sqsubseteq)$ is an ω -algebraic, consistent complete cpo with
- $\perp = \emptyset$
 - The supremum of a directed set S is $\bigcup S$.
 - For X a non-null subset of $R(E)$ we have $\prod X = \bigcap X$.
 - For x in $R(E)$ the element x is isolated in $(R(E), \sqsubseteq)$ iff $M(x)$ is finite.

Proof

(i) Suppose $e \in E$. Then $[e]$ is certainly \leq -left closed and is easily seen to be consistent. Assume $\varepsilon, \varepsilon' \in [e]$ and $\varepsilon \downarrow^L \varepsilon'$. Thus in a picture factoring $\varepsilon \leq e$ and $\varepsilon' \leq e'$ we get:



Unique factorisation gives $\eta = \eta'$ so $\varepsilon, \varepsilon' \leq^L \eta \in [e]$. Thus $[e] \in R(E)$. For e, e' in E it is clear $e \leq e' \Leftrightarrow [e] \subseteq [e']$.

(ii) (a) The null set is clearly in $R(E)$ and it is the \subseteq -minimum element.

(b) Let S be a directed subset of $R(E)$. Clearly if the supremum of S , say $\sqcup S$, exists then $\cup S \subseteq \sqcup S$. Thus it suffices to show $\cup S \in R(E)$. This is trivial.

(c) Suppose $\emptyset \neq X \subseteq R(E)$. Clearly if $\bigcap X \in R(E)$ then $\bigcap X = \prod X$. However $\bigcap X$ is certainly \leq -left closed and consistent and also if $e, e' \in \bigcap X$ with $e \downarrow^L e'$ then for any x in X there exists $e \vee^L e'$ which is in x giving $e \vee^L e'$ in $\bigcap X$.

From (c) it follows that $(R(E), \subseteq)$ is consistent-complete. Suppose for X a subset of $R(E)$ and y in $R(E)$ we have $X \subseteq y$. Then $\bigcap \{y' \mid X \subseteq y'\}$ is in $R(E)$ and equals $\sqcup X$.

(d) Suppose $x \in R(E)$ and $|M(x)| < \infty$. Then as $x = \{e' \in E \mid \exists e \in M(x) e' \leq e\}$ we get x is isolated. Conversely suppose x is an isolated element of $(R(E), \subseteq)$. Assume $A \subseteq M(x)$. Then it may be checked that $(\leq^{-1} A) \cap M(x)$ satisfies properties (i), (ii), (iii) of theorem 9.3.8. Thus $[\leq^{-1} A \cap M(x)] \in R(E)$.

Consider

$$S = \{[\leq^{-1} A \cap M(x)] \mid A \text{ is a finite subset of } M(x)\}.$$

The set S is directed and $x = \cup S$. Thus

$$x = [\leq^{-1} A_1 \cap M(x)] \cup \dots \cup [\leq^{-1} A_n \cap M(x)]$$

for some finite subsets A_1, \dots, A_n of $M(x)$. Therefore

$$M(x) \subseteq \bigcup_{i \leq n} \leq^{-1} A_i$$

As each A_i is finite each $\leq^{-1} A_i$ is finite. Thus $M(x)$ is finite as required.

To show $(R(E), \sqsubseteq)$ is algebraic suppose $x \in R(E)$. Then

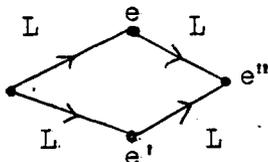
$$x = \bigcup \{ [\leq^{-1} A \cap M(x)] \mid A \text{ is a finite subset of } M(x) \}$$

as above where each element $[\leq^{-1} A \cap M(x)]$ is isolated by (d) above. Finally it is ω -algebraic by (d) as E is countable. ■

In general the cpo $(R(E), \sqsubseteq)$ is not ^{prime algebraic} nor are elements of the form $[e]$ prime, and $\cup \neq \sqcup$ generally. The following simple example suffices.

Example 9.3.11

Suppose E has this form:



Then $[e] \cup [e'] \notin R(E)$ so $[e] \sqcup [e'] = [e'']$. As $\cup \neq \sqcup$ the cpo $(R(E), \sqsubseteq)$ is not prime algebraic. (E is the event structure of $[\mathbb{D}^2 \rightarrow_s \mathbb{D}]$.)

9.4 Images of M are configurations, some "staircase" orderings

Throughout this section we work with a fixed stable event structure E .

Recall theorem 9.3.8. It characterised configurations x in terms of the set $M(x)$ of its \leq^L -maximal elements. It said y was of the form $M(x)$ for some x in $R(E)$ iff

- (i) y is \times -consistent
- (ii) $\forall e, e' \in y \ e \downarrow^L e' \Rightarrow e = e'$
- (iii) $\forall e \in y \ \forall e' \leq^R e \ \exists e'' \in y \ e' \leq^L e''$.

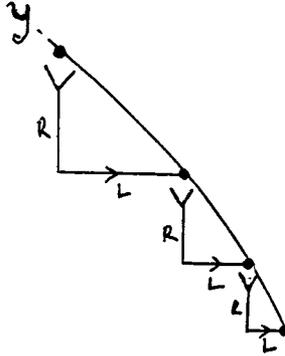
These conditions make y itself look like a configuration.

Conditions (i) and (ii) express the consistency of y .

Condition (iii) suggests events in y are secured with respect to an enabling relation \vdash so that for e in y

$$\{e'' \in y \mid \exists e' \leftarrow^R e \ \& \ e' \leq^L e''\} \vdash e.$$

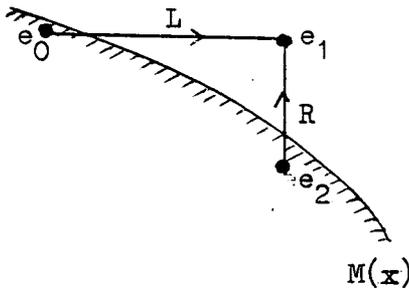
Because $\preceq^{-1}\{e\}$ is always finite we know events really are secured. We can picture the securing of an event in y as:



We have only drawn one "thread" through the securing.

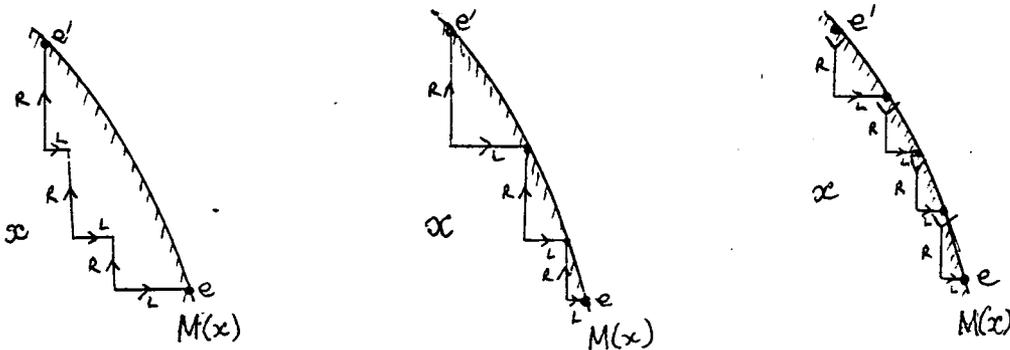
Such "threads" look like staircases. In a sense they represent "relativisations" of \preceq to sets of the form $M(x)$ for x in $R(E)$. They are not restrictions of \preceq as the following example shows. (A more real-life example is the event structure of $[[\mathbb{T} \times \mathbb{O} \rightarrow_s \mathbb{O}] \rightarrow_s \mathbb{O}]$.)

Example 9.4.1



Suppose E consisted of three events as shown. Set $x = \{e_0, e_2\}$. Clearly $e_2 \preceq e_0$ yet $\preceq^{-1}\{e_0\} \cap M(x) = \emptyset$.

There are however three candidates for the relation \preceq relativised to $M(x)$; we might say two events e and e' were in this relation if any of these situations held:



Fortunately they all determine the same relation which we call \preceq_x^M . In proving this we use the following relations.

Definition 9.4.2 ("Staircase" orderings)

Suppose $x \in R(E)$. Define the following relations on E .

$$\begin{aligned} \succ &= \succeq^L \circ \prec^R \quad (e \succ e' : \begin{array}{c} R \swarrow e' \\ \xrightarrow{L} e \end{array}) \\ \succ_x &= \succ \uparrow_{x^2} \quad (e \succ_x e' : \begin{array}{c} R \swarrow e' \\ \xrightarrow{L} e \\ \searrow M(x) \end{array}) \\ \succ_x^M &= \succ \uparrow_{M(x)^2} \quad (e \succ_x^M e' : \begin{array}{c} R \swarrow e' \\ \xrightarrow{L} e \\ \searrow M(x) \end{array}) \\ \preceq_x^1 &= \succeq^L \uparrow_{x^2} \circ \preceq^R \downarrow_{x^2} \quad (e \preceq_x^1 e' : \begin{array}{c} e' \\ \uparrow R \\ \xrightarrow{L} e \end{array}) \\ \preceq_x^M &= \succeq^L \uparrow_{M(x)} \circ \preceq^R \downarrow_{M(x)} \quad (e \preceq_x^M e' : \begin{array}{c} e' \\ \uparrow R \\ \xrightarrow{L} e \end{array}) \end{aligned}$$

Define $\preceq_x^1 = \preceq_x^{1*}$ and $\preceq_x^M = \preceq_x^{M1*}$.

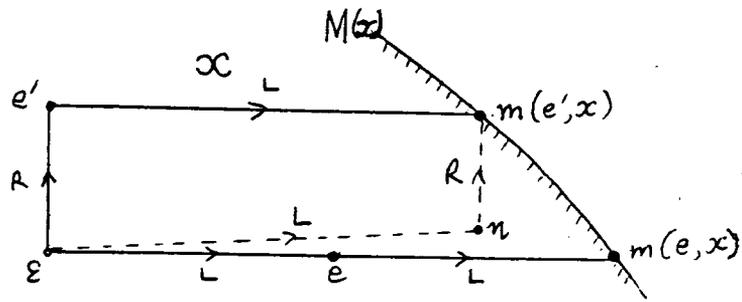
The following lemma shows that a \preceq_x chain determines a unique \preceq_x^M chain as its image in $M(x)$. (This will be important later for the \sqsubseteq^L ordering on configurations.) This is then used to show that the three relativised versions of \preceq above are the same.

Lemma 9.4.3

- (i) For e, e' in x where $x \in R(E)$
 - (a) $e \preceq_x^1 e' \Rightarrow m(e, x) \preceq_x^{M1} m(e', x)$
 - (b) $e \preceq_x^M e' \Rightarrow m(e, x) \preceq_x^M m(e', x)$
- (ii) $\preceq_x \uparrow_{M(x)^2} = \preceq_x^M = \succ_x^{M*}$

Proof

- (i) Suppose $e, e' \in x$ where $x \in R(E)$
 - (a) Assume further that $e \preceq_x^1 e'$ so $e \succeq^L \varepsilon \preceq^R e'$ for some ε .
- We have this picture:



where the dotted line represents the factorisation of $\varepsilon \preceq m(e', x)$.

We have $\eta \downarrow^L e$ so by lemma 9.3.6 we have $m(\eta, x) = m(e, x)$ so $\eta \leq^L m(e, x)$. Thus $m(e, x) \leq^M_x m(e', x)$.

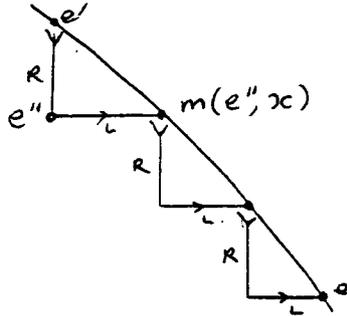
(b) This follows by induction on the number \leq^M_x links in the chain e to e' using (a).

(ii) Part (i) (b) gives $\leq^M_x = \leq^M_x \uparrow M(x)^2$. We now show $\xrightarrow{M}_x^* = \leq^M_x$. Clearly $\xrightarrow{M}_x^* \subseteq \leq^M_x$. We prove conversely

that $(e \leq^M_x e' \Rightarrow e \xrightarrow{M}_x^* e')$

by induction on the well-foundedness of \leq^M_x . For minimal e' it is clear. Otherwise suppose $e \leq^M_x e' \& e \neq e'$. Then by the definition of \leq^M_x we have, for some e'' , that $e \leq^M_x e'' \xrightarrow{R} e'$. Then $m(e'', x) \xrightarrow{M}_x e'$ and, by (i) (b), also $e \leq^M_x m(e'', x)$.

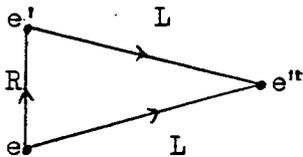
In a picture:



By induction $e \xrightarrow{M}_x^* m(e'', x)$. Thus $e \xrightarrow{M}_x^* e'$ as required. ■

It is quite possible to have $e \xrightarrow{M}_x e'$ and $m(e, x) = m(e', x)$ as the following example shows.

Example 9.4.4



Take $x = [e'']$ in the event structure drawn. Then $e \xrightarrow{M}_x e'$ and $m(e, x) = m(e', x) = e''$.

(This situation occurs in the event structure of $[[\mathbb{O} \rightarrow_s \mathbb{O}] \rightarrow_s \mathbb{O}]$.)

Using the new relation \leq^M_x we can give a characterisation of elements of the form $M(x)$, for x in $R(E)$, as a kind of configuration. We define the appropriate enabling and conflict relations below (Cf. definition 3.3.1).

Definition 9.4.5

Define the stable-conflict relation $\#_s$ by:

$$\mathbb{X}_s = \mathbb{X} \cup (\downarrow^L \setminus 1).$$

Say a subset x of E is \mathbb{X}_s -consistent iff

$$\forall e, e' \in x \neg (e \mathbb{X}_s e').$$

Define the stable-enabling relation $\vdash_s \subseteq \mathcal{F}(E) \times E$

by: $A \vdash_s e$ iff (i) $\forall a \in A a \rightarrow e$
 (ii) A is \mathbb{X}_s -consistent
 (iii) $\forall e' \prec^R e \exists a \in A e' \leq^L a$.

Suppose $e \in E$ and $x \subseteq E$. Say e is \vdash_s -secured in x iff
 $\exists e_0, \dots, e_n \in x e_n = e \ \& \ \forall i \leq n \exists A \subseteq \{e_0, \dots, e_{i-1}\} A \vdash_s e_i$.
 Say x is \vdash_s -secured iff all events in x are \vdash_s -secured in x .

Say a subset x of E is an s -configuration iff

- (i) x is \mathbb{X}_s -consistent
- (ii) x is \vdash_s -secured.

Theorem 9.4.6

$\exists x \in R(E) y = M(x)$ iff y is an s -configuration.

Proof

" \Rightarrow " Suppose $y = M(x)$ for x in $R(E)$. By theorem 9.3.8 y is \mathbb{X}_s -consistent. Suppose $e \in y$. That e is \vdash_s -secured in y follows by induction on $|\prec^{-1}\{e\}|$: first note $\xrightarrow{M}_x^{-1}\{e\} \vdash_s e$; then by induction each element of $\xrightarrow{M}_x^{-1}\{e\}$ is secured so e is secured.

" \Leftarrow " Suppose y is an s -configuration. Then y satisfies (i) and (ii) of theorem 9.3.8 as y is \mathbb{X}_s -consistent. To show (iii) we prove by induction on the well-foundedness of \prec that

$$\forall e' (e' \leq^R e \in y \Rightarrow \exists e'' \in y e' \leq^L e'')$$

Suppose $e' \leq^R e \in y$ and further that $e' \leq^R e'' \prec^R e$ for some e'' (if no such e'' exists the induction hypothesis is obvious). As e is secured in y we have some \mathcal{E} in y such that $e'' \leq^L \mathcal{E}$. In a picture:

Corollary 9.4.9

The set $MR(E)$ ordered by inclusion is a prime-algebraic coherent cpo satisfying axioms F, C, R and V. The complete primes are minimal securings of events.

Proof

By the definition of \vdash_s we have $A \vdash_s e \ \& \ A' \vdash_s e$
 $A \neq A'$ implies a $\times_s a'$ for some a in A and a' in A' . Then use the above result. The complete primes coincide with the complete irreducibles which are minimal securings of events. ■

Note the axioms C and R follow from prime-algebraicity anyhow while axiom V is then a consequence of coherence.

In the next section we look at the structure $(MR(E), \subseteq)$ in more detail. Intuitively it is the set of behaviours ordered by a sub-behaviour relation which will turn out to be Berry's stable ordering; we expect axiom F in such a situation.

9.5 The structure $(R(E), \subseteq^R)$

Again we work with a fixed stable event structure E . We study the inclusion ordering on sets of the form $MR(E)$. As M is 1-1 it is a partial order on $R(E)$ which we call \subseteq^R . (As remarked it is Berry's stable ordering in fact - see section 9.7.)

Definition 9.5.1

For x, y in $R(E)$ define

$x \subseteq^R y$ iff $M(x) \subseteq M(y)$.

We note some simple facts about \subseteq^R ; it is a partial order "extending" \leq^R .

Lemma 9.5.2

- (i) The relation \subseteq^R is a partial order on $R(E)$.
- (ii) $e \in M[e']$ iff $e \leq^R e'$.
- (iii) $e \leq^R e'$ iff $[e] \subseteq^R [e']$.

Proof

(i) Clear as M is 1-1.

(ii) " \Rightarrow " Suppose $e \in M([e'])$. Then $e \leq e'$ so by factorisation for some e'' we have $e \leq^L e'' \leq^R e'$. But e is

\leq^L -maximal in $[e]$ so $e = e''$ giving $e \leq^R e'$.

" \leq " Suppose $e \leq^R e'$. By unique factorisation $e \in M([e'])$.

(iii) This follows from (ii) as $[e] \subseteq^R [e']$ iff $M([e]) \subseteq M([e'])$. ■

From corollary 9.4.9 we know $(R(E), \subseteq^R)$ is a coherent prime algebraic cpo. We now list some properties of the suprema and infima of \subseteq^R . Note that for \subseteq^R -compatible subsets suprema and infima coincide with those for \subseteq .

Lemma 9.5.3 (the sup. and inf. properties of \subseteq^R)

The structure $(R(E), \subseteq^R)$ is a coherent prime algebraic cpo, with $\perp = \emptyset$, such that

(i) If X is a \subseteq^R -compatible subset of $R(E)$ then

$$M(\bigsqcup^R X) = \bigcup MX \quad \& \quad \bigsqcup^R X = \bigcup X = \bigsqcup X.$$

$$\text{and } M(\bigsqcap^R X) = \bigcap MX \quad \& \quad \bigsqcap^R X = \bigcap X = \bigsqcap X.$$

(ii) If S is a \subseteq^R -directed subset of $R(E)$ then $\bigsqcup^R S = \bigcup S$.

Proof

The additional properties (i) and (ii) follow using theorem 9.3.8. ■

Note that in general $\bigsqcap^R X$ does not equal $\bigcap X$ as shown in the following example.

Example 9.5.4

$$e \xrightarrow{L} e'$$

For this event structure (associated with

$$[\emptyset \xrightarrow{s} \emptyset])$$

$$[e] \cap [e'] = [e] \neq \emptyset = [e] \bigsqcap^R [e']$$

There follows an easy characterisation of \subseteq^R -compatibility.

Lemma 9.5.5

For X a subset of $R(E)$, X is \subseteq^R -compatible iff

$$\bigcup X \text{ is } \times\text{-consistent \& } \forall x_1, x_2 \in X \forall e \in x_1 \cap x_2 \\ m(e, x_1) = m(e, x_2).$$

Proof

Use theorem 9.3.8. ■

Corollary 9.5.6

For x, x' in $R(E)$ and e, e' in E , $x \uparrow^R x'$ & $e \in x$ & $e' \in x'$
 & $e \downarrow^L e' \Rightarrow m(e, x) = m(e', x')$

Proof

Use 9.5.5 with 9.3.6. ■

We already know $(R(E), \sqsubseteq^R)$ is prime-algebraic with the complete primes corresponding to minimal securings of events. The next lemma provides an alternative characterisation of the complete primes.

Lemma 9.5.7

Suppose x, y are in $R(E)$. Then

$$x \uparrow^R y \Rightarrow \forall e \in M(x) \cap M(y) \preceq_x^{M-1} \{e\} = \preceq_y^{M-1} \{e\}.$$

Proof

Suppose $x \uparrow^R y$ for x, y in $R(E)$ and that $e \in M(x) \cap M(y)$. It is shown by induction on the well-foundedness of \preceq_x^M that $\preceq_x^{M-1} \{e\} = \preceq_y^{M-1} \{e\}$ using lemma 9.5.5. ■

Another characterisation of the \sqsubseteq^R -complete primes:

Lemma 9.5.8

Let x be in $R(E)$. Then x is a complete prime of $(R(E), \sqsubseteq^R)$ iff $\exists e \in M(x) \forall e' \in M(x) e' \preceq_x^M e$.

Proof

Suppose $x \in R(E)$.

" \Rightarrow " Assume x is a complete prime of $(R(E), \sqsubseteq^R)$. Then $M(x) = \bigcup_{e \in M(x)} \preceq_x^{M-1} \{e\}$ where each set $\preceq_x^{M-1} \{e\}$ satisfies the conditions of theorem 9.3.8. Thus as x is a complete prime $x = \preceq_x^{M-1} \{e\}$ for some e in $M(x)$.

" \Leftarrow " Assume $M(x) = \preceq_x^{M-1} \{e\}$ for some e in $M(x)$. Suppose $x \sqsubseteq^R \bigsqcup^R A$ for some \sqsubseteq^R -compatible subset of $R(E)$. Then $M(x) \subseteq \bigcup_{a \in A} M(a)$. Thus for some a in A we have $e \in M(a)$. By lemma 9.5.7 as $x \uparrow^R a$ we know $\preceq_a^{M-1} \{e\} = \preceq_x^{M-1} \{e\} = M(x)$ so $M(x) \subseteq M(a)$ i.e. $x \sqsubseteq^R a$. Thus x is a complete prime. ■

The above result justifies this definition.

Definition 9.5.9

Denote the set of complete primes of $(R(E), \sqsubseteq^R)$ by $\text{Pr}(R(E))$.

Define $\text{ev}: \text{Pr}(R(E)) \rightarrow E$ by setting $\text{ev}(p)$ equal to the unique event e s.t. $p = \ll_p^{M^{-1}} \{e\}$.

We sum-up some properties of $(R(E), \sqsubseteq^R)$. Note that the \sqsubseteq^R -isolated elements of $(R(E), \sqsubseteq^R)$ are precisely those configurations x such that $M(x)$ is finite; thus they coincide with the isolated elements of $(R(E), \sqsubseteq)$. Any configuration x decomposes into complete primes. From the characterisation above this can be expressed simply in terms of \ll_x^M .

Theorem 9.5.10

The structure $(R(E), \sqsubseteq^R)$ is a coherent ω -prime algebraic cpo satisfying axiom F with $\perp = \emptyset$; the complete primes below x are those elements of the form $[\ll_x^{M^{-1}} \{e\}]$ for e in x . The \sqsubseteq^R -isolated elements are characterised as those configurations x with $M(x)$ finite.

Proof

Clear from 9.4.9 and the results of this section. ■

As the isolated elements of $R(E)$ with respect to the two orders \sqsubseteq and \sqsubseteq^R are the same the following terminology is not ambiguous.

Definition 9.5.11

Define $R(E)^0 = \{x \in R(E) \mid |M(x)| < \infty\}$. Say the elements of $R(E)^0$ are isolated.

9.6 The structure $(R(E), \sqsubseteq^L)$

With an eye to defining stable exponentiation on event structures we introduce a further ordering \sqsubseteq^L on stable configurations. For an event structure E , \leq is defined as $(\leq^L \cup \leq^R)^*$ and it is assumed \leq factors uniquely as $\leq^L \circ \leq^R$. The new ordering \sqsubseteq^L is defined so that \sqsubseteq factors uniquely as $\sqsubseteq^L \circ \sqsubseteq^R$.

Definition 9.6.1

Define the relation \sqsubseteq^L on $R(E)$ by: For x, y in $R(E)$,

$x \sqsubseteq^L y$ iff $x \sqsubseteq y$ & $\forall z (x \sqsubseteq z \sqsubseteq^R y \Rightarrow z = y)$.

If there exists \sqsubseteq^L so that \sqsubseteq factors uniquely as $\sqsubseteq^L \circ \sqsubseteq^R$ the above definition gives it. For arbitrary partial orders instead of \sqsubseteq and \sqsubseteq^R the definition does not necessarily yield a partial order. That \sqsubseteq^L is a partial order will follow soon from the characterisation of \sqsubseteq^L . Unique factorisation follows directly from the fact that \sqsubseteq^R -compatible elements of $R(E)$ have a \sqsubseteq^R -meet *equal to the \sqsubseteq -meet*.

Lemma 9.6.2

For x, y in $R(E)$.

$x \sqsubseteq y \Rightarrow \exists ! z \in R(E) \ x \sqsubseteq^L z \sqsubseteq^R y$.

Proof

Suppose $x, y \in R(E)$ and $x \sqsubseteq y$. Take $z =_{\text{def}} \bigcap^R \{z' \mid x \sqsubseteq z' \sqsubseteq^R y\} = \bigcap \{z' \mid x \sqsubseteq z' \sqsubseteq^R y\} \sqsupseteq x$ by lemma 9.5.3. From the definition of \sqsubseteq^L we get $x \sqsubseteq^L z \sqsubseteq^R y$. The definition of \sqsubseteq guarantees uniqueness. ■

The characterisation of \sqsubseteq^L is suggested by the following simple observation.

Lemma 9.6.3

For x, y in $R(E)$,

$x \sqsubseteq y$ iff $\forall e \in M(x) \ \exists ! e' \in M(y) \ e \sqsubseteq^L e'$.

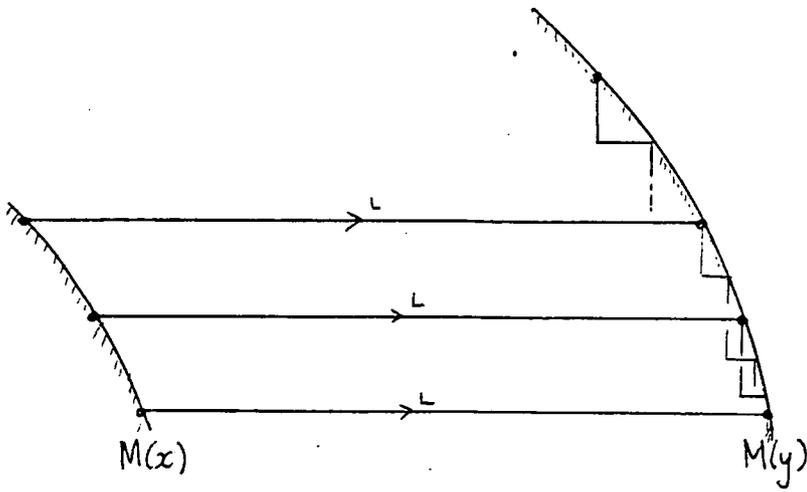
Proof

Suppose $x, y \in R(E)$.

" \Rightarrow " is obvious by lemma 9.3.4.

" \Leftarrow " Suppose $e \in M(x)$. Then $m(e, x) \sqsubseteq^L e'$ for some e' in $M(y)$ giving $e \sqsubseteq^L e'$ so $e \in y$. ■

Note that the event e' is unique in the statement of 9.6.3. We can represent $x \sqsubseteq y$ pictorially



The next theorem characterises $M(z)$ for the unique z such that $x \subseteq^L z \subseteq^R y$ as being the smallest \leq^M_y -left closed subset of $M(y)$ containing the \leq^L -images of $M(x)$.

Theorem 9.6.4 (characterisation of \subseteq^L)

Let x, y be in $R(E)$. Then
 $x \subseteq^L y$ iff (i) $\forall e \in M(x) \exists e' \in M(y) e \leq^L e'$
 (ii) $\forall e' \in M(y) \exists e \in M(x) e' \leq^M_y m(e, y)$.

Proof

Suppose $x, y \in R(E)$.

" \Leftarrow " Assume (i) and (ii) hold. By lemma 9.6.3 we have $x \subseteq y$. Suppose $x \subseteq z \subseteq^R y$. Suppose $e' \in M(y)$. Then $e' \leq^M_y m(e, y)$ for some e in $M(x)$. As $x \subseteq z$ and $z \subseteq^R y$ we have $m(e, y) = m(e, z)$ (by lemma 9.5.5). Then by lemma 9.4.6 we have $e' \in M(z)$. Thus $z = y$ as required.

" \Rightarrow " Suppose $x \subseteq^L y$. Certainly (i) holds by lemma 9.5.3. Suppose (ii) failed i.e. for some e' in $M(y)$ we had

$$\forall e \in M(x) \neg (e' \leq^M_y m(e, y)).$$

Then the set $M(y) \setminus \leq^M_y \{e'\}$ satisfies all conditions of theorem 9.3.8. Thus it defines an element y' of $R(E)$. Clearly $x \subseteq y' \subseteq^R y$ so $x \not\subseteq^L y$, a contradiction. Thus (ii) holds as required. ■

Corollary 9.6.5

The relation \subseteq^L on $R(E)$ is a partial order.

Proof

Reflexivity and antisymmetry now follow easily. To prove transitivity use part (i) (b) of lemma 9.4.3. ■

The ordering \sqsubseteq^L extends \leq^L in this sense:

Corollary 9.6.6

For e, e' in E

$$e \leq^L e' \text{ iff } [e] \sqsubseteq^L [e'].$$

Proof

" \Rightarrow " follows easily using theorem 9.6.3.

" \Leftarrow " Clearly $m(e, [e']) \leq^R e'$. Also by theorem 9.6.3 for some $\mathcal{E} \leq^R e$ we have $m(\mathcal{E}, [e']) = e'$. Lemma 9.4.3 gives $m(\mathcal{E}, [e']) = m(e, [e'])$ so $e \leq^L e'$ as required. ■

From the characterisation of \sqsubseteq^L it follows that any isolated elements of $R(E)$ is \sqsubseteq^L -dominated by only finitely many elements of $R(E)$ which are necessarily isolated.

Corollary 9.6.7

Suppose $x \in R(E)^0$. Then $\{y \in R(E) \mid x \sqsubseteq^L y\}$ is finite and for all $y \in R(E)$ if $x \sqsubseteq^L y$ then $y \in R(E)^0$.

Proof

If $x \in R(E)^0$ then $M(x)$ is finite. Thus $\ll^{-1}M(x)$ is finite. Suppose $x \sqsubseteq^L y$. By the characterisation of \sqsubseteq^L we have $M(y) \subseteq \ll^{-1}M(x)$ so $M(y)$ is finite so y is isolated. As $\ll^{-1}M(x)$ is finite there can only be finitely many such y . ■

At this point it is useful to extend some previous notation. For e an event in a configuration x we use $m(e, x)$ to denote the unique \leq^L -maximal event of x \leq^L -above e . We have extended \leq^L and \leq^R on events to orderings \sqsubseteq^L and \sqsubseteq^R on configurations so that unique factorisation is preserved. Consequently we may extend m .

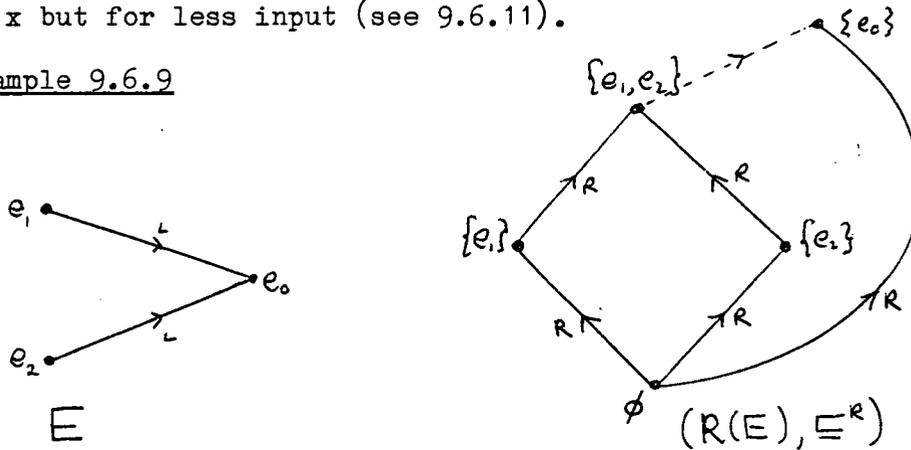
Definition 9.6.8

Suppose x, y are in $R(E)$. For $x \sqsubseteq y$ define $\mu(x, y)$ to be the unique element x' in $R(E)$ such that $x \sqsubseteq^L x' \sqsubseteq^R y$.

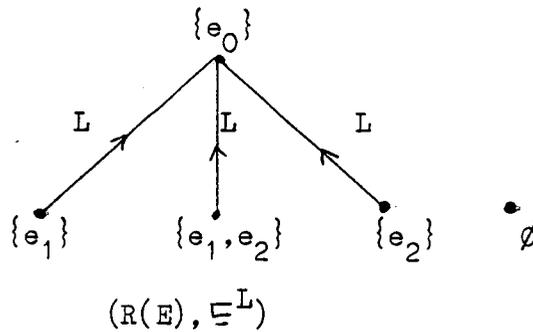
Note $\mu([e], x) = \ll_x^{M^{-1}}\{m(e, x)\}$ for e in x (thus μ gives the prime generated by e in $M(x)$) and also that $m(e, x) = \text{ev}(\mu([e], x))$.

We note some peculiarities of \subseteq^L (it seems a very peculiar ordering from the point of view of denotational semantics). It appears that $x \subseteq^L y$ means the behaviour of y "simulates" that of x but for less input (see 9.6.11).

Example 9.6.9

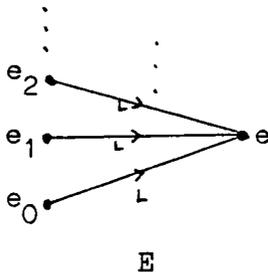


In this example we have drawn an event structure E and alongside it the domain $(R(E), \subseteq^R)$ - the dotted line represents the additional ordering \subseteq^L gives. Below we draw $(R(E), \subseteq^L)$. Note that \emptyset is \subseteq^L -incomparable with all other elements because it is \subseteq^R -minimum.



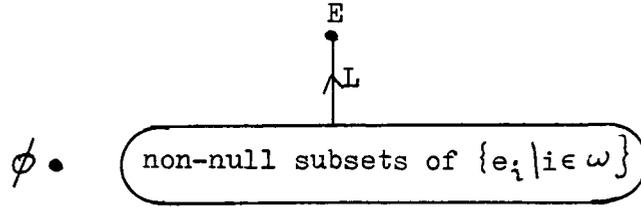
(The event structure E is that associated with stable functions from \mathbb{N} to \mathbb{O} .)

Example 9.6.10



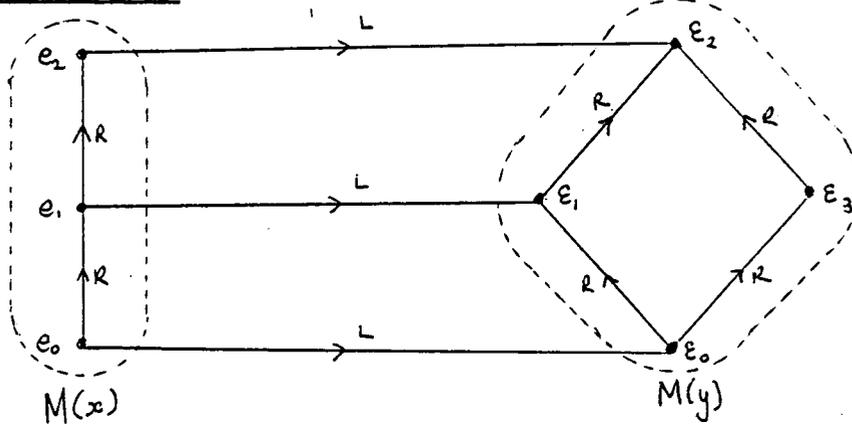
For the event structure E above (associated with $[\mathbb{N} \rightarrow_s \mathbb{O}]$) stable configurations are either subsets of $\{e_i \mid i \in \omega\}$ or the full set E . Ordering by inclusion gives \subseteq . Ordering sets of the first

form by inclusion with E above \emptyset but otherwise incomparable with sets of the first form gives \subseteq^R . Accordingly \subseteq^L looks like:



So the configuration $E \subseteq^L$ -dominates an uncountable set of configurations.

Example 9.6.11



In this event structure take $M(x) = \{e_0, e_1, e_2\}$ and $M(y) = \{\epsilon_0, \epsilon_1, \epsilon_2, \epsilon_3\}$. Then $x \subseteq^L y$. Note ϵ_2 is not \geq^L any event in $M(x)$. (The event structure occurs in $[[T \times \mathbb{O} \rightarrow \mathbb{O}] \rightarrow \mathbb{O}]$ with $e_0 = \epsilon_0 = (\circ, T)$, $e_1 = (\circ \circ, T)$, $e_2 = (\circ \circ \circ, T)$, $\epsilon_1 = (\circ, T)$, $\epsilon_3 = (\circ \circ, T)$ and $\epsilon_2 = (\circ \circ \circ, T)$ where we represent functions from $\mathbb{T} \times \mathbb{O} = \begin{matrix} & \circ & \\ & / \backslash & \\ \circ & & \circ \end{matrix}$ to \mathbb{O} by the minimal points at which they give T . I am grateful to P.L. Curien for this example.)

I do not understand \subseteq^L at all well: The converse relation \supseteq^L has the more intuitive properties (e.g. 9.6.7).

9.7 Stable exponentiation and products of event structures

We have established many properties of $R(E)$ for an event structure E satisfying the axioms of 9.2.1. In particular we defined two partial orderings \subseteq^R and \subseteq^L such that \subseteq , equalling $(\subseteq^L \cup \subseteq^R)^*$, factored uniquely as $\subseteq^L \circ \subseteq^R$. We now define an exponentiation on event structures satisfying the axioms of 9.2.1. The term "exponentiation" will be justified in the next section where we define a category of event structures.

Definition 9.7.1

Let E_0 and E_1 be event structures satisfying the axioms of 9.2.1.

Define $E_0 \rightarrow_s E_1$ to consist of events

$$\{(x, e) \mid x \in R(E_0)^0 \ \& \ e \in E_1\}$$

ordered by \leq^L and \leq^R where

$$(x, e) \leq^L (x', e') \text{ iff } x' \sqsubseteq^R x \ \& \ e \leq^L e'$$

$$(x, e) \leq^R (x', e') \text{ iff } x' \sqsubseteq^L x \ \& \ e \leq^R e'$$

with relation \bowtie given by

$$(x, e) \bowtie (x', e') \text{ iff } x \uparrow^E x' \ \& \ e \bowtie e'.$$

(In the above definition we have not indexed relation symbols to indicate their domain, which above, and in future, should be clear from the context.)

It would be a sick joke if, having got this far, the axioms failed to hold for the exponentiation of event structures. They do. The only difficulties are in showing axiom 4 is true for the exponentiation. From the definitions of the orderings \leq^L and \leq^R on the exponentiation the relation $(x, e) \leq (x', e')$ has two parts; one is $e \leq e'$ in E_1 while the other is $x(\sqsubseteq^L \uparrow R(E_0)^0 \cup \sqsubseteq^R \uparrow R(E_0)^0) * x'$. By previous results an isolated element \sqsubseteq^R -dominates and is \sqsubseteq^L -dominated by only isolated elements. This gives:

Lemma 9.7.2

Let E be an event structure as in 9.2.1. Then the relation $(\sqsubseteq^L \cup \sqsubseteq^R) * \uparrow R(E)^0$ is identical to the relation $(\sqsubseteq^L \uparrow R(E)^0 \cup \sqsubseteq^R \uparrow R(E)^0) *$.

Definition 9.7.3

Let E be an event structure as in 9.2.1. Define \leq on $R(E)$ as $(\sqsubseteq^L \cup \sqsubseteq^R) *$ and \leq^0 as $\leq \uparrow R(E)^0$.

Proofs of properties about \leq on exponentiations will depend on corresponding properties of \leq^0 above holding. For example showing \leq is a partial order on $E_0 \rightarrow_s E_1$ will require that \leq^0 is a partial order on $R(E_0)^0$. In fact the next lemma shows this.

It remains to show $[\text{fix}] = x_i = x'_i$ for all i . Without loss of generality it suffices to show $x_0 = x'_0 = [\text{fix}]$.

Take $e \in M(x'_0)$. Then by repeated use of theorem 9.6.4 characterising \sqsubseteq^L , we deduce from (1) that

$$(2) \quad e = e_0 \leq^M_{x'_0} e'_0 \geq^L e_1 \leq^M_{x'_1} e'_1 \geq^L e_2 \cdots \geq^L e_m \leq^M_{x'_{[m]_n}} e'_m \geq^L e_{m+1} \cdots$$

(here $[m]_n$ is m modulo n)

for some e_i in $M(x_{[i]_n})$ and e'_i in $M(x'_{[i]_n})$ where $i \in \omega$.

The sequence has been continued infinitely by going around and around the loop (1). As $M(x_0)$ is finite and the sequence (2) visits $M(x_0)$ infinitely often there must be e_m, e_q in $M(x_0)$ such that $m < q$ and $[m]_n = [q]_n = 0$ and $e_m = e_q$. Then as \leq is a po, $e_m = e'_m = e_{m+1} = \dots = e_q$. Thus $e_m \in \text{fix}$ so the sequence (2) eventually contains an element of fix . As $\text{fix} \sqsubseteq^R x_i, x'_i$ for all i , using lemma 9.5.6 we have $e_0 (=e)$ must be the earliest element of (2) in fix . But e was chosen to be an arbitrary event in $M(x'_0)$. Thus $M(x'_0) = \text{fix}$. Therefore $x_0 = x'_0 = [\text{fix}]$ as required.

Thus the relation \leq^0 on $R(E)^0$ is a partial order. ■

The next lemma is used to prove 4 (i) holds for the exponentiation of two event structures. It generalises axiom F on $(R(E), \sqsubseteq^R)$ and corollary 9.6.7.

Lemma 9.7.5

Let E be an event structure satisfying the axioms of 9.2.1. Define \leq^0 on $R(E)^0$ as in 9.7.3.

Then for x in $R(E)^0$ we have $\leq^{0-1}\{x\}$ is finite.

Proof

As $x \in R(E)^0$ we have $|M(x)| < \infty$. Also by theorem 9.6.4, characterising \sqsubseteq^L it is clear that $x' \leq^0 x \Rightarrow \forall e' \in M(x') \exists e \in M(x) e' \leq e$. Thus $x' \leq^0 x \Rightarrow M(x') \subseteq \bigcup \{ \leq^{-1}\{e\} \mid e \in M(x) \}$. As $M(x)$ is finite and $\leq^{-1}\{e\}$ is finite for any event e we have $\{x' \mid x' \leq^0 x\}$ is finite, as required. ■

It is not clear that \leq is a partial order ^{on $R(E)$,} at least not from the proof that \leq^0 is.

It now follows that the exponentiation \rightarrow_s on event structures preserves the axioms of 9.2.1.

Theorem 9.7.6

Suppose E_0 and E_1 are event structures satisfying the axioms given in 9.2.1. Then $E_0 \rightarrow_s E_1$ satisfies the axioms too.

Proof

Axiom 1 is clear. Axiom 2 follows as \sqsubseteq^L and \sqsubseteq^R are pos. Axiom 3 (unique factorisation) follows from the unique factorisation of E_1 together with the unique factorisation of \sqsubseteq as $\sqsubseteq^L \circ \sqsubseteq^R$. Axiom 4 (i) follows directly from E_1 satisfying 4 (i) and lemma 9.7.5. Axiom 4 (ii), that \leq is a partial order on $E_0 \rightarrow_s E_1$, follows from the corresponding fact for E_1 and lemma 9.7.4. Axioms 5 and 6 are straightforward. ■

We point out some further axioms which are also preserved by \rightarrow_s .

Proposition 9.7.7

The following axioms may be added (together or separately) to those of 9.2.1 so that a direct analogue of theorem 9.7.6 holds:

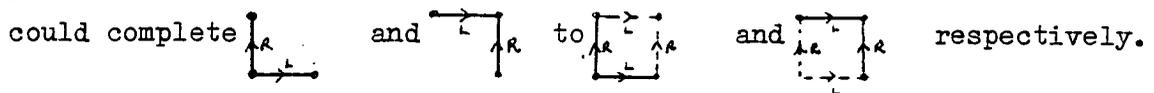
- (i) $e \downarrow^L e' \Rightarrow e \uparrow^L e'$
- (ii) $e \downarrow^L e' \Rightarrow e \wedge^L e'$ exists in E .

Proof

We shall only show how (i) is preserved by \rightarrow_s . Suppose E_0 and E_1 satisfy the axioms of 9.2.1 and (i) above. Suppose $\mathcal{E} \downarrow^L \mathcal{E}'$ in $E_0 \rightarrow_s E_1$. Then \mathcal{E} and \mathcal{E}' have the form $\mathcal{E} = (x, e)$ and $\mathcal{E}' = (x', e')$. As $\mathcal{E} \downarrow^L \mathcal{E}'$ we have $e \downarrow^L e'$. As (i) holds for E_1 we know $e \uparrow^L e'$ i.e. $e, e' \leq^L e''$. By lemma 9.5.3, giving consistent-completeness, $x \sqcap^R x'$ exists. Combined we get

$$\mathcal{E}, \mathcal{E}' \leq^L (x \sqcap^R x', e'')$$

We give an example showing how properties may fail to be preserved by exponentiation. After introducing the axioms we mentioned two "reasonable" further axioms true at zeroth and first order but which were not preserved by \rightarrow_s . Recall the two properties; informally they said that in the event structure we could complete

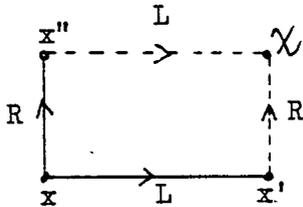


Example 9.7.8

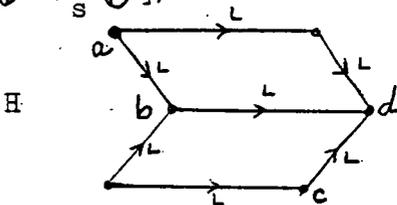
We show the following properties are not preserved by our exponentiation construction:

- (1) $e \sqsubseteq^R e'' \ \& \ e \sqsubseteq^L e' \Rightarrow \exists \mathcal{E} \in E \ e' \sqsubseteq^R \mathcal{E} \ \& \ e'' \sqsubseteq^L \mathcal{E}$
- (2) $e' \sqsubseteq^R e \ \& \ e'' \sqsubseteq^L e \Rightarrow \exists \mathcal{E} \in E \ \mathcal{E} \sqsubseteq^R e'' \ \& \ \mathcal{E} \sqsubseteq^L e'$.

We first show why (2) fails to be preserved by \rightarrow_s . Suppose $e = (x, \eta)$, $e' = (x', \eta')$, $e'' = (x'', \eta'')$ and $e' \sqsubseteq^R e \ \& \ e'' \sqsubseteq^L e$. Then we must have $x \sqsubseteq^L x'$ and $x \sqsubseteq^R x''$ for the isolated x, x', x'' . For (2) to be true we require some isolated \mathcal{X} so that $x'' \sqsubseteq^L \mathcal{X}$ and $x' \sqsubseteq^R \mathcal{X}$. Thus if we can produce an event structure E satisfying (2) but such that for some x, x', x'' with $x'' \sqsupseteq^R x \sqsubseteq^L x'$ there is no \mathcal{X} so that $x' \sqsubseteq^R \mathcal{X} \sqsupseteq^L x''$ we have shown \rightarrow_s does not preserve (2).



Here is a suitable event structure H (it is associated with $[\mathbb{T} \times \mathbb{O} \rightarrow_s \mathbb{O}]$):



Clearly it satisfies (2). Take $M(x) = \{a\}$, $M(x') = \{b\}$ and $M(x'') = \{a, c\}$. Then we have $x'' \sqsupseteq^R x \sqsubseteq^L x'$. However $x' \sqsubseteq^R \mathcal{X}$ implies $x' = \mathcal{X}$ but then we cannot have $x'' \sqsubseteq^L \mathcal{X}$. Thus there is no \mathcal{X} such that $x' \sqsubseteq^R \mathcal{X} \sqsupseteq^L x''$.

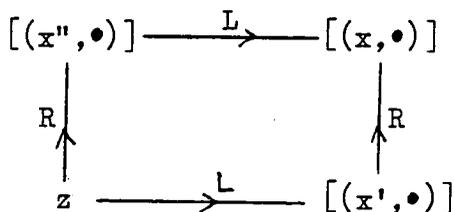
Therefore (2) is not preserved by \rightarrow_s . A further simple observation uses this fact to show (1) cannot be preserved either. Let $\{\bullet\}$ be the event structure consisting of a single event \bullet (It represents \mathbb{O}). Let the event structure H and its configurations x, x', x'' be as above. Then in the event structure $((H \rightarrow_s \{\bullet\}) \rightarrow_s \{\bullet\})$ we have:

$$e'' = ([[x'', \bullet]], \bullet)$$

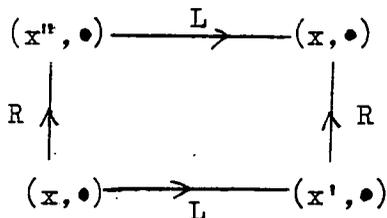


$$e = ([[x, \bullet]], \bullet) \xrightarrow{L} ([[x', \bullet]], \bullet) = e'$$

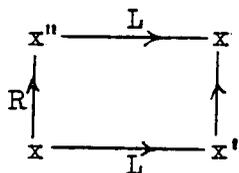
Clearly H satisfies (1). If \rightarrow_s preserved (1) then we could complete the above diagram to a "square" and this would give some z so that in $R(H \rightarrow_s \{\bullet\})$:



However by the characterisation of \subseteq^L there would then be an event (χ, \bullet) in $M(z)$ such that



But then we would have



which we

proved impossible. Thus (1) is not preserved by \rightarrow_s either.

The product of two event structures is defined simply as (disjoint) juxtaposition. The use of the term "product" will be justified in the following section.

Definition 9.7.9

Let $E_i = (E_i, \leq_i^L, \leq_i^R, \#_i)$, for $i = 0, 1$, be two event structures satisfying the axioms in 9.2.1. Define $E_0 \oplus E_1$ to be the event structure $(E_0 \cup E_1, \leq_0^L \cup \leq_1^L, \leq_0^R \cup \leq_1^R, \#_0 \cup \#_1)$ where \cup denotes disjoint union.

Similarly define $\bigoplus_{i \in I} E_i$, where i ranges over an indexing set I , to be

$$\left(\bigcup_i E_i, \bigcup_i \leq_i^L, \bigcup_i \leq_i^R, \bigcup_i \times_i \right)$$

where \bigcup denotes disjoint union.

It is clear that the axioms in 9.2.1 are preserved by countable "products".

Theorem 9.7.10

Let I be a countable set, indexing event structures E_i ($i \in I$) which satisfy the axioms in 9.2.1. Then $\bigoplus_{i \in I} E_i$ satisfies the axioms too.

We point out an alternative way of producing higher type event structures. With the wisdom of hindsight it would be a better way to proceed. From our results in 9.4 it is clear that we could have worked with the s -configurations $M(x)$ rather than the configurations x , for x in $R(E)$. This would have advantages. Firstly our definition of the configurations $R(E)$ is a little unnatural because of condition (ii) in 9.3.2. Secondly the conflict relation \times only imposes a very weak constraint in forming configurations.

Interestingly our work can be paralleled in the following way.

Define event structures instead as being of the form $(E, \leq^L, \leq^R, \Delta)$ - Δ a refl., symm. rel. satisfying all but axiom 6 where Δ is to replace \times_s as the conflict relation determining s -configurations. Let the definition of exponentiation be like 9.7.1 with the one modification that

$$(x, e) \Delta (x', e') \text{ iff } x \uparrow^R x' \ \& \ e \Delta e'.$$

Then the assumption that Δ determines the same s -configurations as \times_s (which equals $\times \cup (\downarrow^L \sim 1)$ remember) is preserved by exponentiation and product. (It is not the case that their being identical is.) Thus the ordering \leq^L is used explicitly in defining the enabling relation but need not be mentioned in defining the conflict relation appropriate to s -configurations.

9.8 The category of stable event structures

In this section we form a category from event structures satisfying the axioms in 9.2.1. We show the category is cartesian closed and in the next section that it determines a cartesian-closed full subcategory of Berry's category of bidomains. Within the category of event structures \rightarrow_s and \oplus will correspond to exponentiation and product thus justifying those terms in the

previous section:

A configuration x of $E_0 \rightarrow_s E_1$ has an obvious interpretation as a function \bar{x} now defined.

Definition and proposition 9.8.1

Suppose E_0 and E_1 are event structures satisfying the axioms of 9.2.1. For x in $R(E_0 \rightarrow_s E_1)$ taking

$$\bar{x}(x_0) = \{e \in E_1 \mid \exists (y, e) \in x \ y \sqsubseteq x_0\} \text{ for } x_0 \text{ in } R(E_0)$$

defines a function $\bar{x}: R(E_0) \rightarrow R(E_1)$ which is continuous with respect to \sqsubseteq and stable with respect to \sqsubseteq^R . In fact $x \mapsto \bar{x}$ defines a 1-1 correspondence between the configurations and such functions. Also

$$M(\bar{x}(x_0)) = \{e \in E_1 \mid \exists (y, e) \in M(x) \ y \sqsubseteq^R x_0\}.$$

Proof

Let E_0 and E_1 be event structures satisfying the axioms of 9.2.1. Suppose $x \in R(E_0 \rightarrow_s E_1)$. From the fact that x is a configuration it follows that for x_0 in $R(E_0)$ the set $\bar{x}(x_0)$ is a configuration of E_0 . Thus \bar{x} is a function: $R(E_0) \rightarrow R(E_1)$. That it is continuous w.r.t. \sqsubseteq follows routinely.

We now show for x_0 in $R(E_0)$ that $M(\bar{x}(x_0)) = \{e \in E_1 \mid \exists (y, e) \in M(x) \ y \sqsubseteq^R x_0\}$. Let e be in $M(\bar{x}(x_0))$. Then there is z with $z \sqsubseteq^R x_0$ and (z, e) in x . The element $m((z, e), x)$ of $M(x)$ has the form (y, e) with $y \sqsubseteq^R x$ showing $e \in$ r.h.s. as required. Conversely suppose for some (y, e) in $M(x)$ we have $y \sqsubseteq^R x_0$. Assume $e' \in \bar{x}(x_0)$ where e' is \leq^L -comparable with e (so $e \downarrow^L e'$). Then $(y', e') \in x$ where for some $y' \sqsubseteq^R x_0$. Clearly as $(y, e) \downarrow^L (y', e')$ in x , $(y', e') \leq^L (y, e)$ so $e' \leq^L e$. Thus $e \in M(\bar{x}(x_0))$ as required.

From the above characterisation of $M(\bar{x}(x_0))$ it follows that \bar{x} is \sqsubseteq^R -continuous. Suppose $z \uparrow^R z'$ for z, z' in $R(E_0)$. Then $z \sqcap z' = z \sqcap^R z'$. For \bar{x} to be stable we further require $\bar{x}(z \sqcap z') = \bar{x}(z) \sqcap \bar{x}(z')$. By \sqsubseteq^R -monotonicity we have $\bar{x}(z \sqcap z') \sqsubseteq^R \bar{x}(z) \sqcap \bar{x}(z')$ where $\bar{x}(z) \sqcap \bar{x}(z') = \bar{x}(z) \cap \bar{x}(z')$. Suppose $e \in \bar{x}(z) \cap \bar{x}(z')$. Then $(y, e), (y', e) \in x$ for some $y \sqsubseteq^R z$ and $y' \sqsubseteq^R z'$. As x is a configuration and $y \uparrow^R y'$ we have $(y \sqcap^R y', e) \in x$. Therefore $e \in \bar{x}(z \sqcap z')$. Thus the sets $\bar{x}(z \sqcap z')$ and $\bar{x}(z) \sqcap \bar{x}(z')$ are equal so \bar{x} is stable.

We now construct an inverse to $x \mapsto \bar{x}$. Suppose $f: R(E_0) \rightarrow R(E_1)$ is continuous w.r.t. \sqsubseteq and stable w.r.t. \sqsubseteq^R . Define

$$\phi(f) = \{(z, e) \in R(E_0)^0 \times E_1 \mid e \in f(z)\}.$$

We show $\phi(f) \in R(E_0 \rightarrow_s E_1)$ and $\overline{\phi(f)} = f$ and $\phi(\bar{x}) = x$.

In showing $\phi(f) \in R(E_0 \rightarrow_s E_1)$ it is easily checked to be \leq -left closed and consistent. Suppose for $(z, e), (z', e')$ in X we have $(z, e) \downarrow^L (z', e')$; then $z \uparrow^R z'$ and $e \downarrow^L e'$. As f is \sqsubseteq^R -monotonic $f(z) \uparrow^R f(z')$. Thus as we have $f(z) \uparrow^R f(z') \& e \in f(z) \& e' \in f(z') \& e \downarrow^L e'$ by lemma 9.5.6 we get $m(e, f(z)) = m(e, f(z'))$. Put $\mathcal{E} = m(e, f(z))$. Then as f is stable w.r.t. \sqsubseteq^R it follows as $\mathcal{E} \in f(z)$ and $\mathcal{E} \in f(z')$ that $\mathcal{E} \in f(z \sqcap z')$. Therefore $(z \sqcap z', \mathcal{E}) \in \phi(f)$ with $(z, e), (z', e') \leq^L (z \sqcap z', \mathcal{E})$ as required to show $\phi(f)$ is a configuration.

As f is continuous $\overline{\phi(f)} = f$. Also by a direct translation of the definitions $\phi(\bar{x}) = x$. Thus the map $x \mapsto \bar{x}$ is 1-1. ■

We now define the category of event structures. Morphisms from E_0 to E_1 are taken to be configurations of $E_0 \rightarrow_s E_1$. Composition $x \bullet y$ is defined so that $\overline{x \bullet y}$ equals $\bar{x} \circ \bar{y}$ the usual function composition on the function \bar{x} and \bar{y} .

Definition and proposition

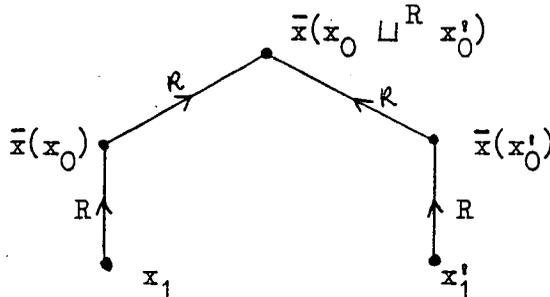
Define \mathbb{E} to be the category consisting of objects event structures E satisfying the axioms in 9.2.1, morphisms E_0 to E_1 being elements of $R(E_0 \rightarrow_s E_1)$ with the following composition denoted \bullet :

For x in $R(E_0 \rightarrow_s E_1)$ and y in $R(E_1 \rightarrow_s E_2)$ define $y \bullet x = \{(x_0, e_2) \in R_s(E_0)^0 \times E_2 \mid \exists (x_1, e_2) \in y \ x_1 \sqsubseteq \bar{x}(x_0)\}$. Then $y \bullet x \in R(E_0 \rightarrow_s E_2)$ and $y \bullet x = \bar{y} \circ \bar{x}$ the usual composition of the functions \bar{y} and \bar{x} . (We call \mathbb{E} the category of stable event structures.)

Proof

First we must check that the definition is correct, that \mathbb{E} is indeed a category. We check that for x in $R(E_0 \rightarrow_s E_1)$ and y in $R(E_1 \rightarrow_s E_2)$ we have $y \bullet x$ in $R(E_0 \rightarrow_s E_2)$. It is easy to check that $y \bullet x$ is \leq -left closed and consistent. Suppose for (x_0, e_2) ,

(x'_0, e'_2) in $y \bullet x$ we have $(x_0, e_2) \downarrow^L (x'_0, e'_2)$ i.e. $x_0 \uparrow^R x'_0$ and $e_2 \downarrow^L e'_2$. We show $(x_0 \sqcap^R x'_0, \varepsilon) \in y \bullet x$ for some $\varepsilon \geq^L e_2, e'_2$. As (x_0, e_2) and (x'_0, e'_2) are in $y \bullet x$ we have for some (x_1, e_2) and (x'_1, e'_2) in y that $x_1 \sqsubseteq \bar{x}(x_0)$ and $x'_1 \sqsubseteq \bar{x}(x'_0)$; clearly by factorisation, without loss of generality, we may assume $x_1 \sqsubseteq^R \bar{x}(x_0)$ and $x'_1 \sqsubseteq^R \bar{x}(x'_0)$. Summarising the facts in a picture:



However as y is a configuration containing (x_1, e_2) and (x'_1, e'_2) with $(x_1, e_2) \downarrow^L (x'_1, e'_2)$ there exists (χ, ε) in y such that $\chi \sqsubseteq^R x_1, x'_1$ and $e_2, e'_2 \leq^L \varepsilon$. As \bar{x} is stable $\bar{x}(x_0 \sqcap^R x'_0) = \bar{x}(x_0) \sqcap^R \bar{x}(x'_0)$. Thus $\chi \sqsubseteq \bar{x}(x_0 \sqcap^R x'_0)$ so $(x_0 \sqcap^R x'_0, \varepsilon) \in y \bullet x$ as required.

Suppose $x \in R(E_0 \rightarrow_s E_1)$ and $y \in R(E_1 \rightarrow_s E_2)$. Then routine manipulation of the definitions gives for any x_0 in $R(E_0)$ that $\overline{y(\bar{x}(x_0))} = \overline{y \bullet x}(x_0)$. Thus $\overline{y \bullet x} = \overline{y} \circ \bar{x}$.

Finally composition is clearly associative as function composition is and each object E has an identity morphism $1_E = \{(x, e) \mid e \in x\}$ in $R(E \rightarrow_s E)$. Thus \mathbb{E} is a category as stated. ■

The category \mathbb{E} is closed under products. Given two event structures E_0 and E_1 in \mathbb{E} a product will be $(E_0 \oplus E_1, \pi_0, \pi_1)$ where the projection function π_i to E_i are obtained by restricting configurations to E_i . (It is well-known that products of E_0, E_1 are isomorphic.)

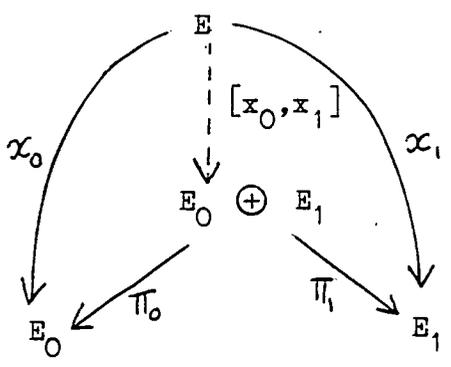
Lemma 9.8.3

The category \mathbb{E} is closed under $(\omega -)$ products. A product of E_0, E_1 in \mathbb{E} will be $(E_0 \oplus E_1, \pi_0, \pi_1)$ where $\pi_i = \{(x, e) \in E_0 \oplus E_1 \rightarrow_s E_i \mid e \in x \cap E_i\}$ for $i = 0, 1$.

Proof

Let E_0 and E_1 be event structures. First note for π_i as

defined above $\pi_i \in R(E_0 \oplus E_1 \rightarrow_s E_2)$. In order that $(E_0 \oplus E_1, \pi_0, \pi_1)$ be a product we require for any x_0 in $R(E \rightarrow_s E_0)$ and x_1 in $R(E \rightarrow_s E_1)$ there exists a unique element $[x_0, x_1]$ in $R(E \rightarrow_s E_0 \oplus E_1)$ such that $x_0 = \pi_0 \bullet [x_0, x_1]$ and $x_1 = \pi_1 \bullet [x_0, x_1]$.

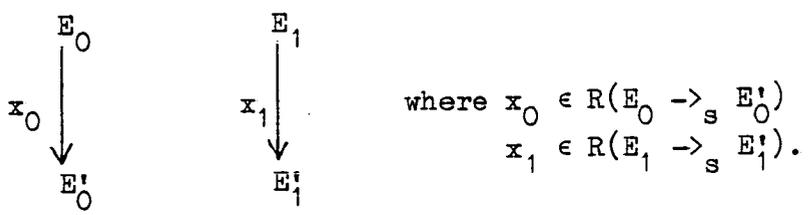


For the above set-up taking $[x_0, x_1] = x_0 \cup x_1$ (where strictly speaking the configurations x_i are formed on the disjoint copies of the events E_0 and E_1 in $E_0 \oplus E_1$) makes the above diagram commute in \mathbb{E} ; the uniqueness of $[x_0, x_1]$ follows by routine manipulation. ■

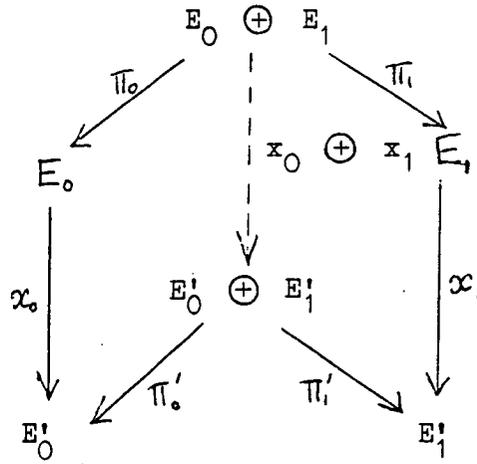
We now give some useful notation.

Definition and Notation 9.8.4

Suppose we have the following set-up in \mathbb{E} :

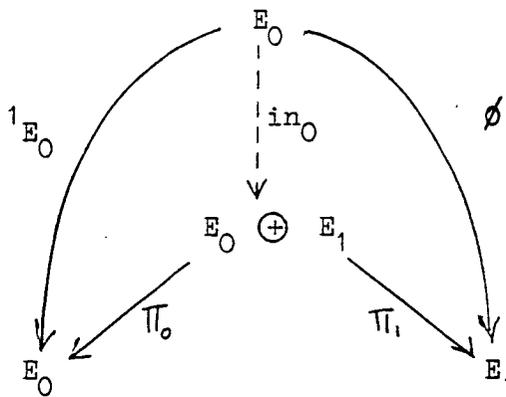


Then certainly by the above result $E_0 \oplus E_1$ and $E'_0 \oplus E'_1$ are products in \mathbb{E} . The operation \oplus extends to a functor. For the morphisms x_0, x_1 above define $x_0 \oplus x_1$ to be the unique map making the following diagram commute:



So using the notation in the above proof $x_0 \oplus x_1$ is $[x_0 \cdot \pi_0, x_1 \cdot \pi_1]$ and has the commutativity properties $\pi'_i \cdot x_0 \oplus x_1 = x_i \cdot \pi_i$ for $i = 0, 1$.

Consider the following diagram in which the null configuration \emptyset is used as a morphism:



Clearly by the properties of product there is a unique morphism in_0 making the above diagram commute. Similarly there is a morphism $in_1: E_1 \rightarrow E_0 \oplus E_1$.

The following observation allows us to simplify notation.

Lemma 9.8.5

Let E_0, E_1 be event structures in \mathbb{E} . Let $\pi_i: E_0 \oplus E_1 \rightarrow E_i$ for $i = 0, 1$ be the projection morphisms introduced in 9.8.3. Then $R(E_0 \oplus E_1)$ is isomorphic to $R(E_0) \times R(E_1)$ consisting of pairs ordered coordinatewise under the map

$$x \longmapsto (\bar{\pi}_0(x), \bar{\pi}_1(x)).$$

Notation 9.8.6

Henceforth we shall identify $R(E_0 \oplus E_1)$ with $R(E_0) \times R(E_1)$ in which the orderings are determined coordinatewise. Thus instead of x in $R(E_0 \oplus E_1)$ we shall often write (x_0, x_1) in $R(E_0) \times R(E_1)$ where $x_0 = \bar{\pi}_0(x)$ and $x_1 = \bar{\pi}_1(x)$. With this identification, the function $\overline{x_0 \oplus x_1}: R(E_0 \oplus E_1) \rightarrow R(E'_0 \oplus E'_1)$ may be expressed as the function $(y_0, y_1) \mapsto (\bar{x}_0(y_0), \bar{x}_1(y_1))$ by simply using the commutativity properties of $x_0 \oplus x_1$.

To show \mathbb{E} is cartesian closed we require the further fact that it is closed under exponentiation. In establishing this we use the following configurations which correspond to application (ap) and curryfication, or abstraction (ab).

Definition and proposition 9.8.7

Suppose E_0, E_1 and E_2 are event structures in \mathbb{E} . Then, with respect to E_0 and E_1 defining

$$\text{ap} = \{((x, x_0), e_1) \in (E_0 \rightarrow_s E_1) \oplus E_0 \rightarrow_s E_1 \mid (x_0, e_1) \in x\}$$

gives $\text{ap} \in R((E_0 \rightarrow_s E_1) \oplus E_0 \rightarrow_s E_1)$.

Also, with respect to E_0, E_1 and E_2 , defining

$$\text{ab} = \{(x, (x_0, (x_1, e_2))) \in (E_0 \oplus E_1 \rightarrow_s E_2) \rightarrow_s (E_0 \rightarrow_s E_1 \rightarrow_s E_2) \mid ((x_0, x_1), e_2) \in x\}$$

gives $\text{ab} \in R((E_0 \oplus E_1 \rightarrow_s E_2) \rightarrow_s (E_0 \rightarrow_s E_1 \rightarrow_s E_2))$.

Proof

The subset ap is clearly \leq -left closed. Suppose $((x, x_0), e), ((x', x'_0), e')$ are in ap and $((x, x_0), e) \downarrow^L ((x', x'_0), e')$. Then $x \uparrow^R x', x_0 \uparrow^R x'_0$ and $e \downarrow^L e'$. Thus $(x_0, e) \in x$ and $(x'_0, e') \in x'$ with $(x_0, e) \downarrow^L (x'_0, e')$ and $x \uparrow^R x'$. By 9.5.6 we have $m((x_0, e), x) = m((x'_0, e'), x')$; call this common event (χ, ε) . Then, as required, we have $((x, x_0), e), ((x', x'_0), e') \leq^L ((x \sqcap^R x'), \chi), \varepsilon \in \text{ap}$.

The proof that ab is a configuration is similar. ■

That the configurations ap and ab do correspond to application and abstraction of functions is justified by the next lemma.

Lemma 9.8.8

For the situation described in 9.8.7

(i) for all (x, x_0) in $R((E_0 \rightarrow_s E_1) \oplus E_0)$

$$\overline{\text{ap}}(x, x_0) = \bar{x}(x_0)$$

(ii) letting y be $\text{ab}(x)$ for x in $R_s(E_0 \oplus E_1 \rightarrow_s E_1)$, for all

(z_0, z_1) in $R(E_0 \oplus E_1)$

$$\overline{y}(z_0)(z_1) = \overline{x}(z_0, z_1).$$

Proof

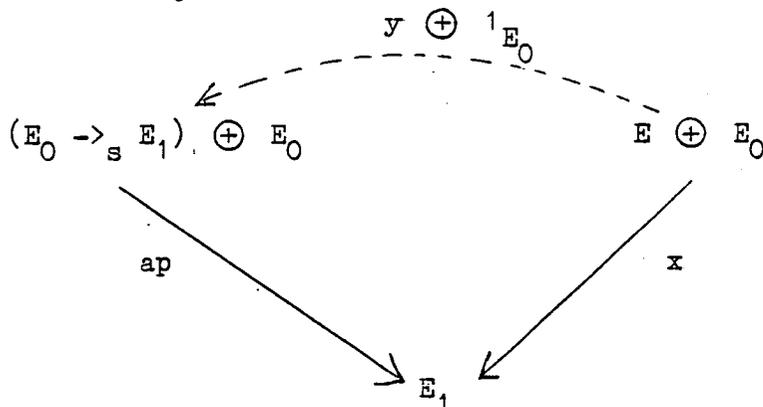
A routine consequence of the definitions. ■

Theorem 9.8.9

The category \mathbb{E} is closed under exponentiation. An exponentiation of E_0, E_1 in \mathbb{E} is $(E_0 \rightarrow_s E_1, ap)$ where ap is as defined in 9.8.7.

Proof

Let E_0 and E_1 be event structures in \mathbb{E} . As in 9.8.7 we have $ap \in R((E_0 \rightarrow_s E_1) \oplus E_0 \rightarrow_s E_1)$. In order for $(E_0 \rightarrow_s E_1, ap)$ to be an exponentiation we require for any E in \mathbb{E} and any x in $R(E \oplus E_0 \rightarrow_s E_1)$ there is a unique y in $R(E \rightarrow_s E_0 \rightarrow_s E_1)$ such that $x = ap \bullet (y \oplus 1_{E_0})$.



The requirement is satisfied by taking $y = \overline{ab}(x)$. Firstly the diagram commutes. Let (z, z_0) be in $R(E \oplus E_0)$. Then

$$\begin{aligned} & \overline{ap \bullet y \oplus 1_{E_0}}(z, z_0) \\ &= \overline{ap \circ y \oplus 1_{E_0}}(z, z_0) \\ &= \overline{ap}(\overline{y}(z), z_0) \\ &= \overline{\overline{y}(z)}(z_0) \quad \text{by lemma 9.8.8 part (i)} \\ &= \overline{x}(z, z_0) \quad \text{by 9.8.8 part (ii)}. \end{aligned}$$

Thus the functions \overline{x} and $\overline{ap \bullet y \oplus 1_{E_0}}$ are equal. As $x \mapsto \overline{x}$ is 1-1 we have the diagram commutes when y is $\overline{ab}(x)$. To establish that this

choice of y is unique assume $x = ap \bullet (w \oplus 1_{E_0})$ for w in $R(E \rightarrow_s (E_0 \rightarrow_s E_1))$. Then as in the above manipulation

$$\overline{w}(z)(z_0) = \overline{y}(z)(z_0)$$

for any (z, z_0) in $R(E \oplus E_0)$. Therefore using the fact that $x \mapsto \overline{x}$ is 1-1 w equals y as required. ■

We note one further fact about the category \mathbb{E} .

Lemma 9.8.10

The category \mathbb{E} has a terminal object, the null event structure.

Proof

Clearly for any event structure E in \mathbb{E} there is a unique morphism \emptyset in $R(E \rightarrow_s \emptyset)$ so the null event structure \emptyset is the terminal object of \mathbb{E} . ■

Collecting facts together we have:

Theorem 9.8.11

The category \mathbb{E} is cartesian closed.

In fact now it follows routinely that the categories $(\mathbb{E}, \subseteq^R)$ and (\mathbb{E}, \subseteq) , obtained by ordering the morphisms by just \subseteq^R or just \subseteq , are \wedge -categories. There are stable event structures representing the domains \mathbb{T} and \mathbb{N} ; the truth values \mathbb{T} are for example represented by $(\{tt, ff\}, 1, 1, \{\{tt, ff\}\})$. By the result of Berry and Curien we have two models for PCF. The one obtained from (\mathbb{E}, \subseteq) is order extensional. We show \mathbb{E} represents a full subcategory of bidomains in the next section.

We end the section with cute characterisations of the application and identity morphisms.

Lemma 9.8.12

The application morphism ap defined in 9.8.7 is characterised by

$$M(ap) = \{((p, x), e) \mid p \in Pr (R(E_0 \rightarrow_s E_1)) \& (x, e) = ev(p)\}.$$

The identity morphism of E in \mathbb{E} is characterised by

$$M(1_E) = \{(p, e) \mid p \in Pr (R(E)) \& e = ev(p)\}.$$

Proof

Simply consequences of \leq^L -maximality. ■

9.9 Cartesian closed categories of domains

We introduce two categories of domains, $R\mathbb{E}$ with objects of the form $(R(E), \sqsubseteq^L, \sqsubseteq^R)$ and $B\mathbb{E}$ with objects of the form $(R(E), \sqsubseteq, \sqsubseteq^R)$. The categories $R\mathbb{E}$ and $B\mathbb{E}$ will be trivially isomorphic as categories and both equivalent as categories to \mathbb{E} . In this sense \mathbb{E} represents them. The category $B\mathbb{E}$ will be a cartesian closed full subcategory of Berry's category of bidomains (BIDOM).

We start with a lemma which is a key result in proving \mathbb{E} is equivalent to the category $R\mathbb{E}$ and also that our future definition of $R\mathbb{E}$ is proper.

Lemma 9.9.1

Suppose E is in \mathbb{E} . Then $\exists e \in E \ x = [e]$ iff (i) x is a \sqsubseteq^R -complete prime and (ii) $\forall x_0, x_1 \ x \sqsubseteq^R x_0 \ \& \ x_1 \sqsubseteq^L x \ \& \ x = x_0 \sqcup x_1 \Rightarrow (x = x_0 \text{ or } x = x_1)$.

Proof

" \Rightarrow " Suppose x is of the form $[e]$ for e an event in event structure E . Then (i) is clear by the characterisations of \sqsubseteq^R -complete primes. Supposing $x_0 \sqsubseteq^R [e], x_1 \sqsubseteq^L [e]$ and $[e] = x_0 \sqcup x_1$ gives $e \in x_0$ or $e \in x_1$. Thus $x_0 = [e]$ or $x_1 = [e]$ as required for (ii) to hold.

" \Leftarrow " Suppose (i) and (ii) hold for x in $R(E)$. If x were not of the form $[e]$ where $e = ev(x)$ then taking $x_1 = [e]$ and $x_0 = [\bigwedge_x^{M-1} \{e'\}]$ for some $e' \in M(x) \setminus [e]$ contradicts (ii). ■

Thus events identified with $[e]$ in $R(E)$ may be picked out as those \sqsubseteq^R -complete primes x with no non-trivial decomposition as $x_0 \sqcup x_1$ with $x_0 \sqsubseteq^R x$ and $x_1 \sqsubseteq^L x$. Having picked out such representatives of events in the domain the orderings \sqsubseteq^L and \sqsubseteq^R restricted to the representatives return \leq^L and \leq^R by lemmas 9.6.6 and 9.5.2 (iii).

We wish to form a category of domains $(R(E), \sqsubseteq^L, \sqsubseteq^R)$ from \mathbb{E} . As morphisms from $R(E_0)$ to $R(E_1)$ we take functions \bar{x} for x in $R(E_0 \rightarrow_s E_1)$. However a little care is needed as distinct event structures may yield the same domain; we want the definition of morphisms in the new category to be independent of the event structures chosen to represent the domains $R(E_0)$ and $R(E_1)$.

Precisely, we require this lemma.

Lemma 9.9.2

Suppose E_0, E'_0 and E_1, E'_1 are in \mathbb{E} . Then $(R(E_i), \subseteq^L, \subseteq^R) = (R(E'_i), \subseteq^L, \subseteq^R)$ for $i = 0, 1$ implies $(R(E_0 \rightarrow_s E_1), \subseteq^L, \subseteq^R) = (R(E'_0 \rightarrow_s E'_1), \subseteq^L, \subseteq^R)$.

Proof

Using 9.9.1 it is clear that the events and orderings of $E_0 \rightarrow_s E_1$ and $E'_0 \rightarrow_s E'_1$ are identical. Suppose $X \in R(E_0 \rightarrow_s E_1)$ and $X \notin R(E'_0 \rightarrow_s E'_1)$. Then this must be because, for some (x, e_0) and (x, e_1) in X , we have $e_0 \not\ll e_1$, where $\not\ll$ is the conflict relation of E'_1 . However e_0, e_1 are in $\bar{X}(x)$ which is in $R(E'_1)$ so consistent, a contradiction. ■

We may now define the category $R\mathbb{E}$ assured the definition is good.

Definition 9.9.3

Define $R\mathbb{E}$ to consist of objects $(R(E), \subseteq^L, \subseteq^R)$ for E in \mathbb{E} with morphisms $R(E_0)$ to $R(E_1)$ precisely the functions \bar{x} for x in $R(E_0 \rightarrow_s E_1)$ with the usual composition.

Clearly by the properties of $x \mapsto \bar{x}$ we have:

Lemma 9.9.4

The structure $R\mathbb{E}$ is a category.

We establish that $R\mathbb{E}$ and \mathbb{E} are equivalent as categories [Mac] so the categorical properties of \mathbb{E} transfer to $R\mathbb{E}$. The category \mathbb{E} represents the category $R\mathbb{E}$.

Proposition 9.9.5

Define $R: \mathbb{E} \rightarrow R\mathbb{E}$ to act on objects by $E \mapsto R(E)$ and on arrows by $x \mapsto \bar{x}$.

$$\begin{array}{ccc}
 E_0 & & R(E_0) \\
 \downarrow x & \xrightarrow{\quad R \quad} & \bar{x} \\
 E_1 & & R(E_1)
 \end{array}$$

Then R is a natural equivalence of categories.

Proof

That R is a functor follows directly from proposition 9.8.1. In [Mac] (theorem 1 page 91) it is shown that R is an equivalence of categories is equivalent to R being full, faithful and dense (R is dense if each object in the codomain category of R is isomorphic to an image object under R.) As R is onto the objects of $R\mathbb{E}$ the functor R is clearly dense. Proposition 9.8.1 shows R is full and faithful. ■

In the above sense the category of event structures \mathbb{E} represents the category of domains $R\mathbb{E}$. If domains of the form $R(E)$ were axiomatised a more impressive representation theorem would hold. From a domain D satisfying the axioms one would obtain an event structure representing it as follows: For events take those elements of D satisfying (i) and (ii) of lemma 9.9.1 ordered by the restrictions of \subseteq^L and \subseteq^R with conflict relation $e \# e'$ iff $\forall x \in D e \subseteq x \Rightarrow e' \not\subseteq x$.

Because of proposition 9.9.5 the categorical properties of transmit to $R\mathbb{E}$.

Proposition 9.9.6

The category $R\mathbb{E}$ is cartesian closed. A product of $R(E_0), R(E_1)$ in $R\mathbb{E}$ is $R(E_0) \times R(E_1)$ the set of pairs having orders \subseteq^L and \subseteq^R determined pointwise with projections the usual set-theoretic projection functions. An exponentiation of $R(E_0), R(E_1)$ is $(R(E_0 \rightarrow_s E_1), \overline{ap})$ where ap is defined in 9.8.7. A terminal object in $R\mathbb{E}$ is $\{\emptyset\}$.

From the category $R\mathbb{E}$ it is easy to construct an isomorphic category which will turn out to be a full subcategory of Berry's category of bidomains (BIDOM). Recall for a domain $(R(E), \subseteq^L, \subseteq^R)$ we have \subseteq equals $(\subseteq^L \cup \subseteq^R)^*$.

Definition 9.9.7

Define $B\mathbb{E}$ to consist of objects $(R(E), \subseteq, \subseteq^R)$ for E in \mathbb{E} with morphisms $R(E_0)$ to $R(E_1)$ which are functions \bar{x} for x in $R(E_0 \rightarrow_s E_1)$ with the usual composition.

Theorem 9.9.8

The structure $B\mathbb{E}$ is a cartesian closed full subcategory of BIDOM, Berry's category of bidomains.

Proof

We conclude $B\mathbb{E}$ is a cartesian closed category directly from theorem 9.9.6 as $R\mathbb{E}$ and $B\mathbb{E}$ are obviously isomorphic categories. The functor establishing this is given by

$$(R(E), \sqsubseteq^L, \sqsubseteq^R) \mapsto (R(E), (\sqsubseteq^L \cup \sqsubseteq^R)^*, \sqsubseteq^R)$$

on objects and as the identity on morphisms; noting we can recover \sqsubseteq^L from \sqsubseteq and \sqsubseteq^R provides the inverse.

We cannot immediately prove the objects of $B\mathbb{E}$ are bidomains as these are defined in terms of morphisms in the category of distributive biopcd's DBIOPCD (see section 8.2). We first show the objects of $B\mathbb{E}$ are distributive biopcd's. Refer to §2.12 and 14 for the axioms on distributive biopcd's. (Throughout this proof we will abbreviate $(R(E), \sqsubseteq, \sqsubseteq^R)$ to $R(E)$.)

Suppose $R(E)$ is an object in $B\mathbb{E}$. The distributivity axiom clearly holds for $R(E)$ by the properties of \sqsubseteq^R , in particular that $(R(E), \sqsubseteq^R)$ is prime algebraic. Of the remaining axioms all but axiom (iv) follow directly. Recall axiom (iv) is:

$$\begin{aligned} &\forall S, S' \quad S, S' \sqsubseteq\text{-directed subsets of } R(E) \\ &\forall s \in S, s' \in S' \quad \exists t \in S, t' \in S' \quad s \sqsubseteq t \ \& \ s' \sqsubseteq t' \ \& \ t \sqsubseteq^R t' \\ &\qquad \Rightarrow \sqcup S \sqsubseteq^R \sqcup S'. \end{aligned}$$

Assume the hypothesis of the axiom. Remember $\sqcup S = \bigcup S$ for \sqsubseteq -directed subsets S . We require $M(\sqcup S) \sqsubseteq M(\sqcup S')$. Take e in $M(\sqcup S)$. Then $e \in M(s)$ for some s in S . As e is \sqsubseteq^L -maximal in $\sqcup S (= \bigcup S)$ we have

$$(1) \quad \forall t \in S \quad s \sqsubseteq t \Rightarrow e \in M(t).$$

By assumption, taking s' some arbitrary elements of S' , we have for some t in S and t' in S'

$$t \sqsubseteq^R t' \ \& \ s \sqsubseteq t \ \& \ s' \sqsubseteq t'.$$

Using (1) we get $e \in M(t')$. Suppose $t' \sqsubseteq t'_1$. Then again by assumption $e \in M(t'_1)$. Thus for $t'_1 \sqsupseteq t$ we have $e \in M(t'_1)$. Thus $e \in M(\sqcup S')$ as required. We conclude the objects of $B\mathbb{E}$ are distributive biopcd's.

The morphisms of DBIOCPD are exactly those functions which are continuous with respect to the extensional order and stable with respect to the stable order. As the objects of $B\mathbb{E}$ are distributive biocpd's from proposition 9.8.1 we get that $B\mathbb{E}$ is a full subcategory of DBIOCPD.

As $B\mathbb{E}$ is a full subcategory of DBIOCPD we know that products and exponentiations in $B\mathbb{E}$ are respectively products and exponentiations in DBIOCPD. Berry's exponentiation is formed from a set of functions which are ordered both pointwise and according to his stable ordering on functions. Ours is defined as a set of configurations ordered by \sqsubseteq (inclusion) and \sqsubseteq^R . However as exponentiations are isomorphic the two constructions of exponentiation give isomorphic domains and, in particular, our ordering \sqsubseteq^R coincides with the stable ordering on functions. (That the ordering \sqsubseteq^R on configurations x induces the stable ordering on functions \bar{x} can be proved directly without using the fact that $B\mathbb{E}$ is a full subcategory.) In view of this fact we use \sqsubseteq^R for Berry's stable ordering on functions.

It remains to show that each $R(E)$ is a bidomain. Recall from definition 8.2.15 that the one further requirement on $R(E)$ is that in DBIOCPD the identity $1_R(E)$ is the \sqsubseteq^R -supremum of a countable \sqsubseteq^R -increasing chain of finite projections w.r.t. \sqsubseteq^R . We have $1_R(E) = \top_E$. The set $M(1_E)$ is certainly countable; enumerate its elements as $e_0, e_1, \dots, e_n, \dots$. Define $X_n = \leq_1^{M-1} \{e_0, \dots, e_n\}$. Then $\{\overline{X_n} \mid n \in \omega\}$ forms the required chain of projections.

Thus $B\mathbb{E}$ is a cartesian closed full subcategory of BIDOM. ■

Corollary 9.9.9

Products and exponentiations in $B\mathbb{E}$ are isomorphic to the products and exponentiations, respectively, in BIDOM. In particular the configurations x in $R(E_0 \rightarrow_s E_1)$ are in 1-1 correspondence with the functions $R(E_0)$ to $R(E_1)$ in BIDOM with \sqsubseteq and \sqsubseteq^R on configurations inducing Berry's extensional and stable orderings on functions.

9.10 Sequential configurations

We have seen how stable event structures determine a full subcategory of bidomains. Thus they yield a stable model for PCF.

Can the method using event structures be refined to construct a fully abstract model of PCF? The definition of suitable event structures and configurations of them must capture the sequential evaluation of PCF; it is hoped that then a fully abstract model will result. This approach has some promise as the results of this section show.

Although we have largely worked with \leq -left closed sets as configurations x it turned out that the \leq^L -maximal elements $M(x)$ could themselves be regarded as another form of configuration. It is this form of configuration which captured the operational behaviour more closely. We noted that all the work of this chapter on stable event structures could be based on a definition of a stable configuration which determined subsets of the form $M(x)$. It is an interesting fact then we can define stable configurations $(M(x))$ as subsets y such that

(i) $\forall e \in y \forall e' \leq^R e \exists e'' \in y \ e' \leq^L e''$ and (ii) y is Δ -consistent where Δ is inherited up the types by $(x, e) \Delta (x', e')$ iff $x \uparrow^R x' \ \& \ e \Delta e'$. Thus the ordering \leq^L is involved in the enabling but need not be mentioned explicitly in the conflict relation.

It is hoped that by adding axioms to 9.2.1 and refining the definition of configuration a category of sequential event structures with sequential configurations can be formed. To capture the operational flavour it seems best to work with the configurations $M(x)$. They should be secured as in (i) above and consistent in some sense. Consistency is open. Firstly we cannot get away with a simple binary relation like Δ as the example 8.2.5 shows. Rather we must work with a consistency relation. There is a chance that it need not explicitly mention \leq^L and be inherited up the types in a way only mentioning \leq^R .

The following modest results at first order add some faith to this approach.

Lemma 9.10.1

Let A and B be concrete domains. Suppose f is a continuous function from A to B . Then f is Kahn-Plotkin sequential iff

$$(*) \ \forall Z \subseteq A \ \forall q \in d(f(\prod Z)) \ ((\forall z \in Z \ f(\prod Z) \underset{q}{\prec} f(z)) \Rightarrow (\exists p \in d(\prod Z) \ \forall z \in Z \ \prod Z \underset{p}{\prec} z)).$$

Proof

" \Rightarrow " Suppose $Z \subseteq A$ and $q \in d(f(\cap Z))$. Assume $\forall z \in Z f(\cap Z) \not\prec_q f(z)$. If Z is null it is obvious so assume Z is non-null. Then from the definition of sequential for some p in $d(\cap Z)$ we have $\forall x \ni \cap Z f(\cap Z) \not\prec_q f(x) \Rightarrow \cap Z \not\prec_p x$. Thus by the assumption on Z we have $\forall z \in Z \cap Z \not\prec_p z$ as required.

" \Leftarrow " Assume (*) above. Suppose $x \in A$ and $q \in d(f(x))$ and that $\exists z \supset x f(x) \not\prec_q f(z)$. Then define $Z = \{z \supset x \mid f(x) \not\prec_q f(z)\}$. It is non-null. We have $x \in \cap Z$.

If $x = \cap Z$ we have $f(x) = f(\cap Z)$ so by (*) above $\exists p \in d(x) \forall z \in Z x \not\prec_p z$. Thus $\exists p \in d(x) \forall z \supset x (f(x) \not\prec_q f(z) \Rightarrow x \not\prec_p z)$ as required.

If $x \subset \cap Z$ then $x \prec x' \subseteq \cap Z$ for some x' . Take $p = [x, x'] \approx$ i.e. take p to be a direction at x filled by x' . Then $\forall z \in Z x \not\prec_p z$ so by the definition of Z we have $\forall z \supset x f(x) \not\prec_q f(z) \Rightarrow x \not\prec_p z$ as required. ■

Proposition 9.10.2

Let $E_i = (E_i, \leq_i, \otimes_i)$ $i = 0, 1$ be event structures so that $\mathcal{L}_e(E_i)$ ($i = 0, 1$) are distributive concrete domains. Define $E_0 \rightarrow E_1$ to be the event structure consisting of events $\mathcal{L}_e(E_0)^0 \times E_1$ ordered by $(x, e) \leq^L (x', e')$ iff $x' \subseteq x$ & $e = e'$
 $(x, e) \leq^R (x', e')$ iff $x' = x$ & $e \leq e'$
 with this consistency relation:

$\text{con}\{(x_i, e_i) \mid i \in I\}$ iff $\forall J \subseteq I \{e_j \mid j \in J\}$ occupy the same direction implies either (i) $\exists d \forall j \in J \cap_{j \in J} x_j \not\prec_d x_j$
 or (ii) $\forall j, k \in J (x_j, e_j) = (x_k, e_k)$.

For y a subset of $E_0 \rightarrow E_1$ say y is a configuration
 iff (i) $\forall e \in y \forall e' \leq^R e' \exists e'' \in y e' \leq^L e''$ (y is secured)
 and (ii) $\text{con}(y)$ (y is consistent)

Then y is a configuration iff y is a sequential function:

$\mathcal{L}(E_0) \rightarrow \mathcal{L}(E_1)$.

(The proof uses the above lemma. In its present state it is inelegant and uninformative, so omitted.)

Chapter 10 Conclusion

In conclusion we summarise the achievements, problems and inadequacies in the work presented here. The inadequacies should guide us in future work to a more complete theory of events in computation.

10.1 Achievements

The unifying role of events has been apparent in this thesis. Even at the most superficial level, the number of introductory chapters, necessary to its development, is a testimony to this. The approach relates to some degree the theories initiated by Petri and Scott and some more specialised work of authors like Kahn and Plotkin, Berry, Lamport and Hewitt. The thesis provides an introduction to apparently diverse fields through following a common theme, the fundamental part played by events in computation.

We have seen how nets, and thus event structures, model computations and receive definite interpretations (section 2.3). In particular this highlighted when extra structure was called for and exhibited the nature of computation, for example, how datatypes were involved in the process of computing.

Through new representation results we linked and compared theories. This established some concepts in common and some points of divergence. In particular it cast the thesis of Petri ("real processes determine K-dense causal nets") and, admittedly far less thoroughly, the thesis of Scott ("computable functions are continuous") in the new light of an event-structure setting. Event structures inject a new venom into the theories of nets and of denotational semantics; for net theory it is a more abstract approach to foundations and for denotational semantics a way of incorporating ideas of behaviour more completely. Specifically we contribute mathematical ideas on states, conditions, expressiveness and extra structure to the foundations of net theory while to denotational semantics we provide more physical realisations of its ideas with some promise of solving full-abstractness problems.

10.2 Problems

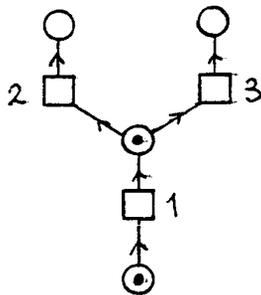
Here are listed some mathematical problems left unsolved in this thesis.

1. (End of section 3.3)

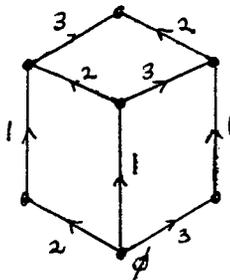
Axiomatise the class of domains represented by event structures of the form (E, \vdash, \mathbb{X}) defined as in 3.3.1 but now with $\mathbb{X} \subseteq \mathcal{Y}(E)$ (the set of finite subsets of E); configurations x are secured as in 3.3 and consistent in a new sense: $\forall A \in \mathbb{X} A \not\subseteq x$.

Subsequently axiomatise the classes of domains Dom_n ($n \in \omega$) represented by event structures of the form (E, \vdash, \mathbb{X}) as above but with restriction: $\forall A \in \mathbb{X} |A| \leq n$. (Note we have represented the domains Dom_2 as then the incompatibility predicate can be replaced by a binary conflict relation).

It might be thought that event structures of the form above relate to transition nets where more than one token may reside on a condition $[N, P, \tau]$. However the domains represented by such event structures satisfy axiom C while those represented by such nets do not, for example:



This time conditions may carry more than one token so although events 2 and 3 cannot occur together initially, they can after event 1 has occurred. The appropriate domain is



which fails axiom C. What is the representation result for domains represented by such nets?

2. (Section 6.3)

Are the expressiveness relations \leq and \leq_1 the same on condition-extensional occurrence nets of finite-depth and satisfying N3?

Let E be an event structure of finite depth. For b in $\mathcal{B}(E)$, characterise those subsets X of $\mathcal{B}(E)$ which satisfy

$$\forall C \in OS(E) \text{ (on } (b, C) \Leftrightarrow \exists ! b' \in X \text{ on } (b', C)).$$

(Such sets X arise for the expressiveness relation \leq_1 - see 6.3.4).

3. (Section 6.4)

Characterise the relation \models (of 6.4.2), for restless events.

4. (Section 7.3)

Can the reachability classes be axiomatised neatly, without using a direct domain analogue of the metric?

5. (Chapter 9)

Can the work of chapter 9 be mimicked for exponentiation corresponding to all continuous functions while maintaining identical definitions of M and \leq^R so that \leq^R is still natural as an ordering on behaviours? (This will involve appropriate axioms on orderings \leq^L and \leq^R)

6. (Section 9.7)

Is the relation \leq (defined in 9.7.3) on all stable configurations a partial order? (It is when restricted to isolated configurations by 9.7.4).

7. (Section 9.8)

What were the key event-structure facts which enabled us to construct a cartesian-closed category of event structures in section 9.8?

8. (Section 9.9)

Axiomatise the domains in RE .

9. (Section 9.10)

Can the full-abstractness problem for PCF be solved on the lines indicated in section 9.10? If so, is there a syntactic characterisation of \subseteq^L ?

10.3 Future work

This thesis has demonstrated the fundamental and unifying role of events in computation. However here is presented only the beginnings of a reasonably complete theory of events; while indicating the scope and depth of such a theory there are several counts on which our work is inadequate or incomplete. This is due, in part, to its exploratory nature and our attempts to relate different approaches. Though there is still a fair deal to be done at this general level much should be learnt by trying to solve specific more well-defined problems within the framework of event structures. Of course solutions to well-chosen problems can throw light on the theory overall. We sketch some future projects. They have various degrees of openness as sometimes basic concepts involved have yet to be formalised to give the problems a strictly mathematical nature.

It would be very satisfying if the full-abstractness problem can be solved on the lines suggested by chapter 9. We need a far clearer understanding of the peculiar \leq^L and \subseteq^L orderings. From our work the configurations of \leq^L -maximal events look like the objects to study. Even if this fairly direct approach fails the approach of Berry and Curien [Ber and Cur] may well succeed and it uses event-structure concepts. At present they have a cartesian-closed category of concrete domains with algorithms as morphisms. Though this does not yield an extensional model they hope to achieve extensionality by a form of quotienting. If successful they will be essentially mapping algorithm configurations (with extra control events) to function configurations which should determine the definitions of higher type event structures and configurations appropriate to PCF, as well as providing some ideas on event structure morphisms.

Another major project is to link-up net and event-structure

ideas with those in Milner's book [Mil]. A prerequisite for replacing synchronisation trees by event structures will be some more general definition of observational equivalence; without it synchronisation trees suffice, as Milner shows. If successful this might yield a mathematical justification of Petri's ideas on the fundamental role of the concurrency relation in parallel computations.

A major inadequacy of the work presented here has been the omission of a systematic treatment of event-structure morphisms. We have seen how to formalise some idea of implementing one event structure by another (5.3) and how to regard one event structure as a datatype involved in another (5.6) using the relations \leq , \leqslant . The relations are close to morphisms. In chapter 7 we used the idea of collapsing a convex subset of events to an event, again suggesting morphisms. In chapter 9 morphisms arose in a different way; they represented continuous functions, essentially by introducing extra causality relations between event structures. All this should be unified. Then for example one might settle the question of whether or not an event structure is physically feasible by demonstrating that it can or cannot be implemented by one which clearly is. (This is like the resultsⁱⁿ 5.3 which showed that being implemented by a finite-width event structure induced restrictions, like countability for instance). Another example: One would expect that event structures of the form (E, \vdash, \otimes) would be "generated" by morphisms from a basic class of the form (E, \leq, \otimes) which assume an event is caused in a unique way. As the definition of observer stands (5.1), time is in a sense outside the theory. Should we not regard recording time-of-occurrence as a computation based on modelling a clock as a process? Then observers themselves would be morphisms within the theory of event structures. Unfortunately many ideas of morphism depend for their naturalness on event structures having additional structure, for example to ensure certain events occur.

Here are some cases where event structures must possess additional structure if they are to model correctly. We have seen how some new idea is needed to distinguish situations where something (like an event occurrence) is inevitable from other

situations (§2.3 and §6.4). A careful modelling of Milner processes on the lines of 2.3A should help clarify things. More speculatively it might be informative to study episodes (see the introduction) which are events without the atomicity restraint; they are a bit like critical regions. And, how can event structures be generalised to continuous processes like example 5.6.5? Perhaps ideas like those of Cardelli [Car] might guide and motivate such a study. Suitable mathematics might be [Nac] and [Cts:at].

REFERENCES

(In these references, LNCS n stands for Lecture Notes in Computer Science, Vol. n, Springer Verlag).

[Arb] M.A. Arbib, E.G. Manes, "Arrows, structures and functors " ACADEMIC PRESS, INC., 1975.

[Bac] J. Back, "Semantics of unbounded non-determinism" ICALP 80 , LNCS 85, 1980.

[Ber] G. Berry, "Modèles complètement adéquats et stable des λ -calculs typés", These de Doctorat d'Etat, Université Paris VII, 1979.

[Ber and Cur] G. Berry, P.L. Curien, "Sequential algorithms on concrete data structures", to appear in TCS.

[Bir] G. Birkhoff, "Lattice theory" in Coll. Pub., vol. 25, Amer. Math. Soc., Providence, R.I., 3rd edition, 1967.

[Bes] E. Best, "The relative strength of K-density" in [N.proc].

[Car] L. Cardelli, "Analog Processes", To appear in Proc. 96 MFCS, Poland 1980.

[Cts. lat.] G. Gierz, K.M. Hofmann, K. Keimel, M. Mislove, D.S. Scott, "A compendium of continuous lattices", to appear.

[Cur] P.L. Curien, "Algorithmes sequential sur structure de données concrètes", Thèse de troisième cycle, Université Paris VII, 1979.

[Gen] H. Genrich, K. Lautenbach, "The analysis of distributed systems by means of predicate/transition-nets", Semantics of Concurrent Computation, Evian, LNCS 70, 1979.

[Gor] M.J.C. Gordon, "The denotational description of programming languages", Springer Verlag, 1979.

[Gra] G. Grätzer, "Lattice theory", W.H. Freeman and Co., San Francisco, 1971.

[Hen] M. Hennessy and R. Milner, "On observing non-determinism and concurrency" ICALP 80, LNCS 85, 1980.

[Hew] C. Hewitt, H. Baker, "Actors and continuous functionals" in: Formal Description of Programming Concepts (ed. E. Neuhold), North Holland, 1978.

[Jen] K. Jensen, "Coloured Petri nets and the invariant method" to appear in TCS.

[Kah and Plo] G. Kahn, G. Plotkin, "Structures de données concrètes", IRIA-LABORIA Report 336, 1978.

[Kah and Mac] G. Kahn and D. MacQueen, "Coroutines and networks of parallel processes", Proc. IFIP Congress, North Holland, 1977.

[Lam] L. Lamport, "Time, clocks and the ordering of events in a distributed system", CACM 21, 1978.

[Mac] S. MacLane, "Categories for the working mathematician", Springer-Verlag, 1971.

[Mil 1] R. Milner, "A calculus of communicating systems", to appear as a LNCS 92, 1980.

[Mil 2] R. Milner, "Fully abstract models of typed λ -calculi", TCS 4, 1977.

[Mos] Y.M. Moschovakis, "Elementary induction on abstract structures", Studies in Logic, Vol. 77, North Holland, 1974.

[Nac] L. Nachbin, "Topology and order", Van Nostrand Mathematical studies, 1965.

[N. proc.] ed. by W. Brauer, "Net theory and Applications", Proceedings of the Advanced Course on General Net Theory of Processes and Systems, LNCS 84, 1979.

[Nie] M. Nielsen, G. Plotkin, G. Winskel, "Petri nets, event structures and domains", Proc. Conf. on Semantics of Concurrent Computation, Evian LNCS 70, 1979. (Also to appear in TCS as one of a selection of papers from the Evian proceedings).

[Par] D. Park, "On the semantics of fair parallelism" Internal report, University of Warwick.

[Pet 1] C.A. Petri, "Non-sequential processes", GMD-ISF Report ISF-77-05, 1977.

[Pet 2] C.A. Petri, "Concurrency as a basis of systems thinking", GMD-ISF Report ISF-78-06, 1978.

[Plo 1] G.D. Plotkin, "LCF considered as a programming language", TCS 5, 1977.

[Plo 2] G.D. Plotkin, "A powerdomain construction", SIAM J. Comp 5, 1976.

[Rog] H. Rogers, Jr. "Theory of recursive functions and effective computability", McGraw-Hill Series in Higher Mathematics, 1967.

[Sac] R.K. Sachs, H. Wu, "General relativity for mathematicians", Springer Verlag, 1976.

[Sco] D.S. Scott, "Datatypes as lattices", SIAM J. Comp. 5, 1976.

[Smy 1] M. Smyth, "Effectively given domains", TCS 5, 1977.

[Smy 2] M. Smyth, "Powerdomains", JCSS16, 1978.

[Wad] C.P. Wadsworth, "Semantic domains for programming languages", Prentice Hall, 1981.