Mobile Processes in Bigraphs

Ole Høgh Jensen

October 2006

Abstract

Bigraphical reactive systems (BRSs) are a formalism for modelling mobile computation. A *bigraph* consists of two combined mathematical structures: a *place graph* that models locality, and a *link graph* that models connectivity. The computation, or behaviour, of systems is modelled as *reaction rules* that prescribe how bigraphs can change dynamically. A notion of bisimulation exists for BRSs, based on a notion of *contextual transitions* derived uniformly from any given set of reaction rules.

In this dissertation we make several extensions to the theory of bigraphs and give two substantial applications, demonstrating how two of the main existing models of mobile computation, the *π-calculus* and the *ambient calculus*, can be accurately represented as particular instances of BRSs.

One of our extensions is *place-sorting*, a refinement of bigraphs that allows certain structural constraints to be imposed. As another extension we refine the existing notion of (strong) bisimilarity into *weak bisimilarity*, closely following the spirit of the similar well-known concept in process calculi; we show how, for contextual transitions, weak bisimilarity can be based elegantly on a method for composing reaction rules. Our third main extension to the theory of bigraphs concerns the adequacy (for proving bisimilarity) of a subclass of the contextual transitions known as *engaged transitions*; we extend previous adequacy theorems to cover a wider range of BRSs, in particular including certain BRSs in which sub-systems can be replicated dynamically.

Our first application of bigraphs demonstrates how (several versions of) the π-calculus can be modelled in a way that accurately represents both the structure and the dynamics of processes. We investigate the contextual transitions (and accompanying bisimilarity) induced on the calculus by the bigraph model, and in particular we compare them to the original transition systems and bisimilarities defined specifically for the calculus, finding substantial (though not complete) agreement. We repeat the exercise for the ambient calculus (mobile ambients), obtaining similar results.

Preface

The work presented in this dissertation was carried out under the supervision of Robin Milner. His influences are manifold: He originated bigraphs and has been central in founding and furthering concurrency and mobility as research areas. He has guided with invaluable inspiration, significant suggestions, attentive advice, and kind criticism. And when the work dragged out unbearingly, his patience and perseverance has been beyond any reasonable expectation.

For all of this, Robin: Thank you.

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. It is not, as a whole or in any of its parts, substantially the same as any work that I have submitted for a degree, diploma, or other qualification at any other university. It does not exceed 60,000 words.

Contents

Chapter 1

Introduction

Information technology pervades our lives. Not only are computers used extensively for tasks such as calculation and word-processing—because they are interconnected, they are becoming at least as important for a growing range of tasks involving communication and information-gathering. On top of that, computer technology is increasingly embedded into all manner of man-made systems not usually associated with computers as such: cell-phones, car engines, factory machinery—to mention but a few, diverse examples.

Thus, computers not only *compute* in the sense of performing calculations; they also *interact*. They interact with people whom they serve, with devices that they monitor and control, and with other computers that they cooperate with. As computer technology proliferates and becomes ubiquitous, the need for interaction only increases: to exchange information, to spread workload, to coordinate the use of resources. Moreover, as wireless devices are carried around, or scripts shipped across the Internet, systems are increasingly required to be flexible and instantly reconfigurable in response to their ever-changing environments. Our systems are not only interactive; they are also *mobile*.

A single computer, say an office PC, is an enormously complex system. Its individual components, such as the CPU, the operating system, or an individual application program, each represent large efforts of hardware or software engineering. Like all engineering of complex systems, these engineering disciplines rely on science and mathematics in order to understand and predict the behaviour of the systems being developed. Computer science, though hardly a mature science, copes well with the challenges involved in building and programming the individual computer. It focuses largely on computation as a sequential process, a series of data manipulations leading toward a final result, and provides theories to reason about the correctness of results and the efficiency with which they are obtained.

However, as computer systems integrate across networks to form what is sometimes described as the *global computer*, and as the technology integrates with physical devices, becoming the *vanishing computer*, systems are growing

even more complex, and the focus is shifting from sequential computation towards computation with any number of processes that run concurrently and engage in complex patterns of interaction and mobility. At present, the theoretical tools for describing and reasoning about such systems are much less well-established than those that apply to sequential computation, and as a result these systems are often developed and deployed without a thorough understanding of how they work—and why they fail.

Providing the proper tools—a scientific theory of interactive and mobile systems—is the broad context in which this dissertation aims to make a contribution. More particularly, we shall study, and extend, a particular formalism called *bigraphs* that has been put forward as a basis for such a theory.

In the present chapter, we give some background to theories of interacting and mobile processes (Sections 1.1–1.3), before introducing bigraphs themselves in Section 1.4. In Section 1.5 we state the purpose of the dissertation more specifically and provide an overview of the main chapters. The present chapter concludes with Section 1.6 that discusses related work.

1.1 Modelling concurrency

Reasoning about the properties of a system, whether it actually exists or is merely contemplated, requires a *model* of the system—a representation in a suitable formalism that allows the application of relevant theories. For example, mechanical systems are typically modelled in terms of idealized components like springs and rigid bodies, and reasoned about using the theory of Newtonian mechanics that applies to such components; similarly, electrical circuits can be modelled in terms of resistors and capacitors, etc., and reasoned about using the theory of electrodynamics.

Computation is typically modelled as a discrete process—a sequence of events that represent individual computation steps. Many different formalisms exist, differing in the features they represent and the theories they accommodate. Especially in the area of interactive and mobile systems the range of formalisms is wide and the relations between them not always clear. This dissertation is concerned with *bigraphs*, a recently proposed formalism that attempts to act as a unifying framework in which many existing formalisms can be seen as special cases.

What makes a good modelling formalism? A model should be—in the words of Einstein—"as simple as possible, but no simpler." The "no simpler" part requires that the formalism can represent all the features that are relevant to the particular properties we want to reason about. The "as simple as possible" part requires, on the other hand, that other features can be omitted. Focusing on relevant features requires *abstraction* and *compositionality*: when reasoning about a large system we should be able to "zoom out" from the details of its components and perform the reasoning in terms of suitably abstract properties of the components; conversely, we should be able to establish these component properties by "zooming in" on each individual component and rea-

soning about it as a system of its own. Thus, a good modelling formalism, apart from being well-chosen for the specific kind of system and properties at hand, should be general enough to encompass different levels of abstraction, and it should support compositional reasoning by providing primitives for describing the system structurally as a combination of components.

One of the earliest formalisms that explicitly models concurrency and interdependence of events is *Petri nets* [31]. Many variants of this formalism exist, but basically a Petri net takes the form of a directed hypergraph whose nodes we call *conditions* and whose edges we call *events*. A *marking* of a net assigns to each node a natural number, interpreted as a number of *tokens* residing at that node. An event can *fire* if each of its source nodes (its *preconditions*) holds at least one token; when the event fires these tokens are removed and a token is added to each of its target nodes (its *post-conditions*).

A Petri net can express several degrees of dependency among events. For example, if two events are in different, unconnected regions of the net, then any firings of these events are completely independent. Another possibility is that a post-condition of one event is a pre-condition of another; then there is a sequential relationship, in that a firing of the first event may trigger the firing of the second.

Yet another possibility is that two events share a pre-condition, in which case there may be *contention* among the two events, which means that either may fire but the firing of one will preclude the firing of the other. This last situation signifies *non-determinism*, as the model does not specify how the choice between the two events is resolved. Non-determinism is often a highly desirable feature in a modelling formalism, even when deterministic systems are modelled, because it allows abstraction of the details involved in choice-making.

The dependencies among events in nets can be captured more abstractly (without explicitly representing conditions) in *event structures* [29].

Petri nets are a special form of *rewrite systems*, a model of computation based on replacing parts of systems according to specified *rewrite rules*. Each rule consists of a *redex* r and a *reactum* r'; it specifies that r, whenever it occurs as a sub-system, can be replaced by r' as a single step of computation. In a Petri net, rewriting is performed on markings. More general forms of rewriting have been studied in the context of *term rewriting* [43], *graph rewriting* [34], and *rewrite logics* [19]. Rewriting is also variously referred to as *reduction* or *reaction*; in this dissertation we generally use the latter. We write $P \to P'$ to denote a reaction step leading from the configuration P to P'.

Rewriting, or reaction, represents the *dynamics* of a system—its evolution over time—and as such forms an *operational semantics*. Such a semantics is useful for analysing many important properties, such as whether a particular (desirable or undesirable) configuration of a system can be reached. However, it does not support compositional reasoning, as the reaction behaviour of a whole system does not typically relate directly to that of its parts. For example, in a model of communicating entities, it is natural to represent message exchange as reaction; this means that an originator of a message and a recipient can engage in a reaction (by exchanging the message), although neither can perform

a reaction on its own.

Compositionality is addressed in *process calculi*, a family of models of concurrent computation originating with Milner's CCS [22] and Hoare's CSP [11]. A process calculus is based on a formal language for the expression of process structure; the exact primitives vary between specific calculi but inevitably include some form of *parallel composition*. Given processes P and Q, their parallel composition, usually written $P \mid Q$, is a more complex process that can be thought of as consisting of P and Q "side by side" so that they can interact with each other and also act independently.

More generally, if a process P occurs as a sub-process of a larger process S, then we can think of S as consisting of P together with a *context* C. The context can be represented as a process expression with a hole, such that the result of plugging P into the hole, usually denoted by $C[P]$, yields the original process S. The composite process $C[P]$ will typically have reactions that arise from interaction between P and C. Thus, compositional reasoning requires knowledge of the ways in which P may "contribute" to such interaction. These contributions are commonly referred to as *observable actions*, taking the view that observing the behaviour of a process is the same as interacting with it.

Many process calculi use a form of operational semantics known as *labelled transitions* that represents observable actions. A labelled transition $P \xrightarrow{\ell} P'$ expresses that P can evolve to P' by performing an observable action represented by the label ℓ. The specific structure and interpretation of labels depends on the calculus in question. The transition relation is normally defined compositionally; that is, the labelled transitions of a process can be inferred from the labelled transitions of its sub-processes. For example, CCS has (amongst others) the inference rule

$$\frac{P \xrightarrow{\alpha} P' \qquad Q \xrightarrow{\bar{\alpha}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

which says that by performing complementary actions α and $\bar{\alpha}$ the two processes P and Q can together perform a "complete" reaction step, represented by a transition with the distinguished label τ that represents a *silent* (i.e., unobservable) action.

Based on the notion of labelled transitions a large number of more abstract semantics have been proposed to capture the notion of observable behaviour, amongst others traces, failures [11], testing [6] and bisimilarity [30, 22]. Many of them take the form of equivalences or preorders over processes, expressing when the behaviours of two processes should be considered identical, or, in the case of preorders, when the behaviour of one process should be considered a refinement of that of the other. The variety among the semantics arises from differences in the treatment of, among other things, non-determinism and silent actions.

There is also some variation in how well the various semantics support compositional reasoning. In the case of equivalences, we ideally want a notion of equivalence that is a *congruence* with respect to all process constructors

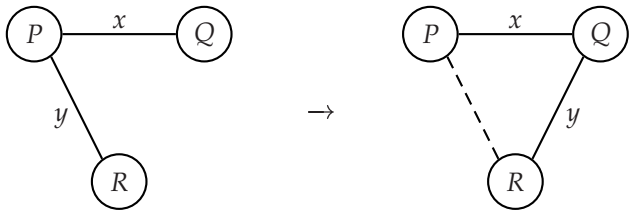


Figure 1.1 Example of mobility in the π-calculus

in a particular calculus (or at least important ones like parallel composition). Congruence is a desirable property because it allows us to "substitute equals for equals", that is, to replace any sub-process by a behaviourally equivalent sub-process without changing the behaviour of the overall system in which they occur. Achieving compositionality is a major challenge in the area of process calculi, as it generally depends on a delicate interplay between, on the one hand, the particular constructions allowed by the process language, and on the other, the distinctions made by the semantics.

1.2 Modelling mobility

The first model of computation specifically intended to model mobile computation is the *π-calculus*, a process calculus developed by Milner, Parrow and Walker [27] as, essentially, an extension of CCS. Reaction in the π-calculus represents binary communication in which the communicated data are themselves (names of) communication channels. This allows mobility to be represented, e.g. as illustrated in Figure 1.1.

The diagram on the left in the figure shows a system consisting of three processes P, Q and R, where P and Q share a communication channel named x, and P and R share a channel named y; there is no channel shared by Q and R, so these two processes cannot communicate directly. The diagram on the right in the figure shows the result of a reaction in which the name y is communicated (along x) from P to Q. After this reaction all three processes share the channel y, but supposing P makes no further use of y we can regard the link between P and R (shown dashed) as being lost. Thus, the reaction has caused the link with R to move from P to Q.

This seemingly very restricted form of dynamics turns out to be surprisingly powerful, as evidenced by the wide variety of phenomena that have been modelled naturally in the calculus; examples can be found both within computer science (e.g. data structures [23] and concurrent objects [32]), and in other areas (e.g. biochemical systems [33] and business processes [41]).

From the point of view of mobility, too, the dynamics may also seem rather restricted, as it allows only communication channels to move. However, referring to the example above, there is no clear distinction between moving y—the only channel providing access to R—and moving R itself. Indeed, results by

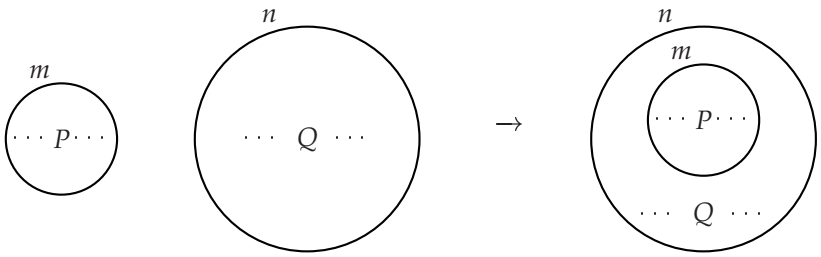


Figure 1.2 Example of mobility in mobile ambients

Sangiorgi [35] indicate that the *higher-order π-calculus*, a version of the calculus where the communicated data are processes, can be encoded naturally and faithfully in the π-calculus itself.

Partly, the lack of a clear distinction between the two notions of mobility can be explained by the fact that the calculus has no explicit notion of *location*. Since movement is change of location, the concept is central in any model of mobile systems. In the π-calculus a model of a distributed system (i.e. a system with locations) might represent the locations as special processes (along with other processes that represent other entities). Several extensions have been proposed that add explicit locations to the π-calculus, e.g. [10, 5]; they all allow some form of process migration between locations and varying degrees of communication between co-located or remote processes.

A somewhat different location model is offered by the *ambient calculus* (also called *mobile ambients*) [4], where locations, referred to as *ambients*, are organized hierarchically, and moreover it is ambients, rather than individual processes, that form the units of migration. The hierarchical location structure mirrors many structures that occur in practice, such as administrative domains (e.g. on the Internet) or the block-structure of many traditional programming languages.

The ambient calculus has three forms of reaction: two that allow movement respectively into and out of an ambient, and one that allows an ambient to dissolve, releasing its content into the surrounding ambient; Figure 1.2 illustrates a reaction in which process P performs the action *enter* n, causing its own ambient (named m) to migrate into the neighbouring ambient n.

There have also been proposals for adding mobility to Petri nets [1, 3]. The idea is to equip tokens with extra data (names) which control how certain reconfigurations of the net topology are performed when events are fired. Obviously, there is a wide range of choices in exactly which forms of dynamical reconfigurability to allow, and it is by no means clear, at a first glance, how to make this choice in such a way that a tractable and suitably expressive notion of mobility is obtained. Remarkably, however, the cited work obtains an exact correspondence with the dynamics of the join-calculus [7], a process calculus that has been developed as, essentially, a restriction of the π-calculus and moti-

vated by considerations for efficient implementation as a distribution primitive for programming languages.

1.3 Process structure and dynamics

The π-calculus inherits most of its process combinators from CCS, and like CCS it has a labelled transition semantics, although its transition rules are considerably more complicated. In other calculi, for example mobile ambients, it is even less clear how to give a satisfactory and reasonably simple semantics in this style. For this reason it has become usual to take a reaction semantics as the primary definition of dynamics and, to some extent, to rely on alternative means for reasoning compositionally about behaviour. One such alternative is Milner and Sangiorgi's notion of *barbed congruence* [28].

It is common to define the reaction relation not over syntactical process terms but instead over a quotient obtained by factoring out a congruence usually referred to as *structural congruence*. The concept, introduced by Milner [23] and strongly inspired by the *chemical abstract machine* [2], can be thought of as a way to remove certain "accidental" distinctions introduced by the textual syntax; for example, one usually asserts commutativity of parallel composition, i.e. the equality $P \mid Q \equiv Q \mid P$, as well as equality of alpha-equivalent terms, i.e. terms that differ only by the choice of bound names. Using structural congruence, the definition of reaction can usually be given as one or a few very simple rules.

Arguably, the simplification in defining reaction comes about because the quotiented system represents *process structure* and leaves the reaction relation to deal purely with dynamics. With the quotient, one moves from a textual syntax toward a graphical syntax; indeed a diagram such as that shown in Figure 1.1 suggests that the processes shown are unordered and thus presupposes commutativity of parallel composition. With the increasing emphasis on topographical aspects demanded by the prospects of tackling global computation, it is reasonable to expect that graphical intuitions for process structure will even become increasingly important.

1.4 Bigraphs

Bigraphical reactive systems [25] were proposed by Milner as a unifying framework for a large range of models of mobile computation. It uses a graphical process structure that treats locality and connectivity as separate, orthogonal aspects, and it uses a reaction relation, given by a set of reaction rules, to specify dynamics.

A *bigraph* combines two graphical structures: a *place graph* (modelling locality) and a *link graph* (modelling connectivity), both defined over a common set of nodes. A typical bigraph is shown in Figure 1.3; its place graph is represented by the nesting of nodes (ovals) within each other and thus forms a tree;

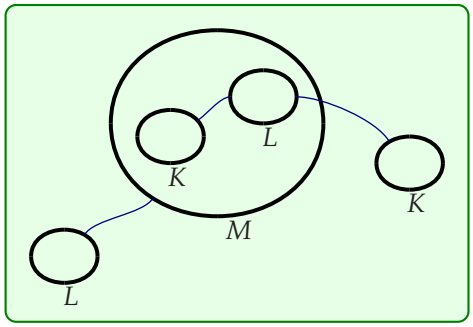


Figure 1.3 Example of a bigraph

its link graph is formed by the arcs between nodes.

Thus, as a formalism, bigraphs are quite general, the only commitment being the two particular graphical structures. The formalism itself implies no particular interpretation of the nodes or of the two structures; such interpretations, along with particular reaction rules, will be given only when the formalism is applied to model a particular class of systems or phenomenon.

Typically, systems contain different kinds of entities that should be distinguishable in the models. Thus, it is often useful to partition the nodes of a bigraph into separate kinds; we do this by equipping each node in a bigraph by a label called a *control*. (In Figure 1.3 the nodes have controls K, L, M). Controls affect the dynamics, as redexes (which themselves are bigraphs) assign particular controls to their nodes. Moreover, controls govern to some degree the formation of the graphical structure, as the formalism allows structural constraints to be imposed that depend on controls. In each particular application, the set of available controls, as well as any structural constraints, are specified by a *signature*; thus a particular bigraphical reactive system (BRS) is specified by a signature together with a set of reaction rules.

In order to support compositional reasoning, bigraphs model not only processes but also process contexts. Instead of making an explicit distinction between these two entities—i.e., introducing both "bigraphs" and "bigraphs with holes"—a more general definition is adopted that equips each bigraph with a multiplicity (called a *width*) on both the inside and outside. An ordinary process is then represented as a bigraph with inner width 0 and outer width 1, and a process context by a bigraph with both widths equal to 1. More generally, the inner width of a bigraph represents its number of holes, and the outer width represents the number of holes it will fill when plugged into a surrounding context.

We can think of the inner and outer widths as a notion of *interface* for bigraphs. (In fact, the full notion of interface incorporates additional information, but at this point we shall not go into the details.) Thus, we talk of the *inner face* and *outer face* of a bigraph. For bigraphs A and B, if the outer face of A coincides with the inner face of B, then the two can be composed by plugging

A into B. This notion of composition gives rise to a category with interfaces as objects and bigraphs as arrows.

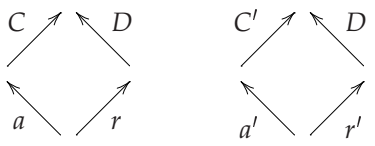
A main challenge in bigraphs is the development of a behavioural theory that supports compositional reasoning. As mentioned above, the standard approach in process calculi is to base behavioural theory on labelled transitions; but transition semantics are traditionally developed in an ad hoc manner for each particular calculus, and in some cases become quite complicated.

In bigraphs we are not presupposing a specific set of reaction rules, and so the behavioural theory must be developed uniformly using only the (unspecified) set of reaction rules as a basis. The central idea for achieving this, which has been a driving force in Milner's work on action structures and action calculi leading up to the present notion of bigraph, is to derive labelled transitions from the reaction relation, taking labels to be small contexts that can "reveal" an observable action. The idea is based on the informal observation that in a typical labelled transition semantics a transition $a \xrightarrow{\ell} a'$ is evidence that a can react with the context by "contributing" ℓ; thus, we expect there is some context C that can contribute the complement of ℓ in order to permit a reaction (or equivalently, silent transition)

$$C \circ a \longrightarrow C' \circ a' \ .$$

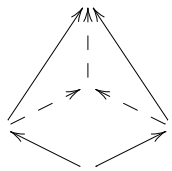
Given only the reaction relation the idea is, essentially, to take this reaction as *defining* a transition $a \xrightarrow{C} C' \circ a'$. Thus we obtain *contextual transitions*, i.e. transitions whose labels are contexts that permit reaction. However, we do not want to include *all* such contextual transitions, as there may be a large number of (arbitrarily large) contexts that reveal "the same" underlying observable action of a; so we ideally would like only a single, preferably small contextual label for each distinct observation.

The latter requirement can be formulated using a variant of the category-theoretic notion of a *pushout*. The key observation is that the above reaction step requires some reaction rule (r, r') and some context D such that $C \circ a = D \circ r$ and $C' \circ a' = D \circ r'$, or equivalently, such that the diagrams



commute. Requiring the square on the left to be pushout amounts to demanding that the arrows C and D are the "smallest" that make the square commute—we can think of it as a least upper bound of the span (a, r). Unfortunately, in the category of bigraphs, many spans do not possess pushouts because, in the terminology of bounds, there is not (even up to isomorphism) a unique least upper bound, but instead there are several bounds that are all minimal but mutually incomparable. By a minimal bound we mean one that

cannot be factored by inscribing arrows as shown by dashed arrows in the diagram below unless the mediating (vertical) arrow is an isomorphism. We give the formal definition of such minimal bounds, called *idempushouts* or *IPOs*, in Chapter 3.



1.5 Purpose and outline of dissertation

In the endeavour to establish bigraphs as a suitable formalism for mobile systems, or perhaps refine them to become one, many different directions must be explored, and it is clear that a behavioural theory based on labelled transitions and behavioural congruence represents just one of those directions. Nevertheless, this is an important direction, as it aims to provide a link with process calculi, the area that arguably has been most successful so far in tackling mobility.

Our purpose in this dissertation, therefore, is to provide accurate bigraphical models of important process calculi for mobility and to study the behavioural congruences induced upon the calculi by the uniform theory of contextual transitions. Our models will represent some of the most substantial applications of bigraphs thus far, and they will require considerable refinements and extensions of the existing work on bigraphs and their behavioural theory. Thus, the work falls in two main parts: one which is involved in developing the general framework of bigraphs, and one which is involved in applying bigraphs to particular models.

Reflecting this division of the work, the dissertation is structured in two parts. Part I, dealing with the general framework, contains Chapters 2–5. Chapter 2 reviews the fundamental and standard concepts of transition systems and bisimilarity; a few non-standard refinements that will be useful for this particular work are also introduced. Chapter 3 reviews the behavioural theory based on contextual transitions; this is done at the level of reactive systems, a generalization of bigraphical reactive systems. In addition, we extend the behavioural theory to also cover *weak bisimilarity*. In Chapters 4 and 5 we define bigraphs and their dynamics, and we review and extend their structural and behavioural theories. Chapter 4 concentrates on structure, whereas Chapter 5 treats dynamics.

Part II gives two main applications of bigraphs. The first is the modelling of (a series of variants of) the π-calculus, and the second is the ambient calculus. It is structured as follows: Chapter 6 gives a brief introduction to the π-calculus itself. In Chapter 7 we give bigraph models for the π-calculus, as a

series of progressively more complete fragments. Chapter 8 studies the transition semantics and corresponding bisimilarity that the uniform theory of contextual transitions induces in this particular instance, and compares this to the standard transition semantics for the calculus. As a second application, the whole exercise is repeated for the ambient calculus; as the approach is similar, the exposition for this second application is condensed into a single chapter (Chapter 9). The conclusion of the dissertation is in Chapter 10.

1.6 Related work

As explained in the preceding sections, the bigraph formalism is strongly inspired by, on the one hand, graph-theoretic approaches to the representation of topographical process structure, and on the other hand, the formal-language-based approach to compositionality via algebraic process structure offered by process calculi. The relationships might be summarized by saying that, from the graph-theoretic point of view, bigraphs offer (1) a particular (but still quite general) structure suited to modelling mobile systems, and (2) a notion of interface and composition that serves as the basis for an algebraic treatment. Similarly, from a process calculus point of view, bigraphs offer (1) a graph-based topographical structure that matches well with structural congruence, and (2) a uniform treatment of the relationship between reaction semantics and transition semantics.

As mentioned, the idea of contextual transitions as a basis of behavioural congruences has existed for some time in the community involved with Milner's action structures and action calculi. Sewell [40] uses the idea to derive labelled transitions for term rewriting systems, and Leifer (in his thesis [16] and in joint work with Milner [18]) studies the issue in the context of their *reactive systems*, a general formalism abstracting away the particular structure of bigraphs. Sassone and Sobociński [39, 42] reformulate this to a version of reactive systems based on 2-categories.

The tile model of Montanari and others [8] generalizes rewrite rules as double-categorical cells (tiles), in which one dimension represents contextual structure (as described above) and the other represents labelled transitions. Thus, tiles describe how contexts transform observable actions (and vice versa), a generalization of Larsen's notion of *action transducer* [14, 15]. Like the bigraphical framework, tiles allow the representation of many different dynamics, but, unlike the present work, they take labelled transitions as primitive.

The usefulness of an algebraic structure has inspired work in both Petri nets and graph rewriting. Winskel [44] advocates a compositional approach to Petri nets based on morphisms between nets, and a similar approach, based on graph morphisms, is widely advocated in graph rewriting, see e.g. [34]. A superficial similarity between these bodies of work and that on bigraphs, is that both employ category theory to some extent. However, the category of bigraphs has interfaces as objects and bigraphs (i.e. structures) as arrows, whereas the other work has structures (nets or graphs) as objects and mor-

phisms between them as arrows. Though a similar category of "uninterfaced" bigraphs and morphisms between them can be defined, and the interfaces of a bigraph can be recovered as certain degenerate morphisms into it, we have not found such a formulation helpful for the present work on bigraphs. In any case, the mentioned work does not address the issue of labelled transitions, which is pivotal for our work.

As mentioned, a main advantage of the contextual transitions for bigraphs is that the associated bisimilarity is guaranteed to be a congruence. In process calculi, where labelled transition relations are defined inductively by sets of inference rules, there has been some work directed at finding conditions, or rule formats, which similarly guarantee the congruence property; see e.g. [9]. Even though the starting points for the transition relations are quite different in the two approaches, there is likely some relation, and it might be interesting to explore the relationship between IPO-based and inductively defined transitions; we do not, however, pursue this question in the present work.

Part I

Bigraphs and their Behavioural Theory

Chapter 2

Transitions and Bisimilarity

The concept of *labelled transition* and the associated notion of *bisimulation* are fundamental to all the work presented in this dissertation. We start in this short chapter by defining these concepts in a manner that suits the way we shall use them in this work. We also review and develop some theory of bisimulation which is motivated by (but not specific to) the behavioural theory of bigraphs; this includes *relative bisimulation*, a notion that arises when working with transition systems that are sub-systems of one another, and *generalized bisimulation*, an adaptation to the present setting of certain proof techniques for bisimilarity.

2.1 Transition systems and bisimilarity

We start by defining transition systems; the definition is a slight generalization of the one that is standard in the literature.

Definition 2.1 (transition system) A *labelled transition system* is a quadruple

$$\mathcal{L} = (A, Lab, app, trans)$$

where

- A is a set of *agents*;
- Lab is a set of *labels*;
- $app \subseteq A \times Lab$ is the *applicability relation*;
- $trans \subseteq app \times A$ is the *transition relation*.

When $(a, \ell) \in app$ we say that the label ℓ *applies to* the agent a. We call an element (a, ℓ, a') of *trans* a *transition*, and we usually write it as $a \xrightarrow{\ell} a'$. ■

The non-standard ingredient in the above definition is the inclusion of the applicability relation and the requirement for each transition $a \xrightarrow{\ell} a'$ that the label ℓ applies to the agent a. The usual kind of transition system is one for

which $app = A \times Lab$; that is, all its labels apply to all its agents. We shall call such a transition system *total*.

The definition of bisimilarity is also essentially the standard one:

Definition 2.2 (bisimilarity) Assume an arbitrary transition system $\mathcal{L}$. A *simulation* is a binary relation $\mathcal{S}$ over agents such that whenever $(a, b) \in \mathcal{S}$ and $a \xrightarrow{\ell} a'$ and ℓ applies to b, then $b \xrightarrow{\ell} b'$ for some b' such that $(a', b') \in \mathcal{S}$. A *bisimulation* is a relation $\mathcal{S}$ such that both $\mathcal{S}$ and its inverse $\mathcal{S}^{-1}$ are simulations. Agents a and b are *bisimilar* if $(a, b) \in \mathcal{S}$ for some bisimulation $\mathcal{S}$. ■

We shall usually denote the bisimilarity relation associated with a transition system $\mathcal{L}$ by $\sim_{\mathcal{L}}$, dropping the index only when $\mathcal{L}$ is understood.

Our definition of (bi)simulation departs from the usual definition only by the extra condition concerning applicability: for a pair (a, b) in a simulation, we only require b to match a transition of a if its label applies to b. When the transition system $\mathcal{L}$ is total, the present definition coincides with the familiar one.

It is easy to show the standard results that bisimilarity $\sim$ is itself a bisimulation (in fact, the largest one), and that it is an equivalence, i.e. it is reflexive, symmetric and transitive. The basic technique for proving a particular bisimilarity result $a \sim b$ is indicated directly by the definition: exhibit a relation $\mathcal{S}$ that contains the pair (a, b) and show that $\mathcal{S}$ is a bisimulation. This proof technique is sound, because, by its definition, bisimilarity includes every bisimulation.

2.2 Relative bisimilarity

We shall sometimes find it useful to define a transition system by "thinning out" a larger transition system, that is, by selecting only a subset of the transitions, typically by restricting to certain classes of agents or certain classes of agent/label-pairs. This gives rise to the notion of *relative bisimilarity* defined in [25]; for completeness, we repeat the definition and give a brief discussion here.

Definition 2.3 (relative bisimulation) We say that a transition system $\mathcal{M}$ is a *sub-transition system* of $\mathcal{L}$, written $\mathcal{M} \preccurlyeq \mathcal{L}$, if $\mathcal{M}$ is component-wise contained in $\mathcal{L}$.

When this holds, a *relative simulation (for $\mathcal{M}$ on $\mathcal{L}$)* is a relation $\mathcal{S}$ over the agents of $\mathcal{L}$ such that whenever $(a, b) \in \mathcal{S}$, then for every transition $a \xrightarrow{\ell} a'$ in $\mathcal{M}$ where ℓ applies to b, there is a transition $b \xrightarrow{\ell} b'$ in $\mathcal{L}$ such that $(a', b') \in \mathcal{S}$. $\mathcal{S}$ is a *relative bisimulation (for $\mathcal{M}$ on $\mathcal{L}$)* when, in addition, $\mathcal{S}^{-1}$ is a relative simulation for $\mathcal{M}$ on $\mathcal{L}$). Agents a and b are *relative bisimilar (for $\mathcal{M}$ on $\mathcal{L}$)*, written $a \sim_{\mathcal{L}}^{\mathcal{M}} b$, if $(a, b) \in \mathcal{S}$ for some relative bisimulation $\mathcal{S}$ (for $\mathcal{M}$ on $\mathcal{L}$).

We say that $\mathcal{M}$ is *adequate (for $\mathcal{L}$)* if $\sim_{\mathcal{L}}^{\mathcal{M}}$ coincides with $\sim_{\mathcal{L}}$ on the agents of $\mathcal{M}$. ■

Note that although the definition uses two different transition systems $\mathcal{M}$ and $\mathcal{L}$, we still require that both of the agents a and b be drawn from the same transition system, namely the sub-transition system $\mathcal{M}$. The extra freedom lies in the matching of transitions: when either agent makes a transition (also drawn from $\mathcal{M}$), the other agent is allowed to make use of the larger transition system $\mathcal{L}$ in performing a matching transition.

Relative bisimilarity is a generalization of the usual notion, as the usual (or, *absolute*) bisimilarity for a transition system $\mathcal{L}$, i.e. the relation $\sim_{\mathcal{L}}$, coincides with relative bisimilarity for $\mathcal{L}$ on itself, i.e. the relation $\sim_{\mathcal{L}}^{\mathcal{L}}$. We shall often drop the qualifier 'relative', when it is clear from the context that we are talking about bisimilarity in the more general sense.

The asymmetry of the matching requirement means that, in general, relative bisimilarity is not transitive and therefore not an equivalence. Thus, it is not in itself directly useful in expressing equivalence of behaviours. Instead, it is useful when $\mathcal{M}$ is adequate for $\mathcal{L}$, for then the smaller transition system $\mathcal{M}$ can be used as the basis of an optimized proof technique for $\sim_{\mathcal{L}}$: In order to show $a \sim_{\mathcal{L}} b$ it is enough to find matching transitions for the subset of $\mathcal{L}$-transitions that are in $\mathcal{M}$, but in constructing the matching transition we still have the freedom of using the larger system $\mathcal{L}$.

Sometimes this freedom is of no use; that is, it may be that every transition in $\mathcal{M}$ can only be matched by another transition in $\mathcal{M}$. A sufficient condition for this is given by the following definition:

Definition 2.4 (definite sub-transition system) For $\mathcal{M} \preccurlyeq \mathcal{L}$ we say that $\mathcal{M}$ is *definite for* $\mathcal{L}$ if $\mathcal{M}$ is determined by a subset $Lab_{\mathcal{M}}$ of the labels $Lab_{\mathcal{L}}$ in $\mathcal{L}$ in the sense that, for an arbitrary transition $a \xrightarrow{\ell} a'$ in $\mathcal{L}$, the transition is in $\mathcal{M}$ iff $\ell \in Lab_{\mathcal{M}}$. ■

In this case, we can immediately deduce that the relative bisimilarity $\sim_{\mathcal{L}}^{\mathcal{M}}$ is an absolute bisimilarity:

Proposition 2.5 If $\mathcal{M}$ is definite for $\mathcal{L}$ then $\sim_{\mathcal{M}}$ and $\sim_{\mathcal{L}}^{\mathcal{M}}$ coincide on the agents of $\mathcal{M}$. If, in addition, $\mathcal{M}$ is adequate for $\mathcal{L}$ then $\sim_{\mathcal{M}}$ and $\sim_{\mathcal{L}}$ coincide on the agents of $\mathcal{M}$. ■

2.3 Generalized bisimulation

As observed above, the basic proof technique for bisimilarity is to exhibit a bisimulation $\mathcal{S}$ that contains the pair of agents we want to show bisimilar. In this section we look at refined proof techniques that aim at reducing the work involved in proving bisimilarity by relaxing the requirement on the relation $\mathcal{S}$. A prominent example is Milner's notion of "bisimulation up to bisimilarity" [22]; for that relation, instead of requiring that matching transitions lead to pairs $(a', b') \in \mathcal{S}$, we only require $(a', b') \in \mathcal{S}^{\sim}$, where $(-)^{\sim}$ denotes closure

under bisimilarity. This proof technique is sound, because it turns out that even the weaker matching requirement implies $\mathcal{S} \subseteq \sim$.

We shall develop a small theory of *generalized bisimulation*, which will encompass several such "up-to" proof techniques. This is essentially an adaptation to the present setting of work by Sangiorgi [37]; we comment on the relationship with that work at the end of the section.

For the remainder of the section we assume fixed, but arbitrary transition systems $\mathcal{L}$ and $\mathcal{M}$ such that $\mathcal{M} \preccurlyeq \mathcal{L}$.

Definition 2.6 (progression) Let $\mathcal{R}$ and $\mathcal{S}$ be binary relations over agents in $\mathcal{L}$. We say that $\mathcal{R}$ *progresses to* $\mathcal{S}$ (for $\mathcal{M}$ on $\mathcal{L}$), written $\mathcal{R} \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{S}$, if, for all $(a, b) \in \mathcal{R}$, whenever $a \xrightarrow{\ell} a'$ in $\mathcal{M}$ and ℓ applies to b then $b \xrightarrow{\ell} b'$ in $\mathcal{L}$ for some b' such that $(a', b') \in \mathcal{S}$. ■

It is clear that bisimilarity (for $\mathcal{M}$ on $\mathcal{L}$), as we defined it in the previous section, can be characterized as the largest symmetric relation $\mathcal{S}$ that progresses to itself (for $\mathcal{M}$ on $\mathcal{L}$).

We note a few immediate properties of the progression relation:

Proposition 2.7

(1) If $\mathcal{R} \subseteq \mathcal{R}' \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{S}' \subseteq \mathcal{S}$, then $\mathcal{R} \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{S}$;

(2) If $\mathcal{R}_i \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{S}$ for all $i \in I$, where I is a possibly infinite index set, then $\bigcup_{i \in I} \mathcal{R}_i \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{S}$. ■

The refined proof techniques will each be represented by some operation $\mathcal{F}$ on relations over agents. Examples of such operations include the familiar *reflexive closure* $(-)^{\mathrm{R}}$ and *transitive reflexive closure* $(-)^*$. The proof technique "up to bisimilarity" mentioned above, is based on an instance of the operation $(-)^{\equiv}$ that closes under an arbitrary equivalence $\equiv$; it is defined as follows:

$$\mathcal{S}^{\equiv} \stackrel{\text{def}}{=} \{(a, b) \mid a \equiv a' \text{ and } b \equiv b' \text{ for some } (a', b') \in \mathcal{S}\} .$$

For operations $\mathcal{F}$ and $\mathcal{G}$, we define their *composition* and *union* as

$$(\mathcal{F} \circ \mathcal{G})(\mathcal{S}) = \mathcal{F}(\mathcal{G}(\mathcal{S}))$$
$$(\mathcal{F} \cup \mathcal{G})(\mathcal{S}) = \mathcal{F}(\mathcal{S}) \cup \mathcal{G}(\mathcal{S}) .$$

For operations $(-)^{\mathsf{a}}$ and $(-)^{\mathsf{b}}$ denoted by superscripts, we write $\mathcal{S}^{\mathsf{a},\mathsf{b}}$ as an abbreviation for $(\mathcal{S}^{\mathsf{a}})^{\mathsf{b}}$.

Definition 2.8 (generalized bisimulation) We say that an operation $\mathcal{F}$ is *allowable* (for $\mathcal{M}$ on $\mathcal{L}$) if $\mathcal{R} \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{S}$ implies $\mathcal{F}(\mathcal{R}) \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{F}(\mathcal{S})$. If $\mathcal{S} \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{F}(\mathcal{S})$ we call $\mathcal{S}$ a *simulation up to* $\mathcal{F}$ (for $\mathcal{M}$ on $\mathcal{L}$). If both $\mathcal{S}$ and $\mathcal{S}^{-1}$ are simulations up to $\mathcal{F}$ (for $\mathcal{M}$ on $\mathcal{L}$), then $\mathcal{S}$ is a *bisimulation up to* $\mathcal{F}$ (for $\mathcal{M}$ on $\mathcal{L}$). If $\mathcal{S}$ is a bisimulation up to some allowable $\mathcal{F}$ then we also call $\mathcal{S}$ a *generalized bisimulation*. ■

The following proposition shows that several basic operations on relations are allowable, and, in the first clause, that they can be combined using composition and union to yield more complex allowable operations. Many other useful operations are allowable only for certain (classes of) transition systems; several such operations will be introduced later, as we define the specific transition systems to which they apply.

Proposition 2.9

(1) If $\mathcal{F}$ and $\mathcal{G}$ are allowable for $\mathcal{M}$ on $\mathcal{L}$, then so are $\mathcal{F} \circ \mathcal{G}$ and $\mathcal{F} \cup \mathcal{G}$;

(2) $(-)^{\mathrm{R}}$ is allowable for $\mathcal{M}$ on $\mathcal{L}$;

Furthermore, if $\mathcal{L}$ is total, then

(3) $(-)^*$ is allowable for $\mathcal{L}$;

(4) $(-)^{\sim_{\mathcal{L}}}$ is allowable for $\mathcal{L}$.

Proof

(1) Follows easily from Proposition 2.7.

(2) Follows from (1) together with the trivial fact that the constant operation that yields the identity relation on agents is allowable.

(3) We must show $\mathcal{R}^* \rightarrowtail_{\mathcal{L}} \mathcal{S}^*$ under the assumption $\mathcal{R} \rightarrowtail_{\mathcal{L}} \mathcal{S}$. Let $(a, b) \in \mathcal{R}^*$. Then $a = c_0 \; \mathcal{R} \cdots \mathcal{R} \; c_n = b$ for some $c_0, \ldots, c_n$. Let $a = c_0 \xrightarrow{\ell} a' = c_0'$. Because we assume that $\mathcal{L}$ is total, the label ℓ applies to each of the c_i, and so by iteration we infer for each $i \geq 1$ a transition $c_i \xrightarrow{\ell} c_i'$ for some c_i' such that $(c_{i-1}' c_i') \in \mathcal{S}$. In particular $b = c_n \xrightarrow{\ell} b' \stackrel{\text{def}}{=} c_n'$, and we have $(a', b') \in \mathcal{S}^*$, as required.

(4) Similar. ∎

To prove bisimilarity (for $\mathcal{M}$ on $\mathcal{L}$) of two agents a and b the idea is to construct a suitable allowable operation $\mathcal{F}$ and a relation $\mathcal{S}$ that contains (a, b), and then prove $\mathcal{S} \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{F}(\mathcal{S})$. This suffices due to the following result:

Theorem 2.10 (generalized bisimulation) Let $\mathcal{S}$ be a generalized bisimulation for $\mathcal{M}$ on $\mathcal{L}$. Then $(a, b) \in \mathcal{S}$ implies $a \sim^{\mathcal{M}}_{\mathcal{L}} b$.

Proof By the assumption we have $\mathcal{S} \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{F}(\mathcal{S})$ for some allowable $\mathcal{F}$. Consider the relation

$$\hat{\mathcal{S}} \stackrel{\text{def}}{=} \bigcup_{n \geq 0} \mathcal{F}^n(\mathcal{S}) \,,$$

where the superscript n denotes repeated application of $\mathcal{F}$. Allowability of $\mathcal{F}$ implies $\mathcal{F}^n(\mathcal{S}) \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \mathcal{F}^{n+1}(\mathcal{S})$, and so by Proposition 2.7 we have $\hat{\mathcal{S}} \rightarrowtail^{\mathcal{M}}_{\mathcal{L}} \hat{\mathcal{S}}$. Thus $\mathcal{S} \subseteq \hat{\mathcal{S}} \subseteq \sim^{\mathcal{M}}_{\mathcal{L}}$, and the result follows. ∎

As stated above, the material in this section closely follows work of Sangiorgi [37], the main difference being our adaptation to relative bisimilarity.

Since relative bisimilarity is not transitive, relation composition does not preserve allowability, unlike the case in Sangiorgi's setting. (Hence, the two last clauses of Proposition 2.9, which implicitly involve relation composition, apply only to absolute bisimilarity, and only to a total transition system.)

Another difference with respect to Sangiorgi's work is that our notion of allowable operation is a simplification of his *respectful functions*: for such a function $\mathcal{F}$ he requires that $\mathcal{R} \subseteq \mathcal{S}$ and $\mathcal{R} \rightarrowtail \mathcal{S}$ implies $\mathcal{F}(\mathcal{R}) \subseteq \mathcal{F}(\mathcal{S})$ and $\mathcal{F}(\mathcal{R}) \rightarrowtail \mathcal{F}(\mathcal{S})$, where our allowability condition leaves out the inclusions. He needs the stronger requirement in order to establish the soundness of an operation that closes a relation under application of certain contexts, an operation that applies in the common situation where the agents are terms in a process calculus. In our setting, we shall also be interested in the context closure operation, but only in connection with our contextual transitions; and here it turns out that the weaker requirement is sufficient. The essential difference is that in our transition systems a transition $C \circ a \xrightarrow{\ell} a'$ will always imply the existence of a related transition of a. When this property does not hold, as is typically the case in process calculi, the inclusion conditions become necessary.

Chapter 3

Reactive Systems and Behavioural Congruence

This chapter reviews reactive systems, essentially as defined in [13], and the behavioural theory based on contextual transitions and strong bisimilarity (also following [13]). In addition, at the end of the chapter, we extend the behavioural theory to include also weak bisimilarity.

The chapter is organized as follows: The first section gives a small sequence of examples that illustrate our notion of contextual transition and also points out that considerable care is needed in order to obtain a suitable structure in which the required IPOs exist. That structure, called s-categories, is presented in Section 3.2. Then Section 3.3 introduces relative pushouts and idempushouts, the formal basis of contextual transitions, which we define in Section 3.4. Sections 3.5 and 3.6 deal with strong and weak bisimilarity, respectively.

3.1 Reaction and transitions

We start by a very simple, non-computational example that illustrates contextual transitions based on IPOs, as described in the introduction. (The formal definition of IPO is given in Definition 3.9.)

Example Let us model a very small ecosystem consisting of two kinds of animals: *predators*, which we shall denote by pred, and *prey animals*, which we shall denote by prey. Dynamics in this system consists of the fact that from time to time a pred may eat a prey. We can model this system as a category with a single object and with arrows being multisets of preds and preys; the identity is the empty multiset, and composition is multiset union. We shall write a multiset $\{a, b, c, \ldots\}$ as $a \mid b \mid c \mid \cdots$ because we think of it as the parallel composition of its elements. A single reaction rule expresses the dynamics:

$$\mathsf{pred} \mid \mathsf{prey} \longrightarrow \mathsf{pred}\ .$$

The category in question has all IPOs. (In fact, IPOs here coincide with the simpler notion of pushout.) They are easy to characterize: the IPO (B_0, B_1) of a span (A_0, A_1) has $B_0 = A_1 - A_0$ and $B_1 = A_0 - A_1$, where minus denotes multiset difference. Thus, typical transitions based on IPOs in this system include

$$\mathsf{pred} \xrightarrow{\mathsf{prey}} \mathsf{pred}$$
$$\mathsf{prey} \xrightarrow{\mathsf{pred}} \mathsf{pred}$$
$$\mathsf{pred} \mid \mathsf{prey} \xrightarrow{id} \mathsf{pred}\ .$$

The first transition is easy to interpret: it expresses that pred, as an agent, can perform the action prey (i.e., eat a prey) and go on being a pred. The second transition expresses that prey can perform pred (i.e., be eaten by a pred). It may seem odd that also in this case the result of the transition is pred—after all, what remains of the agent is nothing! However, with our notion of transition the roles of agent and label are symmetrical, at least in the sense that labels, as well as agents, may leave residuals. In this case, of course, pred is the residual of the label. The third transition expresses *internal activity*: the system that consists of a pred and a prey can perform a reaction that requires no contribution from the environment.

Only rather boring bisimilarities can be proved in this very simple system. For example, we have

$$\mathsf{pred} \sim \mathsf{pred} \mid \mathsf{pred}\ ;$$

this expresses, intuitively, that to an external observer it makes no difference how many preds one has; all that can be observed of any pack of preds is that it eats preys. The proof of the bisimilarity is very simple, as the only transitions of the two agents are

$$\mathsf{pred} \xrightarrow{\mathsf{prey}} \mathsf{pred}$$
$$\mathsf{pred} \mid \mathsf{pred} \xrightarrow{\mathsf{prey}} \mathsf{pred} \mid \mathsf{pred}\ ;$$

thus, the agents are not only bisimilar, they in fact have isomorphic transition systems. ■

Unfortunately, a simple categorical structure, like the category of multisets used in the preceding example, is generally not sufficient as a basis of a behavioural theory based on IPO transitions. To illustrate the problem, let us consider a small extension of our example model.

Example Suppose we want to model an additional kind of dynamics in the system, namely that preds also sometimes eat each other! Then we might add a reaction rule

$$\mathsf{pred} \mid \mathsf{pred} \longrightarrow \mathsf{pred}\ .$$

The extended system has additional IPO transitions such as

$$\mathsf{pred} \xrightarrow{\mathsf{pred}} \mathsf{pred}$$
$$\mathsf{pred} \mid \mathsf{pred} \xrightarrow{id} \mathsf{pred}\ .$$

One would naturally expect the two agents $\mathsf{pred} \mid \mathsf{pred}$ and $\mathsf{pred} \mid \mathsf{prey}$ to be equivalent: both can perform an internal action leading to a single pred, and both can eat either preds or preys both before and after their internal move. But their IPO transitions do not make them bisimilar. The two agents have only the following IPO transitions:

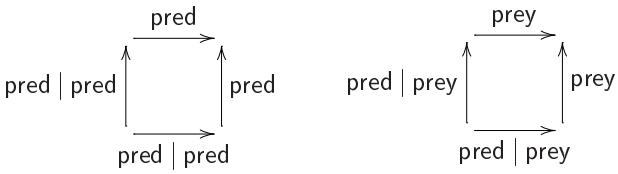
$$\mathsf{pred} \mid \mathsf{pred} \xrightarrow{id} \mathsf{pred} \qquad \mathsf{pred} \mid \mathsf{prey} \xrightarrow{id} \mathsf{pred}$$
$$\mathsf{pred} \mid \mathsf{pred} \xrightarrow{\mathsf{prey}} \mathsf{pred} \mid \mathsf{pred} \qquad \mathsf{pred} \mid \mathsf{prey} \xrightarrow{\mathsf{pred}} \mathsf{pred} \mid \mathsf{pred}\ ;$$

hence, they are not bisimilar. In particular, the transitions

$$\mathsf{pred} \mid \mathsf{pred} \xrightarrow{\mathsf{pred}} \mathsf{pred} \mid \mathsf{pred} \qquad \mathsf{pred} \mid \mathsf{prey} \xrightarrow{\mathsf{prey}} \mathsf{pred} \mid \mathsf{prey}$$

are *not* IPO transitions, although, intuitively, they represent perfectly valid interactions between the agents and their contexts. If the two latter transitions were included, then bisimilarity of the two agents would hold. ■

Why are the two latter transitions not included? This would require the squares



to be IPOs, but they are not. As mentioned above, the simple category we are working with has all IPOs, and in both transitions the agent and the redex coincide and hence have (id, id) as an IPO. Thus, the bounds in the above diagrams are "too large" to be IPOs, and clearly this will be the case whenever an agent contains a particular redex. In this situation the agent needs no contribution from the context to perform a reaction step based on the redex in question, and therefore it will have no non-identity transition based on that redex.

But that is not how we intend contextual transitions to work. If, for example, in the case of the agent $\mathsf{pred} \mid \mathsf{prey}$, the context happens to supply a prey, then it is perfectly possible for the agent to interact with the context, and transitions should reflect this. The purpose of the IPO condition is only to enforce that the label is no larger than necessary for a particular application of a reaction rule; it is not to prevent a redex from occurring "across" the boundary between an agent and the context.

The key concept to understanding the situation is that of *occurrence*. The point is that a redex might occur completely within an agent, but it might also be the case that some of its elements occur in the agent and others occur in the context. (For that matter, it might even be the case that the redex occurs completely outside the agent.)

In order to allow all these possibilities, we need somehow to express the sharing of elements between the agent and the redex, or more generally, between the arrows in our category. Before we discuss how this can be done in general, let us consider how the problem may be solved in the particular system we have been considering.

Example In order to represent occurrences, we need to be able to distinguish individual animals, even when they are of the same species. Therefore, instead of representing a collection of animals as a multiset, we shall now represent it as a set V together with a map

$$spec : V \to \{\mathsf{pred}, \mathsf{prey}\}$$

that assigns a species to each individual $v \in V$. This model is closely related to the multiset model: the latter arises as the quotient under the equivalence that relates isomorphic systems, i.e. systems whose underlying sets are related by a bijection that preserves the species maps. In other words, the multiset model is obtained by "forgetting" the concrete identities of the elements that make up a system.

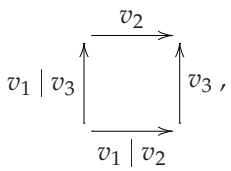
In our concrete model the abstract redex $\mathsf{pred} \mid \mathsf{prey}$ has many instances; one of them is

$$v_1 \mid v_2 \qquad \text{where } spec(v_1) = \mathsf{pred} \text{ and } spec(v_2) = \mathsf{prey}\ .$$

Our abstract agent $\mathsf{pred} \mid \mathsf{prey}$ might, for example, be instantiated as

$$v_1 \mid v_3 \qquad \text{where } spec(v_1) = \mathsf{pred} \text{ and } spec(v_3) = \mathsf{prey}\ .$$

Now, provided $v_2 \neq v_3$, this agent-redex pair has the IPO



which underlies one of our "missing" transitions from above. (The other transition is recovered analogously in our revised model.) ■

In the example we neglected to make clear in exactly which category we are taking the IPO, and unfortunately the answer to this question is not as straightforward as one might have hoped. The problem is how to define composition.

In the example, an arrow is a set of animals, and we have taken the composition of two arrows to be the union of their two underlying sets. This makes sense only when the species maps are compatible; for example, the two arrows

$$\big(\,\{u\}\,,\ u \mapsto \mathsf{pred}\,\big)$$
$$\big(\,\{u\}\,,\ u \mapsto \mathsf{prey}\,\big)$$

cannot be sensibly composed, for what should be the species map of the result?

In general, we shall be working with sets enriched with more structure than just (an equivalent of) the species map, and it turns out generally that composition only makes sense when the underlying sets are *disjoint*. Hence, composition of arrows in our "category" is only a *partial* operation, and therefore our system is not a reactive system according to the definition we have so far been considering, namely, a "proper" category together with a set of reaction rules.

Returning, then, to the general question of how to obtain a satisfactory definition of reactive system, it is clear that our proposed definition, based simply on a category, is insufficient; it needs some refinement in order to handle occurrences properly. There are several ways to address the problem.

One is Leifer's *functorial reactive systems* [16]. The idea in this approach is to set up two categories: one corresponding to the abstract structure (the multisets in our example); the other is a "super-concrete" category where the interfaces (objects) are equipped with information that allows one to keep track of the concrete structure in a way that allows composition to be defined in terms of disjoint union, but still be a total operation. A quotient functor factors out the concrete structure and allows IPO transitions derived in the concrete to be transferred to the abstract. This approach has the disadvantage that the extra information in interfaces makes it awkward to perform some of the manipulations that are necessary in the concrete category when developing the behavioural theory based on IPO transitions.

Another approach is Milner's *s-categorical reactive systems* [12, 13]. This also uses disjoint union in the definition of composition for concrete structures, but without making any changes to the interfaces. Then, as we have seen, composition is only partial, and hence the resulting structure is not a category. However, it turns out that the structure (called an *s-category*) is sufficiently close to the ordinary notion of category to allow the necessary constructions; in particular, RPOs and IPOs can be defined and manipulated much as one would in an ordinary category, and again there is a quotient functor that allows the transfer of transitions and behavioural theory to the abstract (proper) category. The main objection to this approach, therefore, is that it uses a slightly nonstandard categorical notion and represents the concrete structure manifestly in terms of sets, thus mixing a category-theoretical and set-theoretical approach.

An approach that addresses these objections is that of *2-categorical reactive systems* of Sassone and Sobociński [39, 42]. Here, one defines 2-cells ("arrows between arrows") on top of the category of abstract bigraphs. The 2-cells are

isomorphisms that represent how concrete element identities are related between arrows. Instead of working with RPOs and IPOs which are merely commutative diagrams, one works with similar diagrams that form 2-cells. Thus, in a manner of speaking, the concrete structure is represented abstractly! Sobociński's thesis [42] shows in detail how a precategory can be seen as an instance of the more general, and purely category-theoretical, notion of 2-category. For our present purpose, which is to investigate and apply reactive systems whose arrows are bigraphs, the greater abstraction is not directly helpful, as bigraphs are indeed based firmly on set-theory, and thus naturally form a precategory of the required kind. Undoubtedly, however, the 2-categorical approach is valuable for establishing relationships between the theory of reactive systems and other approaches to modelling mobile or concurrent computation, such as *rewriting logic* by Meseguer [19], the *tile logic* of Montanari and others (see, e.g. [8]), and the algebraic approach to *graph rewriting* by Ehrig and others (see e.g. [34]).

For the reasons given, we shall use Milner's definition based on precategories. As mentioned, this involves some non-standard concepts; for completeness, we shall therefore review the full definition, although it requires a certain amount of detail. We devote the following section to this purpose.

3.2 S-categories

In the following, and throughout the dissertation, we write $\vec{e}$ for a vector, or sequence, of elements. If $\vec{e}$ has length n we shall always use the convention of naming the elements $e_0, \ldots, e_{n-1}$. Very often we shall be working with vectors of length 2 (i.e., ordered pairs); in that case we shall use i to range over the indices 0 and 1, and we shall take $\bar{\imath}$ to denote the "opposite" of i, that is $\bar{\imath} \stackrel{\text{def}}{=} i + 1$ modulo 2.

Definition 3.1 (precategory) A *precategory* $\mathbf{C}$ consists of a set of *objects* and for each pair (a, b) of objects a set denoted $\mathbf{C}(a, b)$ whose elements we call *arrows*. We call $\mathbf{C}(a, b)$ a *homset*, and we write $f : a \to b$ for an arrow f in $\mathbf{C}(a, b)$. For each object a there is a distinguished arrow $id_a : a \to a$ called the *identity* at a, and for each triple a, b, c of objects there is a partial map from $\mathbf{C}(a, b) \times \mathbf{C}(b, c)$ to $\mathbf{C}(a, c)$ called *composition*. For $f : a \to b$ and $g : b \to c$ we write their composition (if defined) as $g \circ f : a \to c$. Composition is required to be associative with the identities as units; that is, for arrows

$$a \xrightarrow{f} b \xrightarrow{g} c \xrightarrow{h} d$$

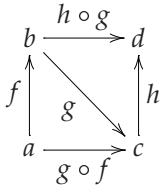
we require that the compositions $h \circ (g \circ f)$ and $(h \circ g) \circ f$ are either both defined and equal, or both undefined; and for

$$a \xrightarrow{f} b$$

we require that the compositions $f \circ id_a$ and $id_b \circ f$ are both defined and equal to f. ■

Thus, 'precategory' is defined just like 'category', except that composition is not required to be a total function. As readers familiar with category theory will recognise, essentially all concepts that we define for precategories in the following are straightforward generalizations of categorical concepts.

Equations involving composition of arrows are conventionally represented by *commutative diagrams*. For example, the equation $h \circ (g \circ f) = (h \circ g) \circ f$ is represented by the diagram



In a precategory, to say that such a diagram commutes means that along any path of arrows all compositions are defined, and moreover, for any two objects (vertices) a and b, and any two paths from a to b, the compositions along the two paths are equal.

An arrow $f : a \to b$ is an *isomorphism* (or *iso*) if there exists an arrow $f^{-1} : b \to a$ such that $f^{-1} \circ f = id_a$ and $f \circ f^{-1} = id_b$.

Definition 3.2 (monoidal precategory) A precategory $\mathbf{C}$ is *(strict, symmetric) monoidal* if it has a partial *tensor product* $\otimes$, a *unit* object ϵ satisfying $\epsilon \otimes a = a = a \otimes \epsilon$ for all a, and for objects $a \otimes b$ and $b \otimes a$ a *symmetry isomorphism* $\gamma_{a,b} : a \otimes b \to b \otimes a$, such that the following equations hold when both sides exist:

(1) $f \otimes (g \otimes h) = (f \otimes g) \otimes h$;

(2) $(g_0 \circ f_0) \otimes (g_1 \circ f_1) = (g_0 \otimes g_1) \circ (f_0 \otimes f_1)$;

(3) $\gamma_{a,\epsilon} = id_a$;

(4) $\gamma_{b,a} \circ \gamma_{a,b} = id_{a \otimes b}$;

(5) $\gamma_{b_0,b_1} \circ (f_0 \otimes f_1) = (f_1 \otimes f_0) \circ \gamma_{a_1,a_0}$ for $\vec{f} : \vec{a} \to \vec{b}$;

(6) $\gamma_{a \otimes b,c} = (\gamma_{a,c} \otimes id_b) \circ (id_a \otimes \gamma_{b,c})$. ■

Definition 3.3 (s-category) A *supported precategory*, or *s-category* for short, is a precategory with the following additional structure:

For each arrow f there is a set $|f|$ called its *support*, such that $|id_a| = \emptyset$. For arrows $f : a \to b$ and $g : b \to c$ the composition $g \circ f$ is defined iff $|f| \cap |g| = \emptyset$, and then $|g \circ f| = |f| \uplus |g|$.

For any arrow $f : a \to b$ and any injective map ρ whose domain includes $|f|$ there is an arrow $\rho(f)$ called a *support translation* of f such that

(1) $\rho(id_a) = id_a$;
(2) $\rho(g \circ f) = \rho(g) \circ \rho(f)$;
(3) $Id_{|f|}(f) = f$;
(4) $(\rho_1 \circ \rho_0)(f) = \rho_1(\rho_0(f))$;
(5) $\rho(f) = (\rho \restriction |f|)(f)$;
(6) $|\rho(f)| = \rho(|f|)$. ■

S-categories are a generalization of categories, for a category can be regarded as an s-category in which every arrow has empty support.

Definition 3.4 (support equivalence) Two parallel arrows $f, g : a \to b$ in an s-category $\mathbf{C}$ are *support equivalent*, written $f \simeq g$, if $\rho(f) = g$ for some support translation ρ. ■

The axioms of an s-category ensure that support equivalence is an equivalence relation. We often write $[f]$ for the support equivalence class $[f]_{\simeq}$ of f.

Definition 3.5 (functor) Given s-categories $\mathbf{C}$ and $\mathbf{D}$, a *functor* $\mathcal{F} : \mathbf{C} \to \mathbf{D}$ consists of two maps: the *object map* assigns to each object a of $\mathbf{C}$ an object $\mathcal{F}(a)$ of $\mathbf{D}$; and the *arrow map* assigns to each arrow $f : a \to b$ of $\mathbf{C}$ an arrow $\mathcal{F}(f) : \mathcal{F}(a) \to \mathcal{F}(b)$ of $\mathbf{D}$ such that

(1) $\mathcal{F}(id_a) = id_{\mathcal{F}(a)}$;
(2) $\mathcal{F}(g \circ f) = \mathcal{F}(g) \circ \mathcal{F}(f)$ if $g \circ f$ defined;
(3) $f \simeq g$ implies $\mathcal{F}(f) \simeq \mathcal{F}(g)$.

$\mathcal{F}$ is *full* if its arrow map is surjective on each homset $\mathbf{D}(\mathcal{F}(a), \mathcal{F}(b))$; it is *faithful* if its arrow map is injective on each homset. If $\mathcal{F}$ is an inclusion then $\mathbf{C}$ is a sub-s-category of $\mathbf{D}$. ■

Definition 3.6 (congruence) Let $\equiv$ be an equivalence defined homset-wise on an s-category $\mathbf{C}$. Then $\equiv$ is a *congruence* on $\mathbf{C}$ if it is preserved by composition; that is $f \equiv f'$ and $g \equiv g'$ implies $g \circ f \equiv g' \circ f'$ whenever both compositions are defined. ■

Proposition 3.7 Support equivalence $\simeq$ is a congruence. ■

When we do not need to work concretely, but only want to work "up to support equivalence", then we may abstract the support set away and move to the *support quotient* $\mathbf{C}/\simeq$—an ordinary category:

Definition 3.8 (support quotient category) For any s-precategory $\mathbf{C}$, the *support quotient category* $\mathbf{C}/\simeq$ is the category whose objects are the objects of $\mathbf{C}$ and whose arrows are support equivalence classes of arrows in $\mathbf{C}$; identity and composition are given by

$$id_a = [id_a]$$
$$[g] \circ [f] = [g \circ f] \,.$$ ■

It is routine to check that $\mathbf{C}/\mathrel{\hat{=}}$ indeed forms a category. In particular, composition of arrows $f : a \to b$ and $g : b \to c$ is always defined, because we can always find support instances of f and g with disjoint supports.

As observed above, we may also regard $\mathbf{C}/\mathrel{\hat{=}}$ as an s-category; thus, because of the equations for identity and composition, the support quotient can be regarded as a functor $[-] : \mathbf{C} \to \mathbf{C}/\mathrel{\hat{=}}$ between s-categories.

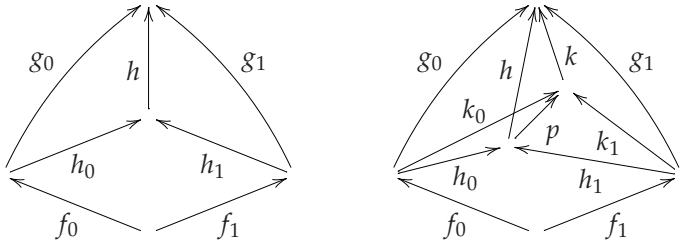
The above definition also makes sense if support equivalence $\mathrel{\hat{=}}$ is replaced by an arbitrary congruence $\equiv$ that includes $\mathrel{\hat{=}}$. Thus, for any such $\equiv$ we can form a quotient category $\mathbf{C}/\equiv$.

3.3 Relative pushouts

We now define relative pushouts and state some basic properties. We presuppose a fixed but arbitrary s-category $\mathbf{C}$.

Definition 3.9 (Relative pushout, idempushout) When the outer square commutes in the diagram on the left below we say that $\vec{g}$ is a *bound* of $\vec{f}$. When the whole diagram commutes we say, moreover, that the triple $(\vec{h}, h)$ is a *relative bound of $\vec{f}$ to $\vec{g}$*.

The relative bound $(\vec{h}, h)$ is a *relative pushout (RPO) of $\vec{f}$ to $\vec{g}$* if for any other relative bound $(\vec{k}, k)$ of $\vec{f}$ to $\vec{g}$ there exists a unique arrow p such that the diagram on the right commutes.



A bound $\vec{h}$ of $\vec{f}$ is an *idempushout (IPO) of $\vec{f}$* if $(\vec{h}, id)$ forms an RPO of $\vec{f}$ to $\vec{h}$. ∎

Proposition 3.10

(1) If an RPO of $\vec{f}$ to $\vec{g}$ exists, then it is unique up to an isomorphism.

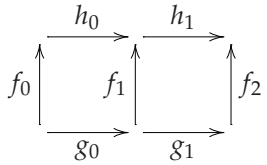
(2) If $(\vec{h}, h)$ is an RPO of $\vec{f}$ to $\vec{g}$, then $\vec{h}$ is an IPO of $\vec{f}$.

(3) If $\vec{h}$ is an IPO of $\vec{f}$ and an RPO exists of $\vec{f}$ to $h \circ \vec{h}$, then $(\vec{h}, h)$ is such an RPO.

(4) Suppose that the diagram below commutes and that there exists an RPO of (f_0, g_0) to $(h_1 \circ h_0, f_2 \circ g_1)$. Then

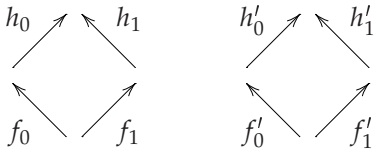
   (a) if both inner squares are IPOs then so is the outer rectangle;

(b) if the left square and the rectangle are IPOs then so is the right square.



(5) (id, f) is an IPO of (f, id). ■

Proposition 3.11 Let the square on the left below be an IPO, and suppose $f_0' \simeq f_0$ and $h_0' \simeq h_0$. Then there exist $f_1' \simeq f_1$ and $h_1' \simeq h_1$ such that also the square on the right is IPO.



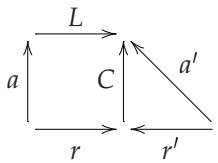
■

3.4 Reactive systems and contextual transitions

Definition 3.12 (reactive system) A *reactive system* is an s-category **C** equipped with a set $\mathcal{R}$ of *reaction rules*, where each reaction rule is a pair (r, r') of arrows with shared codomain; in such a rule we call r the *redex* and r' the *reactum*. We require $\mathcal{R}$ to be closed under support translation; that is, whenever $(r, r') \in \mathcal{R}$ then also $(\rho(r), \rho(r')) \in \mathcal{R}$ for any applicable support translation ρ.

We say that *a reacts to* a', written $a \longrightarrow a'$, whenever $a = C \circ r$ and $a' = C \circ r'$ for some arrow C of **C** and some reaction rule $(r, r') \in \mathcal{R}$. ■

Definition 3.13 (standard transitions) The *standard transition system* for a reactive system $(\mathbf{C}, \mathcal{R})$ has the arrows of **C** as both agents and labels; a label L applies to an agent a iff the composition $L \circ a$ is defined; and there is a transition $a \xrightarrow{L} a'$ iff there is a reaction rule $(r, r') \in \mathcal{R}$ and an arrow C of **C** such that the following diagram commutes and the square forms an IPO:



We say that the transition is *based on* the given redex, reaction rule, or IPO; and also that they *underlie* the transition. ■

When a particular reactive system is understood we denote by ST its standard transition system.

The following properties of standard transitions follow immediately from the definition:

Proposition 3.14

(1) $a \xrightarrow{id} a'$ iff $a \rightarrow a'$;

(2) standard transitions are determined only up to isomorphism; that is, whenever $a \xrightarrow{L} a'$ and ι is an iso composable with L, then also $a \xrightarrow{\iota \circ L} \iota \circ a'$. ■

3.5 Strong bisimilarity

The standard transition system gives rise to a bisimilarity in the usual way:

Definition 3.15 *Strong standard bisimilarity* for a reactive system is the bisimilarity $\sim_{\mathrm{ST}}$ for its standard transition system. ■

We shall often drop the index ST and the qualification 'standard' when there can be no confusion with other transition systems.

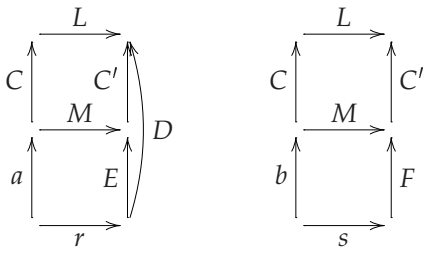
We now come to the result—due to Leifer and Milner [18]—that strong standard bisimilarity is a congruence. We repeat the proof, as the result is fundamental and the argument quite short and elegant. In our present setting, the argument mainly involves proving the allowability of the operation

$$\mathcal{S}^{\mathrm{C}} \stackrel{\text{def}}{=} \{(D \circ a, D \circ b) \mid (a, b) \in \mathcal{S}\},$$

i.e., that exhibiting a bisimulation "up to context" is a sound proof technique.

Lemma 3.16 In a reactive system $(\mathbf{C}, \mathcal{R})$ with all RPOs the operation $(-)^{\mathrm{C}}$ is allowable for the standard transition system.

Proof Suppose $\mathcal{S} \rightarrowtail \mathcal{T}$. We must show $\mathcal{S}^{\mathrm{C}} \rightarrowtail \mathcal{T}^{\mathrm{C}}$. Let $(a, b) \in \mathcal{S}$ and $C \circ a \xrightarrow{L} a'$ with $L \circ C \circ b$ defined; we seek a transition $C \circ b \xrightarrow{L} b'$ for some b' such that $(a', b') \in \mathcal{T}^{\mathrm{C}}$. The transition of $C \circ a$ is based on a reaction rule (r, r') and an IPO in the form of the outer rectangle on the left below such that $a' = D \circ r'$.



By the assumption that **C** has all RPOs we can fill in the rectangle as shown with an RPO (M, E, C') of (a, r) to $(L \circ C, D)$, and by Proposition 3.10 both small squares are IPOs. The lower square underlies a transition $a \xrightarrow{M} a'_0 \stackrel{\text{def}}{=} E \circ r'$. As $(a, b) \in \mathcal{S}$ we then have a transition $b \xrightarrow{M} b'_0$ for some b'_0 such that $(a'_0, b'_0) \in \mathcal{T}$. This transition is based on a reaction rule $(s, s') \in \mathcal{R}$ and an IPO in the form of the lower square on the right above such that $b'_0 = F \circ s'$. We compose this with the upper IPO from the left, as shown; by Proposition 3.10 the resulting rectangle is also IPO. As $\mathcal{R}$ is assumed to be closed under support translation, we can assume without loss of generality that $C' \circ b'_0$ is defined. The IPO rectangle on the right therefore underlies a transition $b \xrightarrow{L} b' \stackrel{\text{def}}{=} C' \circ b'_0$. Moreover, observing that $a' = C' \circ a'_0$, we have $(a', b') \in \mathcal{T}^{\mathbf{C}}$ as required. ■

Now the congruence property follows easily:

Theorem 3.17 (congruence) In a reactive system with all RPOs strong standard bisimilarity is a congruence; that is, if $a \sim b$ then $C \circ a \sim C \circ b$.

Proof By the definition of bisimilarity we have $\sim \rightarrowtail \sim$. Applying the preceding lemma yields $(\sim)^{\mathbf{C}} \rightarrowtail (\sim)^{\mathbf{C}}$; that is, $(\sim)^{\mathbf{C}}$ is a standard bisimulation, and the result follows. ■

3.6 Weak bisimilarity

In applications, strong bisimilarity is often found to be too strong; that is, it distinguishes between systems that it would be reasonable to consider equivalent. Consider the following two examples:

Example Recall the system introduced earlier, which had the following two reaction rules:

$$\begin{aligned} \mathsf{pred} \mid \mathsf{prey} &\longrightarrow \mathsf{pred} \\ \mathsf{pred} \mid \mathsf{pred} &\longrightarrow \mathsf{pred}\ . \end{aligned}$$

For the system consisting of only the first of the two rules we proved that the agents pred and pred | pred are strongly bisimilar. Is this true also in the system consisting of both rules?

The agents have the following standard transitions arising from the second rule:

$$\begin{aligned} \mathsf{pred} &\xrightarrow{\mathsf{pred}} \mathsf{pred} \\ \mathsf{pred} \mid \mathsf{pred} &\xrightarrow{\mathsf{pred}} \mathsf{pred} \mid \mathsf{pred} \\ \mathsf{pred} \mid \mathsf{pred} &\xrightarrow{id} \mathsf{pred}\ . \end{aligned}$$

The latter transition cannot be matched by pred, so the two agents are not strongly bisimilar when we include the second rule. However, it is reasonable, for many purposes, to regard the two agents as equivalent, because their externally observable behaviours are still identical: they eat preds and preys. In other words, they have the same potential for interaction when placed in a larger system. ■

Example For a different example, expressed in the π-calculus, consider the two processes

$$P \stackrel{\text{def}}{=} \overline{x}y \qquad \text{and} \qquad Q \stackrel{\text{def}}{=} \nu z\,(\overline{z}y \mid z(u).\overline{x}u)\,.$$

The process P sends y directly on x; the other process Q instead sends y on a private channel z which it shares with a buffer process that forwards any received data on x. Thus, to someone who interacts with P and Q, but cannot observe how they work internally, their behaviour is the same. (This assumes that the time taken by the buffering in Q is not noticeable, but we are not attempting to address temporal behaviour.) The transitions of P and Q are:

$$P \xrightarrow{\overline{x}y} \mathbf{0} \qquad \text{and} \qquad Q \xrightarrow{\tau} \xrightarrow{\overline{x}y} \mathbf{0}\,.$$

Clearly P and Q are not strongly bisimilar: for example, P has no τ-transition, and so cannot match the τ-transition of Q. ■

In process calculi it is common to represent internal activity as transitions with a distinguished label τ, often referred to as the *silent action*. Thus, a transition $P \xrightarrow{\tau} P'$ signifies a move by P in which the environment takes no part. In our present formulation, such a silent transition corresponds to a reaction step, or equivalently, a standard transition whose label is the identity context.

Silent transitions are the basis for defining *weak bisimilarity*, a version of bisimilarity that abstracts away internal activity. It is based on a derived transition relation $\Rightarrow$ defined as follows:

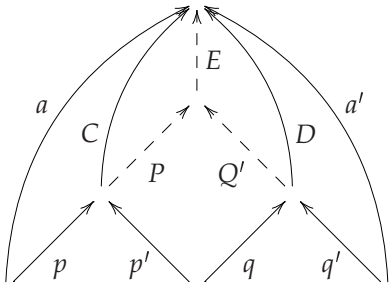
$$\text{Whenever} \quad a \xrightarrow{l_1} \cdots \xrightarrow{l_n} a' \quad \text{then} \quad a \stackrel{\hat{s}}{\Rightarrow} a'\,,$$

where s is the label sequence $(l_1, \ldots, l_n)$, and $\hat{s}$ denotes the sequence obtained from s by removing every occurrence of τ. Weak bisimilarity can be defined simply as the bisimilarity for the transition relation $\Rightarrow$. The more usual—and equivalent—definition uses a mixture of both ordinary and derived transitions; it requires merely that every ordinary transition $\xrightarrow{l}$ can be matched by $\stackrel{l}{\Rightarrow}$. In our present terms the equivalence of the two definitions means that ordinary transitions are adequate for weak transitions.

We now discuss how weak bisimilarity may be defined for reactive systems in a way that fits naturally with our adopted premise that labels are contexts. A central idea connected with our IPO-based transitions is that of an *experiment*: a transition $a \xrightarrow{L} a'$ expresses that a, with the contribution of L from the environment, can perform a reaction step to become a'. It is natural to extend

this idea to *reaction sequences*; a derived transition $a \stackrel{L}{\Rightarrow} a'$ then expresses that a, with the contribution of L from the environment, can perform a sequence of reaction steps to become a'.

Let us consider a reaction sequence $a \rightarrow\!\!\rightarrow a'$ of length 2. The definition of reaction requires for the first reaction step that $a = C \circ p$ for some context C and redex p; the result of the reaction will be $C \circ p'$, where p' is the reactum for p. The second reaction step requires the result of the first reaction to have the form $D \circ q$ for some context D and redex q; the result of the reaction will be $a' = D \circ q'$, where q' is the reactum for q. These equations are represented by the solid arrows in the following diagram:



Filling in the middle square with an RPO (P, Q', E), as indicated by the dashed arrows, we see that it is possible to view the two-step reaction as a single-step reaction that arises from applying a derived reaction rule

$$(r, r') \stackrel{\text{def}}{=} (P \circ p, Q' \circ q')$$

in the context E. Clearly, the construction can be repeated to cater for reaction sequences of arbitrary length.

We can think of the derived rule (r, r') as a composition of the two actual rules (p, p') and (q, q'). Since the construction is based on an IPO of the pair (p', q), the composition can be formed only if these two arrows are consistent, but apart from this there is no constraint on the amount of sharing among their supports. Since, moreover, the IPO arrows P and Q' are composed with p and q', respectively, the composition can be formed only if the supports satisfy

$$|p| \cap |P| = |q'| \cap |Q'| = \emptyset\,. \qquad (*)$$

By construction we have $|p'| \cap |P| = |q| \cap |Q'| = \emptyset$. Moreover, by support translation we can choose P and Q' such that $(|P| \cup |Q'|) - (|p'| \cup |q|)$ is disjoint from $|p| \cup |q'|$. (In fact, as we shall see, in bigraphs the former set is always empty.) Hence $(*)$ will hold if

$$|p| \cap |q| \subseteq |p'| \qquad (1)$$

$$|p'| \cap |q'| \subseteq |q| \qquad (2)$$

The following definition introduces a derived reaction relation based on composition of rules.

Definition 3.18 (weak reaction) We say that reaction rules (p, p') and (q, q') are *compatible* if p' and q are consistent and equations (1) and (2) above hold.

For compatible rules (p, p') and (q, q'), their *composition* $(p, p') \cdot (q, q')$ is defined as the rule $(P \circ p, Q' \circ q')$, where (P, Q') is an IPO of (p', q).

We call a rule of the form (id_I, id_I) an *identity rule*.

For $\mathcal{R}$ a set of rules we define its *weakening*, written $\mathcal{W}(\mathcal{R})$, as the result of adding all identity rules and then closing under composition of compatible rules. We extend $\mathcal{W}$ to a function that sends a reactive system $(\mathbf{C}, \mathcal{R})$ to $(\mathbf{C}, \mathcal{W}(\mathcal{R}))$.

Define the *weak reaction relation* $\Rightarrow$ in $\mathbf{C}$ as the reflection of reaction in $\mathcal{W}(\mathbf{C})$; that is, $a \Rightarrow a'$ in $\mathbf{C}$ iff $a \rightarrow a'$ in $\mathcal{W}(\mathbf{C})$. ■

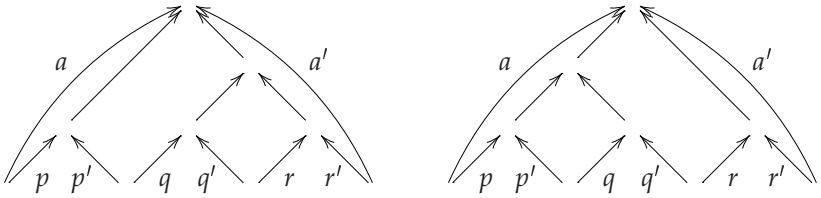
With this definition weak reaction indeed corresponds to sequences of ordinary reaction steps:

Lemma 3.19 In a reactive system with all RPOs it holds that $a \Rightarrow a'$ iff $a \rightarrow^* a'$.

Proof (sketch) The proof hinges on showing that rule composition is associative:

$$(p, p') \cdot \big((q, q') \cdot (r, r')\big) = \big((p, p') \cdot (q, q')\big) \cdot (r, r') \,.$$

This requires showing that the diagram on the left below, with both inner squares IPOs, can be transformed to the diagram on the right, also with both inner squares IPOs.



For this, one merely needs to fill in the long rectangle by an RPO; the composition properties of IPOs (Proposition 3.10) then ensure that the squares on the right are IPOs. ■

We take *weak standard transitions* $a \overset{L}{\Rightarrow} a'$ to be the (ordinary) standard transitions of the weakened system:

Definition 3.20 (weak transitions) In a reactive system $\mathbf{C}$ we define the *weak standard transition system* in $\mathbf{C}$ as the reflection of the standard transition system in $\mathcal{W}(\mathbf{C})$; that is, $a \overset{L}{\Rightarrow} a'$ in $\mathbf{C}$ iff $a \overset{L}{\rightarrow} a'$ in $\mathcal{W}(\mathbf{C})$. ■

When the reactive system concerned is understood we denote its weak standard transition system by WST, and we shall continue to use the notation

$a \stackrel{L}{\Rightarrow} a'$ for its transitions. We shall refer to the associated bisimilarity $\sim_{\mathrm{WST}}$ as *weak standard bisimilarity* and normally write it as $\approx$.

Since the weak standard transitions in $\mathbf{C}$ are exactly the standard transitions in $\mathcal{W}(\mathbf{C})$, and $\mathcal{W}$ is the identity on structure, the congruence property (Theorem 3.17) for strong standard bisimilarity carries over immediately to weak bisimilarity:

Corollary 3.21 (congruence) In a reactive system with all RPOs weak standard bisimilarity is a congruence; that is, if $a \approx b$ then $C \circ a \approx C \circ b$. ■

The following lemma records the basic properties of the weak transition relation.

Lemma 3.22 In a reactive system with all RPOs the following hold:

(1) $a \stackrel{id}{\Rightarrow} a$.

(2) If $a \xrightarrow{L} a'$ then $a \stackrel{L}{\Rightarrow} a'$;

(3) If $a \stackrel{L_1}{\Rightarrow} \cdots \stackrel{L_n}{\Rightarrow} a'$ and $L = L_n \circ \cdots \circ L_1$ then $a \stackrel{L}{\Rightarrow} a'$;

(4) If $a \stackrel{L}{\Rightarrow} a'$ then $a \xrightarrow{L_1} \cdots \xrightarrow{L_n} a'$ for some $L_1, \cdots, L_n$ such that $L = L_n \circ \cdots \circ L_1$.

Proof (sketch) Each property is an immediate consequence of the construction of the weakened rule set $\mathcal{W}(\mathcal{R})$: (1) holds because $\mathcal{W}(\mathcal{R})$ contains all identity rules; (2) holds because $\mathcal{W}(\mathcal{R})$ contains every rule in $\mathcal{R}$; (3) holds because $\mathcal{W}(\mathcal{R})$ is closed under rule composition; (4) holds because every rule in $\mathcal{W}(\mathcal{R})$ is the composition of a sequence of rules from $\mathcal{R}$. ■

The properties established in the lemma can be summarized by saying that a weak standard transition arises as a (possibly empty) sequence of standard transitions, and that the labels of these transitions are composed to form the label of the weak transition. Taking the composition of the labels does the job of abstracting away silent transitions, because silent transitions have identity labels. In fact it does more, because labels formed in this way may have more than one decomposition into non-identity labels; this potentially allows greater freedom in constructing matching transitions than under the usual definition in process calculi. As we shall see in Part II, the greater freedom turns out not to make any difference in the π-calculus. It does in some systems, however; a simple example is given below.

First, we establish that our notions of weak and strong bisimilarity have the same relationship as they do in process calculi, namely that strongly bisimilar agents are also weakly bisimilar, and that standard transitions are adequate for the weak standard transitions. As mentioned earlier, the latter property validates the familiar proof technique for weak bisimilarity, where only the "strong" transitions of an agent are matched (with weak transitions) by the other agent.

Theorem 3.23 (weak and strong bisimilarity) In a reactive system that has all RPOs the following hold:

(1) Strong standard bisimilarity implies weak standard bisimilarity; that is, $\sim \subseteq \approx$;

(2) The standard transition system is adequate for weak standard transitions; that is, $\approx^{\mathrm{ST}} = \approx$.

Proof By Lemma 3.22(2) we have $\mathrm{ST} \preccurlyeq \mathrm{WST}$, from which we directly obtain the following inclusions:

$$\sim \subseteq \approx^{\mathrm{ST}} \qquad (*)$$
$$\approx \subseteq \approx^{\mathrm{ST}} . \qquad (**)$$

Assertion (1) of the theorem will follow from $(*)$ together with assertion (2). For (2) the inclusion in one direction is given by $(**)$. Thus, it only remains to show $\approx^{\mathrm{ST}} \subseteq \approx$. We show that $\approx^{\mathrm{ST}}$ is a weak bisimulation. Suppose $a \approx^{\mathrm{ST}} b$ and $a \stackrel{L}{\Rightarrow} a'$ with $L \circ b$ defined; we must find a transition $b \stackrel{L}{\Rightarrow} b'$ such that $a' \approx^{\mathrm{ST}} b'$. By Lemma 3.22(4) there is a sequence of standard transitions

$$a \xrightarrow{L_1} \cdots \xrightarrow{L_n} a'$$

such that $L = L_n \circ \cdots \circ L_1$. Then, appealing repeatedly to the assumption $a \approx^{\mathrm{ST}} b$, we obtain

$$b \stackrel{L_1}{\Longrightarrow} \cdots \stackrel{L_n}{\Longrightarrow} b'$$

for some b' such that $a' \approx^{\mathrm{ST}} b'$. It follows from Lemma 3.22(3) that $b \stackrel{L}{\Rightarrow} b'$. This completes the proof. ∎

Example Returning to our system with predators and prey, we can now prove the weak standard bisimilarity

$$\mathsf{pred} \approx \mathsf{pred} \mid \mathsf{pred} .$$

The only interesting part of the proof concerns the matching of the transition

$$\mathsf{pred} \mid \mathsf{pred} \xrightarrow{id} \mathsf{pred} .$$

As observed earlier, the other agent, pred, cannot match this with an ordinary standard transition, but it has a weak standard transition

$$\mathsf{pred} \stackrel{id}{\Longrightarrow} \mathsf{pred}$$

based on the reaction rule with empty redex and reactum. ∎

Example As promised, we also give an example where weak transition matching is not used merely to remove identity labels. Suppose the system is enlarged to also contain a third species, bigpred, for which we add the reaction rules

$$\begin{aligned} \mathsf{bigpred} \mid \mathsf{prey} &\longrightarrow \mathsf{bigpred} \\ \mathsf{bigpred} \mid \mathsf{prey} \mid \mathsf{prey} &\longrightarrow \mathsf{bigpred} \\ \mathsf{bigpred} \mid \mathsf{pred} &\longrightarrow \mathsf{bigpred}\ ; \end{aligned}$$

thus, a bigpred behaves like a pred, except that it sometimes eats two preys at a time. We can prove

$$\mathsf{pred} \approx \mathsf{bigpred}\ .$$

The only transition that cannot be matched directly is

$$\mathsf{bigpred} \xrightarrow{\mathsf{prey}\mid\mathsf{prey}} \mathsf{bigpred}\ ;$$

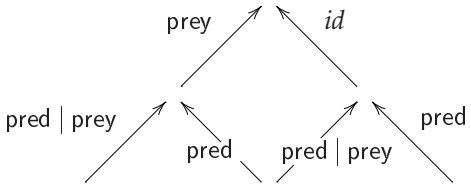
no standard transition of pred matches this transition. Instead we use the weak standard transition

$$\mathsf{pred} \stackrel{\mathsf{prey}\mid\mathsf{prey}}{\Longrightarrow} \mathsf{pred}\ ;$$

this transition reflects that there is a standard transition sequence

$$\mathsf{pred} \xrightarrow{\mathsf{prey}} \xrightarrow{\mathsf{prey}} \mathsf{pred}$$

with the same composite label. Formally, the weak transition is based on the composite reaction rule derived from the following IPO (in fact, pushout):



This example exploits the freedom in transition matching that is allowed by composition of labels. Under the definition that is usual in process calculi we are only allowed to insert extra *id*-transitions. In this case that is not enough; the matching transition can only be obtained by combining two non-*id*-transitions.

It is worth noting that the weak bisimilarity holds only because a bigpred can consume, in one step, either one prey or two; if it always consumed two preys at a time, it could not match the transition $\mathsf{pred} \xrightarrow{\mathsf{prey}} \mathsf{pred}$. ■

Chapter 4

Bigraphs: Structure

4.1 Introduction

A bigraph models two different aspects of a system: locality (place structure) and connectivity (link structure). As an example, the bigraph in Figure 4.1 represents a system in which people are located (physically) in rooms within buildings and at the same time can connect (virtually) by conducting mobile phone calls. The nesting of nodes shows that each room is located in a building and each person is located in a room; the arcs between person nodes represent phone connections between the respective persons.

An example of dynamics is shown in Figure 4.2, which depicts a reaction step in which a person moves from one room to another in the same building; the phone call he is conducting persists.

This example has three kinds of nodes, representing buildings, rooms and people, respectively. Exactly what kinds of nodes are available, and what kinds of reactions they engage in, will of course depend on the system being modelled. Thus, each particular bigraphical reactive system (BRS) is determined by, on the one hand, a *signature* that determines the node kinds, and on the other, a set of *reaction rules* that determine dynamics. We concentrate in this chapter on the structure of bigraphs and defer the treatment of dynamics to the next chapter.

In the present example, the signature has three elements: building, room and person; we refer to these as *controls*. Each node is assigned a control, and this assignment influences both dynamics and structure. Thus, the dynamics allows people to move between rooms, but not, for example, rooms to move between buildings. As for structure, links can exist between person nodes, but not, for example, between a building node and a room node.

To elaborate on the latter point, linking takes place not between nodes directly, but rather between *ports*, which we can think of as connection points residing on the boundaries of nodes. Each node has a specific number of ports; we refer to this number as the *arity* of the node. The arity of a node is deter-

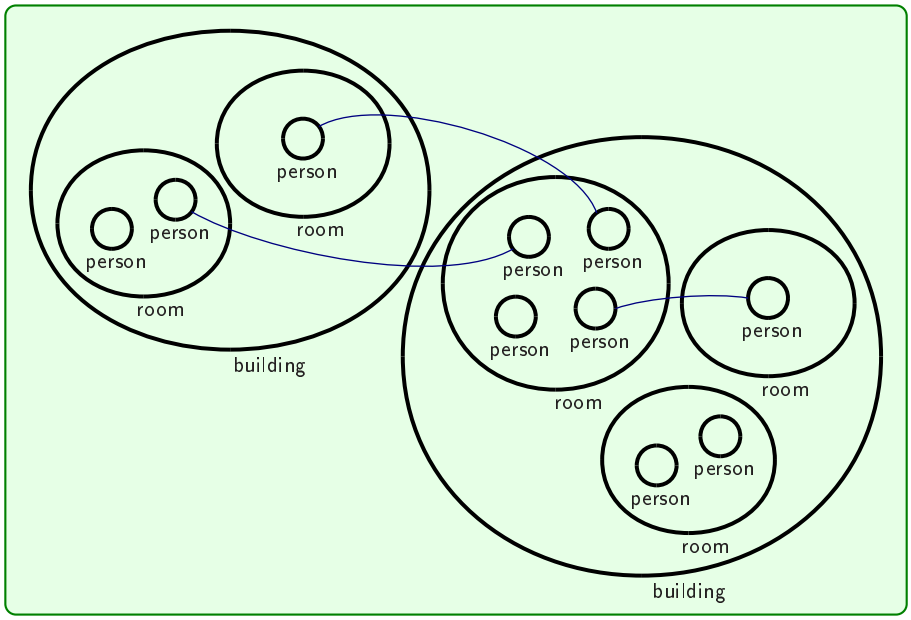


Figure 4.1 Example of a bigraph

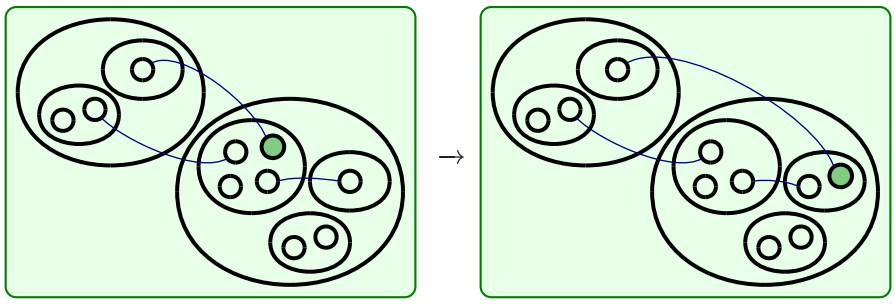


Figure 4.2 A reaction step

mined via its control by the signature. Thus, in the present example the linking constraint is automatically satisfied by declaring that person nodes have arity 1 and that building and room nodes have arity 0.

In our example, controls govern not only linking structure but also place structure; for example, it would not make sense to have a building node within a person node. Such constraints on the place structure will be provided for in *place-sorted bigraphs*, which we shall define as a refined notion of the simpler *basic bigraphs* that we introduce first.

A similar refinement, *link-sorted bigraphs*, can be defined to capture constraints on link structure. An instance is the additional constraint we have implicitly placed on links in our example, namely that a link always connects exactly two ports. (In basic bigraphs a link connects an arbitrary number of ports.) We shall not deal with link-sorting in this dissertation, because none of the applications in Part II require it; the topic is dealt with by Leifer and Milner in [17].

Yet another refinement, which along with place-sorting will be needed in Part II, is *binding*, which can be seen as a joint constraint on place and link structure. To introduce it, let us consider a refinement of the model in our example, in which the connectivity by mobile phones is modelled with a bit more detail. In reality, when two people are conducting a call, their phones are not connected directly; instead, each connects to a nearby base station that handles the onwards connectivity. Let us imagine that a room can be equipped with a base station and that each person can connect (via their phone) only to a base station in that particular room. (Admittedly, this assumption is not entirely realistic for normal mobile telephony, where base stations tend to cover a larger and less strictly delimited geographical area; it might be quite realistic, however, in a system that combines telephony with a capability for tracking people within buildings.)

Our refined model might look as shown in Figure 4.3, where for clarity we include just one building node. Base stations are represented by nodes with the new control base of arity 2. Thus, each base node has two separate ports, one for linking it to other base nodes, and one for linking it to person nodes in its room. In the figure, the latter port of each base node is indicated by a small circle, which indicates that these ports are *binding*, as explained below. Figure 4.4 shows a reaction step similar to the one shown earlier; the moving person node is now required to establish a link to a new base node.

Figure 4.3 shows several possibilities for the linking structure: In one room there is no base station, so people in that room have no connections; another room has a base station, but a person in the room is nevertheless unconnected (perhaps his phone is switched off). The main point, however, is that the bigraph satisfies the structural constraints that each person is linked to at most one base node and that each base node is linked only to person nodes within its surrounding room node. We can capture these constraints by declaring the person-linking port of a base node to be a *binding port* that has the surrounding room node as its *scope*; then the structural constraints follow from imposing the natural binding discipline that every ordinary (i.e., non-binding) port is bound

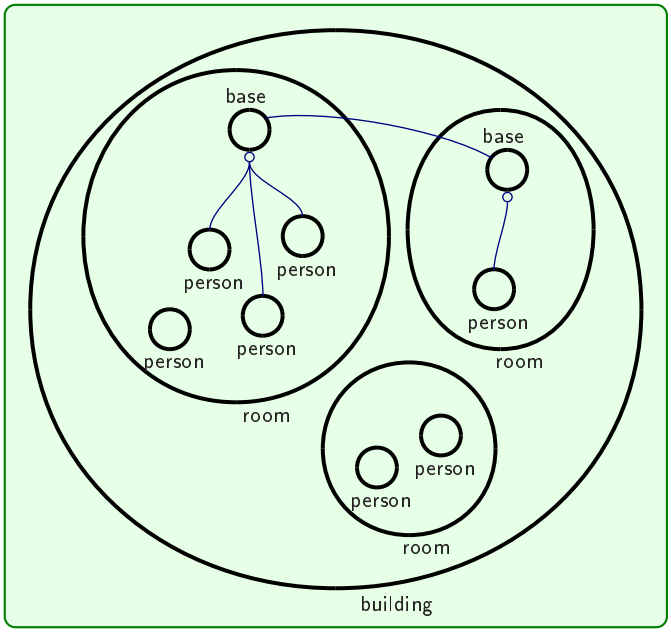


Figure 4.3 A bigraph with binding

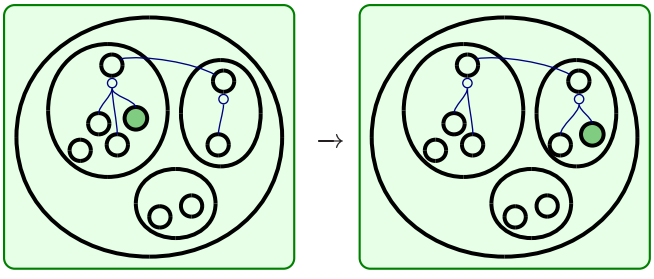


Figure 4.4 A reaction step

by at most one binding port and lies within its scope. We shall see in Part II that the usual notion of name binding in process calculi corresponds exactly to this form of port binding.

As was the case for place-sorting, we shall introduce binding as a refinement of a simpler notion of bigraph that we introduce first. We shall refer to bigraphs without binding as *pure bigraphs*. Thus, our simplest notion of bigraph will be basic, pure bigraphs.

We shall return to our example model in the following chapter on bigraph dynamics. For now, we embark on a more detailed exposition of the structural aspects of bigraphs. As stated, we shall at first concentrate on basic, pure bigraphs. We begin by defining the corresponding notion of signature, as follows:

Definition 4.1 (signature) A *(basic, pure) signature* Σ consists of a set $\mathcal{K}$, whose elements we call *controls*, and a map $ar : \mathcal{K} \to \mathbb{N}$ that assigns to each control $K \in \mathcal{K}$ an *arity* $m = ar(K)$. When the signature is understood we write $K : m$ to indicate the arity assignment. ■

When a signature ascribes an arity (and later, other properties) to a particular control it is natural to think of this property as being "inherited" by any bigraph node with that control. Accordingly, we often talk about, for example, the arity of a node when, to be strictly correct, we should be referring to the arity of its control.

As previously mentioned, a bigraph arises as a combination of a *place graph* and a *link graph*, so the definition of bigraph is most conveniently given in terms of these two constituent structures. Thus, in the following two sections we introduce place graphs and link graphs, respectively; then in Section 4.4 we define (concrete basic pure) bigraphs, and in Section 4.5 we review their structural theory, mainly as regards relative pushouts. After that, Sections 4.6 and 4.7 introduce the two structural refinements we have mentioned: binding and place-sorting. Section 4.8 shows how abstract bigraphs are obtained from the concrete ones, and in the last section of this chapter we introduce some additional concepts and notation in preparation for the treatment of dynamics, and for applications.

4.2 Place graphs

In the definition of place graphs (and throughout the dissertation) we treat a finite ordinal n as the set $\{0, \ldots, n-1\}$, and we use '+' to denote disjoint union (coproduct) of sets.

Definition 4.2 (place graph) A *(concrete) place graph* over Σ takes the form

$$A = (V, ctrl, prnt) : m \to n \,,$$

where

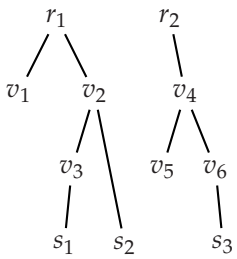


Figure 4.5 Example of a place graph

- m and n are finite ordinals whose elements we call, respectively, the *sites* and *roots* of G;
- V is a set of *nodes*;
- $ctrl : V \to \mathcal{K}$ is the *control map*;
- $prnt : m + V \to n + V$ is the *parent map*.

We call $m + V$ the *places* of A. For a place w, if $prnt(w) = w'$ then we say that w' is the *parent* of w, and w a *child* of w'. Distinct places with the same parent are called *siblings*. If $w' = prnt^{(k)}(w)$ for some $k > 0$ we say that w' is an *ancestor* of w, and w a *descendant* of w', and we write $w' >_A w$. We require ancestry to be irreflexive on nodes; that is, $prnt^{(k)}(v) \neq v$ for all $k > 0$ and $v \in V$. We say that a root or node is *barren* if it has no descendants. ■

The domain and codomain of the parent map, together with irreflexivity, ensure that the places of A form a forest, i.e. a set of trees, in which the sites m are among the leaves and moreover (justifying the terminology) the elements of n are the roots.

Figure 4.5 shows an example of a place graph. It has roots r_1 and r_2, sites s_1, s_2 and s_3, and nodes $v_1, \ldots, v_6$; the nodes v_1 and v_5 are barren.

Given two place graphs

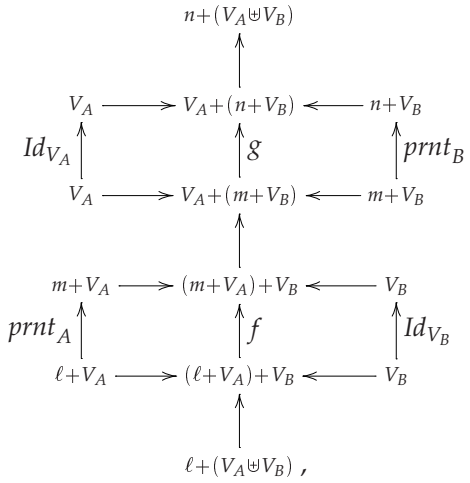
$$A = (V_A, ctrl_A, prnt_A) : \ell \to m$$
$$B = (V_B, ctrl_B, prnt_B) : m \to n\ ,$$

we can form their composition $B \circ A : \ell \to n$, provided the node sets V_A and V_B are disjoint. The composition is obtained, informally speaking, by "planting" the roots of A in the sites of B. More precisely, the construction is as follows: For the node set take the disjoint union $V_A \uplus V_B$ of the component node sets, and for the control map take the union map

$$ctrl_A \uplus ctrl_B : V_A \uplus V_B \to \mathcal{K}\ .$$

The parent map $prnt$ must be formed as a combination of the component parent maps $prnt_A$ and $prnt_B$. For a place w that is either a site in ℓ or a node in V_A it should map w to $prnt_A(w)$, unless this yields a root r in m, in which case the

result should be $prnt_B(r)$. For a node v in V_B it should map v to $prnt_B(v)$. Thus, we obtain *prnt* as the vertical composition of maps in the diagram



where the horizontal arrows are injections into the sums, the anonymous vertical arrows are the evident bijections, and f and g are the unique maps that make the squares commute. Thus, abusing notation by ignoring the bijections, we have

$$prnt = (Id_{V_A} + prnt_B) \circ (prnt_A + Id_{V_B}) .$$

Composition of place graphs gives rise to an s-category as follows:

Definition 4.3 The s-category $\hat{}\mathbf{PLG}(\Sigma)$ has finite ordinals as objects and concrete place graphs as arrows. The support $|A|$ of a place graph A is its node set. The identity at m is the empty place graph with identity parent map, and composition is defined as detailed above. ■

It is straightforward to verify that $\hat{}\mathbf{PLG}(\Sigma)$ as defined does indeed satisfy the conditions of an s-category; for example, by the constructions above, composition is indeed defined just when the supports are disjoint.

We now detail another operation on place graphs called *tensor product*. First, for any two ordinals m_0 and m_1 we define the tensor product $m_0 \otimes m_1$ as the (arithmetic) sum $m_0 + m_1$. Then, given two place graphs

$$A_i = (V_i, ctrl_i, prnt_i) : m_i \to n_i \qquad (i = 0, 1) ,$$

the tensor product $A_0 \otimes A_1 : m_0 \otimes m_1 \to n_0 \otimes n_1$ is defined iff $V_0 \cap V_1 = \emptyset$, and is then given as follows: Its nodes are $V_0 \uplus V_1$, and its control map is

$$ctrl_0 \uplus ctrl_1 : V_0 \uplus V_1 \to \mathcal{K} .$$

For the parent map *prnt* we also take, essentially, a disjoint union map; here the sites and roots of A_1 must be "shifted" upwards by m_0 and n_0, respectively;

thus, we take *prnt* to be the unique arrow that makes

$$\begin{array}{ccccc} n_0+V_0 & \longrightarrow & (n_0\otimes n_1)+(V_0\uplus V_1) & \longleftarrow & n_1+V_1 \\ {\scriptstyle prnt_0}\uparrow & & \uparrow{\scriptstyle prnt} & & \uparrow{\scriptstyle prnt_1} \\ m_0+V_0 & \longrightarrow & (m_0\otimes m_1)+(V_0\uplus V_1) & \longleftarrow & m_1+V_1 \end{array}$$

commute, where the horizontal arrows are the evident injections that map each $s \in m_0$ to $s \in m_0 \otimes m_1$ and each $s \in m_1$ to $m_0 + s \in m_0 \otimes m_1$, and similarly for the n_i.

It is straightforward to verify that $\hat{}\mathbf{PLG}(\Sigma)$ and tensor product (with unit 0) form a strict monoidal s-category.

4.3 Link graphs

For the definition of link graphs we presuppose an infinite set of *names*.

Definition 4.4 (link graph) A *(concrete) link graph* over Σ takes the form

$$A = (V, E, ctrl, link) : X \to Y\ ,$$

where

- X and Y are finite sets of names whose elements we call, respectively, the *inner names* and *outer names* of A;
- V is a set of *nodes*;
- E is a set of *edges*;
- $ctrl : V \to \mathcal{K}$ is the *control map*;
- $link : X + P_V \to Y + E$ is the *link map*.

Here P_V denotes the *ports* of A, defined as follows: first, for each node $v \in V$ the ports of v is the ordinal $P_v \stackrel{\text{def}}{=} ar(ctrl(v))$; then P_V is the sum $\sum_{v\in V} P_v$.

We call $X + P_V$ the *points* of A, and $Y + E$ its *links*. For a point p, if $link(p) = l$ then we say that the link l *contains* p. Distinct points in the same link are called *peers*. We call a link *idle* if it contains no points. A link graph is *lean* if it has no idle edges. A link is *open* if it is a name, and *closed* if it is an edge. The attributes 'open' and 'closed' also apply to the points in the links. ■

The requirement that the supply of names is infinite is technically important; it means that whatever collection of interfaces we are considering for a particular purpose, it is always possible to choose a *fresh* name, that is, a name that does not occur in any of the given interfaces.

Figure 4.6 shows an example of a link graph. It has inner names x_1, x_2 and x_3, nodes $v_1, \ldots, v_6$, and outer names y_1 and y_2. There are three closed links, two of which contain inner names x_1 and x_3, respectively, and one that connects v_1 and v_5. There is one open link, namely the outer name y_1; the outer name y_2 is idle.

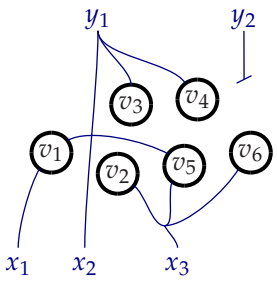


Figure 4.6 Example of a link graph

We can compose link graphs in a manner completely analogous to that for place graphs. We detail the construction for completeness. Given two link graphs

$$A = (V_A, E_A, ctrl_A, link_A) : X \to Y$$
$$B = (V_B, E_B, ctrl_B, link_B) : Y \to Z\,,$$

provided $V_A \cap V_B = \emptyset$ and $E_A \cap E_B = \emptyset$, we can form the composition $B \circ A : X \to Z$ as follows: For the node and edge sets we take the disjoint unions $V_A \uplus V_B$ and $E_A \uplus E_B$ of the component sets, and for the control map we take

$$ctrl_A \uplus ctrl_B : V_A \uplus V_B \to \mathcal{K}\,.$$

The link map *link* will map a point p that is either an inner name in X or a port in P_{V_A} to $link_A(x)$, unless this yields a name y in Y, in which case the result will be $prnt_B(y)$. For a port p in P_{V_B} it should map p to $link_B(p)$. More precisely, we obtain *link* as the vertical composition of maps in the diagram

$$\begin{array}{ccccc}
 & & Z+(E_A \uplus E_B) & & \\
 & & \uparrow & & \\
E_A & \longrightarrow & E_A+(Z+E_B) & \longleftarrow & Z+E_B \\
Id_{E_A}\uparrow & & \uparrow g & & \uparrow link_B \\
E_A & \longrightarrow & E_A+(Y+P_{V_B}) & \longleftarrow & Y+P_{V_B} \\
 & & \uparrow & & \\
Y+E_A & \longrightarrow & (Y+E_A)+P_{V_B} & \longleftarrow & P_{V_B} \\
link_A\uparrow & & \uparrow f & & \uparrow Id_{V_B} \\
X+P_{V_A} & \longrightarrow & (X+P_{V_A})+P_{V_B} & \longleftarrow & P_{V_B} \\
 & & \uparrow & & \\
 & & X+(P_{V_A \uplus V_B})\,; & &
\end{array}$$

again, the horizontal arrows are injections, and the anonymous vertical arrows bijections. In approximate notation,

$$link = (Id_{E_A} + link_B) \circ (link_A + Id_{V_B}) .$$

Definition 4.5 The s-category $\hat{}\mathbf{LIG}(\Sigma)$ has finite sets of names as objects and concrete link graphs as arrows. The support $|A|$ of a link graph A is the indexed sum $V + E$ of its nodes V and its edges E. The identity at X is the empty link graph with identity link map, and composition is defined as detailed above. ■

It is straightforward to check that $\hat{}\mathbf{LIG}(\Sigma)$ is a well-defined s-category.

For any two name sets X_0 and X_1 the tensor product $X_0 \otimes X_1$ is defined iff $X_0 \cap X_1 = \emptyset$, and is then given as the disjoint union $X_0 \uplus X_1$. Given two link graphs

$$A_i = (V_i, E_i, ctrl_i, link_i) : X_i \to Y_i \qquad (i = 0, 1) ,$$

the tensor product $A_0 \otimes A_1 : X_0 \otimes X_1 \to Y_0 \otimes Y_1$ is defined iff $X_0 \otimes X_1$ and $Y_0 \otimes Y_1$ are defined and $V_0 \cap V_1 = E_0 \cap E_1 = \emptyset$; it is then given as follows: Its nodes are $V_0 \uplus V_1$; its edges are $E_0 \uplus E_1$; its control map is

$$ctrl_0 \uplus ctrl_1 : V_0 \uplus V_1 \to \mathcal{K} ;$$

and its link map *link* is the unique arrow that makes

$$\begin{array}{ccccc} Y_0 + E_0 & \longrightarrow & (Y_0 \otimes Y_1) + (E_0 \uplus E_1) & \longleftarrow & Y_1 + V_1 \\ \uparrow{\scriptstyle link_0} & & \uparrow{\scriptstyle link} & & \uparrow{\scriptstyle link_1} \\ X_0 + P_{V_0} & \longrightarrow & (X_0 \otimes X_1) + P_{V_0 \uplus V_1} & \longleftarrow & X_1 + P_{V_1} \end{array}$$

commute, where the horizontal arrows are the evident injections.

It is straightforward to verify that $\hat{}\mathbf{LIG}(\Sigma)$ and tensor product (with unit $\emptyset$) form a strict monoidal s-category.

4.4 Bigraphs

We are now ready for the main definition:

Definition 4.6 (bigraph) A *(concrete, basic, pure) bigraph* over Σ takes the form

$$A = (V, E, ctrl, prnt, link) : I \to J ,$$

where $I = \langle m, X \rangle$ is its *inner face* and $J = \langle n, Y \rangle$ its *outer face*, and where the substructures

$$A^{\mathrm{P}} \stackrel{\text{def}}{=} (V, ctrl, prnt) : m \to n$$
$$A^{\mathrm{L}} \stackrel{\text{def}}{=} (V, E, ctrl, link) : X \to Y$$

form, respectively, a place graph and a link graph over Σ. ■

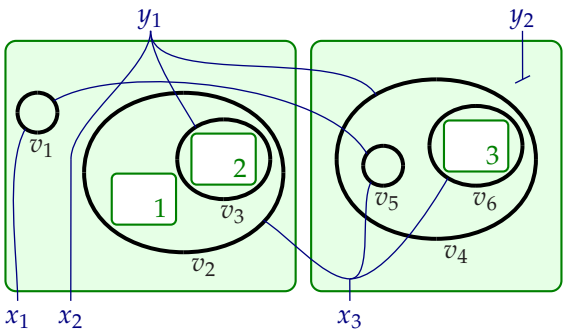


Figure 4.7 The bigraph formed by combining the place graph in Figure 4.5 and the link graph in Figure 4.6

Thus, whenever a place graph $A^{\mathrm{P}} : m \to n$ and a link graph $A^{\mathrm{L}} : X \to Y$ have identical node sets and control maps, they combine to form a bigraph $A : \langle m, X\rangle \to \langle n, Y\rangle$; we then refer to A as the *combination* of A^{P} and A^{L}, and denote it by $\langle A^{\mathrm{P}}, A^{\mathrm{L}}\rangle$. For example, the place graph in Figure 4.5 and the link graph in Figure 4.6 combine to form the bigraph shown in Figure 4.7. When relationships and attributes such as ancestry, peerness, openness, leanness, etc., hold of (the constituents of) A^{P} or A^{L}, then we also take them to hold of (the constituents of) A. Thus, we naturally talk of, say, A being lean, a link being open in A, or a place having a particular ancestor in A.

The fact that a bigraph A consists of an underlying pair $\langle A^{\mathrm{P}}, A^{\mathrm{L}}\rangle$ makes it easy to compose bigraphs: For bigraphs $A = \langle A^{\mathrm{P}}, A^{\mathrm{L}}\rangle : H \to I$ and $B = \langle B^{\mathrm{P}}, B^{\mathrm{L}}\rangle : I \to J$ the composition $B \circ A : H \to J$ is defined iff both of the compositions $B^{\mathrm{P}} \circ A^{\mathrm{P}}$ and $B^{\mathrm{L}} \circ A^{\mathrm{L}}$ are defined, and it is then given as their combination $\langle B^{\mathrm{P}} \circ A^{\mathrm{P}}, B^{\mathrm{L}} \circ A^{\mathrm{L}}\rangle$.

Definition 4.7 The s-category $\hat{}\mathbf{BIG}(\Sigma)$ has interfaces as objects and concrete, basic, pure bigraphs as arrows. The support $|A|$ of a bigraph A is the indexed sum $V + E$ of its nodes V and its edges E. The identity at X is the empty bigraph with identity place- and link-map; composition is defined by composition of the constituent place- and link-graphs, as described above. ■

The check that $\hat{}\mathbf{BIG}(\Sigma)$ is a well-defined s-category is routine.

For bigraphs $A_i : I_i \to J_i$ $(i = 0, 1)$ the tensor product is defined iff both of the products $A_0^{\mathrm{P}} \otimes A_1^{\mathrm{P}}$ and $A_0^{\mathrm{L}} \otimes A_1^{\mathrm{L}}$ are defined, and it is then given as their combination $\langle A_0^{\mathrm{P}} \otimes A_1^{\mathrm{P}}, A_0^{\mathrm{L}} \otimes A_1^{\mathrm{L}}\rangle$. It is straightforward to verify that $\hat{}\mathbf{BIG}(\Sigma)$ and tensor product (with unit $\epsilon \stackrel{\text{def}}{=} \langle 0, \emptyset\rangle$) form a strict monoidal s-category.

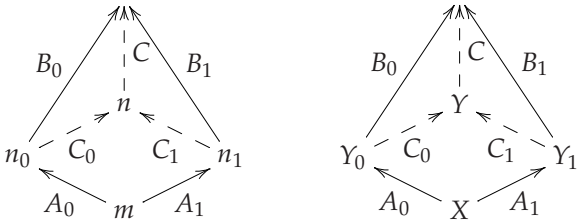
For an interface $I = \langle m, X\rangle$ we call m its *width*, and sometimes denote it by $width(I)$. For a bigraph $A : I \to J$ we call $width(I)$ its *inner width* and $width(J)$ its *outer width* (or sometimes simply its *width*). We let $width(A)$ denote the function between ordinals $width(I)$ and $width(J)$ that sends every site $s \in width(I)$ to the unique root $r \in width(J)$ which is an ancestor of s in A. Thus, *width* constitutes

a functor into the category of ordinals and functions.

4.5 Relative pushouts

In this section we review results showing that the s-category $\hat{}\mathbf{BIG}(\Sigma)$ of concrete, basic, pure bigraphs has all RPOs. This property is crucial when we want to use $\hat{}\mathbf{BIG}(\Sigma)$ as the basis for deriving contextual transitions whose bisimilarity is a congruence. To facilitate further study of the transition systems we obtain, we also give characterizations of the IPOs that may arise from a given span of bigraphs (an agent and a redex). Just as composition of bigraphs was defined in terms of composition of the constituent place- and link-graphs, so RPOs and IPOs of bigraphs are formed by performing the constructions separately on the place- and link-graphs. We therefore give a series of propositions that characterize RPOs and IPOs in each s-category. We shall omit the proofs; they can all be found in [13] along with illustrative examples.

We first characterize RPOs in $\hat{}\mathbf{PLG}(\Sigma)$ and $\hat{}\mathbf{LIG}(\Sigma)$. The characterizations assume commuting squares as indicated by the solid arrows in the diagrams below, and show how to construct RPOs $(\vec{C}, C)$ as indicated by the dashed arrows.



The constructions for the two categories are very similar, but a little involved; the reader may find the following informal description helpful:

In both cases the idea is to choose $\vec{C}$ to be "as small as possible" for the inner squares to commute, and to have the parent or link maps of $(\vec{C}, C)$ mimic those of $\vec{B}$. When we say "as small as possible," we mean that

- each C_i should have only the necessary nodes (and edges), that is, only those that occur in $A_{\bar{i}}$ but not in A_i;
- each C_i should be as disjoint as possible; that is, it should only make places siblings, or make points peers, when this is required for commutativity;
- in the case of link graphs, each C_i should be as open as possible; that is, it should only close a link when this is required for commutativity.

Proposition 4.8 (Place graph RPOs) Every commutative square in $\hat{}\mathbf{PLG}(\Sigma)$ has an RPO. For the square indicated by the solid arrows in the diagram above on the left, its RPO $(\vec{C}, C)$ is characterized up to a bijection on m as follows:

For $i = 0, 1$, let A_i have nodes U_i, let B_i have nodes V_i, let C_i have nodes W_i, let C have nodes W, let

$$P_i \stackrel{\text{def}}{=} \{r \in n_i \mid B_i(r) \notin U_0 \cup U_1\} ,$$

and let $\equiv$ be the least equivalence on $P_0 + P_1$ that relates $r_0 \in P_0$ with $r_1 \in P_1$ whenever $A_0(w) = r_0$ and $A_1(w) = r_1$ for some $w \in m + (U_0 \cap U_1) \setminus U$. Then

$$n \cong (P_0 + P_1)/\equiv$$

and, writing $\hat{r}$ for the place in n that corresponds to the $\equiv$-equivalence class of $r \in P_i$,

$$\begin{aligned}
W_i &= U_{\bar{\imath}} \setminus U_i \\
W &= V_0 \cap V_1 \\
\text{for } r \in n_i : \quad C_i(r) &= \begin{cases} \hat{r} & \text{if } r \in P_i \\ B_i(r) & \text{if } r \notin P_i \end{cases} \\
\text{for } v \in V \setminus V_i : \quad C_i(v) &= \begin{cases} \hat{r} & \text{if } A_{\bar{\imath}}(v) = r \in n_{\bar{\imath}} \\ B_i(v) & \text{if } A_{\bar{\imath}}(v) \notin n_{\bar{\imath}} \end{cases} \\
\text{for } \hat{r} \in n : \quad C(\hat{r}) &= B_i(r) \quad \text{where } r \in P_i \\
\text{for } v \in W \setminus V : \quad C(v) &= B_i(v) .
\end{aligned}$$

■

Proposition 4.9 (Link graph RPOs) Every commutative square in $\hat{}\mathbf{LIG}(\Sigma)$ has an RPO. For the square indicated by the solid arrows in the diagram above on the right, its RPO $(\vec{C}, C)$ is characterized up to a bijection on Y as follows: For $i = 0, 1$, let A_i have nodes U_i and edges E_i, let B_i have nodes V_i and edges F_i, let C_i have nodes W_i and G_i, let C have nodes W and edges G, let

$$Y_i' = \{y \in Y_i \mid B_i(y) \notin F\} ,$$

and let $\equiv$ be the least equivalence on $Y_0' + Y_1'$ that relates $y_0 \in Y_0'$ with $y_1 \in Y_1'$ whenever $A_0(p) = y_0$ and $A_1(p) = y_1$ for some $p \in X + P_{U_0 \cap U_1}$. Then

$$Y \cong (Y_0' + Y_1')/\equiv$$

and, writing $\hat{y}$ for the place in Y that corresponds to the $\equiv$-equivalence class

of $y \in Y_i'$,

$$
\begin{aligned}
W_i &= U_{\bar{\imath}} \setminus U_i \\
W &= V_0 \cap V_1 \\
G_i &= E_{\bar{\imath}} \setminus E_i \\
G &= F_0 \cap F_1 \\
\text{for } y \in Y_i: \qquad C_i(y) &= \begin{cases} \hat{y} & \text{if } y \in Y_i' \\ B_i(y) & \text{if } y \notin Y_i' \end{cases} \\
\text{for } p \in P_{V \setminus V_i}: \qquad C_i(p) &= \begin{cases} \hat{y} & \text{if } A_{\bar{\imath}}(p) = y \in Y_{\bar{\imath}} \\ B_i(p) & \text{if } A_{\bar{\imath}}(p) \notin Y_{\bar{\imath}} \end{cases} \\
\text{for } \hat{y} \in Y: \qquad C(\hat{y}) &= B_i(y) \quad \text{where } y \in Y_i' \\
\text{for } p \in P_{W \setminus V}: \qquad C(p) &= B_i(p)\,.
\end{aligned}
$$

■

We then use the characterizations in $\hat{}\mathbf{PLG}(\Sigma)$ and $\hat{}\mathbf{LIG}(\Sigma)$ to characterize RPOs among bigraphs:

Proposition 4.10 (Bigraph RPOs) Every commutative square in $\hat{}\mathbf{BIG}(\Sigma)$ has an RPO. Each RPO arises from RPOs in $\hat{}\mathbf{PLG}(\Sigma)$ and $\hat{}\mathbf{LIG}(\Sigma)$; that is, $(\vec{C}, C)$ is an RPO of $\vec{A}$ to $\vec{B}$ in $\hat{}\mathbf{BIG}(\Sigma)$ iff $(\vec{C}^{\mathrm{P}}, C^{\mathrm{P}})$ is an RPO of $\vec{A}^{\mathrm{P}}$ to $\vec{B}^{\mathrm{P}}$ in $\hat{}\mathbf{PLG}(\Sigma)$ and $(\vec{C}^{\mathrm{L}}, C^{\mathrm{L}})$ is an RPO of $\vec{A}^{\mathrm{L}}$ to $\vec{B}^{\mathrm{L}}$ in $\hat{}\mathbf{LIG}(\Sigma)$. ■

We now turn to the characterization of IPOs. The IPOs of a span $\vec{A}$ comprise, essentially, the set obtained by taking the RPO of every possible bound $\vec{B}$ of $\vec{A}$. This set will be empty if no bound exists for $\vec{A}$, so let us first consider under what conditions a bound exists. We say that the span $\vec{A}$ is *consistent* if it has a bound.

Proposition 4.11 A span $\vec{A} : m \to \vec{n}$ of place graphs is consistent iff it satisfies the following conditions:

(1) Each shared node v is assigned the same control in both of the A_i;

(2) If $A_i(w)$ is a shared node, then the place w is either a site $s \in m$ or a shared node, and in either case $A_{\bar{\imath}}(w) = A_i(w)$;

(3) If w is a shared place and $A_i(w)$ is an unshared node, then $A_{\bar{\imath}}(w)$ is a root $r \in n_{\bar{\imath}}$, and if also $A_{\bar{\imath}}(w') = A_{\bar{\imath}}(w)$, then the place w' is shared and $A_i(w') = A_i(w)$. ■

Proposition 4.12 A span $\vec{A} : X \to \vec{Y}$ of link graphs is consistent iff it satisfies the following conditions:

(1) Each shared node v is assigned the same control in both of the A_i;

(2) If $A_i(p)$ is a shared edge, then the point p is either an inner name $x \in X$ or a shared port, and in either case $A_{\bar{\imath}}(p) = A_i(p)$;

(3) If p is a shared place and $A_i(p)$ is an unshared edge, then $A_{\bar{\imath}}(p)$ is an outer name $y \in Y_{\bar{\imath}}$, and if also $A_{\bar{\imath}}(p') = A_{\bar{\imath}}(p)$, then the point p' is shared and $A_i(p') = A_i(p)$. ■

Proposition 4.13 A span $\vec{A} : I \to J$ of bigraphs is consistent iff the spans $\vec{A}^{\mathrm{P}}$ and $\vec{A}^{\mathrm{L}}$ of the constituent place- and link-graphs are consistent. ■

Now, let us suppose $\vec{A}$ is consistent and ask ourselves what its IPOs look like. Referring to the diagrams on page 50, the question is how the RPO $(\vec{C}, C)$ varies, as we vary $\vec{B}$ through the range of possible bounds.

It turns out that the only variation arises because barren roots and idle names in each of the A_i can typically be mapped into C_i in several distinct ways. For convenience, let us consider the situation in place graphs; it is analogous for link graphs. The reason barren roots are special is that the way a barren root r of A_i is mapped to a parent in C_i is not determined by the span $\vec{A}$ itself; we can get an IPO either by mapping r to a distinct root $\hat{r}$ that we include among the roots n of $\vec{C}$, or by mapping it into an arbitrary node. In the latter case we say that r is *elided (from A_i into C_i)*. Thus, an IPO is determined by choosing, for each $i = 0, 1$, a subset Q_i of the barren roots among n_i together with an *elision* η_i, which is a map from Q_i into the nodes of C_i.

Proposition 4.14 (Place graph IPOs) Let $\vec{A} : m \to \vec{n}$ be a span of consistent place graphs, and for $i = 0, 1$ let A_i have nodes U_i. The IPOs of $\vec{A}$ are exactly the cospans $\vec{C} : \vec{n} \to n$ that can be constructed as follows: For $i = 0, 1$, choose an arbitrary subset Q_i of n_i such that every $r \in Q_i$ is barren in A_i, and choose a map $\eta_i : Q_i \to U_{\bar{\imath}} \setminus U_i$. Let

$$P_i = \{r \in n_i \setminus Q_i \mid \forall w \in m + (U_0 \cap U_1) \,.\, A_i(w) = r \Rightarrow A_{\bar{\imath}}(w) \in n_{\bar{\imath}}\} \,,$$

and let $\equiv$ be the least equivalence on $P_0 + P_1$ that relates $r_0 \in P_0$ with $r_1 \in P_1$ whenever $A_0(w) = r_0$ and $A_1(w) = r_1$ for some $w \in m + (U_0 \cap U_1)$. Let

$$n \cong (P_0 + P_1)/{\equiv} \,;$$

let each C_i have nodes W_i, where

$$W_i = U_{\bar{\imath}} \setminus U_i \,;$$

and, writing $\hat{r}$ for the place in n that corresponds to the $\equiv$-equivalence class of $r \in P_i$, let

$$\text{for } r \in n_i : \qquad C_i(r) = \begin{cases} \hat{r} & \text{if } r \in P_i \\ \eta_i(r) & \text{if } r \in Q_i \\ A_{\bar{\imath}}(w) & \text{otherwise, where } w \in m + (U_0 \cap U_1) \\ & \quad \text{and } A_i(w) = r \end{cases}$$

$$\text{for } v \in U_{\bar{\imath}} \setminus U_i : \quad C_i(v) = \begin{cases} \hat{r} & \text{if } A_{\bar{\imath}}(v) = r \in n_{\bar{\imath}} \\ A_{\bar{\imath}}(v) & \text{if } A_{\bar{\imath}}(v) \notin n_{\bar{\imath}} \,. \end{cases}$$

■

The characterization of IPOs for link graphs is analogous:

Proposition 4.15 (Link graph IPOs) Let $\vec{A} : X \to \vec{Y}$ be a span of consistent link graphs, and for $i = 0, 1$ let A_i have nodes U_i and edges E_i. The IPOs of $\vec{A}$ are exactly the cospans $\vec{C} : \vec{Y} \to Y$ that can be constructed as follows: For $i = 0, 1$, choose an arbitrary subset Q_i of Y_i such that every $y \in Q_i$ is idle in A_i, and choose a map $\eta_i : Q_i \to E_{\bar{\imath}} \setminus E_i$. Let

$$Y_i' = \{y \in Y_i \setminus Q_i \mid \forall p \in X + P_{U_0 \cap U_1} \,.\, A_i(p) = y \Rightarrow A_{\bar{\imath}}(p) \in Y_{\bar{\imath}}\} \,,$$

and let $\equiv$ be the least equivalence on $Y_0' + Y_1'$ that relates $y_0 \in Y_0'$ with $y_1 \in Y_1'$ whenever $A_0(p) = y_0$ and $A_1(p) = y_1$ for some $p \in X + P_{U_0 \cap U_1}$. Let

$$Y \cong (Y_0' + Y_1')/{\equiv} \,;$$

let each C_i have nodes W_i and edges G_i, where

$$\begin{aligned} W_i &= U_{\bar{\imath}} \setminus U_i \\ G_i &= E_{\bar{\imath}} \setminus E_i \,; \end{aligned}$$

and, writing $\hat{y}$ for the name in Y that corresponds to the $\equiv$-equivalence class of $y \in Y_i'$, let

$$\text{for } y \in Y_i : \qquad C_i(y) = \begin{cases} \hat{y} & \text{if } y \in Y_i' \\ \eta_i(y) & \text{if } y \in Q_i \\ A_{\bar{\imath}}(p) & \text{otherwise, where } p \in Z + P_{U_0 \cap U_1} \\ & \quad \text{and } A_i(p) = y \end{cases}$$

$$\text{for } p \in P_{U_{\bar{\imath}} \setminus U_i} : \quad C_i(p) = \begin{cases} \hat{y} & \text{if } A_{\bar{\imath}}(p) = y \in Y_{\bar{\imath}} \\ A_{\bar{\imath}}(p) & \text{if } A_{\bar{\imath}}(p) \notin Y_{\bar{\imath}} \,. \end{cases}$$

■

Again, we take advantage of the characterizations in $\hat{}\mathbf{PLG}(\Sigma)$ and $\hat{}\mathbf{LIG}(\Sigma)$ to obtain a characterization for $\hat{}\mathbf{BIG}(\Sigma)$:

Proposition 4.16 (Bigraph IPOs) In $\hat{}\mathbf{BIG}(\Sigma)$ the IPOs are exactly those that arise from $\hat{}\mathbf{PLG}(\Sigma)$ and $\hat{}\mathbf{LIG}(\Sigma)$; that is, $\vec{C}$ is an IPO of $\vec{A}$ in $\hat{}\mathbf{BIG}(\Sigma)$ iff $\vec{C}^{\mathrm{P}}$ is an IPO of $\vec{A}^{\mathrm{P}}$ in $\hat{}\mathbf{PLG}(\Sigma)$ and $\vec{C}^{\mathrm{L}}$ is an IPO of $\vec{A}^{\mathrm{L}}$ in $\hat{}\mathbf{LIG}(\Sigma)$. ■

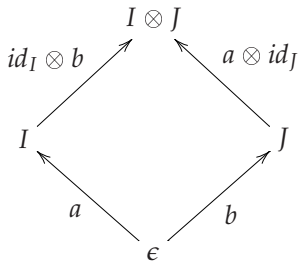
As mentioned above, the only variation among the IPOs of a given span arises because of elisions. We can therefore state the following.

Proposition 4.17 (Unique IPOs) In $\hat{}\mathbf{PLG}(\Sigma)$, $\hat{}\mathbf{LIG}(\Sigma)$ or $\hat{}\mathbf{BIG}(\Sigma)$, let $\vec{A}$ be a consistent span. Then $\vec{A}$ has an IPO $\vec{C}$ with no elisions from $\vec{A}$ into $\vec{C}$, and moreover $\vec{C}$ is determined uniquely by this property, up to an isomorphism on its outer face. In particular, if the A_i have no barren roots or idle names, then they have a unique IPO (up to isomorphism), and hence a pushout. ■

We conclude this section with a proposition about IPOs and tensor product:

Proposition 4.18 In $\hat{}\mathbf{PLG}(\Sigma)$, $\hat{}\mathbf{LIG}(\Sigma)$ or $\hat{}\mathbf{BIG}(\Sigma)$, let $\vec{C}$ be an IPO of $\vec{A}$ and $\vec{D}$ be an IPO of $\vec{B}$, where the supports of the two IPOs are disjoint. Then, provided the tensor products exist, $(C_0 \otimes D_0, C_1 \otimes D_1)$ is an IPO of $(A_0 \otimes B_0, A_1 \otimes B_1)$. ■

A particular form of such a tensorial IPO is the following, assuming a and b have disjoint support.



4.6 Binding bigraphs

In this section we present the first of the two refinements we want to apply to our basic, pure notion of bigraph. As already discussed, other refinements have been, or are likely to be, useful for various applications, and we start this section with some introductory remarks that apply generally to such refinements.

The kind of refinement we are considering always involves some *formation discipline*, i.e. some constraint on how bigraphs may be formed. As a simple example, one might impose a formation discipline that requires every root and every node in a bigraph to have an even number of nodes (and any number of sites) among its children. Under this constraint the identities are well-formed, and well-formedness is preserved by composition; therefore the well-formed bigraphs form a sub-s-category. (Moreover, since well-formedness is also preserved by tensor product, the sub-s-category is monoidal.) We shall always insist on using constraints that generate (monoidal) sub-s-categories, as this is crucial to preserving most of the structural theory of basic, pure bigraphs. Thus, for example, the constraint that every root and every node has an *odd* number of nodes among its children is excluded, as this property is not preserved by composition.

Typically a formation discipline involves some form of enrichment of the signature and of interfaces. An example in which the signature is enriched (but interfaces are not) is *atomicity*, one of the ingredients from [13] that we have left out of our basic definition of bigraph. For this discipline, we enrich the signature Σ to declare for each of its controls K whether K is *atomic* or *non-atomic*; well-formedness then requires that each atomic node is barren, a condition which is vacuously satisfied by identities and easily seen to be preserved by composition and tensor product.

The atomicity discipline is uncomplicated, in the sense that it does not affect the construction of RPOs. More precisely, as the reader may easily verify, if

$(\vec{C}, C)$ is an RPO from $\vec{A}$ to $\vec{B}$, and all of the arrows $\vec{A}$ and $\vec{B}$ respect atomicity, then so do the arrows $\vec{C}$ and C. This situation is partly due to the fact that the atomicity discipline leaves interfaces unchanged. As we said, however, many formation disciplines also enrich interfaces, as this is often necessary in order to ensure that composition preserves well-formedness. Typically, this does affect the RPO construction, but only mildly. Indeed, for the refinements we present here, we find that only little extra effort is required to characterize their RPOs.

Turning more specifically to binding, recall the example given in Section 4.1 where we introduced base nodes. Such a node had a binding port that was linked to other nodes that resided in the same surrounding node as the base node itself; thus, the *scope* of the binding was (the contents of) the surrounding node. This form of binding we shall call *outward binding*; it contrasts another form of binding that we shall find useful, namely *inward binding*, in which the scope of a binding port is (the contents of) the node on which the port is located.

(We could, in principle, make do with only outward binding, as it allows inward binding to be encoded, essentially by replacing each inward-binding port on a node boundary by a node (inside the boundary) with a single outward-binding port. Place-sorting, as introduced in the following section, can be used to ensure that exactly one such outward-binding node is introduced per inward-binding port. For convenience, however, we shall here take both forms of bindings as primitive, allowing nodes in general to have both inward- and outward-binding ports.)

We first equip signatures with binding, as follows:

Definition 4.19 (binding signature) A *(basic) binding signature* is a (basic) pure signature Σ extended with maps $\mathit{bind}, \mathit{inner} : \mathcal{K} \to \mathbb{N}$ which assign to each control $K : m$ a *binding arity* $h = \mathit{bind}(K)$ and an *inward binding arity* $i = \mathit{inner}(K)$ such that $0 \leq i \leq h \leq m$. We then say that K has *free arity* $k = m - h$ and *outward binding arity* $o = h - i$, and we write $K : (i) \to (o)k$ to indicate this arity assignment. (If $i = 0$ we may write simply $K : (o)k$, and similarly if $o = 0$ or $k = 0$ we may omit (o) or k, respectively.) ■

A node v of arity $(i) \to (o)k$ has $i + o + k$ ports. The first i of them (i.e. those with indices $0, \ldots, i-1$) we call *inward-binding ports*; the next o of them (those with indices $i, \ldots, i+o-1$) we call *outward-binding ports*; and the last k (those with indices $i+o, \ldots, i+o+k-1$) we call *free ports*. We also say that all the ports are *located* at v.

For a binding signature Σ we denote by $\mathcal{U}_\mathrm{b}(\Sigma)$ the *underlying pure signature*, i.e. the pure signature obtained by discarding the binding maps.

Definition 4.20 (binding bigraph) A *binding interface* is an interface I enriched with a *location map*

$$\mathit{loc} : X \to m + \{\perp\} ,$$

where X is the name set and m the width of I. If $\mathit{loc}(x) = s \in m$ we say that x is *located* at s, or *local* (to s); if $\mathit{loc}(x) = \perp$ we say that x is *global*. For a binding interface I we denote by $\mathcal{U}_\mathrm{b}(I)$ the underlying *pure interface* obtained by discarding the location map.

Given a (basic) binding signature Σ, a (basic) *binding bigraph* $A : I \to J$ over Σ consists of a (basic) pure bigraph $A : \mathcal{U}_b(I) \to \mathcal{U}_b(J)$ over $\mathcal{U}_b(\Sigma)$, subject to the following constraints: Declare the *binders* of A to be its local names together with the binding ports of all its nodes. Give each binder a *scope* as follows: the scope of a name located at a root r is r; the scope of an inward-binding port located at a node v is v; and the scope of an outward-binding port located at a node v is the parent of v (a node or a root). Then require A to satisfy the following condition, which we call the *scope rule*: If p is a binder with scope w (a node or a root) then every peer q of p must be located at a place u (a site or a node) such that $w >_A u$. ■

Note, in particular, that the scope rule ensures that a link can have at most one binder.

For a binding interface I with location map *loc* we often abuse notation slightly and write $I(x)$ for the location $loc(x)$ of a name x in I.

The scope rule is satisfied by the identities and preserved by composition and tensor product; thus the binding bigraphs over Σ form a monoidal sub-s-category of $\hat{}\mathbf{BIG}(\mathcal{U}_b(\Sigma))$; we denote this by $\hat{}\mathbf{BBG}(\Sigma)$. Thus, the mapping $\mathcal{U}_b$ on interfaces extends to a forgetful functor

$$\mathcal{U}_b : \hat{}\mathbf{BBG}(\Sigma) \to \hat{}\mathbf{BIG}(\mathcal{U}_b(\Sigma))$$

that embeds the binding bigraphs among the pure bigraphs. The fact that the binding discipline consists in enriching interfaces and constraining the allowable bigraphs can be clearly expressed in terms of $\mathcal{U}_b$. (Recall that a functor is *faithful* if it is injective on each homset, and *full* if it is surjective on each homset.)

Proposition 4.21 The forgetful functor $\mathcal{U}_b$ is faithful, but not full; on interfaces it is surjective, but not injective. ■

Thus, the binding bigraphs are a subset of the pure bigraphs, in the sense that the binding bigraphs are those pure bigraphs that satisfy the scope rule (according to the particular way their signature and interfaces have been enriched with binding). However, in another sense the reverse is true, namely that pure bigraphs are a proper subset of the binding bigraphs; for a pure signature can be regarded as a special case of a binding signature (in which all binding arities happen to be 0), and a pure interface as a special case of a binding interface (in which all names happen to be global). From now on, we shall work in $\hat{}\mathbf{BBG}(\Sigma)$ for arbitrary Σ, but all results will apply also in the *global* sub-s-category of $\hat{}\mathbf{BBG}(\Sigma)$—i.e., the homsets whose interfaces are global; hence, by the preceding remarks, everything applies also to pure bigraphs.

We call a binding bigraph a *globalizer* if its image under $\mathcal{U}_b$ is an identity; we use $\gamma, \delta, \varepsilon$ to range over globalizers. In a globalizer the outer face is at least as global as the inner face, in the following sense:

Proposition 4.22 In a globalizer $\gamma : I \to J$ the interfaces I and J have the same width m and names X, and whenever $J(x) = s$ for $x \in X$ and $s \in m$ then $I(x) = s$. ■

Definition 4.23 (minimally global) Let $\mathcal{A}$ be a set of bigraphs with common outer face J. We say that $\mathcal{A}$ is *minimally global* if for every globalizer γ such that every $A \in \mathcal{A}$ can be expressed in the form $\gamma \circ A'$ it holds that $\gamma = id_J$. We also say that a bigraph A is minimally global if the singleton set $\{A\}$ is minimally global. ■

Informally, this definition says that $\mathcal{A}$ is minimally global if every global name in the outer face J is forced to be global by the structure of the bigraphs in $\mathcal{A}$. The following characterization points out the two ways this forcing may occur.

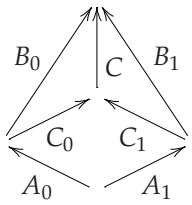
Proposition 4.24 $\mathcal{A}$ is minimally global iff for each global name x in the outer face J one of the following hold:

(a) $A(y) = x$ for some global inner name y in some $A \in \mathcal{A}$;

(b) $A(p) = B(q) = x$ for some points p and q in some (not necessarily distinct) $A, B \in \mathcal{A}$ such that the region of p in A and the region of q in B are distinct roots of J. ■

Using the notion of minimal globality, we now extend the characterization of RPOs and IPOs to binding bigraphs with the following three propositions. Again, we omit the proofs, which can be easily adapted from [13], which gives the same propositions although not formulated explicitly in terms of minimal globality.

Proposition 4.25 (binding RPO) Let Σ be a binding signature. Then $\hat{}\mathbf{BIG}(\Sigma)$ has all RPOs. Moreover, in the diagram below $(\vec{C}, C)$ is an RPO of $\vec{A}$ to $\vec{B}$ iff $\mathcal{U}_{\mathrm{b}}(\vec{C}, C)$ is an RPO of $\mathcal{U}_{\mathrm{b}}(\vec{A})$ to $\mathcal{U}_{\mathrm{b}}(\vec{B})$ and the pair $\vec{C}$ is minimally global.



■

Proposition 4.26 (consistency) A span $\vec{A}$ in $\hat{}\mathbf{BIG}(\Sigma)$ is consistent iff its image $\mathcal{U}_{\mathrm{b}}(\vec{A})$ is consistent and $\vec{A}$ satisfies the following additional condition:

If a shared point p is bound and closed in A_i but open in $A_{\bar{\imath}}$,
then $A_{\bar{\imath}}(p)$ is a local name. ■

Proposition 4.27 (binding IPO) Given a span $\vec{A} : I \to \vec{J}$ in $\hat{}\mathbf{BIG}(\Sigma)$, a cospan $\vec{C} : \vec{J} \to J$ is an IPO of $\vec{A}$ iff $\mathcal{U}_{\mathrm{b}}(\vec{C})$ is an IPO of $\mathcal{U}_{\mathrm{b}}(\vec{A})$ and the pair $\vec{C}$ is minimally global. ■

In Propositions 4.25 and 4.27 the consequent implies that $\mathcal{U}_{\mathrm{b}}(\vec{C})$ is an IPO and $\vec{C}$ is minimally global. When $\mathcal{U}_{\mathrm{b}}(\vec{C})$ is an IPO then the characterization of minimal globality (Proposition 4.24) can be strengthened for $\vec{C}$ to using only condition (a) of the proposition:

Proposition 4.28 If $\mathcal{U}_{\mathrm{b}}(\vec{C})$ is an IPO of some span $\mathcal{U}_{\mathrm{b}}(\vec{A})$ and $\vec{C} : \vec{I} \to J$ is minimally global, then for each global name x in J there is a global name y in one of the I_i such that $C_i(y) = x$. ■

4.7 Place-sorted bigraphs

The bigraph example that we discussed in the introduction required a formation discipline that ruled out non-sensical nesting such as buildings within persons. Like atomicity, which was mentioned in the previous section, this discipline constrains the parent map, and thus is really a formation discipline for place graphs. In our applications we shall need several similar place graph disciplines. Typically, unlike e.g. atomicity, they will require an enrichment of interfaces, as well as an enrichment of the signature. The general concept is defined as follows:

Definition 4.29 (Place-sorting) Let Σ be a basic signature, and let Θ be a set. A Θ-*place-sorted interface* is a basic interface I extended with a map

$$sort : width(I) \to \Theta\,,$$

which assigns a *sort* $\alpha \in \Theta$ to each place. A basic bigraph over Σ is Θ-*place-sorted* if its interfaces are Θ-place-sorted.

A *place-sorting (discipline)* for Σ is a pair

$$\mathcal{S} = (\Theta, \Phi)\,,$$

where Φ is a condition on the place graphs of Θ-place-sorted bigraphs. We require Φ to be satisfied by the identities and preserved by composition and tensor product. We say that a Θ-sorted place graph *respects* $\mathcal{S}$, or is *($\mathcal{S}$-)well-sorted*, if it satisfies Φ.

We call the signature Σ' obtained by extending Σ with $\mathcal{S}$ a *place-sorted signature*, and we denote by $\mathbf{BBG}(\Sigma')$ the s-category of Θ-sorted interfaces and $\mathcal{S}$-well-sorted bigraphs. ■

We said above that a typical place-sorting will enrich both the signature and interfaces, but the definition does not make explicit provision for an enrichment of the signature. However, in many applications such an enrichment will be inherent in the sorting condition Φ; a typical statement of Φ first declares certain attributes of the various controls in the signature and then expresses the constraint itself in terms of these attributes.

For a place-sorted signature Σ with sorting $\mathcal{S}$ we denote by $\mathcal{U}_{\mathcal{S}}(\Sigma)$ the *underlying basic signature* obtained by discarding $\mathcal{S}$. This readily extends to a forgetful functor

$$\mathcal{U}_{\mathcal{S}} : \hat{\mathbf{}}\mathbf{BBG}(\Sigma) \to \hat{\mathbf{}}\mathbf{BBG}(\mathcal{U}_{\mathcal{S}}(\Sigma)) ,$$

which we call a *sorting functor*; it discards the sort maps from interfaces and acts as an embedding of the well-sorted bigraphs.

Proposition 4.30 A sorting functor is faithful, but not necessarily full; on interfaces it is surjective, but not necessarily injective. ∎

From now on we assume a fixed, but arbitrary, $\mathcal{S}$-sorted signature Σ with controls $\mathcal{K}$ and $\mathcal{S} = (\Theta, \Phi)$.

The definitions above describe how sorting is added to a basic, i.e. unsorted, signature. Such a signature can be equivalently regarded as a special case of a sorted signature, namely one subject to the *trivial sorting* whose sort set is a singleton and whose sorting condition is *true* (i.e., satisfied by every bigraph). In view of this, it is not difficult to see how the definitions could be modified so as to add a sorting $\mathcal{S} = (\Theta, \Phi)$ to a signature already containing a (possibly non-trivial) sorting $\mathcal{S}' = (\Theta', \Phi')$: The resulting sorting, which we call the *combination* of $\mathcal{S}$ and $\mathcal{S}'$, has sorts $\Theta \times \Theta'$, and its sorting condition is the conjunction $\Phi \wedge \Phi'$. We then have sorting functors

$$\hat{\mathbf{}}\mathbf{BBG}(\Sigma) \xrightarrow{\mathcal{U}_{\mathcal{S}}} \hat{\mathbf{}}\mathbf{BBG}(\mathcal{U}_{\mathcal{S}}(\Sigma)) \xrightarrow{\mathcal{U}_{\mathcal{S}'}} \hat{\mathbf{}}\mathbf{BBG}(\mathcal{U}_{\mathcal{S}'}(\mathcal{U}_{\mathcal{S}}(\Sigma))) .$$

Thus, when we do not state otherwise, we shall allow a sorting $\mathcal{S}$ to apply also to non-basic underlying bigraphs, and we then take $\mathcal{U}_{\mathcal{S}}$ to denote the functor that discards only $\mathcal{S}$ from the combined sorting.

We call an $\mathcal{S}$-well-sorted bigraph a *resorting (for $\mathcal{S}$)* if its image under $\mathcal{U}_{\mathcal{S}}$ is an identity; we shall use ϕ, ψ to range over resortings.

Definition 4.31 (minimally sorted) Let $\mathcal{A}$ be a set of well-sorted bigraphs with common outer face J. We say that $\mathcal{A}$ is *minimally sorted* if for every resorting ϕ such that every $A \in \mathcal{A}$ can be expressed in the form $\phi \circ A'$ it holds that ϕ is an iso. We also say that a well-sorted bigraph A is minimally sorted if the singleton set $\{A\}$ is minimally sorted. ∎

It turns out that not every sorting Σ provides a suitable basis for the behavioural theory that we develop in the next chapter; in particular, $\hat{\mathbf{}}\mathbf{BBG}(\Sigma)$ may lack RPOs. For this reason, we impose some mild conditions on the sortings we shall consider:

Definition 4.32 A sorting $\mathcal{S}$ is *safe* if it satisfies the following conditions:

(1) For every well-sorted bigraph A there is a minimally sorted A_0 and a resorting ϕ such that $A = \phi \circ A_0$;

(2) Whenever $\mathcal{U}_{\mathcal{S}}(A) = P \otimes Q$ there are unique B and C such that $\mathcal{U}_{\mathcal{S}}(B) = P$, $\mathcal{U}_{\mathcal{S}}(C) = Q$, and $A = B \otimes C$;

(3) Whenever $\mathcal{U}_{\mathcal{S}}(A) = Q \circ P$ there are (not necessarily unique) B and C such that $\mathcal{U}_{\mathcal{S}}(B) = P$, $\mathcal{U}_{\mathcal{S}}(C) = Q$, and $A = C \circ B$;

(4) Whenever B is minimally sorted and $\mathcal{U}_{\mathcal{S}}(A) = Q \circ \mathcal{U}_{\mathcal{S}}(B)$, where A and B have the same inner face, then there is a unique C such that $\mathcal{U}_{\mathcal{S}}(C) = Q$ and $A = C \circ B$;

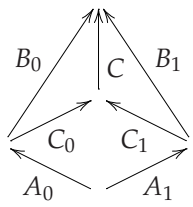
(5) Every span of resortings has a pushout whose arrows are also resortings. ■

It is straightforward to check the following:

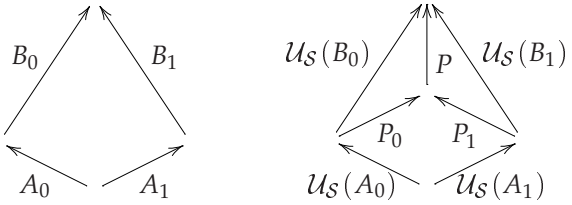
Proposition 4.33 The trivial sorting is safe, and so is the combination of any two safe sortings. ■

Safety is sufficient to ensure that $\hat{}\mathbf{BBG}(\Sigma)$ has all RPOs:

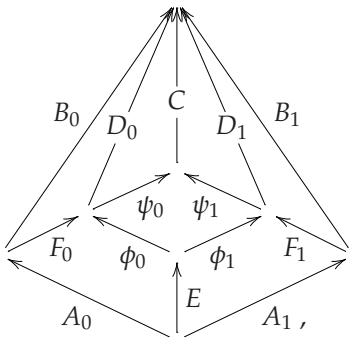
Theorem 4.34 (well-sorted RPO) Let Σ be safely sorted. Then $\hat{}\mathbf{BBG}(\Sigma)$ has all RPOs. Moreover, in the diagram below $(\vec{C}, C)$ is an RPO of $\vec{A}$ to $\vec{B}$ iff $\mathcal{U}_{\mathcal{S}}(\vec{C}, C)$ is an RPO of $\mathcal{U}_{\mathcal{S}}(\vec{A})$ to $\mathcal{U}_{\mathcal{S}}(\vec{B})$ and the pair $\vec{C}$ is minimally sorted.



Proof To show that $\hat{}\mathbf{BBG}(\Sigma)$ has all RPOs, we consider an arbitrary commutative square as shown below on the left and construct an RPO $(\vec{C}, C)$ of $\vec{A}$ to $\vec{B}$. Since $\hat{}\mathbf{BBG}(\mathcal{U}_{\mathcal{S}}(\Sigma))$ has all RPOs, there is an RPO $(\vec{P}, P)$ of $\mathcal{U}_{\mathcal{S}}(\vec{A})$ to $\mathcal{U}_{\mathcal{S}}(\vec{B})$ in $\hat{}\mathbf{BBG}(\mathcal{U}_{\mathcal{S}}(\Sigma))$, as shown on the right.

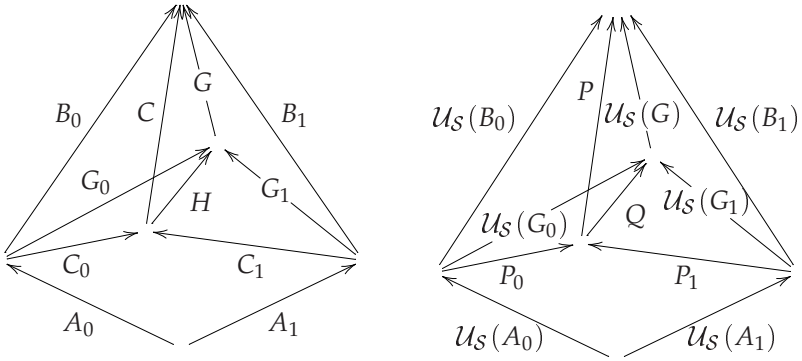


We fill in the diagram on the left as



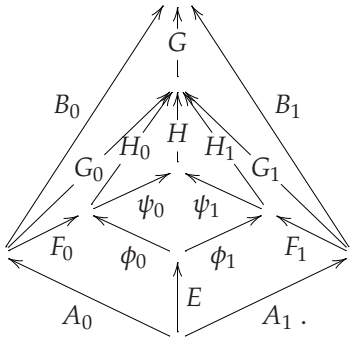
where the arrows on the inside arise as follows: $\vec{F}$ and E are minimally sorted pre-images of $\vec{P}$ and $\vec{P} \circ \mathcal{U}_{\mathcal{S}}(\vec{A})$, respectively; $\vec{D}$ are the unique pre-images of P making the outside triangles commute; $\vec{\phi}$ are resortings making the lower squares and the square consisting of $\vec{\phi}$ and $\vec{D}$ commute; $\vec{\psi}$ form a pushout of $\vec{\phi}$, thus making the small square commute; and C is the unique arrow making the inside triangles commute. For $i = 0, 1$, let $C_i \stackrel{\text{def}}{=} \psi_i \circ F_i$. Then $(\vec{C}, C)$ forms a relative bound of $\vec{A}$ to $\vec{B}$.

To show that our relative bound is an RPO, suppose that $(\vec{G}, G)$ is also a relative bound of $\vec{A}$ to $\vec{B}$. We must show that there is a unique arrow mediating between the two relative bounds, as indicated in the diagram on the left below.



The image of $(\vec{G}, G)$ in $\hat{}\mathbf{BBG}(\mathcal{U}_{\mathcal{S}}(\Sigma))$ is a relative bound of $\mathcal{U}_{\mathcal{S}}(\vec{A})$ to $\mathcal{U}_{\mathcal{S}}(\vec{B})$, and so by the RPO property of $(\vec{P}, P)$ there is a unique mediating arrow Q from $(\vec{P}, P)$ to $\mathcal{U}_{\mathcal{S}}(\vec{G}, G)$, as indicated in the diagram on the right above. Since $C_i = \psi_i \circ F_i$ for $i = 0, 1$ and each F_i is minimally sorted, there are (unique) arrows $\vec{H}$ such that $H_i \circ F_i = G_i$; and since E is minimally sorted, the square

formed by $\vec{\phi}$ and $\vec{H}$ commutes in



Since $\vec{\psi}$ is a pushout of $\vec{\phi}$ there is a (unique) arrow H, as indicated, mediating from $\vec{\psi}$ to $\vec{H}$, and hence also from $(\vec{C}, C)$ to $(\vec{G}, G)$. It is unique in doing so, for otherwise, by faithfulness of $\mathcal{U}_{\mathcal{S}}$, the arrow Q would not be unique in mediating from $(\vec{P}, P)$ to $\mathcal{U}_{\mathcal{S}}(\vec{G}, G)$. Hence, $(\vec{C}, C)$ is an RPO as claimed.

Each of the F_i is minimally sorted, and by the pushout property the pair $\vec{\psi}$ is minimally sorted. It follows that the pair $\vec{C} = \vec{\psi} \circ \vec{F}$ is minimally sorted. Moreover, by the construction of $(\vec{C}, C)$, its image under $\mathcal{U}_{\mathcal{S}}$ is an RPO. Thus, the RPO we have constructed satisfies the conditions stated in the theorem, and by the uniqueness of RPOs up to isomorphism, so does every RPO. ■

We no longer have a tight characterization of consistency. The following provides a sufficient, but not necessary, condition.

Proposition 4.35 A span $\vec{A}$ in $\hat{}\mathbf{BIG}(\Sigma)$ is consistent if its image $\mathcal{U}_{\mathcal{S}}(\vec{A})$ is consistent and each A_i is minimally sorted. ■

For a consistent span, we obtain an IPO by taking an IPO in $\hat{}\mathbf{BIG}(\mathcal{U}_{\mathcal{S}}(\Sigma))$ and equipping it with a minimal sorting:

Theorem 4.36 (well-sorted IPO) Given a span $\vec{A} : I \to \vec{J}$ in $\hat{}\mathbf{BIG}(\Sigma)$, a cospan $\vec{C} : \vec{J} \to J$ is an IPO of $\vec{A}$ iff $\mathcal{U}_{\mathcal{S}}(\vec{C})$ is an IPO of $\mathcal{U}_{\mathcal{S}}(\vec{A})$ and the pair $\vec{C}$ is minimally sorted.

Proof By faithfulness of $\mathcal{U}_{\mathcal{S}}$, the square formed by $\vec{A}$ and $\vec{C}$ commutes iff its image under $\mathcal{U}_{\mathcal{S}}$ does. By definition, $\vec{C}$ is an IPO of $\vec{A}$ iff $(\vec{C}, id)$ is an RPO of $\vec{A}$ to $\vec{C}$, and by Theorem 4.34 this holds iff $\mathcal{U}_{\mathcal{S}}(\vec{C})$ is an IPO of $\mathcal{U}_{\mathcal{S}}(\vec{A})$ and the pair $\vec{C}$ is minimally sorted. ■

From now on we shall assume a fixed signature Σ that is safely sorted, but otherwise arbitrary. In particular this includes the possibility that it is trivially sorted, and therefore all results apply also to basic bigraphs.

4.8 Abstract bigraphs

In applications of bigraphs, we shall usually be interested only in the structure (placement and linking) of bigraphs; then we want to abstract away concrete node and edge identities and work "up to support equivalence". For this purpose we define in this section the category of *abstract bigraphs* by quotienting $\hat{\mathbf{B}}\mathbf{BG}(\Sigma)$ by support equivalence $\simeq$, using Definition 3.8.

The reason we have taken great care to first set up the concrete s-category $\hat{\mathbf{B}}\mathbf{BG}(\Sigma)$ is not only that it is the basis for defining the abstract structure, but also that the abstract structure is inadequate for some purposes. As argued in the previous chapter, the derivation of contextual transitions is one such purpose.

In defining our notion of abstract bigraph we shall, in fact, abstract away not only node and edge identities but also idle edges, because we have no use for them. We therefore use the following equivalence:

Definition 4.37 (Lean support equivalence) Two parallel concrete bigraphs $A, B : I \to J$ are *lean support equivalent*, written $A \mathrel{\dot{\simeq}} B$, iff $A' \simeq B'$, where A' and B' are the bigraphs obtained from A and B by discarding all idle edges. ■

Clearly, lean support equivalence $\mathrel{\dot{\simeq}}$ is a congruence, and thus allows the following definition:

Definition 4.38 The category $\mathbf{BBG}(\Sigma)$ of *abstract bigraphs* has interfaces as objects and $\mathrel{\dot{\simeq}}$-equivalence classes of bigraphs as arrows. The *lean support quotient functor*

$$[\![-]\!] : \hat{\mathbf{B}}\mathbf{BG}(\Sigma) \to \mathbf{BBG}(\Sigma)$$

is the identity on objects and sends each concrete bigraph A to its lean support equivalence class $[\![A]\!] = [A]_{\mathrel{\dot{\simeq}}}$. ■

Why did we not save the trouble of factoring out idle edges by simply disallowing them already in the definition of concrete bigraphs? The reason is that idle edges sometimes arise from composition. (An idle edge arises in $B \circ A$ if B has an edge that contains only inner names and all these names are idle in A.) Allowing idle edges therefore smooths the structural theory of concrete bigraphs—in particular RPO and IPO constructions.

This leads naturally to another question: When we factor out idle edges, why do we keep idle *names*? The reason for this is that in applications names are sometimes made idle by reaction. (Our models of the π-calculus and the ambient calculus in later chapters are examples of this.) Our definition of reaction requires outer faces to be invariant under reaction, and therefore names that become idle as a result of reaction cannot simply be dropped.

Support equivalence $\simeq$ on concrete bigraphs is really a notion of isomorphism. (This should not be confused with isos *in* $\hat{\mathbf{B}}\mathbf{BG}(\Sigma)$; an iso in $\hat{\mathbf{B}}\mathbf{BG}(\Sigma)$ is an empty bigraph whose place and link maps are bijections.) Though we shall not do so, it is possible to represent isomorphism between bigraphs in categorical terms; for this one needs to work in the category with bigraphs as *objects*

and suitable morphisms of bigraphs as arrows. Indeed, this category holds interest, too; it corresponds closely to the categories of graphs and graph morphisms that are employed in the field of *graph rewriting* [34]. Sobociński [42] has investigated the representation of bigraphs and transitions based on IPOs in this setting.

4.9 Further structural theory and notation

In this section, which concludes the chapter on bigraph structure, we introduce some additional notation, terminology and basic results which will be useful in the following chapter on dynamics and in the applications we give in Part II. The material presented here is largely based on [13], although some definitions differ slightly to suit the present material; the proofs of all propositions are easily adapted from [13].

Open bigraphs

We say that a bigraph A is *open* if every *free link*, i.e. every link without a binding port, is open in A. The following properties of open bigraphs are easily proved:

Proposition 4.39

(1) $B \circ A$ is open iff A and B are open;

(2) $A \otimes B$ is open iff A and B are open;

(3) Every open bigraph is lean;

(4) In an IPO $\vec{C}$ of $\vec{A}$, if A_i is open then so is $C_{\bar{\imath}}$; ■

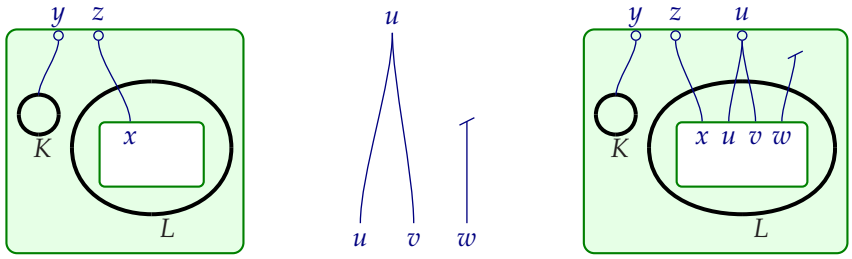
It follows from clauses (1) and 2, together with the observation that identities are open, that the open bigraphs form sub(-s)-categories of $\mathbf{BBG}(\Sigma)$ and $\hat{}\mathbf{BBG}(\Sigma)$; we denote these by $\mathbf{BBG}_{\mathrm{o}}(\Sigma)$ and $\hat{}\mathbf{BBG}_{\mathrm{o}}(\Sigma)$. By clause (4), $\hat{}\mathbf{BBG}_{\mathrm{o}}(\Sigma)$ has all RPOs.

Interfaces and ground bigraphs

Recall that (in the presence of both binding and place-sorting) an interface has the form $\langle m, X, loc, sort\rangle$, where $loc : X \to m + \{\bot\}$ and $sort : m \to \Theta$ are the location and sorting maps, respectively. It is often inconvenient to work with the maps explicitly; to avoid it, we shall often write the above interface as

$$\langle \alpha_0(X_0), \ldots, \alpha_{m-1}(X_m - 1), Y\rangle\,,$$

where for each $i \in m$ it holds that $\alpha_i = sort(i)$ and $X_i = \{x \in X \mid loc(x) = i\}$, and $Y = \{x \in X \mid loc(x) = \bot\}$. When the sorting is trivial we drop the α_i, writing just $\langle (X_0), \ldots, (X_m - 1), Y\rangle$, but if one or more consecutive X_i are empty



A : $(x) \to (yz)$ (a)

$\omega = {}^{u}/_{uv} \mid /w$: $uvw \to u$ (b)

$A \triangleleft \omega$: $(xuvw) \to (yzu)$ (c)

Figure 4.8 Examples of a prime, a wiring, and an insertion.

we usually replace the remaining components by their multiplicity, writing e.g. $\langle (X_0), 2, (X_3), Y \rangle$ for $\langle (X_0), (\emptyset), (\emptyset), (X_3), Y \rangle$. If all X_i are empty, i.e. all names are global, then we say that the interface itself is global; conversely, if all names are local (i.e. $Y = \emptyset$) we call the interface *local*. In the latter case, we write the interface as $\langle \alpha_0(X_0), \ldots, \alpha_{m-1}(X_m - 1) \rangle$, dropping the global name component from the notation.

According to these conventions, the empty interface ϵ can be variously written as e.g. 0 or $\emptyset$. We say that a bigraph is *ground* if its inner face is ϵ. We shall use lower-case letters to range over ground bigraphs, and for a ground bigraph a we shall write $a : I$ as an abbreviation of $a : \epsilon \to I$. Moreover, as a slight abuse of notation, we shall often take I to denote the (unique) empty ground bigraph with outer face I.

Primes

An interface is *prime* if it has width 1. Thus, in the notation just introduced, it has the form $\langle \alpha(X), Y \rangle$. For a local prime interface we usually drop the angle brackets, writing it in the form $\alpha(X)$ (or (X), or 1, as appropriate). For a global prime interface, however, we retain the brackets to avoid ambiguity; thus, $\langle Y \rangle$ denotes a global prime, whereas Y denotes an interface of width 0.

We say that a bigraph is *prime* if its outer face is prime and its inner face is local. Figure 4.8(a) shows an example of a prime bigraph $A : (x) \to (yz)$.

Wirings

A bigraph of width 0 is a *wiring*. Having no roots, a wiring can have no nodes and no sites, and its interfaces consist only of sets of global names. We use ω, ζ to range over wirings. Every function $\sigma : X \to Y$ represents a wiring; such wirings, which we call *substitutions*, are open. We write the empty substitution from ϵ to X as $X : \epsilon \to X$; and given vectors $\vec{x}$ and $\vec{y}$ of equal length and with

the x_i distinct, we write $\vec{y}/\vec{x}$ for the surjective substitution $x_i \mapsto y_i$. In addition to substitutions, we have for each name x the *closure* $/x : x \to \epsilon$. For $X = \{x_1, \ldots x_n\}$ we write $/X$ for the multiple closure $/x_1 \otimes \cdots \otimes /x_n : X \to \epsilon$. Given two wirings $\vec{\omega} : \vec{X} \to \vec{Y}$ with X_0 and X_1 disjoint, we write $\omega_0 \mid \omega_1 : X_0 \otimes X_1 \to Y_0 \cup Y_1$ for the wiring obtained as the union of their link maps. Figure 4.8(b) shows an example of a wiring.

Given a wiring $\omega : X \to Y$ and a local prime $A : \alpha(Z) \to \beta(W)$ the *insertion* of ω into A is defined iff X and Z are disjoint. The result, written $A \triangleleft \omega : \alpha(XZ) \to \beta(Y \cup W)$, has the nodes and parent map of A and its link map is the union of those of A and ω. Figure 4.8(c) shows an example of a bigraph obtained by such an insertion. We abbreviate $id_\alpha \triangleleft \omega$ as (ω).

Discrete bigraphs

A bigraph is *discrete* if every free link is open and has exactly one point. Hence, a discrete bigraph is open and lean, and the link map on its free points forms a bijection with its global names.

A bigraph $A : I \to J$ is *robustly discrete* if for any globalizer $\gamma : J \to J'$ the composition γA is discrete. Hence, a robustly discrete bigraph is discrete and moreover its open bound points are in bijection with its local names.

Discrete bigraphs and wirings complement each other as follows:

Proposition 4.40 Every bigraph A can be expressed uniquely (up to an isomorphism on the inner face of ω) in the form $(id_I \otimes \omega) \circ D$, where ω is a wiring, D is discrete, and I is local. ■

We call this unique decomposition of A its *discrete normal form*.

Proposition 4.41

(1) If $\vec{C}$ is an IPO of $\vec{A}$ and A_i is discrete, then so is $C_{\bar{i}}$.

(2) If $D \circ A = (\omega \otimes id_I) \circ E$ with D and E discrete, then $(D, \omega \otimes id_I)$ is an IPO of (A, E). ■

Proposition 4.42 Let $d : I$ be ground and discrete with width m. Then d has a unique factorization

$$d = d_0 \otimes \cdots \otimes d_{m-1} ,$$

where each d_i is discrete and prime. ■

Products

We define two notions of product of bigraphs that refine tensor product. The first, called *parallel product*, differs from tensor product only in allowing its arguments to share global outer names; the effect of the product is to coalesce identically named links. The second, called *prime product*, additionally coalesces all regions of the result into a single region, thus yielding a prime.

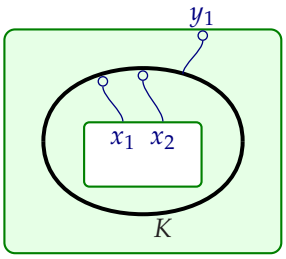


Figure 4.9 An ion $K_{x_1x_2(y_1)}$ for the control $K : 2 \to 1$

Definition 4.43 (parallel and prime products) Let $A_i : I_i \to J_i \otimes X$ $(i = 0, 1)$ be abstract bigraphs with $\vec{J}$ local. The *parallel product* $A_0 \parallel A_1$ is defined iff $I_0 \otimes I_1$ and $J_0 \otimes J_1$ are defined; it is given by

$$A_0 \parallel A_1 \stackrel{\text{def}}{=} (\sigma_0^{-1} \mid \sigma_1^{-1})(\sigma_0 A_0 \otimes \sigma_1 A_1) : I_0 \otimes I_1 \to J_0 \otimes J_1 \otimes (X_0 \cup X_1) ,$$

where $\vec{\sigma} : X \to \vec{Y}$ are any isos such that $Y_0 \cap Y_1 = \emptyset$.

Suppose moreover that all places in $\vec{J}$ have the same sort α, and let Z be the (necessarily disjoint) union of their local name sets. Then the *prime product* $A_0 \mid A_1$ is defined iff the unique empty bigraph $C : J_0 \otimes J_1 \to \alpha(Z)$ that acts as the identity on names is well-sorted; it is then given by

$$A_0 \mid A_1 \stackrel{\text{def}}{=} (C \otimes (X_0 \cup X_1)) \circ (A_0 \parallel A_1) : I_0 \otimes I_1 \to \alpha(Z) . \qquad \blacksquare$$

We earlier used the symbol '|' to denote an operation that combines wirings; in overloading the symbol to denote also prime product, we adopt the convention that the latter operation is indicated iff at least one of the two operands has width larger than zero.

Proposition 4.44 Substitutions commute with parallel and prime product; that is,

$$\begin{aligned} \sigma\,(A \parallel B) &= \sigma\, A \parallel \omega\, B \\ \sigma\,(A \mid B) &= \sigma\, A \mid \omega\, B . \end{aligned} \qquad \blacksquare$$

Ions and molecules

We shall sometimes wish to distinguish the class of nodes that have no outward-binding ports; we call such nodes *ionic*. Thus, according to the notation introduced earlier for arities, ionic nodes have arity $h \to k$ for some $h, k \geq 0$.

Given a control $K : h \to k$ and name vectors $\vec{x}$ and $\vec{y}$ of length h and k, respectively, we can form the *ion* $K_{\vec{y}(\vec{x})} : (\vec{x}) \to (\vec{y})$, which we define as the abstract bigraph consisting of a single K-node that is the parent of the single site and with bijective linking that connects the h inner-binding ports to the

inner names $\vec{x}$, and the k non-binding ports to the outer names $\vec{y}$. Figure 4.9 shows an example.

Given, moreover, a prime $P : I \to (Z\vec{x})$, the composite $(K_{\vec{y}(\vec{x})} \triangleleft id_Z) \circ P$ is a *molecule*. The insertion of id_Z handles the "extra" names Z of P. Note that while the sets $\vec{x}$ and Z are assumed to be disjoint, names may be shared among $\vec{y}$ and Z; the links represented by such shared names are coalesced in the result. For notational convenience we shall usually abbreviate the above molecule as $K_{\vec{y}(\vec{x})}\, P$. We use μ to range over molecules.

Place-sorting conditions may exclude certain ions and molecules. In particular, an atomic control $K : k$ has no ion, and the molecule $K_{\vec{y}}\, P$ is well-sorted only when P is an empty prime (Z); a molecule of this form we call an *atom*.

Chapter 5

Bigraphs: Dynamics

Having dealt with the structure of bigraphs in the preceding chapter, we now turn to their dynamics. In the first section of this chapter we introduce the notion of a (parametric) bigraphical reactive system; in the next section we add *activeness*, to account for the distinction between contexts that preserve reaction and contexts that do not. The two first sections deal with concrete bigraphs; in the third section we show how transitions can be transferred from a concrete BRS to an abstract one. Finally, the fourth section deals with an important technique, which we shall employ in the applications in Part II, for reducing substantially the number of transitions that must be considered in bisimulation proofs.

5.1 Bigraphical reactive systems

In the preceding chapter we have defined the s-category $\hat{}\mathbf{Bbg}(\Sigma)$ of concrete bigraphs, and we have argued that it is suitable as the basis of deriving contextual transitions; in particular, we have established that it possesses RPOs, provided Σ is safely sorted. (Throughout this chapter, we shall continue to assume a fixed signature Σ which is safely sorted but otherwise arbitrary.)

From the s-category $\hat{}\mathbf{Bbg}(\Sigma)$ we obtain a BRS by equipping it with a set of reaction rules, i.e. any set consisting of pairs of bigraphs with shared outer face in $\hat{}\mathbf{Bbg}(\Sigma)$. Figures 5.1–5.3 show examples of reaction rules for the systems discussed in the introduction of the previous chapter; they all model the movement of a person from one room to another, with a few differences that we discuss below.

First, we note that all of the rules are *parametric*, as the redexes and reactums have sites, shown as numbered boxes; this indicates that, in addition to the nodes shown explicitly, the room nodes can contain arbitrary constellations of further person nodes. Thus, the rules are actually *rule schemata* and represent infinite families of actual rules (called *ground rules*), each of which arises from composing both redex and reactum with a parameter (in these cases of

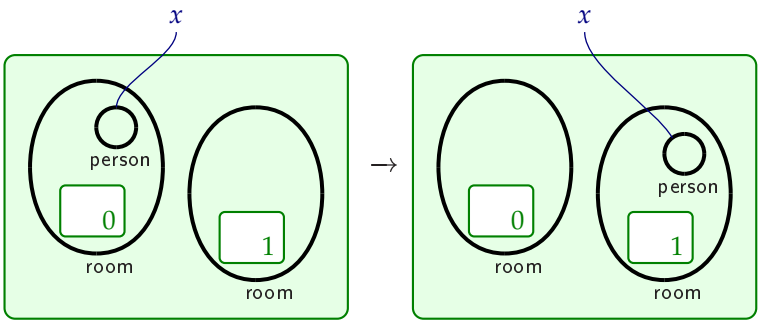


Figure 5.1 Reaction rule for person movement

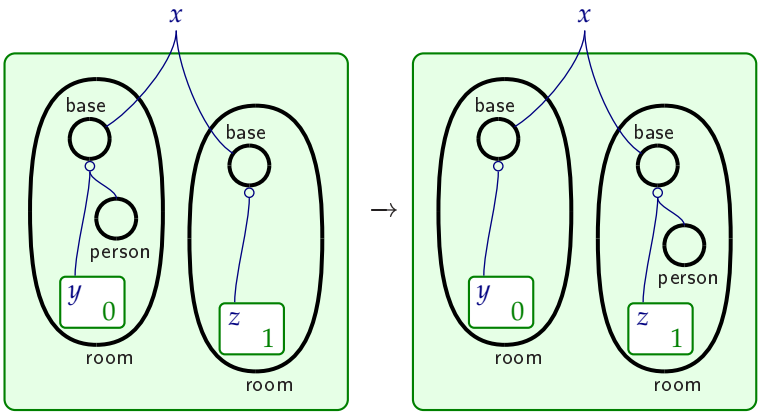


Figure 5.2 Reaction rule for person movement (with base stations)

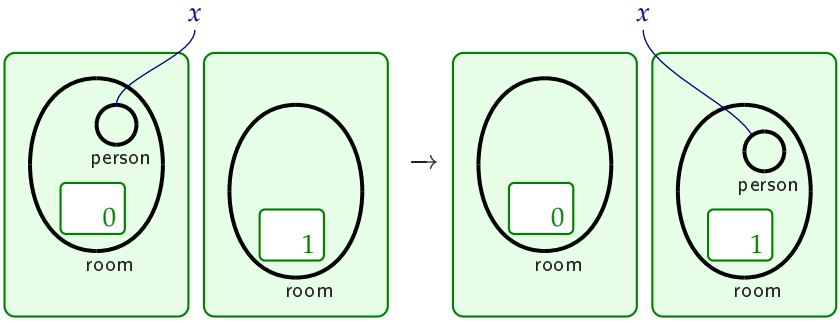


Figure 5.3 Reaction rule for "wide" person movement

width 2).

In the first two rules the redexes and reactums have (outer) width 1 and thus (necessarily) place the two room nodes in the same region. This ensures that any context in which the rules can be applied must place the two rooms within the same building node, and so the rules give rise only to reactions in which a person moves between rooms in the same building. (Reactions of this kind were shown in figures 4.2 and 4.4, and indeed those two reactions can be generated by the rules in figures 5.1 and 5.2, respectively.)

In contrast, the rule in Figure 5.3 allows also movement between rooms in separate buildings. It differs from the rule in Figure 5.1 only by having width 2 and placing the two room nodes in separate regions, thus removing the requirement that a context places the two rooms in the same building node (though it may still do so). A similar change to the rule in Figure 5.2 would allow inter-building movement in the system with base stations.

As for linking, we may note that the three rules each contains an open link in the form of the outer name x; this allows the context in which a rule is applied to add further connections (ports) to the links in question. In contrast, the bound links that appear within the room nodes in Figure 5.2 are closed and therefore not accessible to the context; they are, however, linked to the sites via the inner names y and z, respectively, and therefore accessible for further connectivity within the parameter.

In general, we denote by $\hat{}\mathbf{Bbg}(\Sigma, \mathcal{R})$ the reactive system with $\hat{}\mathbf{Bbg}(\Sigma)$ as its s-category and $\mathcal{R}$ as its set of reaction rules. We shall write $\mathrm{ST}(\Sigma, \mathcal{R})$ and $\mathrm{WST}(\Sigma, \mathcal{R})$ to denote, respectively, the standard transition system and the weak standard transition system in $\hat{}\mathbf{Bbg}(\Sigma, \mathcal{R})$.

Because $\hat{}\mathbf{Bbg}(\Sigma)$ has all RPOs, we can apply the congruence theorems 3.17 and 3.21 directly:

Corollary 5.1 (congruence) Strong and weak standard bisimilarity are congruences in $\hat{}\mathbf{Bbg}(\Sigma, \mathcal{R})$. ■

As mentioned above, we expect for many purposes to be interested mainly in abstract bigraphs and to use concrete bigraphs only as a tool in obtaining a satisfactory transition semantics. For example, in modelling a particular system—such as the process calculi we model in later chapters—one will typically build the model in $\mathbf{Bbg}(\Sigma)$ (for some suitable choice of signature Σ) and also express the reaction rules $\mathcal{R}$ in $\mathbf{Bbg}(\Sigma)$, thus forming a reactive system $\mathbf{Bbg}(\Sigma, \mathcal{R})$. One will then want to derive standard and weak standard transition systems in $\hat{}\mathbf{Bbg}(\Sigma, \hat{}\mathcal{R})$, for a suitable preimage $\hat{}\mathcal{R}$ of $\mathcal{R}$, and to map these transition systems back into $\mathbf{Bbg}(\Sigma, \mathcal{R})$ using the lean support quotient functor $[\![-]\!]$. Figure 5.4 depicts these steps. It shows also, as may be recalled from Section 3.6, that the weak standard transition system WST arises as the reflection of the (ordinary) standard transition system ST of the derived system $\mathcal{W}(\hat{}\mathbf{Bbg}(\Sigma), \hat{}\mathcal{R})$.

However, the figure leaves out a few further aspects that we must also address. First, as we have pointed out earlier, the set $\mathcal{R}$ of reaction rules will

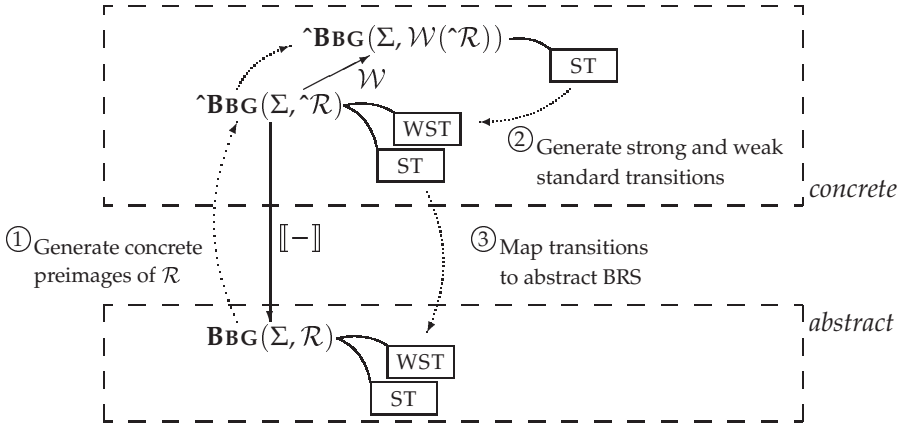


Figure 5.4 Derivation of transition systems for $\mathbf{BBG}(\Sigma, \mathcal{R})$

typically have a certain amount of structure that arises from the rules being *parametric*; we shall want to reflect this parametricity when we transfer the rules to $\hat{}\mathbf{BBG}(\Sigma)$, as this is necessary for the theory of *engaged transitions* that we develop in Section 5.4. Secondly, we have so far made no account of *activeness*, i.e. the distinction between contexts that preserve reaction and contexts that do not; we shall do this by creating suitable variants of the standard transitions systems, called *active* transition systems. The active transitions for $\hat{}\mathbf{BBG}(\Sigma)$ will themselves arise as the standard transitions of a derived system $\hat{}\mathbf{BBG}(\Sigma^{act})$ with a functor $\mathcal{U}_{act}$ into $\hat{}\mathbf{BBG}(\Sigma)$. Thus, to represent the typical situation, we refine the picture as shown in Figure 5.5.

We treat parametricity first, and deal with activeness in the next section. When a parametric reaction rule with redex R and reactum R' is applied to yield a reaction step, each site in R' must be filled with a region of the parameter. To handle this we shall use a map from the inner width of R' to the inner width of R. We call this map an *instantiation*; it will specify, for each site in R', the site in R from which the parameter region will be obtained. The map is not required to be surjective; this allows for discard of parameter regions. Neither is it required to be injective; this allows for replication.

We say that a BRS is *linear* if the instantiations of all its rules are bijections, and we say that it is *affine* if the instantiations are all injective. Thus an affine system has no replication, and a linear system has neither replication nor discard.

Definition 5.2 (instantiation) Let $I = I_0 \otimes \cdots \otimes I_{m-1}$ and $J = J_0 \otimes \cdots \otimes J_{n-1}$ be local interfaces with each I_i and J_j prime. An *instantiation* $\eta :: I \to J$ consists of a function $\bar{\eta} : n \to m$ and for each $j \in n$ an iso $\eta_j : I_{\bar{\eta}(j)} \to J_j$. For any X this determines a map

$$\eta : Gr(I \otimes X) \to Gr(J \otimes X)$$

in $\hat{}\mathbf{BBG}(\Sigma)$ as follows: For $a : I \otimes X$ express it in prime discrete normal form as

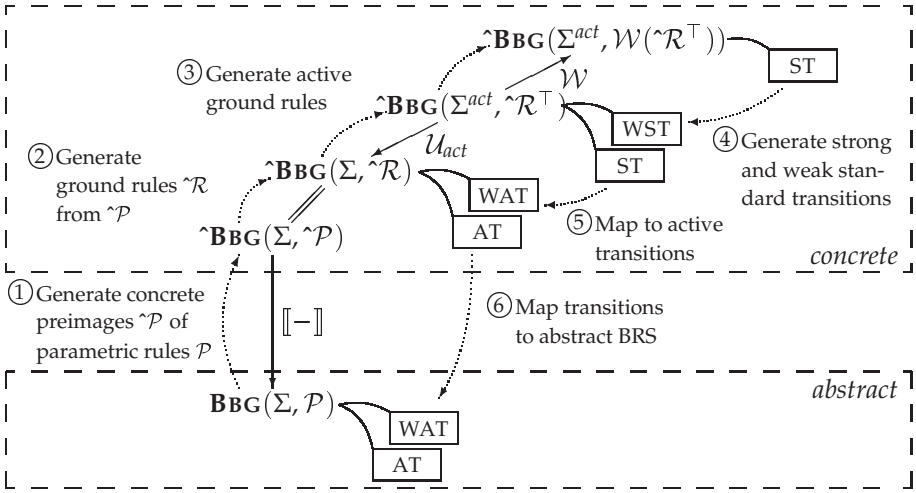


Figure 5.5 Refined derivation of transition systems for $\mathbf{BBG}(\Sigma, \mathcal{P})$

$a = (id_I \otimes \omega) \circ (d_0 \otimes \cdots \otimes d_{m-1})$; then take $\eta(a) : J \otimes X$ to be lean and satisfy

$$[\![\eta(a)]\!] = [\![\omega]\!] \left([\![\eta_0 \circ d_{\bar{\eta}(0)}]\!] \parallel \cdots \parallel [\![\eta_{n-1} \circ d_{\bar{\eta}(n-1)}]\!]\right) .$$ ■

We go to $\mathbf{BBG}(\Sigma)$ to form the parallel product for $\eta(a)$, because parallel product in $\hat{\mathbf{B}}\mathbf{BG}(\Sigma)$ requires disjoint support, and thus the right-hand side in the above equation would not be well-defined for non-injective $\bar{\eta}$ if expressed in $\hat{\mathbf{B}}\mathbf{BG}(\Sigma)$. The effect of taking the product in $\mathbf{BBG}(\Sigma)$ and transporting the result to $\hat{\mathbf{B}}\mathbf{BG}(\Sigma)$ is to force separate support instances to be chosen when multiple copies of a prime d_i occur in $\eta(a)$.

Note that the instance $\eta(a)$ is defined by manipulating only the discrete part of a. This avoids an ambiguity; for if we allowed replication of a prime with closed links, then it would not be clear if the link should be shared between the replica, or if each should have its own copy of the link. With the definition we made, any closed link will be in ω, not in the discrete primes d_i, and such a link will always be shared between replica. If the effect we want is to create a private link for each replica, then we must use links that are *bound* rather than closed. The modelling of replicated input in the π-calculus (Section 7.3) provides an example of this.

A few fundamental properties about instantiation will be useful later on. (Recall for Proposition 5.4 that $(-)^{\mathrm{R}}$ denotes reflexive closure, $(-)^*$ denotes reflexive transitive closure, and $(-)^{\mathrm{C}}$ denotes closure under arbitrary context.)

Proposition 5.3 Wiring commutes with instantiation; that is, $\omega\, \eta(a) = \eta(\omega\, a)$. ■

Proposition 5.4 Let C be a context with prime inner face and outer width m, let a and b be primes such that $C \circ a$ and $C \circ b$ are defined, and let $\eta :: m \to n$ be an

instantiation. Suppose $(a, b) \in \mathcal{S}$, where $\mathcal{S}$ is a relation closed under support translation.

(1) If a and b are open then $(\eta(C \circ a), \eta(C \circ b)) \in \mathcal{S}^{\mathrm{C},*}$;

(2) If η is injective then $(\eta(C \circ a), \eta(C \circ b)) \in \mathcal{S}^{\mathrm{C},\mathrm{R}}$. ■

The first clause requires a and b to be open; the property would fail if they were allowed to have closed links, precisely because instantiation does not replicate closed links. In the second clause a and b are unconstrained, because the condition on η prevents replication.

We are now ready for the definition of parametric reaction rule.

Definition 5.5 (parametric reaction) A *parametric reaction rule* is a triple

$$(R, R', \eta)$$

where $R : I \to J$ is the *(parametric) redex*; $R' : I' \to J$ the *(parametric) reactum*; and $\eta :: I \to I'$ an instantiation. The interfaces I and I' are required to be local.

The parametric reaction rule generates a set of ground reaction rules consisting of the minimally global pairs (r, r') such that

$$\gamma \circ r = (R \otimes id_X) \circ d$$
$$\gamma \circ r' = (R' \otimes id_X) \circ \eta(d)$$

for some discrete $d : I \otimes X$ and globalizer γ.

The bigraphical reactive system is *parametric* if its reaction rules $\mathcal{R}$ are exactly the ground rules generated from some set $\mathcal{P}$ of parametric reaction rules. ■

A parametric reaction rule specifies the exact width of the parameter, because it uses the place structure to express discard and replication. It does not similarly demand a particular set of names, apart from those needed to form links between the parameter and the parametric redex itself. By adjoining the identity id_X for an arbitrary set X of names, we allow unconstrained linking between the parameter and the context, and indeed, via shared links in the context, between points within (possibly different regions in) the parameter. It is therefore no limitation on the achievable ground reaction rules that the parameter is required to be discrete; this requirement, however, helps to simplify some of the behavioural theory we shall develop.

The role of the globalizer γ is to allow ground rules where the parameter can link to the context via local links; this provides the greatest possible generality, because the context can always make local links global, but not the other way around.

5.2 Bigraphs with activeness

In applications it is often not the case that all contexts preserve reaction. In the π-calculus, for example, the semantics does not permit reaction underneath an

input or output prefix. To capture such situations, previous work on bigraphs, e.g. [13], uses the notion of *activeness* in the definitions of reactive system and BRS. In particular, it is part of the usual definition of a signature that it declares each control either active or passive; this distinction is extended to contexts, by declaring a context active only if no site has a passive control among its ancestors. The definition of the reaction relation, then, allows a reaction $C \circ r \longrightarrow C \circ r'$ only for C active.

Here, we have so far worked with a simpler (more "economical") version of BRS that omits activeness as a primitive notion. As a result, our contexts always preserve reaction, and this means our systems are less expressive. In this section we shall show how to recover the missing expressiveness by means of a simple place-sorting discipline. The idea is to equip each BRS with a companion BRS, which is like the original except its formation rules allow only contexts that satisfy certain activeness conditions. We retain the original BRS for the general structure but obtain its dynamics from the refined BRS, which, itself, is just an instance of our "economical" (but place-sorted) kind of BRS.

As before, we presuppose a fixed, safely sorted, but otherwise arbitrary signature Σ with controls $\mathcal{K}$. Together with Σ we assume a map

$$act : \mathcal{K} \to \{\mathsf{act}, \mathsf{pas}\}$$

that declares each control to be either *active* or *passive*. We extend Σ with activeness-sorting as follows:

Definition 5.6 (activeness-sorting) The *activeness-sorting discipline* $\mathcal{S}_{act}$ has sort set

$$\Theta_{act} = \{\mathsf{act}, \mathsf{pas}\} ;$$

thus, in an activeness-sorted interface each place is declared either *active* or *passive*. The sorting condition Φ_{act} states:

every ancestor (node or root) of an active site must itself be active . ■

We denote by Σ^{act} the signature obtained by extending Σ with $\mathcal{S}_{act}$. Recall from Section 4.7 that this means the resulting sorting is obtained by combining $\mathcal{S}_{act}$ with any sorting already present in Σ. There is a sorting functor

$$\mathcal{U}_{act} : \hat{}\mathbf{BBG}(\Sigma^{act}) \to \hat{}\mathbf{BBG}(\Sigma) ,$$

which discards the activeness assignment from interfaces and acts as an embedding on bigraphs. In this section, when we talk of bigraphs and interfaces *with activeness* we mean those in $\hat{}\mathbf{BBG}(\Sigma^{act})$; similarly, we refer to those in $\hat{}\mathbf{BBG}(\Sigma)$ as being *without activeness*.

Let I be an interface without activeness and of width m. An activeness assignment for I can be represented as a set $\lambda \subseteq m$ containing exactly the active places. We use the notation I^λ for the resulting activeness-sorted interface. We say that the interface I^λ is *active* if $\lambda = m$, and *passive* if $\lambda = \emptyset$.

Proposition 5.7 Let $A : I^\lambda \to J^\mu$ be a bigraph with activeness. Then its width function satisfies $width(A)(\lambda) \subseteq \mu$. If moreover A is minimally sorted (w.r.t. activeness) then $width(A)(\lambda) = \mu$. For a resorting $\phi : I^\lambda \to I^\mu$ it holds that $\lambda \subseteq \mu$. ■

From every bigraph $A : I \to J$ without activeness we can construct a bigraph $A^\mu : I^\emptyset \to J^\mu$ with activeness for an arbitrary $\mu \subseteq n$ where $n = width(J)$. It respects activeness vacuously, because the inner face is passive. For A^n, whose outer face is active, we shall also use the notation $A^\top$.

Every ground bigraph a with activeness can be written in the form $\phi \circ {a_0}^\emptyset$ where ϕ is an activeness-resorting and $a_0 = \mathcal{U}_{act}(a)$.

The characterization of RPOs in a sorted BRS (Theorem 4.34) specializes to activeness as follows:

Theorem 5.8 (RPOs and activeness) Activeness is a safe sorting. Hence, provided Σ is safely sorted

(1) The category $\hat{}\mathbf{BBG}(\Sigma^{act})$ of bigraphs with activeness has all RPOs;

(2) $(\vec{C}, C)$ is an RPO of $\vec{A}$ to $\vec{B}$ iff $\mathcal{U}_{act}(\vec{C}, C)$ is an RPO of $\mathcal{U}_{act}(\vec{A})$ to $\mathcal{U}_{act}(\vec{B})$ and every root in the outer face of $\vec{C}$ has an active site among its descendants in one of the C_i.

(3) $(\vec{C})$ is an IPO of $\vec{A}$ iff $\mathcal{U}_{act}(\vec{C})$ is an IPO of $\mathcal{U}_{act}(\vec{A})$ and every root in the outer face of $\vec{C}$ has an active site among its descendants in one of the C_i. ■

As a corollary of this result, strong and weak standard bisimilarity are congruences in any reactive system over $\hat{}\mathbf{BBG}(\Sigma^{act})$. Our main interest, however, is not in the dynamics of $\hat{}\mathbf{BBG}(\Sigma^{act})$ itself, but rather in using it to define a notion of activeness-respecting behaviour in $\hat{}\mathbf{BBG}(\Sigma)$. The following definition, which applies to bigraphs in $\hat{}\mathbf{BBG}(\Sigma)$, will be central:

Definition 5.9 (active bigraph) Let A be a bigraph without activeness. We say that A is *active* (relative to *act*) if every node in A that has a site among its descendants is active, i.e. if its control K satisfies $act(K) = \mathsf{act}$. ■

Proposition 5.10 A bigraph $A : I \to J$ in $\hat{}\mathbf{BBG}(\Sigma)$ is active iff for every subset $\lambda \subseteq width(I)$ of its sites there exists a subset $\mu \subseteq width(J)$ of its roots and a bigraph $A_0 : I^\lambda \to J^\mu$ in $\hat{}\mathbf{BBG}(\Sigma^{act})$ such that $\mathcal{U}_{act}(A_0) = A$. ■

From $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$ we want to generate an activeness-sorted BRS in such a way that redexes are always required to occur in an active context. We do this by equipping every redex with an active outer face; thus, we shall use the rule set

$$\mathcal{R}^\top \stackrel{\text{def}}{=} \{(r^\top, (r')^\top) \mid (r, r') \in \mathcal{R}\} .$$

Thus, starting from $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$ and *act*, we generate $\hat{}\mathbf{BBG}(\Sigma^{act}, \mathcal{R}^\top)$ and map the dynamics of this derived system back into $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$. For reaction, the definition is as follows:

Definition 5.11 (active reaction) The *active reaction relation* for $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$, denoted by $\rightarrow_{act}$, is the image under $\mathcal{U}_{act}$ of reaction in $\hat{}\mathbf{BBG}(\Sigma^{act}, \mathcal{R}^{\top})$; that is, it is the smallest relation such that whenever $a \rightarrow a'$ in $\hat{}\mathbf{BBG}(\Sigma^{act}, \mathcal{R}^{\top})$ then $\mathcal{U}_{act}(a) \rightarrow_{act} \mathcal{U}_{act}(a')$.

Similarly, the *weak active reaction relation* for $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$, denoted by $\Rightarrow_{act}$, is the image under $\mathcal{U}_{act}$ of reaction in $\hat{}\mathbf{BBG}(\Sigma^{act}, \mathcal{W}(\mathcal{R}^{\top}))$ ■

Active reaction can be straightforwardly characterized in terms of $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$ itself, as follows, where we write $\mathcal{W}_{act}(\mathcal{R})$ as an abbreviation of $\mathcal{U}_{act}(\mathcal{W}(\mathcal{R}^{\top}))$.

Proposition 5.12 $a \rightarrow_{act} a'$ (resp. $a \Rightarrow_{act} a'$) iff there is a rule $(r, r') \in \mathcal{R}$ (resp. $\mathcal{W}_{act}(\mathcal{R})$) and an active context C such that $a = C \circ r$ and $a' = C \circ r'$. ■

We also want to transfer transitions from $\hat{}\mathbf{BBG}(\Sigma^{act}, \mathcal{R}^{\top})$ to $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$. For this we define the following class of transition systems:

Definition 5.13 (active transition) An *active transition system* for $\hat{}\mathbf{BBG}(\Sigma)$ is a transition system whose agents are drawn from the ground arrows of $\hat{}\mathbf{BBG}(\Sigma)$ and whose labels are drawn from the arrows of $\hat{}\mathbf{BBG}(\Sigma^{act})$. ■

We are mainly interested in the particular active transition systems whose transitions are the images of standard and weak standard transitions in $\hat{}\mathbf{BBG}(\Sigma^{act})$. In fact, we take the images of only those transitions whose labels have passive inner face. (It turns out that the resulting bisimilarities are unaffected by this restriction.)

Definition 5.14 (standard active transitions) The *standard active transition system* for $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$, denoted by $\mathrm{AT}(\Sigma, \mathcal{R})$, has all ground arrows of $\hat{}\mathbf{BBG}(\Sigma)$ as agents, and its transition relation is the least such that $\mathcal{U}_{act}(a) \xrightarrow{L} \mathcal{U}_{act}(a')$ in $\mathrm{AT}(\Sigma, \mathcal{R})$ whenever $a \xrightarrow{L} a'$ in $\mathrm{ST}(\Sigma^{act}, \mathcal{R}^{\top})$ and L has passive inner face. We refer to its associated bisimilarity as *strong active bisimilarity*.

Similarly, the *weak standard active transition system* for $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$, denoted by $\mathrm{WAT}(\Sigma, \mathcal{R})$, has all ground arrows of $\hat{}\mathbf{BBG}(\Sigma)$ as agents, and its transition relation is the least such that $\mathcal{U}_{act}(a) \stackrel{L}{\Rightarrow} \mathcal{U}_{act}(a')$ in $\mathrm{WAT}(\Sigma, \mathcal{R})$ whenever $a \stackrel{L}{\Rightarrow} a'$ in $\mathrm{WST}(\Sigma^{act}, \mathcal{R}^{\top})$ and L has passive inner face. We refer to its associated bisimilarity as *weak active bisimilarity*. ■

Henceforth, the active bisimilarities $\sim_{\mathrm{AT}}$ and $\sim_{\mathrm{WAT}}$ will often be abbreviated as $\sim$ and $\approx$, respectively.

As the label L of a transition $a \xrightarrow{L} a'$ in $\mathrm{AT}(\Sigma, \mathcal{R})$ is a context of $\hat{}\mathbf{BBG}(\Sigma^{act})$, the transition is not contextual in the sense of the discussion in Section 1.4. But it is "almost contextual" in the sense that the only extra information carried by L, compared to its image $\mathcal{U}_{act}(L)$ is the activeness assignment on its outer face.

A transition in $\mathrm{AT}(\Sigma, \mathcal{R})$ can be written in the form $a \xrightarrow{L^{\lambda}} a'$, where L is now taken to be a context of $\hat{}\mathbf{BBG}(\Sigma)$. If we view a transition in this form as a quadruple (a, L, λ, a'), then we recover exactly the notion of standard transition

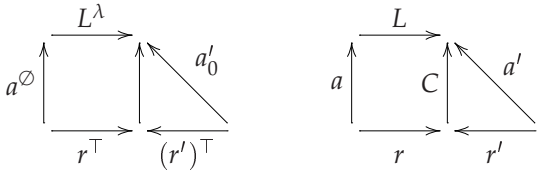
employed in [13], where the notation $a \xrightarrow{L}_{\lambda} a'$ is used. This claim is substantiated by the following theorem in which clause (3) corresponds closely to the definition of standard transitions in [13]. The theorem also relates standard active transitions to certain IPOs in $\hat{}\mathbf{Bbg}(\Sigma^{act})$.

Theorem 5.15 The following three statements are equivalent:

(1) $a \xrightarrow{L^{\lambda}} a'$ is in $\mathrm{AT}(\Sigma, \mathcal{R})$ (resp. $\mathrm{WAT}(\Sigma, \mathcal{R})$);

(2) There is a ground reaction rule $(r, r') \in \mathcal{R}$ (resp. $\mathcal{W}_{act}(\mathcal{R})$) and a commuting diagram in $\hat{}\mathbf{Bbg}(\Sigma^{act})$ as shown below on the left such that the square is an IPO and $a' = \mathcal{U}_{act}(a'_0)$;

(3) There is a ground reaction rule $(r, r') \in \mathcal{R}$ (resp. $\mathcal{W}_{act}(\mathcal{R})$) and a commuting diagram in $\hat{}\mathbf{Bbg}(\Sigma)$ as shown below on the right such that the square is an IPO and $\lambda = \mathit{width}(C)(m)$ where m is the inner width of C.



Proof The equivalence of clause (1) and (2) follows immediately from the definition of standard active transitions. For the equivalence of (2) and (3) we use the fact that an IPO of $(a^{\emptyset}, r^{\top})$ in $\hat{}\mathbf{Bbg}(\Sigma^{act})$ is related to an IPO of (a, r) in $\hat{}\mathbf{Bbg}(\Sigma)$ as detailed in Theorem 4.34. In particular, since the outer face of $r^{\top}$ is active, C must be active in order to respect activeness. Moreover, since the outer face of $a^{\emptyset}$ is passive, a root s must be active in L (i.e. $s \in \lambda$) just if s has a site as a descendant in C; the latter requirement is equivalent to $\lambda = \mathit{width}(C)(m)$. ∎

Let us discuss briefly the role of the activeness assignment λ in an active transition. Recall first that (ordinary) standard transitions are always preserved by context; that is, if $a \xrightarrow{L} a'$ is a standard transition, and if C is a context consistent with L and such that $C \circ a$ is defined, then we can infer another standard transition $C \circ a \xrightarrow{M} C' \circ a'$, where (M, C') is an IPO of (C, L). Standard *active* transitions, on the other hand, are not always preserved by context, because a context C may block an underlying redex. The activeness assignment λ in a standard active transition can be thought of as a specification of the condition under which a context C preserves the transition, namely that every site $s \in \lambda$ is given active ancestors in C.

Representing the activeness condition as part of the transition is what ensures congruence of the active bisimilarities. In [13] the congruence proof is given for the strong case, making direct use of the activeness condition. Here, having established the relationship between transitions in $\mathrm{AT}(\Sigma, \mathcal{R})$ and IPOs in $\hat{}\mathbf{Bbg}(\Sigma^{act})$, we can prove the result without explicit analysis of activeness.

Lemma 5.16 $(-)^{\mathrm{C}}$ is allowable for $\mathrm{AT}(\Sigma, \mathcal{R})$ and $\mathrm{WAT}(\Sigma, \mathcal{R})$.

Proof An easy adaptation of the proof of Lemma 3.16, using Theorem 5.15(2) to link transitions in $\mathrm{AT}(\Sigma, \mathcal{R})$ and $\mathrm{WAT}(\Sigma, \mathcal{R})$ with IPOs in $\hat{}\mathbf{BBG}(\Sigma^{act})$. ■

Theorem 5.17 (congruence) Strong and weak active bisimilarity are congruences in $\hat{}\mathbf{BBG}(\Sigma, \mathcal{R})$.

Proof By the definition of bisimulation we have $\sim_{\mathrm{AT}} \rightarrowtail_{\mathrm{AT}} \sim_{\mathrm{AT}}$. Applying Lemma 5.16 yields $(\sim_{\mathrm{AT}})^{\mathrm{C}} \rightarrowtail_{\mathrm{AT}} (\sim_{\mathrm{AT}})^{\mathrm{C}}$; that is, $(\sim_{\mathrm{AT}})^{\mathrm{C}}$ is an active bisimulation, and the result for $\sim_{\mathrm{AT}}$ follows. The argument for $\approx_{act}$ is analogous. ■

We also obtain the equivalent of Theorem 3.23 for the active bisimilarities:

Theorem 5.18

(1) Strong active bisimilarity implies weak active bisimilarity; that is, $\sim \subseteq \approx$;

(2) The standard active transition system is adequate for weak standard active transitions; that is, $\approx^{\mathrm{AT}} = \approx$.

Proof The proof proceeds analogously to that of Theorem 3.23. The crux is to show, under the assumption $a \approx^{\mathrm{AT}} b$, that every transition $a \stackrel{L^\lambda}{\Longrightarrow} a'$ in $\mathrm{WAT}(\Sigma, \mathcal{R})$ has a matching transition $b \stackrel{L^\lambda}{\Longrightarrow} b'$ also in $\mathrm{WAT}(\Sigma, \mathcal{R})$ such that $a' \approx^{\mathrm{AT}} b'$. The transition of a arises from a sequence of transitions

$$a^\emptyset = a_0 \xrightarrow{L_1} a_1 \cdots \xrightarrow{L_n} a_n = (a')^\lambda \quad \text{in } \mathrm{ST}(\Sigma_{act}, \mathcal{R}^\top)$$

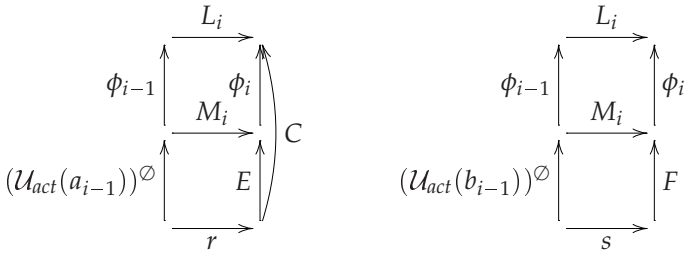
such that $L^\lambda = L_n \circ \cdots \circ L_1$. Since (for $i > 1$) the labels L_i do not necessarily have passive inner faces, we cannot relate these transitions directly to ones in $\mathrm{AT}(\Sigma, \mathcal{R})$ in order to apply the assumption $a \approx^{\mathrm{AT}} b$. Instead, we construct a sequence

$$b^\emptyset = b_0 \xrightarrow{L_1} b_1 \cdots \xrightarrow{L_n} b_n \quad \text{in } \mathrm{WST}(\Sigma_{act}, \mathcal{R}^\top) = \mathrm{ST}(\Sigma_{act}, \mathcal{W}_{act}(\mathcal{R}))$$

such that $\mathcal{U}_{act}(a_i) \approx^{\mathrm{AT}} \mathcal{U}_{act}(b_i)$ for $i = 1, \ldots, n$; this will suffice by taking $b' \stackrel{\mathrm{def}}{=} \mathcal{U}_{act}(b_n)$.

We construct the sequence for b inductively. Consider the transition $a_{i-1} \xrightarrow{L_i} a_i$. By the induction hypothesis we have $\mathcal{U}_{act}(a_{i-1}) \approx^{\mathrm{AT}} \mathcal{U}_{act}(b_{i-1})$. Moreover, there is a resorting ϕ_{i-1} such that $a_{i-1} = \phi_{i-1} \circ (\mathcal{U}_{act}(a_{i-1}))^\emptyset$ and $b_{i-1} = \phi_{i-1} \circ (\mathcal{U}_{act}(b_{i-1}))^\emptyset$. Hence, the IPO underlying the transition of a_{i-1} has the form of the outer rectangle on the left below, where (r, r') is the underlying

reaction rule and $a_i = C_i \circ r'$.



We fill in the rectangle with an RPO as shown. Since the upper square thus created is IPO, ϕ_i is a resorting, and so $\mathcal{U}_{act}(a_i) = \mathcal{U}_{act}(E \circ r')$. Hence, by Theorem 5.15 there is a transition $\mathcal{U}_{act}(a_{i-1}) \xrightarrow{M_i} \mathcal{U}_{act}(a_i)$ in $\mathrm{AT}(\Sigma, \mathcal{R})$. Then, using $\mathcal{U}_{act}(a_{i-1}) \approx^{\mathrm{AT}} \mathcal{U}_{act}(b_{i-1})$, we infer $\mathcal{U}_{act}(b_{i-1}) \stackrel{M_i}{\Longrightarrow} \mathcal{U}_{act}(b_i)$ in $\mathrm{WAT}(\Sigma, \mathcal{R})$ for some b_i such that $\mathcal{U}_{act}(a_i) \approx^{\mathrm{AT}} \mathcal{U}_{act}(b_i)$. By Theorem 5.15 there is a reaction rule (s, s') in $\mathcal{W}_{act}(\mathcal{R})$ and an IPO in the form of the lower square on the right above such that $\mathcal{U}_{act}(b_i) = \mathcal{U}_{act}(F \circ s')$. We compose this IPO with the upper IPO from the left, as shown. From the resulting IPO we infer the required transition $b_{i-1} \stackrel{L_i}{\Longrightarrow} b_i$ in $\mathrm{WST}(\Sigma_{act}, \mathcal{R}^{\top})$. ■

5.3 Transitions for abstract BRSs

We have now established all the connections required for the concrete part of Figure 5.5, and we are ready to deal with the first and last step shown in the figure, i.e. the steps that form the connections between the abstract BRS and the concrete ones.

For the first step we need to generate a suitable concrete preimage of the parametric reaction rules $\mathcal{P}$ in $\mathbf{BBG}(\Sigma, \mathcal{P})$. We do this by simply including every lean, concrete instance of the abstract rules:

$$\hat{\mathcal{P}} \stackrel{\text{def}}{=} \{(R, R', \eta) \mid ([\![R]\!], [\![R']\!], \eta) \in \mathcal{P} \text{ and } R, R' \text{ lean}\} ;$$

The following definition then gives the last step:

Definition 5.19 The *standard active transition system* in $\mathbf{BBG}(\Sigma, \mathcal{P})$, denoted by $\mathrm{AT}(\Sigma, \mathcal{P})$, is the image under $[\![-]\!]$ of the standard active transition system in $\hat{}\mathbf{BBG}(\Sigma, \hat{}\mathcal{P})$; that is, it is the least such that $[\![a]\!] \xrightarrow{[\![L]\!]^{\lambda}} [\![a']\!]$ in $\mathrm{AT}(\Sigma, \mathcal{P})$ whenever $a \xrightarrow{L^{\lambda}} a'$ in $\mathrm{AT}(\Sigma, \hat{}\mathcal{P})$.

The *weak standard active transition system* in $\mathbf{BBG}(\Sigma, \mathcal{P})$, which we denote by $\mathrm{WAT}(\Sigma, \mathcal{P})$, is similarly defined as the image under $[\![-]\!]$ of the weak standard active transition system in $\hat{}\mathbf{BBG}(\Sigma, \hat{}\mathcal{P})$. ■

We now come to the main congruence result for BRSs, namely that the (strong and weak) active bisimilarities are congruences in the abstract system $\mathbf{BBG}(\Sigma, \mathcal{P})$. This result will, of course, be based on the congruence result for the active bisimilarities in $\hat{}\mathbf{BBG}(\Sigma, \hat{}\mathcal{P})$. We first prove that the bisimilarities for $\mathrm{AT}(\Sigma, \mathcal{P})$ and $\mathrm{WAT}(\Sigma, \mathcal{P})$ coincide with the images of $\mathrm{AT}(\Sigma, \hat{}\mathcal{P})$ and $\mathrm{WAT}(\Sigma, \hat{}\mathcal{P})$, respectively:

Lemma 5.20 In $\hat{}\mathbf{BBG}(\Sigma, \mathcal{P})$ suppose $\mathcal{P}$ is closed in the sense that whenever $(R, R', \eta) \in \mathcal{P}$ and $S \eqslantless R$ then $(S, S', \eta) \in \mathcal{P}$ for some $S' \eqslantless R'$. Then

(1) $a \sim b$ in $\hat{}\mathbf{BBG}(\Sigma, \mathcal{P})$ iff $[\![a]\!] \sim [\![b]\!]$ in $\mathbf{BBG}(\Sigma, [\![\mathcal{P}]\!])$;

(2) $a \approx b$ in $\hat{}\mathbf{BBG}(\Sigma, \mathcal{P})$ iff $[\![a]\!] \approx [\![b]\!]$ in $\mathbf{BBG}(\Sigma, [\![\mathcal{P}]\!])$.

Proof

(1) For ($\Rightarrow$) we show that the relation

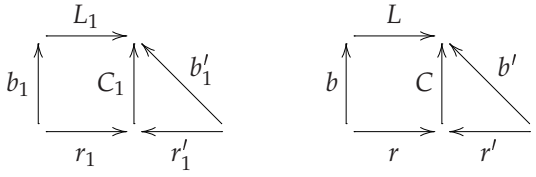
$$\mathcal{S} = \{([\![a]\!], [\![b]\!]) \mid a \sim b \text{ in } \hat{}\mathbf{BBG}(\Sigma, \mathcal{P})\}$$

is an active bisimulation in $\mathbf{BBG}(\mathcal{K}, [\![\mathcal{P}]\!])$. Let $a \sim b$, and let $[\![a]\!] \xrightarrow{L^\lambda} p'$ in $\mathrm{AT}(\Sigma, [\![\mathcal{P}]\!])$. The transition has a preimage $a \xrightarrow{M^\lambda} a'$ in $\mathrm{AT}(\Sigma, \mathcal{P})$ with $[\![M]\!] = L$ and $[\![a']\!] = p'$. By assumption there is then also a transition $b \xrightarrow{M^\lambda} b'$ in $\mathrm{AT}(\Sigma, \mathcal{P})$ for some b' such that $a' \sim b'$. The image of this transition in $\mathrm{AT}(\Sigma, [\![\mathcal{P}]\!])$ is $[\![b]\!] \xrightarrow{L^\lambda} q' \stackrel{\text{def}}{=} [\![b']\!]$, and we have $(p', q') \in \mathcal{S}$.

For ($\Leftarrow$) we show that the relation

$$\mathcal{S} = \{(a, b) \mid [\![a]\!] \sim [\![b]\!] \text{ in } \mathbf{BBG}(\Sigma, [\![\mathcal{P}]\!])\}$$

is an active bisimulation in $\hat{}\mathbf{BBG}(\Sigma, \mathcal{P})$. Let $[\![a]\!] \sim [\![b]\!]$ and $a \xrightarrow{L^\lambda} a'$ in $\mathrm{AT}(\Sigma, \mathcal{P})$. The image in $\mathrm{AT}(\Sigma, [\![\mathcal{P}]\!])$ of the transition is $[\![a]\!] \xrightarrow{[\![L]\!]^\lambda} p' \stackrel{\text{def}}{=} [\![a']\!]$. By assumption there is then also a transition $[\![b]\!] \xrightarrow{[\![L]\!]^\lambda} q'$ in $\mathrm{AT}(\Sigma, [\![\mathcal{P}]\!])$ for some q' such that $p' \sim q'$. This transition has a preimage $b_1 \xrightarrow{{L_1}^\lambda} b_1'$ in $\mathrm{AT}(\Sigma, \mathcal{P})$ with $b_1 \eqslantless b$, $L_1 \eqslantless L$ and $[\![b_1']\!] = q'$. Let (r_1, r_1') be the ground reaction rule underlying the transition of b_1; then its underlying diagram has the form shown on the left:



By Proposition 3.11 we can find $r \eqslantless r_1$ and $C \eqslantless C_1$ such that the square in the diagram on the right is also an IPO, and by the closedness assumption we can find $r' \eqslantless r_1'$ such that (r, r') is a ground rule. Then taking $b' \stackrel{\text{def}}{=} C \circ r'$, as indicated, we have $b \xrightarrow{L^\lambda} b'$ in $\mathrm{AT}(\Sigma, \mathcal{P})$ and $[\![b']\!] = [\![b_1']\!] = q'$. Recalling that $[\![a']\!] = p'$ and $p' \sim q'$, we also have $(a', b') \in \mathcal{S}$, and we are done.

(2) Using Proposition 3.11 it is straightforward to prove that the closedness property of $\mathcal{P}$ carries over to $\mathcal{W}(\mathcal{P})$. Then the proof proceeds exactly as for (1), except that one appeals to reaction rules and diagrams in $\mathcal{W}(\hat{}\mathbf{BBG}(\Sigma,\mathcal{P}))$, rather than in $\hat{}\mathbf{BBG}(\Sigma,\mathcal{P})$. ■

Theorem 5.21 (congruence) Strong and weak active bisimilarity are congruences in $\mathbf{BBG}(\Sigma,\mathcal{P})$.

Proof Let $p \sim q$ in $\mathbf{BBG}(\Sigma,\mathcal{P})$; we must show $C \circ p \sim C \circ q$ for an arbitrary context C. By its construction, $\hat{}\mathcal{P}$ is closed in the sense of the preceding lemma, so by the lemma there exist bisimilar a and b in $\hat{}\mathbf{BBG}(\Sigma,\hat{}\mathcal{P})$ such that $[\![a]\!] = p$ and $[\![b]\!] = q$. By surjectivity of $[\![-]\!]$ there exists D composable with a and b such that $[\![D]\!] = C$. But bisimilarity is a congruence in $\hat{}\mathbf{BBG}(\Sigma,\hat{}\mathcal{P})$, so we have $D \circ a \sim D \circ b$. Using Lemma 5.20 again, we then obtain $C \circ p \sim C \circ q$, as required. ■

5.4 Engaged transitions

Even though the standard active transition systems are minimal in the sense of being based on IPOs they nevertheless contain transitions which do not provide useful information about their agents and could therefore be removed without affecting the associated bisimilarities. For example, suppose an agent a and a ground redex r have no shared nodes; then the pair (a, r) have a tensorial IPO $(r \otimes id, id \otimes a)$, and hence there is a transition $a \xrightarrow{r \otimes id} a \otimes r'$, where r' is the reactum for r. Such tensorial transitions are clearly superfluous, as a matching transition can be constructed for any agent. Intuitively, a transition is interesting only if the agent contributes to the underlying reaction. In a parametric BRS we would therefore like to limit attention to those transitions where the agent and the underlying parametric redex share at least one node. We shall call such transitions *engaged*; and indeed we are able show, under certain assumptions on the BRS in question, that engaged transitions are adequate.

The first assumption we shall impose is *shallowness*:

Definition 5.22 (shallow) A bigraph A is *shallow* if $E \circ A = B \circ F$ with E and F empty implies $B = E \circ A \circ F'$ for some F' such that $F' \circ F = id$. ■

Lemma 5.23 The following conditions, taken in conjunction, are sufficient for A to be shallow:

(a) No two sites are siblings and no two inner names are peers;

(b) Every node has a site among its children and every edge contains an inner name;

(c) Every site has a node as its parent and every inner name is closed;

(d) Every local inner name is bound;

(e) $\phi \circ A = B \circ \psi$ for resortings ϕ and ψ implies that ψ is an isomorphism.

Conditions (a) and (b) are necessary for shallowness in basic bigraphical reactive systems, but not in all systems with sorting; the remaining conditions are necessary in all systems. ■

We need to impose several conditions in addition to shallowness. The class of BRS for which we shall prove adequacy of engaged transitions are the *simple* BRSs, defined as follows:

Definition 5.24 (simple BRS) A bigraph A is *simple* if it is prime, open and shallow and has no barren roots or idle names. A BRS is *simple* if it is parametric with all parametric redexes simple. ■

Among the simplicity conditions only that of primeness is really severe. None of the others restrict the "essence" of a redex, namely the prescription of a constellation of nodes, either in terms of place or link structure; instead they restrict only the ways in which interfaces of such constellations may be expressed. Primeness, on the other hand, excludes the possibility of redexes that arise purely from the link structure, something we would certainly wish to allow. The applications that appear in Part II of this dissertation, however, do fall naturally within our class of simple BRS. Moreover, the techniques we use to obtain the results in the present section rely heavily on primeness, and it remains an open (and apparently quite difficult) problem to obtain similar results for the non-prime case.

One striking property of ground primes is that they do not need global names; that is, as long as the internal structure of a ground prime obeys the scope rule, then so does the bigraph as a whole, even if every outer name is local. We can express the property as follows:

Proposition 5.25 A prime ground bigraph is minimally global iff it is local, i.e. has no global names. ■

This property is special to ground primes: in a non-ground bigraph a name x may be linked with a global inner name, and in a non-prime bigraph a name x may be linked to several ports belonging to different regions; in either case the scope rule is violated if x is local.

The definition of parametric BRS requires that every ground rule be minimally global. In a simple BRS all ground rules are prime, so for this class the requirement simply means they must be local:

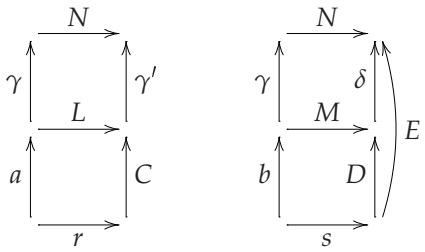
Proposition 5.26 In a simple BRS every ground rule is local. ■

For simple BRSs we now establish a bisimulation proof technique we shall call "up to globalization." It is based on the following operation on relations over ground bigraphs:

$$\mathcal{S}^{\mathrm{G}} \stackrel{\text{def}}{=} \{(a, b) \mid (\gamma \circ a, \gamma \circ b) \in \mathcal{S} \text{ for some globalizer } \gamma\} .$$

Lemma 5.27 In a simple BRS the operation $(-)^{\mathrm{G}}$ is allowable for active transitions.

Proof We must show $\mathcal{R}^{\mathrm{G}} \rightarrowtail \mathcal{S}^{\mathrm{G}}$ under the assumption $\mathcal{R} \rightarrowtail \mathcal{S}$. Let $(\gamma \circ a, \gamma \circ b) \in \mathcal{R}$ and $a \xrightarrow{L^\lambda} a'$ with $L \circ b$ defined; we seek a transition $b \xrightarrow{L^\lambda} b'$ for some b' such that $(a', b') \in \mathcal{S}^{\mathrm{G}}$. Let (r, r') be the ground reaction rule underlying the transition of a; then the underlying IPO has the form of the lower square on the left in the following diagram, and $a' = C \circ r'$.



The upper square is formed by taking an IPO (N, γ') of (γ, L). Since both the lower square and the outer square form IPOs, it follows from Proposition 4.25 L and N differ only in the locality assignments of their interfaces; moreover, since r is local, it also follows that in both L and N a name is global only if it is linked to a global inner name. The outer square underlies a transition $\gamma \circ a \xrightarrow{N^\lambda} a'_1 \stackrel{\text{def}}{=} \gamma' \circ a'$. As $(\gamma \circ a, \gamma \circ b) \in \mathcal{R}$ we then also have $\gamma \circ b \xrightarrow{N^\lambda} b'_1$ for some b'_1 such that $(a'_1, b'_1) \in \mathcal{S}$. Let (s, s') be the ground rule underlying this transition; then the underlying IPO has the form of the outer square on the right above, and $b'_1 = E \circ s'$. We take an RPO (M, D, δ) of (b, s) to $(N \circ \gamma, E)$. Since the lower inner square thus formed is an IPO it follows from Theorem 4.25 that M and N differ only in the locality assignments of their interfaces, and in M a name is global only if it is linked to a global inner name. Hence, $M = L$ and $\delta = \gamma'$, and so the lower square underlies a transition $b \xrightarrow{L^\lambda} b' \stackrel{\text{def}}{=} D \circ s'$, and we calculate

$$\begin{aligned} b'_1 &= E \circ s' \\ &= \gamma' \circ D \circ s' \\ &= \gamma' \circ b' \,; \end{aligned}$$

it follows that $(a', b') \in \mathcal{S}^{\mathrm{G}}$ as required. ∎

We now define engaged transitions formally. The definition is straightforward in concrete bigraphs where the property of shared nodes can be expressed directly. We also want to apply the notion in abstract bigraphs, and to do this we simply transfer engaged transitions using the lean support quotient functor linking the two categories.

Definition 5.28 (engaged transition) In a concrete parametric BRS a transition $a \xrightarrow{L^\lambda} a'$ is *engaged* if it can be based on a parametric redex that shares at least one node with the agent a.

In the abstract parametric BRS $\mathbf{BBG}(\Sigma, \mathcal{P})$ a transition $a \xrightarrow{L^\lambda} a'$ is *engaged* if it is the image under the lean support quotient functor of an engaged transition in $\hat{}\mathbf{BBG}(\Sigma, \hat{}\mathcal{P})$.

In either BRS we denote by PE the transition system of prime agents and engaged transitions, and by POE the sub-transition system of PE whose agents are open. ■

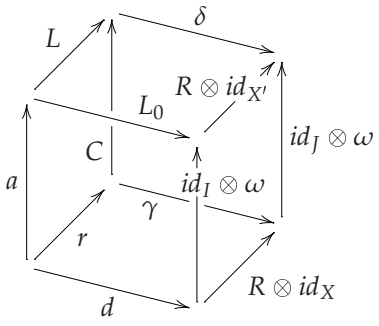
For a transition $a \xrightarrow{L^\lambda} a'$ in PE or POE the agents a and a' are both prime, and hence both interfaces of the label L are prime. It follows that λ, being a non-empty subset of the roots of L, is the singleton 1. We shall therefore omit λ and simply write such a transition in the form $a \xrightarrow{L} a'$.

To show the adequacy of engaged transitions we must show $\sim^{\mathrm{PE}} = \sim$ on ground primes (and $\sim^{\mathrm{POE}} = \sim$ on open ground primes). The crux of the proof is to show that a non-engaged transition $a \xrightarrow{L^\lambda} a'$ can be matched by any agent b, irrespective of any relationship between a and b. To facilitate this, we establish in the following lemma that non-engaged transitions fall into two classes: either their underlying IPO is tensorial, or the IPO satisfies a more involved condition that implies, roughly, that the agent is completely included within the parameter of the redex.

Lemma 5.29 Let $R : I \to J$ be a simple parametric redex, and let r be a ground redex generated from it; that is, $\gamma \circ r = (R \otimes id_X) \circ d$ for some globalizer γ and discrete parameter $d : I \otimes X$. Let a be prime and suppose it shares no nodes with R. Then any IPO (L, C) of (a, r) satisfies one of the following (up to an iso on the outer face of L and C):

(a) $L = r \otimes id$ and $C = id \otimes a$; or

(b) there is a commutative cube of the form



Proof Let (L, C) be an IPO of (a, r). We establish either (a) or (b), according to whether a has any nodes not shared with d.

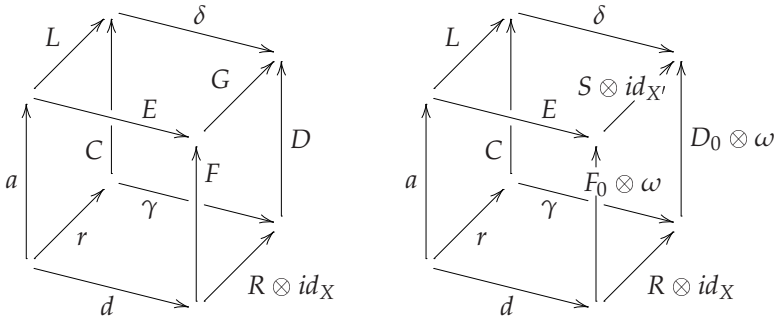
Case $|a| \not\subseteq |d|$: In this case we claim that $|a| \cap |r| = \emptyset$. For the sake of contradiction, suppose there are nodes u and v in a of which only u is in r. Since a does not share with R, the node u must be in d, and v in C. By shallowness of R there is a node w in R which is an ancestor of u in $C \circ r$, but not of v. But u and v are both in the single region of a, so in $L \circ a$ the node w, which is in L, must be an ancestor either of both of u and v or of neither. This is a contradiction, as $C \circ r = L \circ a$.

Since a shares no nodes with r, the pair (a, r) has a tensorial IPO. We argue that this is unique (up to isomorphism) by showing there can be no elisions:

The assumptions on R and d ensure that r is outer-surjective; this ensures there can be no elisions from r into C. Since a is non-empty, its root cannot be elided into L; the scope rule then prevents elisions of local names into L. Finally, since r is open, then so is L; hence there can be no elisions of global names into L.

We conclude that (L, C) is a tensorial IPO of (a, r), from which (a) follows.

Case $|a| \subseteq |d|$: Let (δ, D) be an IPO of (C, γ); this is unique up to isomorphism since there are no elisions either from or into globalizers. Now let (E, F, G) be an RPO of (a, d) to $(\delta \circ L, D \circ (R \otimes id_X))$; this yields a diagram of the form shown on the left below with all squares on the four vertical faces IPOs.



The inclusion $|a| \subseteq |d|$ ensures that F and D are empty. Consider the "diagonal" $D \circ (R \otimes id_X) = G \circ F$ of the right-hand face. Its inner names are the global names X together with the names of the inner face I of R, all local. The latter must all be bound in R, because R is shallow; hence they are bound in the diagonal. The names X, on the other hand, are global, and therefore must be free in the diagonal. Hence, no name $x \in X$ can have a sibling among the names of I. From this it follows that F has the form $F_0 \otimes \omega$ for some wiring $\omega : X \to X'$. Then the pair $(F, R \otimes id_X)$ has an IPO of the form $(S \otimes id_{X'}, D_0 \otimes \omega)$, where (S, D_0) is an IPO of (F_0, R). Moreover, there can be no elisions from ω into S (since R is open, and hence S is open); no elisions from F_0 into $id_{X'}$; and no elisions from $R \otimes id_X$ (since it is outer-surjective). It follows that every IPO of $(F, R \otimes id_X)$ has the form stated, and so we can refine the right-hand face of the cube to yield the diagram shown on the right above.

Since $D_0 \circ R = S \circ F_0$ the shallowness assumption on R stipulates that $S = D_0 \circ R \circ F_0'$ for some F_0' such that $F_0' \circ F_0 = id$. The IPO property of the square

on the right implies that (up to an isomorphism) $D_0 = id_J$. Taking $L_0 = (F_0' \otimes id_{X'}) \circ E$ we have

$$\begin{aligned} L_0 \circ a &= (F_0' \otimes id_{X'}) \circ E \circ a \\ &= (F_0' \otimes id_{X'}) \circ F \circ d \\ &= (F_0' \otimes id_{X'}) \circ (F_0 \otimes \omega) \circ d \\ &= \omega\, d\,. \end{aligned}$$

Thus, we have constructed the cube postulated in (b). ∎

We are now ready to prove the adequacy theorem itself. If falls in three parts: Clause (1) states the adequacy result for open primes; this result requires only the simplicity condition. Clause (2) states the result for all primes; this result needs a further condition, namely that the BRS be affine. We comment on the reason for the additional condition after the proof. Finally, clause (3) gives the analogous result to clause (2) for weak bisimilarity. There is no straightforward extension of clause (1) to the weak case. Again, we comment on the reason after the proof.

Theorem 5.30 (adequacy)

(1) In a simple BRS the engaged transitions over open primes are adequate for standard active transitions; that is, $\sim^{\mathrm{POE}}$ coincides with $\sim$ on open primes.

(2) In an affine simple BRS the engaged transitions over primes are adequate for standard active transitions; that is, $\sim^{\mathrm{PE}}$ coincides with $\sim$ on primes.

(3) In an affine simple BRS the engaged transitions over primes are adequate for weak standard active transitions; that is, $\approx^{\mathrm{PE}}$ coincides with $\approx$ on primes.

Proof The inclusion $\sim \subseteq \sim^{\mathrm{POE}}$ on open primes, and the inclusions $\sim \subseteq \sim^{\mathrm{PE}}$ and $\approx \subseteq \approx^{\mathrm{PE}}$ on all primes, are immediate.

For the reverse inclusion in clause (1) we show that $\sim^{\mathrm{POE}}$ is a generalized active bisimulation. Suppose $a \sim^{\mathrm{POE}} b$ and $a \xrightarrow{L} a'$ such that $L \circ b$ is defined. We seek a transition $b \xrightarrow{L} b'$ for some b' such that $(a', b') \in \mathcal{F}(\sim^{\mathrm{POE}})$ for some allowable $\mathcal{F}$. If the transition of a is engaged, then the required transition of b follows readily from the assumption $a \sim^{\mathrm{POE}} b$. If the transition of a is not engaged then one of the two clauses of Lemma 5.29 applies.

If clause (a) holds, we have $L = r \otimes id$ and $a' = r' \otimes a$, where (r, r') is the ground redex underlying the transition. We then form a tensorial IPO $(r \otimes id, id \otimes b)$ of (b, r); this underlies a transition $b \xrightarrow{L} b' \stackrel{\mathrm{def}}{=} r' \otimes b$, and we have $(a', b') \in (\sim^{\mathrm{POE}})^{\mathsf{C}}$.

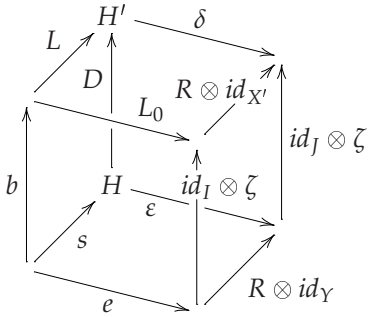
If instead clause (b) of Lemma 5.29 holds, then the parametric and ground rules (R, R', η) and (r, r') underlying the transition are related as shown by the cube given in the lemma, together with the equation

$$\gamma \circ r' = (R' \otimes id_X) \circ \eta(d)\,.$$

Moreover, we have $a' = C \circ r'$. From this we calculate

$$\begin{aligned}
\delta \circ a' &= \delta \circ C \circ r' \\
&= (id_J \otimes \omega) \circ \gamma \circ r' \\
&= (id_J \otimes \omega) \circ (R' \otimes id_X) \circ \eta(d) \\
&= (R' \otimes id_{X'}) \circ (id_I \otimes \omega) \circ \eta(d) \\
&= (id_{X'} \otimes R') \circ \eta(\omega\, d) \\
&= (id_{X'} \otimes R') \circ \eta(L_0 \circ a)\,.
\end{aligned}$$

For the matching transition of b we shall construct the following diagram in several stages.



First, let $\zeta\, e$ be the discrete normal form of $L_0 \circ b$; this provides the square on the front face, which is IPO by Proposition 4.41(2). The square on the right-hand face is a tensorial IPO. Hence, the "diagonal square" composed of the two IPOs is itself an IPO. The top square is copied from the cube for a. The remaining arrows s, ε and D are determined by choosing their common interface H. We take this to be a copy of the outer face $J \otimes Y$ of $R \otimes id_Y$, except we equip it with a different location map. As H is prime, the location map must merely designate each name either local or global. We do this as follows: Let a name x be local if either (1) it is local in $J \otimes Y$ or (2) the name $(id_J \otimes \zeta)(x)$ is local in the inner face H' of δ; if neither of these conditions hold, let x be global. The scope rule is obeyed by ε because of the first condition, by D because of the second condition, and by s because it is ground and prime.

We now argue that the square forming the left-hand face of the cube is IPO. Since the diagonal square is IPO, it is enough, by Theorem 4.25, to show that each name global in H' is linked to a global inner name in either of L and D. Suppose that x' is global in H', and therefore also in the outer face $J \otimes X'$ of δ. If x' is not linked to a global inner name in L, then for the diagonal square to be IPO it must be linked to a global inner name x in $id_J \otimes \zeta$. Then x is also global in H and linked to x' in D. Thus, the square on the left is indeed IPO.

Now pick s' such that $s' = (id_Y \otimes R') \circ \eta(e)$; then (s, s') forms a ground reaction rule, and together with the IPO just established it underlies a transition $b \xrightarrow{L} b' \stackrel{\text{def}}{=} D \circ s'$. By a similar calculation to that for $\delta \circ a'$ one finds

$$\delta \circ b' = (id_{X'} \otimes R') \circ \eta(L_0 \circ b)\,.$$

The agents a and b are open, and so by Proposition 5.4(1) we have

$$(\eta(L_0 \circ a), \eta(L_0 \circ b)) \in (\sim^{\mathrm{POE}})^{\mathrm{C},*} ,$$

and hence

$$(a', b') \in (\sim^{\mathrm{POE}})^{\mathrm{C},\mathrm{G},*} .$$

This completes the proof that $\sim^{\mathrm{POE}}$ is a generalized active bisimulation, and hence the proof of clause (1) in the theorem.

The proof of the inclusion $\sim^{\mathrm{PE}} \subseteq \sim$ for clause (2) proceeds analogously. The only difference is that in the last step we apply clause (2) of Proposition 5.4 and get $a' \ (\sim^{\mathrm{PE}})^{\mathrm{C},\mathrm{G},\mathrm{R}} \ b'$.

For the inclusion $\approx^{\mathrm{PE}} \subseteq \approx$ in clause (3) it suffices by Theorem 5.18(2) to show that $\approx^{\mathrm{PE}}$ is a bisimulation for standard active transitions relative to weak standard active transitions. Again, the proof proceeds analogously, and as in the previous case we use the allowable operation $(-)^{\mathrm{C},\mathrm{G},\mathrm{R}}$. ■

As can be seen from the proof, it would be straightforward to extend the second clause of the theorem to cover the non-affine case if Proposition 5.4(1) applied also to non-open primes, but unfortunately it does not. Indeed, an example is given in [26] of a non-affine simple BRS in which PE is not adequate.

We also lack a result extending clause (1) to the weak case. In the presence of replication it seems unavoidable to do the bisimulation proof "up to transitive closure." This operation is not allowable for relative bisimilarities, and therefore the present proof does net extend smoothly. Attempting to establish directly that $\approx^{\mathrm{PE}}$ is an (absolute) weak active bisimulation does not succeed, essentially because the agent may be replicated in each of the (unboundedly many) transitions underlying a weak transition. We conjecture that clause (1) does not extend to the weak case without further conditions, but we leave the question open.

Part II

Models of Mobile Processes

Chapter 6

The π-Calculus

In this chapter we give an overview of the syntax and semantics of the π-calculus[27]. The purpose is two-fold: first, to introduce the calculus to readers who are not familiar with it, and secondly, to clarify exactly which versions of the calculus we shall consider. As part of the latter, we point out a few details in which our versions differ from the most standard ones. For more thorough treatments of the π-calculus see for example the books by Milner [24] and Sangiorgi and Walker [38].

6.1 Syntax

The π-calculus has terms generated from the following abstract syntax:

$$\begin{array}{rcll} P, Q & ::= & M \mid P \mid Q \mid \nu z\, P \mid !x(z).P & \text{(processes)} \\ M, N & ::= & \overline{x}y.P \mid x(z).P \mid M + N \mid \mathbf{0} & \text{(sums)}\,, \end{array}$$

where x, y, z are *names*, drawn from an infinite set. The *prefixes* $\overline{x}y$ and $x(z)$ denote capabilities for *output* (of *datum* y on *channel* x) and *input* (on *channel* x), respectively. We shall use π to range over $\overline{x}y$ and $x(z)$. A sum containing several prefixed processes denotes a *choice* between the capabilities involved; the empty sum '$\mathbf{0}$' denotes no choices, i.e. *inaction*. We often drop $\mathbf{0}$ from a prefix, writing just π for the process $\pi.\mathbf{0}$. The operator '$\mid$' denotes *parallel composition*, and 'ν' denotes *restriction*; the latter makes z private to P in $\nu z\, P$.

Replication '!' allows infinite behaviour. Usually it is present in a general form in which $!P$ denotes an unbounded number of copies of P in parallel. Here we limit replication to apply to input-prefixed processes. This retains sufficient expressive power for most practical applications of the calculus, and at the same time it allows replication to be treated dynamically rather than structurally, as we shall detail below. The theory of BRSs developed in the preceding chapters turns out to be sufficient for modelling the dynamic version

of replication; modelling the structural version is harder, and we leave that problem for future work.

In each of $x(z).P$, $!x(z).P$ and $\nu z\, P$ the displayed occurrence of the name z is *binding* with *scope* P. We often refer to a binding occurrence as a *binder*, for short. An occurrence of a name is *bound* if it lies within the scope of a binder; it is *free* if it is neither bound nor binding. We write $\mathrm{names}(P)$ for the set of names that occur in P, and $\mathrm{fn}(P)$ for the set of names that have a free occurrence.

If two processes differ only in the choice of bound names, then we say they are *alpha-equivalent*; either can then be obtained from the other by *alpha-conversion*.

We write $\{^x/_y\}P$ for the result of replacing the free occurrences of y by x in P in a way that avoids capture of x; more precisely, the result is obtained by first alpha-converting P to some P' with no binding occurrence of x, and then replacing each free occurrence of y by x in P'.

A *process context* is a process term in which one process subterm has been left out. We use the notation $[\cdot]$ to represent the "hole" left by the missing subterm. For a context C we write $C[P]$ for the process resulting from "plugging" the process P into the hole of C. Note that we are requiring the hole to occur in a position such that $C[P]$ is well-formed for an arbitrary process term P; thus, $x(z).[\cdot]$ and $P \mid [\cdot] \mid Q$ are examples of process contexts, but $x(z).P + [\cdot]$ is not, as its hole can only be filled by a sum.

An equivalence relation $\mathcal{R}$ over processes is a *process congruence* if it is preserved by all process contexts; that is, if $(P, Q) \in \mathcal{R}$ implies $(C[P], C[Q]) \in \mathcal{R}$.

6.2 Structural congruence

Structural congruence $\equiv$ is the smallest process congruence that includes alpha-equivalence and satisfies the following axioms:

$$
\begin{aligned}
(L + M) + N &\equiv L + (M + N) \\
M + N &\equiv N + M \\
M + \mathbf{0} &\equiv M \\
\\
(P \mid Q) \mid R &\equiv P \mid (Q \mid R) \\
P \mid Q &\equiv Q \mid P \\
P \mid \mathbf{0} &\equiv P && (-\mathbf{0}) \\
\\
\nu x\, \nu y\, P &\equiv \nu y\, \nu x\, P \\
\nu z\, (P \mid Q) &\equiv P \mid \nu z\, Q && \text{if } z \notin \mathrm{fn}(P) \\
\nu z\, (M + \pi.P) &\equiv M + \pi.\nu z\, P && \text{if } z \notin \mathrm{fn}(M) \cup \mathrm{names}(\pi) && (\nu\text{-}\pi) \\
\nu z\, \mathbf{0} &\equiv \mathbf{0}\,. && && (-\nu)
\end{aligned}
$$

Usually, there is one or more axioms for handling replication; we have no such axioms, since—as mentioned above—we shall instead treat replication dynamically.

Structural congruence can be viewed as removing "accidental" distinctions that are introduced merely by the need to express process structure in textual syntax. The technical purpose of this is to allow a simpler definition of process behaviour than is possible on the raw syntax. Neither of the three axioms marked with tags in the right margin is needed for this purpose, as will be made clear below. In fact, of the three, only $(-\mathbf{0})$ and $(-\nu)$ are normally included; they allow expressions to be simplified by removing superfluous occurrences of $\mathbf{0}$ and ν. In particular, the natural property

$$\nu z\, P \equiv P \qquad \text{if } z \notin \mathrm{fn}(P)$$

requires both axioms. The axiom $(\nu\text{-}\pi)$ allows the scope of a restriction to vary, provided no free names are captured, or bound names freed; it naturally complements the similar axiom for parallel composition.

We shall use several variants of structural congruence that arise from different choices of which of the three axioms to include; we shall write them as follows, where for each one we list the axioms included:

Relation		Axioms	
$\equiv_{\mathbf{0},\nu}$	$(-\mathbf{0})$	$(-\nu)$	$(\nu\text{-}\pi)$
$\equiv_{\nu}$		$(-\nu)$	$(\nu\text{-}\pi)$
$\equiv$			none

Structural congruence induces a notion of *normal form*. We define *process normal form (PNF)*, *open process normal form (OPNF)*, and *sum normal form (SNF)*, by mutual recursion as expressions of the following forms:

PNF:	$\nu Z\, P$	P an OPNF, $Z \subseteq \mathrm{fn}(P)$
OPNF:	$M_1 \mid \cdots \mid M_m \mid$ $!x_1(z_1).P_1 \mid \cdots \mid !x_n(z_n).P_n$	$m > 0$, each M_i a SNF, $n \geq 0$, each P_1 a PNF
SNF:	$\pi_1.P_1 + \cdots + \pi_p.P_p$	$p \geq 0$, each P_i a PNF .

For $Z = z_1, \ldots, z_k$ we take $\nu Z\, P$ to be an abbreviation of $\nu z_1 \cdots \nu z_k\, P$; if Z is empty there is no restriction. The empty SNF ($p = 0$) is $\mathbf{0}$. For $\equiv_{\nu}$ we drop the requirement $Z \subseteq \mathrm{fn}(P)$ for a PNF, but add the requirement that each P_i in an OPNF is an OPNF, not merely a PNF. For $\equiv_{\mathbf{0},\nu}$ we do the same, plus we allow empty OPNFs ($m = 0$) but require unary sums ($p = 1$). The following then applies to each variant of structural congruence.

Proposition 6.1 For every process there is a structurally congruent PNF. For every open process there is a structurally congruent OPNF. For every sum there is a structurally congruent SNF. All these normal forms are unique up to a reordering of terms in parallel compositions and sums. ∎

6.3 Reaction

The behaviour of processes is expressed using two different styles of operational semantics. The simplest is in terms of *reactions* (or *reductions*) of the form $P \to P'$. The other style is labelled transitions; we return to that below.

The reaction semantics for our variant of π-calculus is based on two fundamental kinds of reaction, expressed by two reaction rules:

$$\begin{aligned} (\overline{x}y.P + M) \mid (x(z).Q + N) &\to P \mid \{y/z\}Q && \text{(communication)} \\ (\overline{x}y.P + M) \mid !x(z).Q &\to P \mid \{y/z\}Q \mid !x(z).Q && \text{(replicating com.)} . \end{aligned}$$

The first rule expresses the passage of a datum y along the channel x from a process with the output capability $\overline{x}y$ to another process with the input capability $x(z)$. In the result on the right-hand side, the two (now exhausted) capabilities have been removed together with the (now preempted) alternatives M and N, and the datum y has replaced the place-holder z in the body Q of the recipient. The second rule represents our dynamic handling of replication; it is similar to the communication rule, except for two things: first, the replicated recipient has no alternative capabilities; second, and more importantly, because the recipient is replicated it is not exhausted, and therefore it is still present after the reaction, along with a substitution-instance of its body Q.

The reaction relation $\to$ is the least relation over processes which (1) admits the two reaction rules; (2) is preserved by parallel composition and restriction; and (3) is closed under structural congruence. The two latter requirements mean that, whenever $P \to P'$, then also $P \mid Q \to P' \mid Q$ and $\nu z\, P \to \nu z\, P'$, and if in addition $P \equiv Q$ and $P' \equiv Q'$, then also $Q \to Q'$.

Reaction can be characterized as follows in terms of process normal forms. The statement applies equally for each of our variants of structural congruence and its corresponding variant of normal form.

Proposition 6.2 $P \to P'$ iff P and P' have PNFs of the forms

$$\begin{aligned} P &\equiv \nu Z\, ((\overline{x}y.Q + M) \mid (x(z).R + N) \mid S) \\ P' &\equiv \nu Z\, (Q \mid \{y/z\}R \mid S) \end{aligned}$$

or of the forms

$$\begin{aligned} P &\equiv \nu Z\, ((\overline{x}y.Q + M) \mid (!x(z).R) \mid S) \\ P' &\equiv \nu Z\, (Q \mid \{y/z\}R \mid !x(z).R \mid S) . \end{aligned}$$ ∎

6.4 Transitions and bisimilarity

Reaction expresses the behaviour of complete systems, but not the interaction among parts of a system. For the latter task, it is common to use one of several

transition systems defined over processes. We define here the *early transition system*, given by the inference rules below. Its labels are of the forms $\overline{x}y$ (*free output*), xy (*free input*), and $x(z)$ (*bound output*); for the purpose of the side conditions of the rules, we declare the names x and y to be free in $\overline{x}y$ and xy, whereas in $x(z)$ the name x is free and z bound.

$$\overline{x}y.P \xrightarrow{\overline{x}y} P \qquad x(z).P \xrightarrow{xy} \{y/z\}P \qquad !x(z).P \xrightarrow{xy} \{y/z\}P \mid !x(z).P$$

$$\frac{M \xrightarrow{\ell} P'}{M + N \xrightarrow{\ell} P'}$$

$$\frac{P \xrightarrow{\ell} P'}{P \mid Q \xrightarrow{\ell} P' \mid Q} \quad \mathrm{bn}(\ell) \cap \mathrm{fn}(Q) = \emptyset \qquad \frac{P \xrightarrow{\ell} P'}{\boldsymbol{\nu} z\, P \xrightarrow{\ell} \boldsymbol{\nu} z\, P'} \quad z \notin \mathrm{names}(\ell)$$

$$\frac{P \xrightarrow{\overline{x}y} P' \quad Q \xrightarrow{xy} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

$$\frac{P \xrightarrow{\overline{x}z} P'}{\boldsymbol{\nu} z\, P \xrightarrow{\overline{x}(z)} P'} \quad z \neq x \qquad \frac{P \xrightarrow{\overline{x}(z)} P' \quad Q \xrightarrow{xz} Q'}{P \mid Q \xrightarrow{\tau} \boldsymbol{\nu} z\, (P' \mid Q')} \quad z \notin \mathrm{fn}(Q)$$

All rules involving summation and parallel composition also have symmetric versions with the roles of P and Q swapped.

The labels of transitions express actions. The rules in the top row translate prefixed capabilities directly into actions; the rule for summation in the second row expresses discard of preempted alternatives; the rules in the third row express that transitions, like reaction, are preserved by parallel composition and restriction; the rule in the fourth row expresses how an output action and an input action communicate to form a *silent* action, marked by the distinguished label τ. The two remaining rules, called the 'open' and 'close' rule, respectively, handle *scope extrusion*—the situation when a private name is sent out of its present scope: in the 'open' rule, the restriction is lifted and a bound output is inferred; in the 'close' rule, a bound output communicates with an input, and the restriction is reinstated, now with a scope that includes the recipient.

The bisimilarity associated with the early transition system is called *(strong) early bisimilarity*; we shall denote it by $\overset{\mathrm{e}}{\sim}$. It is not a process congruence, as it is not always preserved by input prefix. (It is preserved by all other process constructions.) The reason it may fail to be preserved by an input prefix is that an input can lead to non-injective substitutions of names, and such substitutions can destroy bisimilarity. For example, it is easy to prove the bisimilarity $P \overset{\mathrm{e}}{\sim} Q$, where

$$P = \overline{x} \mid y$$
$$Q = \overline{x}.y + y.\overline{x}\,.$$

(We are omitting the data from the inputs and outputs; they can be taken to be arbitrary fresh names.) But $\{x/y\}P \overset{e}{\approx} \{x/y\}Q$, as only $\{x/y\}P$ has a τ-transition. It follows, for example, that $u(x).P \overset{e}{\approx} u(x).Q$.

From $\overset{e}{\sim}$ we obtain *(strong) early congruence*, written $\sim$, by letting $P \sim Q$ iff $\sigma P \overset{e}{\sim} \sigma Q$ for every name substitution σ; it is the largest process congruence included in $\overset{e}{\sim}$.

A variant of the early transition system is the *late transition system*, in which the rules for input, replicated input, and communication, are replaced by

$$x(z).P \xrightarrow{x(z)} P \qquad !x(z).P \xrightarrow{x(z)} P \mid !x(z).P$$

$$\frac{P \xrightarrow{\overline{x}y} P' \quad Q \xrightarrow{x(z)} Q'}{P \mid Q \xrightarrow{\tau} P' \mid \{y/z\}Q'}$$

The form of input $x(z)$ employed here is called *bound input*, as opposed to the *free input* xy employed in the early system. The bound input transition does not substitute the received name for its place-holder, as is done in the free input; instead, the late system performs the substitution when the input communicates with an output to form a silent transition. The bisimilarity associated with the late transition system is called *(strong) ground bisimilarity*;[1] we denote it by $\overset{g}{\sim}$. This relation strictly includes $\overset{e}{\sim}$. Unlike $\overset{e}{\sim}$, it is preserved by input contexts, because bound inputs do not involve substitutions. It is far from being a congruence, however; in particular it is not preserved by parallel composition. (For example, for the processes P and Q considered above we have $u(x).P \overset{g}{\sim} u(x).Q$, but $\overline{u}y \mid u(x).P \overset{g}{\not\sim} \overline{u}y \mid u(x).Q$.) For this reason, ground bisimilarity is considered to be of little practical value, at least in the full calculus. It does have theoretical interest, however; in particular we shall make use of the property that it "captures" strong early congruence in the same sense that $\overset{e}{\sim}$ does, namely that strong early congruence is the largest process congruence included within it.

While several other bisimilarities and behavioural congruences exist for the π-calculus, early congruence gains prominence by the fact that it coincides with *barbed congruence*[28]. This congruence, whose formal definition we omit, expresses in a fairly canonical manner the distinguishing power of an observer expressed in the calculus itself.

The reaction semantics and the two transition semantics all agree, in the following sense:

Proposition 6.3

(1) Structural congruence $\equiv$ is included in early bisimilarity $\overset{e}{\sim}$, and hence also in ground bisimilarity $\overset{g}{\sim}$;

[1] *Strong late bisimilarity*, a relation very close to strong early bisimilarity, does not arise directly from the late transition system; in matching an input $P \xrightarrow{x(y)} P'$ with another input $Q \xrightarrow{x(y)} Q'$ it requires not only P' and Q' to be related, but also $\sigma P'$ and $\sigma Q'$ for every name substitution σ.

(2) $P \to P'$ iff $P \xrightarrow{\tau} \equiv P'$ in either transition system;

These properties hold regardless of which axioms, among the three discussed above, are included in defining structural congruence. ■

6.5 Subcalculi

In the following chapters we shall model the π-calculus in bigraphs. We build up the model in three stages, starting with a core calculus without summation and replication, then adding summation, and last adding replication. To be precise, the subcalculi we shall consider are the following:

- *Finite π-calculus*, or fπ for short, is obtained by omitting the replicated input construct $!x(y)$ from the syntax;
- *Strict, finite π-calculus*, or sfπ for short, is obtained by omitting both replicated input and the summation construct $M + N$; in the absence of '+' the two-sorted syntax collapses to a single-sorted syntax:

$$P, Q \;::=\; \overline{x}y.P \;\mid\; x(z).P \;\mid\; P \mid Q \;\mid\; \nu z\, P \;\mid\; \mathbf{0}\,.$$

The small succession of calculi—from sfπ through fπ to full π—serves rather well to demonstrate progressively more advanced features of bigraphical reactive systems.

We shall also model the *asynchronous* π-calculus, which we shall call aπ for short. This arises from a different syntactical restriction, namely that only $\mathbf{0}$ can be prefixed by output and that outputs cannot occur in sums. To be precise, the syntax is

$$\begin{aligned} P, Q \;&::=\; M \mid \overline{x}y \mid P \mid Q \mid \nu z\, P \mid !x(z).P && \text{(processes)} \\ M, N \;&::=\; x(z).P \mid M + N \mid \mathbf{0} && \text{(sums)}\,. \end{aligned}$$

In this calculus, outputs can only be combined with other processes by parallel composition; therefore a process cannot "feel" when its outputs are received, neither by the release of a prefixed subprocess nor by the preemption of alternatives in a sum. This is the sense in which the calculus is asynchronous.

Remarkably, all the usual variants of strong bisimilarity (including early and ground) coincide on aπ and form a process congruence. However, the picture is complicated in this calculus by the existence of *asynchronous bisimilarity* $\overset{\text{a}}{\sim}$; it is defined like early bisimilarity, except in an *asynchronous bisimulation* $\mathcal{S}$ the matching condition for input transitions of a pair $(P, Q) \in \mathcal{S}$ is weakened as follows:

$$\begin{aligned} &P \xrightarrow{x(z)} P' \text{ implies} \\ &\qquad \text{either } Q \xrightarrow{x(z)} Q' \text{ for some } Q' \text{ such that } (P', Q') \in \mathcal{S} \\ &\qquad \text{or } Q \xrightarrow{\tau} Q' \text{ for some } Q' \text{ such that } (P', Q' \mid \overline{x}z) \in \mathcal{S}\,. \end{aligned}$$

Thus, compared to the usual definition, there is an additional way of matching an input transition; it applies when the input reinstates the consumed output particle immediately, and in that case allows the input to be matched by a silent transition. The motivation is that an asynchronous observer, unable to observe the input directly, cannot tell the difference between the two behaviours. Here are some examples of processes that are asynchronous bisimilar, but not early bisimilar:

$$x(z).\overline{x}z + \tau \overset{\text{a}}{\sim} \tau$$
$$!x(z).\overline{x}z \mid !\tau \overset{\text{a}}{\sim} !\tau\ ,$$

where τ denotes any process that performs a single τ-transition, e.g. $\nu w\,(\overline{w} \mid w)$.

Asynchronous bisimilarity is also a congruence in aπ, and it enjoys the same status that early congruence does in the other calculi, namely that it coincides with the "native" barbed congruence (i.e. the one based on contexts expressible in the calculus itself).

Chapter 7

Modelling the π-Calculus

We are now ready to show how the π-calculus can be represented as a bigraphical reactive system. In the first three sections of this chapter, we build the model of the full calculus in stages, tackling first the subcalculi sfπ and fπ; this allows a gradual introduction of the features needed to handle summation and replication. Finally, in the last section we model the asynchronous calculus, aπ.

7.1 The core calculus

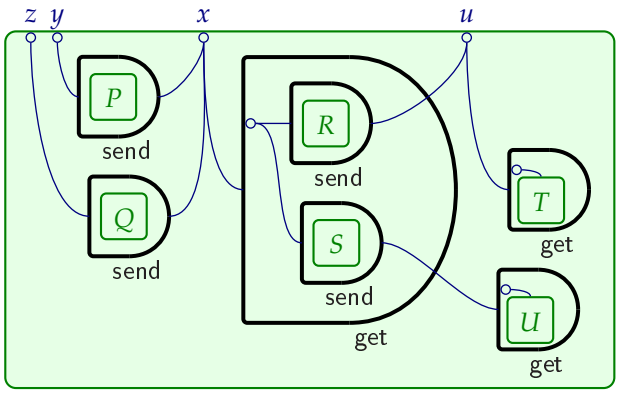
We start by modelling the strict, finite π-calculus, sfπ. We can view this as a core π-calculus, consisting only of the two "dynamic ingredients" (input and output) and the two "structural ingredients" (parallel composition and restriction). Structure is orthogonal to dynamics, in the sense that reaction preserves, and is preserved by, parallel composition and restriction. Thus, we can view the structural ingredients as providing the medium in which the dynamic ingredients perform.

As might be expected from these observations, we shall introduce controls to represent input and output, and we shall use place and link structure to represent parallel composition and restriction. The calculus employs binding (in the input prefix), so we shall use binding bigraphs. (Restriction is also a binder in the calculus, but—for now—we shall model restriction without using binding.) The calculus also has activeness, because reaction is allowed only at "top-level", not underneath prefixes. Accordingly, we shall work in bigraphs with activeness, and declare the controls for both input and output to be passive.

Formally, we define our bigraphical reactive system as

$$\mathbf{BBG}_{\mathrm{sf}\pi} = \mathbf{BBG}(\Sigma_{\mathrm{sf}\pi}, \mathcal{R}_{\mathrm{sf}\pi}) \,,$$

where the signature $\Sigma_{\mathrm{sf}\pi}$ is given by the following definition. (We give the reaction rule set $\mathcal{R}_{\mathrm{sf}\pi}$ later.)



$$\overline{x}y.P \mid \overline{x}z.Q \mid \nu w\,(x(v).(\overline{u}v.R \mid \overline{w}v.S) \mid u(v).T \mid w(v).U)$$

Figure 7.1 Example of a bigraph for a π-calculus process

Definition 7.1 The signature $\Sigma_{\mathrm{sf}\pi}$ is binding, trivially sorted, and has activeness; its controls are

$$\begin{array}{ll} \mathsf{send} : 0 \to 2 & \text{(passive)} \\ \mathsf{get} : \ 1 \to 1 & \text{(passive)} \end{array}$$

■

Figure 7.1 shows an example of how we shall model a process in sfπ. The meta-parameters P, Q, ... represent sub-bigraphs for the identically named sub-process terms. Thus, prefixing is modelled by nesting, that is, by place structure. The connectivity that the calculus expresses with names is handled in the bigraph by links; open names in the process correspond to open links (= outer names) in the bigraph, and the restricted name w is represented by the closed link connecting the send-node containing S with the get-node containing U. Note also that each of the two kinds of node we employ has two ports; we refer to one as the *channel port* and the other as the *datum port*. In a send-node both ports are non-binding; we shall use the convention of drawing the channel port on the curved side of a send-node, and the datum port on the flat side. In a get-node the channel port is non-binding, and the datum port binding. We draw both on the flat side of the node; they can be distinguished by the small circle that indicates a binding port—hence, in this case, the datum port. The scope rule dictates that the link of the datum port may not cross the node boundary; in other words, all its remaining points must lie inside the node.

In the process expression in Figure 7.1, the exact placement of the restriction νw is somewhat arbitrary: assuming that w has no other occurrences in the expression than those displayed, we could have moved the sub-process term $u(v).T$ outside the scope of the restriction, or $\overline{x}y.P$ and $\overline{x}z.Q$ inside, without changing the expression up to structural congruence. Similar considerations apply to the order of parallel components. The bigraph does not have this ambiguity; the representation of a process does not depend on the kind of syntactical detail that structural congruence factors out. We make this claim

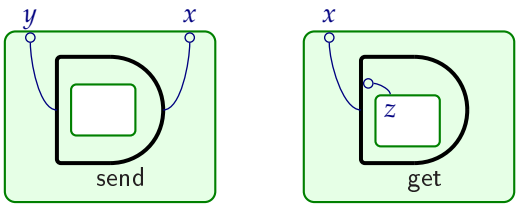


Figure 7.2 Ions in $\mathbf{BBG}_{\mathrm{sf}\pi}$

precise in the structural correspondence theorem below, where we show that the bigraph model recovers (a particular version of) structural congruence.

In modelling processes we shall use bigraphs which—like the one in the example above—are prime and local. In other words, we shall use the homsets of the form $(\epsilon, (X))$ in $\mathbf{BBG}(\Sigma_{\mathrm{sf}\pi})$. We shall refer to bigraphs in these homsets as *process bigraphs*, and we shall use p, q, r, s, t to range over them.

Figure 7.2 shows ions built from the two controls of $\Sigma_{\mathrm{sf}\pi}$. As shown, the ion send_{xy} has the channel port linked to x and the datum port to y, and the ion $\mathsf{get}_{x(z)}$ has the datum port linked to the inner name z.

In the following lemma, recall from Section 4.9 that $(/Z)\,p$ denotes closure of the local names Z in p.

Lemma 7.2 In $\mathbf{BBG}(\Sigma_{\mathrm{sf}\pi})$ every bigraph with outer face (X) belonging to one of the following classes can be expressed in the forms given:

Process bigraphs:	$(/Z)\,p$	$p : (XZ)$ open
Open process bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n \geq 0$, each $\mu_i : (X)$ open
Open molecules:	$\mathsf{send}_{xy}\,p$	$p : (X)$ open
	$\mathsf{get}_{x(z)}\,p$	$p : (Xz)$ open .

Moreover, these forms are unique up to the choice of fresh names Z, z and the ordering of the μ_i.

Proof For a process bigraph q, we form Z and p such that $q = (/Z)\,p$ as follows: choose Z consisting of fresh names in bijection with the edges of q and form p from q by adding Z to the local names of its outer face, mapping each point in each edge e instead to the corresponding name in Z, and removing all free edges. It is routine to verify that $(/Z)\,p = (/Z)\,p'$ implies $p = p'$, from which the uniqueness property follows.

For an open process bigraph $p : (X)$, the molecules $\mu_1, \ldots, \mu_n$ such that $p = \mu_1 \mid \cdots \mid \mu_n$ can be formed as follows: decompose the place graph of p uniquely into factors each consisting of a child node of the root together with all its descendants; then equip each factor with names (X) and a link map that maps each port exactly as in p.

The forms of an open molecule are immediate, as $\Sigma_{\mathrm{sf}\pi}$ allows only send- and get-nodes. ∎

We are now ready to give the translation of processes into bigraphs:

Definition 7.3 The translation $\mathsf{S}_{(X)}[\![-]\!]$ maps every process P of sfπ such that $\mathrm{fn}(P) \subseteq X$ into the homset $(\epsilon, (X))$ of $\mathbf{Bbg}(\Sigma_{\mathrm{sf}\pi})$ as follows:

$$\begin{aligned}
\mathsf{S}_{(X)}[\![\overline{x}y.P]\!] &= \mathsf{send}_{xy}\,\mathsf{S}_{(X)}[\![P]\!] \\
\mathsf{S}_{(X)}[\![x(z).P]\!] &= \mathsf{get}_{x(w)}\,\mathsf{S}_{(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{S}_{(X)}[\![P \mid Q]\!] &= \mathsf{S}_{(X)}[\![P]\!] \mid \mathsf{S}_{(X)}[\![Q]\!] \\
\mathsf{S}_{(X)}[\![\nu z\, P]\!] &= (/w)\,\mathsf{S}_{(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{S}_{(X)}[\![\mathbf{0}]\!] &= (X)\,.
\end{aligned}$$

■

As anticipated, output and input are modelled directly with the two controls. Parallel composition '$\mid$' is modelled as prime product '$\mid$' (which justifies overloading the symbol), and restriction is modelled as name closure. The nil process is modelled by (X); recall from Section 4.9 that this denotes the empty ground prime with names X that are all local. In the translation of binders we alpha-convert to a fresh name w in order to ensure that the resulting bigraph has the required names.

We now prove that our model uses *all* bigraphs in the homsets $(\epsilon, (X))$, and that it recovers $\equiv_{\mathbf{0},\nu}$, i.e. the version of structural congruence that allows restrictions to be pushed past prefixes.

Theorem 7.4 (structural correspondence)

(1) The map $\mathsf{S}_{(X)}[\![-]\!]$ is surjective onto the homset $(\epsilon, (X))$ of $\mathbf{Bbg}(\Sigma_{\mathrm{sf}\pi})$;

(2) $P \equiv_{\mathbf{0},\nu} Q$ iff $\mathsf{S}_{(X)}[\![P]\!] = \mathsf{S}_{(X)}[\![Q]\!]$.

Proof We start by considering the implication ($\Rightarrow$) of (2). Since $\mathsf{S}_{(X)}[\![-]\!]$ is compositional it suffices to prove each axiom of structural congruence separately. They are all straightforward; for example, the axiom $P \mid Q \equiv Q \mid P$ follows by showing $p \mid q = q \mid p$, which is immediate, since the prime product '$\mid$' merges regions and is defined in terms of union on nodes, edges, and names.

To prove (1) and (2)($\Leftarrow$) we define a reverse translation from process bigraphs into sfπ. More precisely, we use the unique forms of Lemma 7.2 and define translations $\widehat{\mathsf{S}}[\![-]\!]$, $\widehat{\mathsf{S}}_1[\![-]\!]$ and $\widehat{\mathsf{S}}_2[\![-]\!]$ by mutual recursion from process bigraphs, open process bigraphs, and open molecules, respectively, as follows:

$$\begin{aligned}
\widehat{\mathsf{S}}[\![(/Z)\,p]\!] &= \nu Z\,\widehat{\mathsf{S}}_1[\![p]\!] \qquad (p \text{ open}) \\
\widehat{\mathsf{S}}_1[\![\mu_1 \mid \cdots \mid \mu_n]\!] &= \widehat{\mathsf{S}}_2[\![\mu_1]\!] \mid \cdots \mid \widehat{\mathsf{S}}_2[\![\mu_1]\!] \\
\widehat{\mathsf{S}}_2[\![\mathsf{send}_{xy}\,p]\!] &= \overline{x}y.\widehat{\mathsf{S}}_1[\![p]\!] \\
\widehat{\mathsf{S}}_2[\![\mathsf{get}_{x(z)}\,p]\!] &= x(z).\widehat{\mathsf{S}}_1[\![p]\!]\,.
\end{aligned}$$

It is routine to verify that

$$\mathsf{S}_{(X)}[\![\widehat{\mathsf{S}}[\![p]\!]]\!] = p \quad \text{and} \quad \widehat{\mathsf{S}}[\![\mathsf{S}_{(X)}[\![P]\!]]\!] \equiv_{\mathbf{0},\nu} P$$

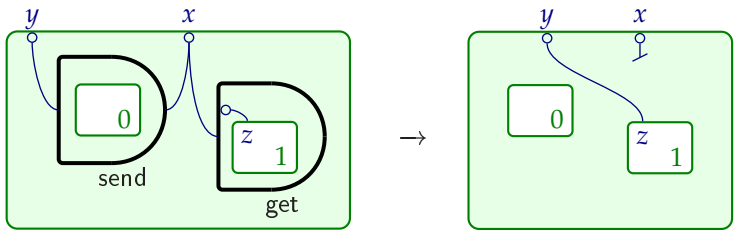


Figure 7.3 The communication rule for strict, finite π-calculus

for all process bigraphs p and processes P. Clause (1) of the theorem follows from the first identity, and (2)($\Leftarrow$) follows from the other identity together with (2)($\Rightarrow$). ■

Having established structural correspondence we now turn to the dynamics of our model. We shall represent the dynamics using a single parametric reaction rule, cf. Definition 5.5. Recall from page 67 that $\lhd$, as used below, denotes insertion of a wiring.

Definition 7.5 The rule set $\mathcal{R}_{\mathrm{sf}\pi}$ consists of the single parametric reaction rule (R, R', η), where the instantiation η is the identity, and

$$R = \mathsf{send}_{xy} \mid \mathsf{get}_{x(z)}$$
$$R' = (id_1 \mid id_1 \lhd {}^{y}/_{z}) \lhd x\,,$$

both with inner face $\langle 1, (z) \rangle$ and outer face (xy). ■

The rule is illustrated in Figure 7.3.

To establish dynamic correspondence, we start by characterizing reaction in $\mathbf{BBG}_{\mathrm{sf}\pi}$. This is made easy by the fact that there are no active controls, and thus reaction is preserved only by contexts that retain the redex at the outermost level.

Lemma 7.6 $p \longrightarrow p'$ in $\mathbf{BBG}_{\mathrm{sf}\pi}$ iff p and p' are of the forms

$$p = (/Z)\,(\mathsf{send}_{xy}\, q \mid \mathsf{get}_{x(z)}\, s \mid t)$$
$$p' = (/Z)\,(q \mid ({}^{y}/_{yz})\, s \mid t)\,.$$ ■

Using this, it is straightforward to establish that the model recovers reaction exactly:

Theorem 7.7 (dynamic correspondence) $P \longrightarrow P'$ iff $\mathsf{S}_{(X)}[\![P]\!] \longrightarrow \mathsf{S}_{(X)}[\![P']\!]$.

Proof As we saw in Section 6 we have $P \longrightarrow P'$ iff P and P' have $\equiv_{\mathbf{0},\nu}$-normal forms as follows, where by alpha-conversion we can assume $(Z \uplus \{z\}) \cap X =$

$\emptyset$:

$$P \equiv_{\mathbf{0},\nu} \nu Z\,(\overline{x}y.Q \mid x(z).S \mid T)$$
$$P' \equiv_{\mathbf{0},\nu} \nu Z\,(Q \mid \{y/z\}S \mid T)\,.$$

By Theorem 7.4 this is equivalent to

$$\begin{aligned}
\mathsf{S}_{(X)}[\![P]\!] &= (/Z)\,(\mathsf{send}_{xy}\,\mathsf{S}_{(XZ)}[\![Q]\!] \mid \mathsf{get}_{x(z)}\,\mathsf{S}_{(XZz)}[\![S]\!] \mid \mathsf{S}_{(XZ)}[\![T]\!] \\
\mathsf{S}_{(X)}[\![P']\!] &= (/Z)\,(\mathsf{S}_{(XZ)}[\![Q]\!] \mid \mathsf{S}_{(XZ)}[\![\{y/z\}S]\!] \mid \mathsf{S}_{(XZ)}[\![T]\!]) \\
&= (/Z)\,(\mathsf{S}_{(XZ)}[\![Q]\!] \mid (y/yz)\,\mathsf{S}_{(XZz)}[\![S]\!] \mid \mathsf{S}_{(XZ)}[\![T]\!])\,.
\end{aligned}$$

By the preceding lemma these equations hold iff $\mathsf{S}_{(X)}[\![P]\!] \to \mathsf{S}_{(X)}[\![P']\!]$. ∎

This correspondence result, together with the structural correspondence theorem (Theorem 7.4), shows that $\mathbf{BBG}_{\mathrm{sf}\pi}$ provides an accurate model of the strict, finite π-calculus viewed as a reactive system.

7.2 Summation

The reaction relation we defined for $\mathbf{BBG}_{\mathrm{sf}\pi}$ is linear, because reaction in the core calculus neither discards nor replicates subterms. In fπ reaction is not linear; its reaction rule

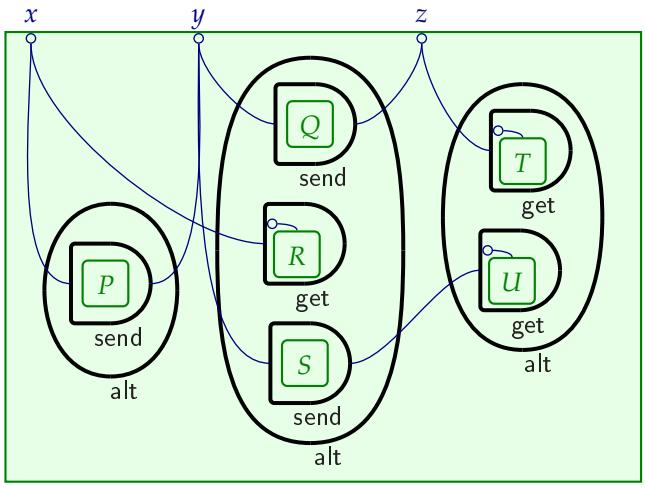
$$(\overline{x}y.P + M) \mid (x(z).Q + N) \to P \mid \{y/z\}Q$$

discards the (arbitrarily large) subterms M and N. To model this behaviour in a bigraphical reactive system we need a reaction rule that discards some parts of its parameter, and we need a way to identify exactly which parts to discard.

In the calculus this is of course what sums do; they are, essentially, collections of prefixed processes, and when an input-prefixed process and an output-prefixed process perform a reaction step it is exactly all the other summands in their respective sums that are discarded. In the bigraph model each summand will be a send- or a get-node (together with its descendants), so to be able to discard the right parts of the parameter in a reaction step we need to group the nodes that belong to the same sum. An obvious (and, as we shall see, perfectly satisfactory) way of doing this is to enclose them in a node. We therefore introduce a new control, called alt, for this purpose. An alt-node should act simply as a container for the summands and therefore needs no linking; accordingly, we shall give alt the arity 0. Figure 7.4 shows an example of how we shall model a process with summation using alt-nodes.

Having introduced alt-nodes to represent sums we can model the communication rule as shown in Figure 7.5. Note how in each sum the arbitrarily many summands being discarded are represented neatly as a single region of the parameter.

However, alt-nodes allow us to combine nodes in ways that are undesirable, either because they do not correspond to processes, or because they create ambiguities. Figure 7.6 gives an example, where two things are wrong:



$$\nu w\,(\overline{y}x.P \mid (\overline{z}y.Q + x(v).R + \overline{w}y.S) \mid (z(v).T + w(v).U))$$

Figure 7.4 Example of a bigraph for a process with summation

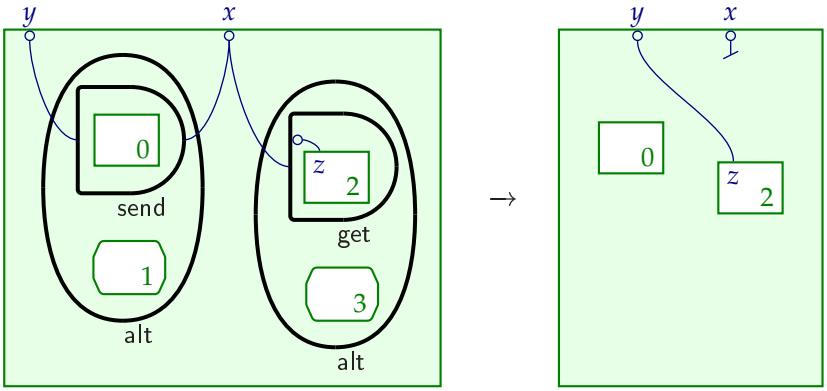


Figure 7.5 The communication rule for π-calculus with summation

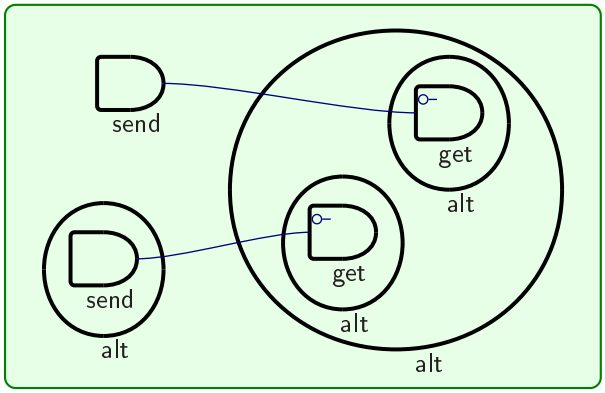


Figure 7.6 Example of an ill-formed bigraph with summation

- It has two alt-nodes directly inside another alt-node. This is not meaningful, since sums in the calculus are flat—a sum cannot itself act as a summand.
- It has a send-node and an alt-node at the same level. This might be meaningful if the send-node is interpreted as a unary sum, but in that case it is ambiguous whether to represent a unary sum like this or with an enclosing alt-node. The reaction rule, as shown in Figure 7.5, assumes the latter; it does not allow an alt-less send-node to react.

The solution is place-sorting. It allows us to dictate that the place-hierarchy should alternate strictly between levels with only alt-nodes and levels with only send- and get-nodes. Such a formation discipline avoids the kind of problems pointed out in the example, and it corresponds exactly to the two sorts (processes and sums) in the syntax of the calculus.

Thus, we shall use a sorted bigraphical reactive system

$$\mathbf{BBG}_{\mathrm{f}\pi} = \mathbf{BBG}(\Sigma_{\mathrm{f}\pi}, \mathcal{R}_{\mathrm{f}\pi}) \,.$$

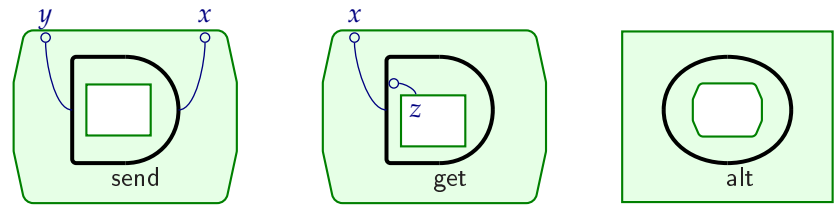
Its structure is determined by the signature $\Sigma_{\mathrm{f}\pi}$:

Definition 7.8 The signature $\Sigma_{\mathrm{f}\pi}$ is binding and has activeness; its controls are

$$\begin{array}{lll} \mathsf{send} : & 0 \to 2 & \text{(passive)} \\ \mathsf{get} : & 1 \to 1 & \text{(passive)} \\ \mathsf{alt} : & 0 \to 0 & \text{(passive)} \,. \end{array}$$

It has sorts pr and sum. The sorting condition stipulates the following:

- A pr-root or a send- or get-node may only have pr-sites and alt-nodes among its children; it may not be barren.
- A sum-root or an alt-node may only have sum-sites and send- and get-nodes among its children. ■



$\mathsf{send}_{xy} : \mathsf{pr} \to \mathsf{sum}(xy)$ $\mathsf{get}_{x(z)} : \mathsf{pr}(z) \to \mathsf{sum}(x)$ $\mathsf{alt} : \mathsf{sum} \to \mathsf{pr}$

Figure 7.7 Ions in $\mathbf{BBG}_{\mathrm{f}\pi}$

Figure 7.7 shows ions for the controls of $\Sigma_{\mathrm{f}\pi}$. The two first are just as for $\Sigma_{\mathrm{sf}\pi}$, except their interfaces are now equipped with sorts. The new one, alt, contains a sum-site and has a pr-region. Sorts are not explicitly indicated in the pictures, but they are suggested by slight differences in the shapes used for regions and sites.

In $\mathbf{BBG}(\Sigma_{\mathrm{f}\pi})$ we shall reserve the term *process bigraph* for bigraphs in the homset $(\epsilon, \mathsf{pr}(X))$; the bigraphs in $(\epsilon, \mathsf{sum}(X))$ we shall call *sum bigraphs*. We separate the molecules into *process molecules* and *sum molecules*, according to sort.

Lemma 7.9 In $\mathbf{BBG}(\Sigma_{\mathrm{f}\pi})$ every bigraph with names (X) belonging to one of the following classes can be expressed in the forms given:

Process bigraphs:	$(/Z)\, p$	$p : \mathsf{pr}(XZ)$ open
Open process bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n > 0$, each $\mu_i : \mathsf{pr}(X)$ open
Open sum bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n \geq 0$, each $\mu_i : \mathsf{sum}(X)$ open
Open process molecules:	$\mathsf{alt}\, m$	$m : \mathsf{sum}(X)$ open
Open sum molecules:	$\mathsf{send}_{xy}\, p$	$p : \mathsf{pr}(X)$ open
	$\mathsf{get}_{x(z)}\, p$	$p : \mathsf{pr}(Xz)$ open .

Moreover, these forms are unique up to the choice of fresh names Z, z and the ordering of the μ_i.

Proof A straightforward adaptation of the proof of Lemma 7.2 ∎

Definition 7.10 The translations $\mathsf{F}_{\mathsf{pr}(X)}[\![P]\!]$ and $\mathsf{F}_{\mathsf{sum}(X)}[\![M]\!]$ into the homsets

$(\epsilon, \mathsf{pr}(X))$ and $(\epsilon, \mathsf{sum}(X))$ of $\mathbf{BBG}(\Sigma_{f\pi})$ are defined by mutual recursion:

$$\begin{aligned}
\mathsf{F}_{\mathsf{pr}(X)}[\![M]\!] &= \mathsf{alt}\,\mathsf{F}_{\mathsf{sum}(X)}[\![M]\!] \\
\mathsf{F}_{\mathsf{pr}(X)}[\![P \mid Q]\!] &= \mathsf{F}_{\mathsf{pr}(X)}[\![P]\!] \mid \mathsf{F}_{\mathsf{pr}(X)}[\![Q]\!] \\
\mathsf{F}_{\mathsf{pr}(X)}[\![\nu z\, P]\!] &= (/w)\,\mathsf{F}_{\mathsf{pr}(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{F}_{\mathsf{sum}(X)}[\![\overline{x}y.P]\!] &= \mathsf{send}_{xy}\,\mathsf{F}_{\mathsf{pr}(X)}[\![P]\!] \\
\mathsf{F}_{\mathsf{sum}(X)}[\![x(z).P]\!] &= \mathsf{get}_{x(w)}\,\mathsf{F}_{\mathsf{pr}(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{F}_{\mathsf{sum}(X)}[\![M+N]\!] &= \mathsf{F}_{\mathsf{sum}(X)}[\![M]\!] \mid \mathsf{F}_{\mathsf{sum}(X)}[\![N]\!] \\
\mathsf{F}_{\mathsf{sum}(X)}[\![\mathbf{0}]\!] &= \mathsf{sum}(X)\,.
\end{aligned}$$

■

Note that we model the empty sum **0** as an empty bigraph of sort sum, while **0** as a process is modelled as a bigraph with a single, barren alt-node. We discuss these choices below.

We now prove the structural correspondence theorem. It is similar to that for sfπ and tells us that (1) the sorting discipline ensures that our process translation is still surjective, i.e. that there are no ground, prime, local bigraphs that do not make sense as a process; and (2) our model validates $\equiv_\nu$, but not $\equiv_{\mathbf{0},\nu}$, since it fails the structural congruence axiom ($-\mathbf{0}$) that allows **0** to be removed from a parallel composition. The reason for the latter is that the process **0** is now modelled with a non-empty bigraph.

Theorem 7.11 (structural correspondence)

(1) The map $\mathsf{F}_{\mathsf{pr}(X)}[\![-]\!]$ is surjective onto the homset $(\epsilon, \mathsf{pr}(X))$ of $\mathbf{BBG}(\Sigma_{f\pi})$, and the map $\mathsf{F}_{\mathsf{sum}(X)}[\![-]\!]$ is surjective onto $(\epsilon, \mathsf{sum}(X))$;

(2) $P \equiv_\nu Q$ iff $\mathsf{F}_{\mathsf{pr}(X)}[\![P]\!] = \mathsf{F}_{\mathsf{pr}(X)}[\![Q]\!]$.

Proof The proof proceeds as for Theorem 7.4, except the reverse translation now is

$$\begin{aligned}
\widehat{\mathsf{F}}[\![(/Z)\,p]\!] &= \nu Z\,\widehat{\mathsf{F}}_1[\![p]\!] \qquad (p \text{ open}) \\
\widehat{\mathsf{F}}_1[\![\mu_1 \mid \cdots \mid \mu_n]\!] &= \widehat{\mathsf{F}}_3[\![\mu_1]\!] \mid \cdots \mid \widehat{\mathsf{F}}_3[\![\mu_n]\!] \\
\widehat{\mathsf{F}}_2[\![\mu_1 \mid \cdots \mid \mu_n]\!] &= \widehat{\mathsf{F}}_3[\![\mu_1]\!] + \cdots + \widehat{\mathsf{F}}_3[\![\mu_n]\!] \\
\widehat{\mathsf{F}}_3[\![\mathsf{alt}\,m]\!] &= \widehat{\mathsf{F}}_2[\![m]\!] \\
\widehat{\mathsf{F}}_3[\![\mathsf{send}_{xy}\,p]\!] &= \overline{x}y.\widehat{\mathsf{F}}_1[\![p]\!] \\
\widehat{\mathsf{F}}_3[\![\mathsf{get}_{x(z)}\,p]\!] &= x(z).\widehat{\mathsf{F}}_1[\![p]\!]\,.
\end{aligned}$$

Again the result follows by verifying that the two translations are mutually inverse:

$$\mathsf{F}_{\mathsf{pr}(X)}[\![\widehat{\mathsf{F}}[\![p]\!]]\!] = p \quad \text{and} \quad \widehat{\mathsf{F}}[\![\mathsf{F}_{\mathsf{pr}(X)}[\![P]\!]]\!] \equiv_\nu P\,.$$

■

It is tempting to change the process translation to make $\mathsf{F}_{\mathsf{pr}(X)}[\![\mathbf{0}]\!]$ the empty bigraph in order to recover the axiom $(-\mathbf{0})$. To avoid ambiguity we would then have to disallow barren sum-roots and barren alt-nodes, and we would thus be prevented from modelling empty sums. Such a change would, in fact, correspond exactly to changing the syntax of the calculus in such a way that $\mathbf{0}$ occurred at the level of processes rather than the level of sums. The resulting syntax would be perfectly acceptable from the point of view that it would generate the same terms, up to structural congruence. We avoid this solution, however, because empty sums are often convenient. Most notably, without empty sums the reaction rule of the calculus, in the form that we have stated it, would not permit a unary sum to react, as this would require either of the sums M and N in the rule to be instantiated to $\mathbf{0}$. It would seem necessary, then, to add reaction rules with one or both of these sums absent, both in the calculus and the model. Such a duplication of rules would be a nuisance and complicate analysis considerably.

We now give the formal definition of $\mathcal{R}_{\mathrm{f}\pi}$ that consists of the single reaction rule already presented in Figure 7.5.

Definition 7.12 The rule set $\mathcal{R}_{\mathrm{f}\pi}$ consists of the single parametric reaction rule (R, R', η), where

$$R = \mathsf{alt}\left(\mathsf{send}_{xy} \mid id_{\langle \mathsf{sum} \rangle}\right) \mid \mathsf{alt}\left(\mathsf{get}_{x(z)} \mid id_{\langle \mathsf{sum} \rangle}\right) : \langle \mathsf{pr}, \mathsf{sum}, \mathsf{pr}(z), \mathsf{sum} \rangle \to \mathsf{pr}(xy)$$
$$R' = \left(id_{\langle \mathsf{pr} \rangle} \mid id_{\langle \mathsf{pr} \rangle} \lhd {}^{y}/_{z}\right) \lhd x : \langle \mathsf{pr}, \mathsf{pr}(z) \rangle \to \mathsf{pr}(xy)$$

and the instantiation η is given by

j	$\bar{\eta}(j)$	η_j
0	0	id_{pr}
1	2	$id_{\mathsf{pr}(z)}$

■

Note how non-linearity of the reaction rule is represented as non-surjectivity of the instantiation. The instantiation is still injective, and hence reaction in $\mathbf{BBG}_{\mathrm{f}\pi}$ is affine.

The characterization of reaction and the dynamic correspondence theorem are straightforward extensions of the corresponding results from the previous section.

Lemma 7.13 $p \to p'$ in $\mathbf{BBG}_{\mathrm{f}\pi}$ iff p and p' are of the forms

$$p = (/Z)\left(\mathsf{alt}\left(\mathsf{send}_{xy}\, q \mid m\right) \mid \mathsf{alt}\left(\mathsf{get}_{x(z)}\, s \mid n\right) \mid g\right)$$
$$p' = (/Z)\left(q \mid ({}^{y}/_{yz})\, s \mid g\right).$$

■

Theorem 7.14 (dynamic correspondence) $P \to P'$ iff $\mathsf{F}_{\mathsf{pr}(X)}[\![P]\!] \to \mathsf{F}_{\mathsf{pr}(X)}[\![P']\!]$.

■

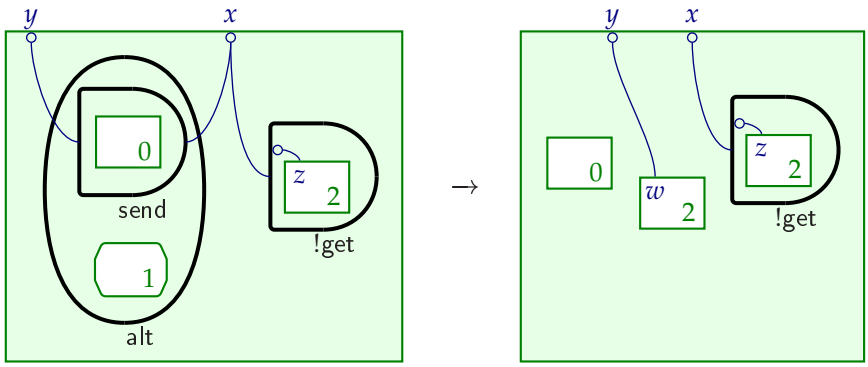


Figure 7.8 The replicating communication rule for π-calculus

7.3 Replication

We now complete the model of the π-calculus with synchronous communication by adding replicated input to the structure and the replicating communication rule to the dynamics. Essentially, we need a new control, which we shall call !get, and a new reaction rule of the form shown in Figure 7.8. It is different from the non-replicating rule in Figure 7.5 in two ways:

- First, the input itself is not consumed in the reaction step; instead an extra copy of all its children is created. This difference in dynamics is, of course, the whole point of the extension from fπ to full π.
- The second difference is structural: the syntax of our calculus does not allow replicated inputs to occur as summands, and therefore the !get-node occurs in a pr-position, unlike the get-node that occurs in sum-position.

Thus, the required extension to the model seems quite straightforward. In a replicating bigraphical reactive system, however, we must be careful with closed links. Figure 7.9 illustrates the problem. The nodes inside the !get-node share a closed link, and the question is what should happen when they are copied: Do we want the link to be shared between the copies, as in the upper reaction, or do we want each copy to have its own private link, as in the lower reaction? This question is equivalent to asking which of the following two processes should be represented by the bigraph on the left in the figure:

$$P = \overline{x}y.R \mid \nu z\, !x(u).(\overline{z}u.S \mid z(v).T)$$
$$Q = \overline{x}y.R \mid !x(u).\nu z\, (\overline{z}u.S \mid z(v).T)\ ;$$

the distinction is important, as P and Q indeed have different reactions in the π-calculus:

$$P \to P' = R \mid \nu z\, \big(\overline{z}y.S \mid z(v).T \mid !x(u).(\overline{z}u.S \mid z(v).T)\big)$$
$$Q \to Q' = R \mid \nu w\, (\overline{w}y.S \mid w(v).T) \mid \nu z\, \big(!x(u).(\overline{z}u.S \mid z(v).T)\big)\ .$$

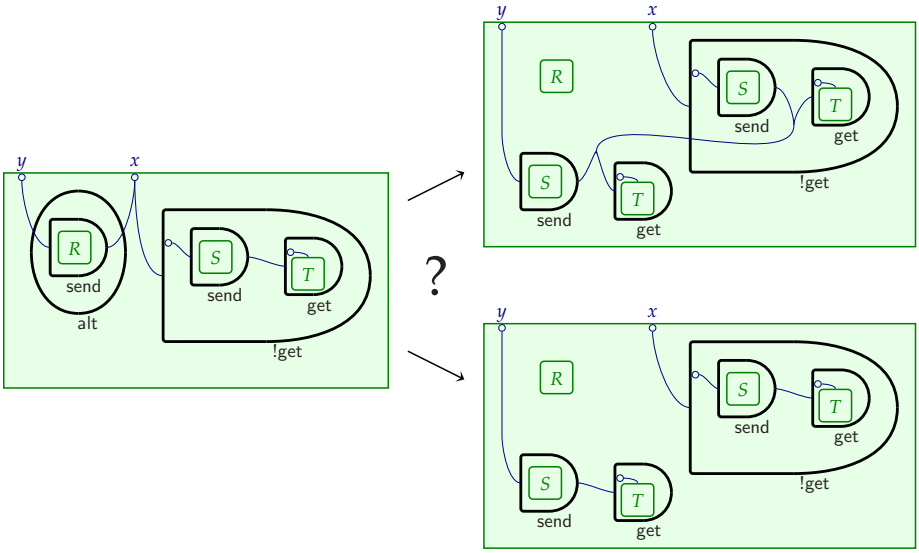


Figure 7.9 The ambiguity of replicating closed links

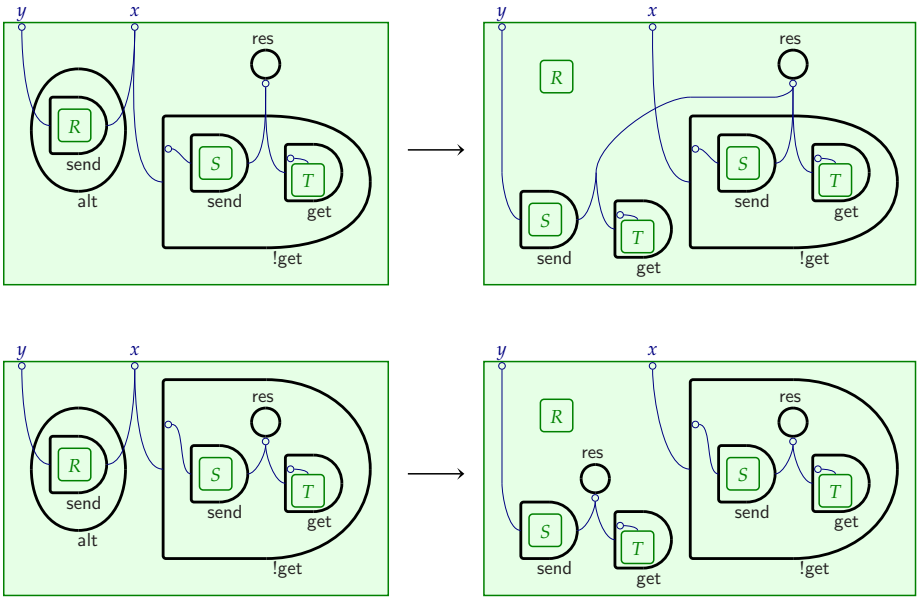


Figure 7.10 Resolving the ambiguity with restriction-nodes

The results P' and Q' correspond to the two bigraphs on the right in the figure.

Of course, our definition of parameterized reaction (Definition 5.5) resolves the apparent ambiguity, because it requires parameters to be discrete and thus, in particular, disallows name closures in the parameter. Name closures are therefore never copied, and as a result the bigraph in the figure will, in fact, behave as P, not as Q; so in the figure it is the upper reaction that will be chosen. One way to interpret this is to think of a name closure as an entity that has no definite position in a bigraph but rather "floats" to the outermost level.

Nevertheless, the problem remains that our model must allow us to represent also processes like Q, and we therefore need to refine the modelling of restriction. The solution we shall adopt is to model restriction with a node instead of name closure. This has the desired effect of "fixing" each restriction in a definite position relative to other nodes.

Figure 7.10 shows how P and Q are now modelled as distinct bigraphs; P corresponds to the upper bigraph on the left, and Q corresponds to the lower bigraph on the left. The single port of the res-node is outward-binding; this ensures that the scope of the restriction does not extend outside the surrounding node—the !get-node in the case of Q. The purpose of a res-node is simply to provide the binding port; it should never be the parent of other nodes, so we declare it to be atomic. Furthermore, we now have no use for name closure, so in order to avoid ambiguity we specify the open subcategory for our model:

$$\mathbf{BBG}_\pi = \mathbf{BBG}_{\mathrm{o}}(\Sigma_\pi, \mathcal{R}_\pi) \,.$$

Definition 7.15 The signature Σ_π is binding and has activeness and atomicity; its controls are

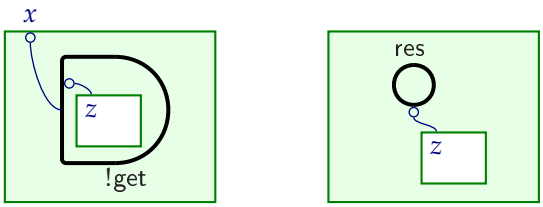
$$\begin{array}{lll} \mathsf{send} : 0 \to 2 & & \text{(passive)} \\ \mathsf{get} : \ 1 \to 1 & & \text{(passive)} \\ \mathsf{alt} : \ 0 \to 0 & & \text{(passive)} \\ !\mathsf{get} : 1 \to 1 & & \text{(passive)} \\ \mathsf{res} : \ (1) & & \text{(atomic)} \,. \end{array}$$

It has sorts pr and sum. The sorting condition stipulates the following:

- A pr-root or a send-, get- or !get-node may only have pr-sites and alt-, !get- and res-nodes among its children; it may not be barren.
- A sum-root or an alt-node may only have sum-sites and send- and get-nodes among its children. ■

Figure 7.11 shows on the left an ion for the control !get. The other new control, res, is not ionic, as it has outward binding; thus, the bigraph $\mathsf{res}_{(z)}$ on the right is not an ion. However, in analogy with our notational convention for ions, we shall write $\mathsf{res}_{(z)}\ p$ as an abbreviation of $(\mathsf{res}_{(z)} \triangleleft id_Z) \circ p$. Moreover, for $Z = \{z_1, \ldots, z_n\}$ we shall use $\mathsf{res}_{(Z)}$ as an abbreviation of $\mathsf{res}_{(z_1)} \cdots \mathsf{res}_{(z_n)}$; this is unambiguous, since the restriction nodes are unordered.

In $\mathbf{BBG}_{\mathrm{o}}(\Sigma_\pi)$ we again take the process bigraphs and sum bigraphs to be the local primes with sorts pr and sum, respectively. We shall call a bigraph



$!\mathsf{get}_{x(z)} : \mathsf{pr}(z) \to \mathsf{sum}(x)$ $\quad$ $\mathsf{res}_{(z)} : \mathsf{pr}(z) \to \mathsf{pr}$

Figure 7.11 Additional elementary bigraphs in $\mathbf{BBG}_\pi$

unrestricted if it has no restriction nodes at the outer level, that is, occurring as a child of the root.

Lemma 7.16 In $\mathbf{BBG}_{\mathrm{o}}(\Sigma_\pi)$ every bigraph with names (X) belonging to one of the following classes can be expressed in the forms given:

Process bigraphs:	$\mathsf{res}_{(Z)}\ p$	$p : \mathsf{pr}(XZ)$ unrestricted
Unrestricted process bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n > 0$, each $\mu_i : \mathsf{pr}(X)$
Sum bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n \geq 0$, each $\mu_i : \mathsf{sum}(X)$
Process molecules:	$\mathsf{alt}\ m$	$m : \mathsf{sum}(X)$
	$!\mathsf{get}_{x(z)}\ p$	$p : \mathsf{pr}(Xz)$
Sum molecules:	$\mathsf{send}_{xy}\ p$	$p : \mathsf{pr}(X)$ unrestricted
	$\mathsf{get}_{x(z)}\ p$	$p : \mathsf{pr}(Xz)$ unrestricted .

Moreover, these forms are unique up to the choice of fresh names Z, z and the ordering of the μ_i.

Proof A straightforward adaptation of the proof of Lemma 7.2 ∎

Definition 7.17 The translation $\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!]$ into the homset $(\epsilon, \mathsf{pr}(X))$ of $\mathbf{BBG}_{\mathrm{o}}(\Sigma_\pi)$ and the translation $\mathsf{P}_{\mathsf{sum}(X)}[\![M]\!]$ into the homset $(\epsilon, \mathsf{sum}(X))$ are defined by mutual recursion:

$$
\begin{aligned}
\mathsf{P}_{\mathsf{pr}(X)}[\![M]\!] &= \mathsf{alt}\,\mathsf{P}_{\mathsf{sum}(X)}[\![M]\!] \\
\mathsf{P}_{\mathsf{pr}(X)}[\![P \mid Q]\!] &= \mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] \mid \mathsf{P}_{\mathsf{pr}(X)}[\![Q]\!] \\
\mathsf{P}_{\mathsf{pr}(X)}[\![\nu z\, P]\!] &= \mathsf{res}_{(w)}\,\mathsf{P}_{\mathsf{pr}(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{P}_{\mathsf{pr}(X)}[\![!x(z).P]\!] &= !\mathsf{get}_{x(w)}\,\mathsf{P}_{\mathsf{pr}(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{P}_{\mathsf{sum}(X)}[\![\overline{x}y.P]\!] &= \mathsf{send}_{xy}\,\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] \\
\mathsf{P}_{\mathsf{sum}(X)}[\![x(z).P]\!] &= \mathsf{get}_{x(w)}\,\mathsf{P}_{\mathsf{pr}(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{P}_{\mathsf{sum}(X)}[\![M + N]\!] &= \mathsf{P}_{\mathsf{sum}(X)}[\![M]\!] \mid \mathsf{P}_{\mathsf{sum}(X)}[\![N]\!] \\
\mathsf{P}_{\mathsf{sum}(X)}[\![\mathbf{0}]\!] &= \mathsf{sum}(X)\ .
\end{aligned}
$$

∎

Two points are worth noting in the structural correspondence theorem below. First, because we are now using the open subcategory we still have surjectivity. Second, the "non-floating" representation of restriction has a side-effect: It means that the structural congruence axiom (ν-π) is no longer validated; this is not surprising as this axiom exactly allows a restriction to float past prefixes. We also lose the axiom ($-\nu$) that allows the removal of a restriction with empty scope; such a restriction will now be represented as an isolated res-node.

Theorem 7.18 (structural correspondence)

(1) The map $\mathsf{P}_{\mathsf{pr}(X)}[\![-]\!]$ is surjective onto the homset $(\epsilon, \mathsf{pr}(X))$ of $\mathbf{BBG}_{\mathrm{o}}(\Sigma_\pi)$, and the map $\mathsf{P}_{\mathsf{sum}(X)}[\![-]\!]$ is surjective onto $(\epsilon, \mathsf{sum}(X))$;

(2) $P \equiv Q$ iff $\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] = \mathsf{P}_{\mathsf{pr}(X)}[\![Q]\!]$.

Proof The proof proceeds as for Theorem 7.4, using the following reverse translation:

$$\begin{aligned}
\widehat{\mathsf{P}}[\![\mathsf{res}_{(Z)}\ p]\!] &= \nu Z\, \widehat{\mathsf{P}}_1[\![p]\!] \qquad (p \text{ unrestricted}) \\
\widehat{\mathsf{P}}_1[\![\mu_1 \mid \cdots \mid \mu_n]\!] &= \widehat{\mathsf{P}}_3[\![\mu_1]\!] \mid \cdots \mid \widehat{\mathsf{P}}_3[\![\mu_n]\!] \\
\widehat{\mathsf{P}}_2[\![\mu_1 \mid \cdots \mid \mu_n]\!] &= \widehat{\mathsf{P}}_3[\![\mu_1]\!] + \cdots + \widehat{\mathsf{P}}_3[\![\mu_n]\!] \\
\widehat{\mathsf{P}}_3[\![\mathsf{alt}\ m]\!] &= \widehat{\mathsf{P}}_2[\![m]\!] \\
\widehat{\mathsf{P}}_3[\![!\mathsf{get}_{x(z)}\ p]\!] &= !x(z).\widehat{\mathsf{P}}[\![p]\!] \\
\widehat{\mathsf{P}}_3[\![\mathsf{send}_{xy}\ p]\!] &= \overline{x}y.\widehat{\mathsf{P}}[\![p]\!] \\
\widehat{\mathsf{P}}_3[\![\mathsf{get}_{x(z)}\ p]\!] &= x(z).\widehat{\mathsf{P}}[\![p]\!]\ .
\end{aligned}$$

Again the result follows by verifying that the two translations are mutually inverse:

$$\mathsf{P}_{\mathsf{pr}(X)}[\![\widehat{\mathsf{P}}[\![p]\!]]\!] = p \quad \text{and} \quad \widehat{\mathsf{P}}[\![\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!]]\!] \equiv P\ . \qquad \blacksquare$$

We have already discussed the additional reaction rule: the formal definition is as follows:

Definition 7.19 The rule set $\mathcal{R}_\pi$ consists of the single rule in $\mathcal{R}_{\mathrm{f}\pi}$ and the additional parametric reaction rule (R, R', η), where

$$\begin{aligned}
R &= \mathsf{alt}\ (\mathsf{send}_{xy} \mid id_{\langle \mathsf{sum} \rangle}) \mid !\mathsf{get}_{x(z)} : \langle \mathsf{pr}, \mathsf{sum}, \mathsf{pr}(z) \rangle \to \langle \mathsf{pr}(xy) \rangle \\
R' &= id_{\langle \mathsf{pr} \rangle} \mid id_{\langle \mathsf{pr} \rangle} \triangleleft {}^{y}\!/_{w} \mid !\mathsf{get}_{x(z)} : \langle \mathsf{pr}, \mathsf{pr}(w), \mathsf{pr}(z) \rangle \to \langle \mathsf{pr}(xy) \rangle
\end{aligned}$$

and the instantiation η is given by

j	$\bar{\eta}(j)$	η_j
0	0	$id_{\langle \mathsf{pr} \rangle}$
1	2	$id_{\langle \mathsf{pr} \rangle} \triangleleft {}^{w}\!/_{z}$
2	2	$id_{\langle \mathsf{pr}(z) \rangle}$

$\blacksquare$

Note how the non-affinity of the new reaction rule is represented as a non-injective instantiation.

Lemma 7.20 $p \to p'$ in $\mathbf{BBG}_\pi$ iff p and p' are of the forms

$$p = \mathsf{res}_{(Z)}\ (\mathsf{alt}\ (\mathsf{send}_{xy}\, q \mid m) \mid \mathsf{alt}\ (\mathsf{get}_{x(z)}\, s \mid n) \mid g)$$
$$p' = \mathsf{res}_{(Z)}\ (q \mid (y/yz)\, s \mid g)$$

or of the forms

$$p = \mathsf{res}_{(Z)}\ (\mathsf{alt}\ (\mathsf{send}_{xy}\, q \mid m) \mid !\mathsf{get}_{x(z)}\, s \mid g)$$
$$p' = \mathsf{res}_{(Z)}\ (q \mid (y/yz)\, s \mid !\mathsf{get}_{x(z)}\, s \mid g)\,. \qquad \blacksquare$$

Dynamic correspondence is proved along the same lines as the preceding dynamic correspondence theorems.

Theorem 7.21 (dynamic correspondence) $P \to P'$ iff $\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] \to \mathsf{P}_{\mathsf{pr}(X)}[\![P']\!]$. $\blacksquare$

7.4 Asynchrony

The last version of π-calculus we shall model is aπ, in which communication is made asynchronous by restricting outputs to the form $\overline{x}y.\mathbf{0}$ (abbreviated $\overline{x}y$) and by disallowing outputs to appear in sums. The first restriction is straightforwardly represented in the BRS model by declaring the send-control *atomic*. The second restriction requires an obvious change to the sorting condition. The reaction rules, depicted in Figure 7.12, reflect these restrictions, but are otherwise as before.

The definitions and correspondence results are similar to those in the preceding section. For the model of aπ we define

$$\mathbf{BBG}_{\mathrm{a}\pi} = \mathbf{BBG}_{\mathrm{o}}(\Sigma_{\mathrm{a}\pi}, \mathcal{R}_{\mathrm{a}\pi})\,.$$

Definition 7.22 The signature $\Sigma_{\mathrm{a}\pi}$ is binding and has activeness and atomicity; its controls are

send :	$0 \to 2$	(atomic)
get :	$1 \to 1$	(passive)
alt :	$0 \to 0$	(passive)
!get :	$1 \to 1$	(passive)
res :	(1)	(atomic)\,.

It has sorts pr and sum. The sorting condition stipulates the following:

- A pr-root or a get- or !get-node may only have pr-sites and alt-, send-, !get- and res-nodes as children; it may not be barren.
- A sum-root or an alt-node may only have sum-sites and get-nodes as children. $\blacksquare$

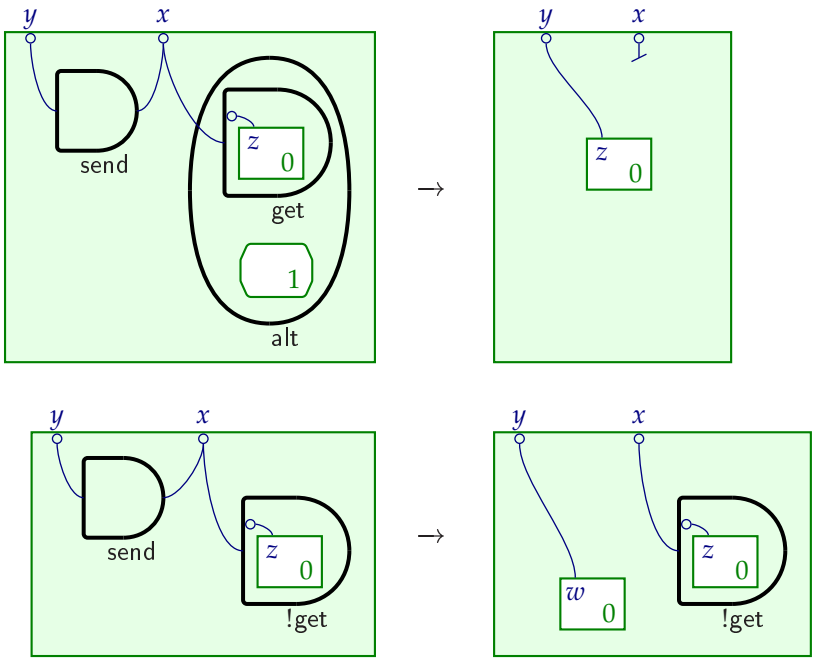
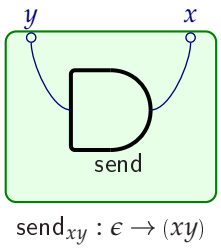


Figure 7.12 Reaction rules for asynchronous π-calculus



$\mathsf{send}_{xy} : \epsilon \to (xy)$

Figure 7.13 The atom for the control send

Figure 7.13 shows the atom that arises from the atomic control send. (There is no longer an ion for this control.)

In $\mathbf{BBG}_{\mathrm{o}}(\Sigma_{\mathrm{a}\pi})$ we again take the process bigraphs and sum bigraphs to be the local primes with sorts pr and sum, respectively.

Lemma 7.23 In $\mathbf{BBG}_{\mathrm{o}}(\Sigma_{\mathrm{a}\pi})$ every bigraph with outer face (X) belonging to one of the following classes can be expressed in the forms given:

Process bigraphs:	$\mathsf{res}_{(Z)}\, p$	$p : \mathsf{pr}(XZ)$ unrestricted
Unrestricted process bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n > 0$, each $\mu_i : \mathsf{pr}(X)$
Sum bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n \geq 0$, each $\mu_i : \mathsf{sum}(X)$
Process molecules:	$\mathsf{alt}\ \mathrm{m}$	$m : \mathsf{sum}(X)$
	send_{xy}	
	$!\mathsf{get}_{x(z)}\, p$	$p : \mathsf{pr}(Xz)$
Sum molecules:	$\mathsf{get}_{x(z)}\, p$	$p : \mathsf{pr}(Xz)$ unrestricted .

Moreover, these forms are unique up to the choice of fresh names Z, z and the ordering of the μ_i.

Proof A straightforward adaptation of the proof of Lemma 7.2 ∎

Definition 7.24 The translation $\mathsf{A}_{\mathsf{pr}(X)}[\![P]\!]$ into the homset $(\epsilon, \mathsf{pr}(X))$ of $\mathbf{BBG}_{\mathrm{o}}(\Sigma_{\mathrm{a}\pi})$ and the translation $\mathsf{A}_{\mathsf{sum}(X)}[\![M]\!]$ into the homset $(\epsilon, \mathsf{sum}(X))$ are defined by mutual recursion:

$$
\begin{aligned}
\mathsf{A}_{\mathsf{pr}(X)}[\![M]\!] &= \mathsf{alt}\ \mathsf{A}_{\mathsf{sum}(X)}[\![M]\!] \\
\mathsf{A}_{\mathsf{pr}(X)}[\![P \mid Q]\!] &= \mathsf{A}_{\mathsf{pr}(X)}[\![P]\!] \mid \mathsf{A}_{\mathsf{pr}(X)}[\![Q]\!] \\
\mathsf{A}_{\mathsf{pr}(X)}[\![\nu z\, P]\!] &= \mathsf{res}_{(w)}\, \mathsf{A}_{\mathsf{pr}(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{A}_{\mathsf{pr}(X)}[\![\overline{x}y]\!] &= \mathsf{send}_{xy} \\
\mathsf{A}_{\mathsf{pr}(X)}[\![!x(z).P]\!] &= !\mathsf{get}_{x(w)}\, \mathsf{A}_{\mathsf{pr}(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{A}_{\mathsf{sum}(X)}[\![x(z).P]\!] &= \mathsf{get}_{x(w)}\, \mathsf{A}_{\mathsf{pr}(Xw)}[\![\{w/z\}P]\!] \\
\mathsf{A}_{\mathsf{sum}(X)}[\![M + N]\!] &= \mathsf{A}_{\mathsf{sum}(X)}[\![M]\!] \mid \mathsf{A}_{\mathsf{sum}(X)}[\![N]\!] \\
\mathsf{A}_{\mathsf{sum}(X)}[\![\mathbf{0}]\!] &= \mathsf{sum}(X)\ .
\end{aligned}
$$
∎

Theorem 7.25 (structural correspondence)

(1) The map $\mathsf{A}_{\mathsf{pr}(X)}[\![-]\!]$ is surjective onto the homset $(\epsilon, \mathsf{pr}(X))$ of $\mathbf{BBG}_{\mathrm{o}}(\Sigma_{\mathrm{a}\pi})$, and the map $\mathsf{A}_{\mathsf{sum}(X)}[\![-]\!]$ is surjective onto $(\epsilon, \mathsf{sum}(X))$;

(2) $P \equiv Q$ iff $\mathsf{A}_{\mathsf{pr}(X)}[\![P]\!] = \mathsf{A}_{\mathsf{pr}(X)}[\![Q]\!]$.

Proof The proof proceeds as for Theorem 7.4, using the following reverse translation:

$$\begin{aligned}
\widehat{\mathsf{A}}[\![\mathsf{res}_{(Z)}\ p]\!] &= \nu Z\, \widehat{\mathsf{A}}_1[\![p]\!] \qquad (p \text{ unrestricted}) \\
\widehat{\mathsf{A}}_1[\![\mu_1 \mid \cdots \mid \mu_n]\!] &= \widehat{\mathsf{A}}_3[\![\mu_1]\!] \mid \cdots \mid \widehat{\mathsf{A}}_3[\![\mu_n]\!] \\
\widehat{\mathsf{A}}_2[\![\mu_1 \mid \cdots \mid \mu_n]\!] &= \widehat{\mathsf{A}}_3[\![\mu_1]\!] + \cdots + \widehat{\mathsf{A}}_3[\![\mu_n]\!] \\
\widehat{\mathsf{A}}_3[\![\mathsf{alt}\ m]\!] &= \widehat{\mathsf{A}}_2[\![m]\!] \\
\widehat{\mathsf{A}}_3[\![\mathsf{send}_{xy}]\!] &= \overline{x}y \\
\widehat{\mathsf{A}}_3[\![!\mathsf{get}_{x(z)}\ p]\!] &= !x(z).\widehat{\mathsf{A}}[\![p]\!] \\
\widehat{\mathsf{A}}_3[\![\mathsf{get}_{x(z)}\ p]\!] &= x(z).\widehat{\mathsf{A}}[\![p]\!]\,.
\end{aligned}$$

Again the result follows by verifying that the two translations are mutually inverse:

$$\mathsf{A}_{\mathsf{pr}(X)}[\![\widehat{\mathsf{A}}[\![p]\!]]\!] = p \quad \text{and} \quad \widehat{\mathsf{A}}[\![\mathsf{A}_{\mathsf{pr}(X)}[\![P]\!]]\!] \equiv P\,. \qquad \blacksquare$$

Definition 7.26 The rule set $\mathcal{R}_{a\pi}$ consists of the two rules (R_1, R_1', η_1) and (R_2, R_2', η_2), where

$$\begin{aligned}
R_1 &= \mathsf{send}_{xy} \mid \mathsf{alt}\,(\mathsf{get}_{x(z)} \mid id_{\langle \mathsf{sum} \rangle}) : \langle \mathsf{pr}(z), \mathsf{sum} \rangle \to \langle \mathsf{pr}(xy) \rangle \\
R_1' &= (id_{\langle \mathsf{pr} \rangle} \lhd {}^{y}\!/_{z}) \lhd x : \langle \mathsf{pr}(z) \rangle \to \langle \mathsf{pr}(xy) \rangle \\
R_2 &= \mathsf{send}_{xy} \mid !\mathsf{get}_{x(z)} : \langle \mathsf{pr}(z) \rangle \to \langle \mathsf{pr}(xy) \rangle \\
R_2' &= id_{\langle \mathsf{pr} \rangle} \lhd {}^{y}\!/_{w} \mid !\mathsf{get}_{x(z)} : \langle \mathsf{pr}(w), \mathsf{pr}(z) \rangle \to \langle \mathsf{pr}(xy) \rangle
\end{aligned}$$

and the instantiations η_1 and η_2 are given by

j	$\bar{\eta}_1(j)$	$(\eta_1)_j$
0	0	$id_{\langle \mathsf{pr}(z) \rangle}$

j	$\bar{\eta}_2(j)$	$(\eta_2)_j$
0	0	$id_{\langle \mathsf{pr} \rangle} \lhd {}^{w}\!/_{z}$
1	0	$id_{\langle \mathsf{pr}(z) \rangle}$

$\blacksquare$

Lemma 7.27 $p \to p'$ in $\mathbf{BBG}_{a\pi}$ iff p and p' are of the forms

$$\begin{aligned}
p &= \mathsf{res}_{(Z)}\,(\mathsf{send}_{xy} \mid \mathsf{alt}\,(\mathsf{get}_{x(z)}\, s \mid n) \mid g) \\
p' &= \mathsf{res}_{(Z)}\,(({}^{y}\!/_{yz})\, s \mid g)
\end{aligned}$$

or of the forms

$$\begin{aligned}
p &= \mathsf{res}_{(Z)}\,(\mathsf{send}_{xy} \mid !\mathsf{get}_{x(z)}\, s \mid g) \\
p' &= \mathsf{res}_{(Z)}\,(({}^{y}\!/_{yz})\, s \mid !\mathsf{get}_{x(z)}\, s \mid g)\,.
\end{aligned}$$

$\blacksquare$

Theorem 7.28 (dynamic correspondence) $P \longrightarrow P'$ iff $\mathsf{A}_{\mathsf{pr}(X)}[\![P]\!] \longrightarrow \mathsf{A}_{\mathsf{pr}(X)}[\![P']\!]$. ■

We have now completed the models of the full π-calculus both with synchronous and asynchronous communication, and we have obtained satisfactory structural and dynamic correspondence results. In the next chapter we study the behavioural theory induced on the π-calculus by our model as an instance of the theory developed in preceding chapters at the general level of bigraphs. In particular we shall compare this induced theory to the existing, well-developed behavioural theory of the π-calculus.

Chapter 8

Contextual Transitions for the π-Calculus

In this chapter we study the contextual transition systems (and the associated bisimilarities) that our bigraphical model induces on the π-calculus.

A typical use that we envisage for the behavioural theory of bigraphs is to provide transition systems and behavioural congruences to systems that lack such behavioural theory and whose behaviour is expressed purely by means of a reaction relation. That, of course, is not the situation for the π-calculus. Here, the behavioural theory induced by the bigraphical model will complement an already existing, very well-developed behavioural theory that has been carefully tailored to the specific features of the calculus. We can hardly expect the general bigraphical theory, when instantiated to the π-calculus, to provide improvements on the existing theory. Rather, we can see the existing theory as a standard by which the bigraphical theory can be measured. If we are to have confidence in the bigraphical theory when applying it to other, less well-explored systems, then we should hope at least to find it not substantially inferior when we apply it here in familiar territory.

For notational convenience, we shall in this chapter often use π-calculus syntax to denote process bigraphs. Thus, for example,

$$\begin{array}{rll} \overline{x}y.p \mid \nu w\, !x(z).q & \text{abbreviates} & \mathsf{alt}\ \mathsf{send}_{xy}\ p \mid \mathsf{res}_{(u)}\ !\mathsf{get}_{x(z)}\ q \\ p \mid q \mid \mathbf{0} & \text{abbreviates} & p \mid q \mid \mathsf{alt}\ \langle \mathsf{sum} \rangle \\ m + n + \mathbf{0} & \text{abbreviates} & \mathsf{alt}\ (m \mid n)\ . \end{array}$$

Note that an expression like the last one, which according to the π-calculus syntax could be read as either a process or a sum, will always be taken to denote a process bigraph; hence the alt-node in the defining expression.

Note also that a single expression may denote different bigraphs in the BRSs for different sub-calculi. For example, $\nu w\, p$ denotes $\mathsf{res}_{(w)}\ p$ in $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\mathrm{a}\pi}$, while in $\mathbf{BBG}_{\mathrm{sf}\pi}$ and $\mathbf{BBG}_{\mathrm{f}\pi}$ it denotes $(/w)\ p$.

8.1 Bisimilarity

We have modelled the π-calculus and its subcalculi in abstract BRSs with activeness assignments, and therefore the behavioural equivalences of interest will be the (strong and weak) active bisimilarities $\sim$ and $\approx$. These are the bisimilarities for the standard active transition systems AT and WAT (Definition 5.19), defined by taking the lean support quotients of the standard active transition systems in the corresponding s-category of concrete bigraphs.

We start by ascertaining that no ambiguity is created by the fact that we translate each process into many homsets.

Proposition 8.1 Let $X, Y \supseteq \mathrm{fn}(P, Q)$. Then

$$\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] \sim \mathsf{P}_{\mathsf{pr}(X)}[\![Q]\!] \quad \text{iff} \quad \mathsf{P}_{\mathsf{pr}(Y)}[\![P]\!] \sim \mathsf{P}_{\mathsf{pr}(Y)}[\![Q]\!] \,.$$

The analogous result holds also for $\approx$, and both results also hold for the subcalculi.

Proof Let $W = \mathrm{fn}(P, Q)$ and $Z = X - W$. Then $\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!]$ has the form $\mathsf{P}_{\mathsf{pr}(W)}[\![P]\!] \triangleleft Z$. Thus, the result will follow from proving the more general property that $p \sim q$ iff $p \triangleleft z \sim q \triangleleft z$ for an arbitrary fresh z. This property is easily established by providing suitable bisimulations (in one direction, bisimulation up to bisimilarity), and using the IPO characterization of Section 4.5. ■

It is then easy to establish that the two bisimilarities, being congruences for $\mathbf{BBG}_\pi$, are also process congruences:

Theorem 8.2 (process congruence) Strong and weak active bisimilarity $\sim$ and $\approx$ are process congruences; that is, for any process context C in π and any name set $Y \supseteq \mathrm{fn}(C[P], C[Q])$,

$$\text{if } \mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] \sim \mathsf{P}_{\mathsf{pr}(X)}[\![Q]\!] \text{ then } \mathsf{P}_{\mathsf{pr}(Y)}[\![C[P]]\!] \sim \mathsf{P}_{\mathsf{pr}(Y)}[\![C[Q]]\!] \,,$$

and similarly for $\approx$ and for the subcalculi.

Proof From compositionality of the map $\mathsf{P}_{\mathsf{pr}(X)}[\![-]\!]$ (Definition 7.17) it follows that

$$\mathsf{P}_{\mathsf{pr}(Y)}[\![C[P]]\!] = D \circ \mathsf{P}_{\mathsf{pr}(YZ)}[\![P]\!]$$
$$\mathsf{P}_{\mathsf{pr}(Y)}[\![C[Q]]\!] = D \circ \mathsf{P}_{\mathsf{pr}(YZ)}[\![Q]\!]$$

for some context D and name set Z. By the preceding proposition $\mathsf{P}_{\mathsf{pr}(YZ)}[\![P]\!] \sim \mathsf{P}_{\mathsf{pr}(YZ)}[\![Q]\!]$ iff $\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] \sim \mathsf{P}_{\mathsf{pr}(X)}[\![Q]\!]$. The result then follows directly from the general congruence result for active bisimilarity (Theorem 5.21). The same argument applies to $\approx$ and for the subcalculi. ■

8.2 Engaged transitions

For further study of the contextual bisimilarities we want to apply the theory of engaged transitions developed in Section 5.4; this will help us obtain manageable transition systems. Let us first define the appropriate notion of engaged transition:

Definition 8.3 In each of $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$, $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\pi}$ let EP denote the transition system that consists of engaged transitions over process bigraphs. ■

Lemma 8.4 The engaged transitions $p \xrightarrow{L} p'$ in $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$, $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\mathrm{a}\pi}$ are exactly those of the forms given by the cases in Table 8.1, up to a bijection on the outer names of L and p'.

Proof The engaged transitions for $\mathbf{BBG}_{\mathrm{sf}\pi}$ fall in three groups, according to how the send- and get-node of the parametric redex are distributed between the agent and the label. The case in which the agent has the send-node leads to case (a1) of the table, and the case in which the agent has the get-node leads to case (a2). When the agent has both nodes, there are two cases, according to whether the nodes are linked within the agent or not, leading to cases (a3) and (a4) respectively. In the latter, the label contains a non-injective substitution that provides the linking.

Cases (b1–b4) arise similarly for $\mathbf{BBG}_{\mathrm{f}\pi}$. The full calculus $\mathbf{BBG}_{\pi}$ has two redexes (with and without replication), and thus, in addition to cases (b1–b4), gives rise to four further cases, (b5–b8). The asynchronous calculus $\mathbf{BBG}_{\mathrm{a}\pi}$ also has two redexes; it gives rise to the analogous cases (c1–c8). ■

Adequacy and definiteness of EP are immediate:

Lemma 8.5 In each of $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$, $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\mathrm{a}\pi}$

(1) EP is adequate; that is, $\sim^{\mathrm{EP}} = \sim$ on process bigraphs;

(2) EP is definite; hence, $\sim_{\mathrm{EP}} = \sim$ on process bigraphs.

Proof

(1) Each of the four BRSs is easily verified to be simple. The result is therefore obtained directly from Theorem 5.30, using clause (1) of the theorem for $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\mathrm{a}\pi}$, and using clause (2) for $\mathbf{BBG}_{\mathrm{sf}\pi}$ and $\mathbf{BBG}_{\mathrm{f}\pi}$.

(2) In each BRS, every parametric redex has exactly two nodes at top-level, i.e. as the children of the root. From this observation, it is easy to verify that a transition is engaged iff its label L has exactly one node at top-level. Thus, the engagedness of a transition depends only on its label, so the sub-transition system of engaged transitions is definite. ■

Thus, in order to prove that process bigraphs p and q are active bisimilar it suffices to establish an engaged bisimulation (i.e. bisimulations for engaged transitions).

Case	p	L	p'
(a1)	$\nu Z\,(\overline{x}y.q \mid g)$	$(x(z).s \lhd id) \lhd id_x$	$\nu Z\,(q \mid (y/z)\,s \mid g)$
(a2)	$\nu Z\,(x(z).s \mid g)$	$(\overline{x}y.q \lhd id) \lhd id_x$	$\nu Z\,(q \mid (y/z)\,s \mid g)$
(a3)	$\nu Z\,(\overline{x}y.q \mid x(z).s \mid g)$	id	$\nu Z\,(q \mid (y/yz)\,s \mid g)$
(a4)	$\nu Z\,(\overline{x}y.q \mid u(z).s \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,(q \mid (y/yz)\,s \mid g)$
(b1)	$\nu Z\,((\overline{x}y.q + m) \mid g)$	$((x(z).s + n) \lhd id) \lhd id_x$	$\nu Z\,(q \mid (y/z)\,s \mid g)$
(b2)	$\nu Z\,((x(z).s + n) \mid g)$	$((\overline{x}y.q + m) \lhd id) \lhd id_x$	$\nu Z\,(q \mid (y/z)\,s \mid g)$
(b3)	$\nu Z\,((\overline{x}y.q + m) \mid x(z).s + n) \mid g)$	id	$\nu Z\,(q \mid (y/yz)\,s \mid g)$
(b4)	$\nu Z\,((\overline{x}y.q + m) \mid (u(z).s + n) \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,(q \mid (y/yz)\,s \mid g)$
(b5)	$\nu Z\,((\overline{x}y.q + m) \mid g)$	$(!x(z).s \lhd id) \lhd id_x$	$\nu Z\,(q \mid (y/z)\,s \mid !x(z).s \mid g)$
(b6)	$\nu Z\,(!x(z).s \mid g)$	$((\overline{x}y.q + m) \lhd id) \lhd id_x$	$\nu Z\,(q \mid (y/z)\,s \mid !x(z).s \mid g)$
(b7)	$\nu Z\,((\overline{x}y.q + m) \mid !x(z).s \mid g)$	id	$\nu Z\,(q \mid (y/yz)\,s \mid !x(z).s \mid g)$
(b8)	$\nu Z\,((\overline{x}y.q + m) \mid !u(z).s \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,(q \mid (y/yz)\,s \mid !x(z).s \mid g)$
(c1)	$\nu Z\,(\overline{x}y \mid g)$	$((x(z).s + n) \lhd id) \lhd id_x$	$\nu Z\,((y/z)\,s \mid g)$
(c2)	$\nu Z\,((x(z).s + n) \mid g)$	$(\overline{x}y \lhd id) \lhd id_x$	$\nu Z\,((y/z)\,s \mid g)$
(c3)	$\nu Z\,(\overline{x}y \mid (x(z).s + n) \mid g)$	id	$\nu Z\,((y/yz)\,s \mid g)$
(c4)	$\nu Z\,(\overline{x}y \mid (u(z).s + n) \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,((y/yz)\,s \mid g)$
(c5)	$\nu Z\,(\overline{x}y \mid g)$	$(!x(z).s \lhd id) \lhd id_x$	$\nu Z\,((y/z)\,s \mid !x(z).s \mid g)$
(c6)	$\nu Z\,(!x(z).s \mid g)$	$(\overline{x}y \lhd id) \lhd id_x$	$\nu Z\,((y/z)\,s \mid !x(z).s \mid g)$
(c7)	$\nu Z\,(\overline{x}y \mid !x(z).s \mid g)$	id	$\nu Z\,((y/yz)\,s \mid !x(z).s \mid g)$
(c8)	$\nu Z\,(\overline{x}y \mid !u(z).s \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,((y/yz)\,s \mid !x(z).s \mid g)$

Table 8.1 Forms of an engaged transition $p \xrightarrow{L} p'$.
Cases (a1–4) apply to $\mathbf{BBG}_{\mathrm{sf}\pi}$; cases (b1–4) apply to $\mathbf{BBG}_{\mathrm{f}\pi}$; cases (b1–8) apply to $\mathbf{BBG}_{\pi}$; cases (c1–8) apply to $\mathbf{BBG}_{\mathrm{a}\pi}$. In all cases the label $L = L_0 \lhd \sigma$ must have L_0 robustly discrete with names not among Z.

For $\mathbf{BBG}_{\mathrm{sf}\pi}$ and $\mathbf{BBG}_{\mathrm{f}\pi}$, which are affine, we can apply Theorem 5.30(3) to extend the above results to weak bisimilarity, but we lack results to extend the results for the full calculus to weak bisimilarity. For the remainder of the chapter we shall therefore concentrate only on strong bisimilarity, although all results have straightforward extensions to weak bisimilarity in the cases of $\mathbf{BBG}_{\mathrm{sf}\pi}$ and $\mathbf{BBG}_{\mathrm{f}\pi}$.

8.3 Trim transitions

Even though limiting the transition system to engaged transitions provides a substantial reduction in both the number of transitions and size of the labels, an examination of Table 8.1 reveals that there is still some redundancy. To remove at least some of the redundancy we shall define an optimized version of the engaged transition system. We begin with an informal discussion of the problem.

The most obvious redundancy concerns the sum n in the label of cases (b1) and (c1) and the sum m in cases (b2) and (c2): they are discarded and therefore have no effect on the result of the transition. One might therefore just as well always choose them to be the empty sum bigraph $\mathbf{0}$ (or indeed any other constant sum bigraph).

Another redundancy can be found in cases (a2), (b2) and (b6) when we consider the role of q, which occurs underneath the output prefix in the label. Though q is not discarded, it can nevertheless be shown to be redundant, essentially because it plays no important role in the result p' of the transition. The reasoning is this: Since q is part of the label it is assumed not to have names among Z. This allows us, say in case (b2), to write p' in the form $q \mid \boldsymbol{\nu} Z\,((y/z)\,s \mid g)$. Let $p'_0 = \mathbf{0} \mid \boldsymbol{\nu} Z\,((y/z)\,s \mid g)$; this is the value that p' takes in case $q = \mathbf{0}$. Then the general expression for p' can be rewritten as $(q \mid id) \circ p'_0$ (up to an extra occurrence of $\mathbf{0}$). Thus, the effect of choosing an arbitrary value for q, compared with choosing $q = \mathbf{0}$, is essentially to add "extra context" in p'. Active bisimilarity is a congruence, and therefore the freedom of including extra context does not contribute to the distinguishing power. Hence, we are justified in always choosing $q = \mathbf{0}$.

One might be tempted to try to argue analogously that s is redundant in cases (a1), (b1) and (b5), but there is an important difference: If we choose s to always be $\mathbf{0}$ then the datum y sent by p would never occur in p' or L, and then the transition system could not distinguish, say, the processes $p = \overline{x}v$ and $p = \overline{x}w$; these are clearly not bisimilar, unless $v = w$. It is possible, of course, that s can be limited to some small set of choices (some of which would necessarily have z occurring free), but such a result has proved difficult to obtain, and we shall not pursue it here.

Yet another source of redundancy concerns replicated input. Comparing the result p' in the cases (b1) and (b5) we find, by a similar argument to that above, that case (b5) is redundant because $!x(z).s$ can be pulled out of the restriction and thus represents "extra context" compared with case (b1). A simi-

lar observation applies to cases (c1) and (c5). Thus, it is not necessary to have any labels with replicated input at top-level.

Finally, Table 8.1 characterizes L and p' only up to a bijection on their outer names. Thus, each of the forms given in the table can be varied by composing both L and p' by an arbitrary bijective substitution, which gives rise to unintuitive "renaming" transitions like

$$\overline{x}y \mid \overline{x}v \xrightarrow{(x(z) \lhd {}^{u}/_{x}) \lhd id_u} \overline{u}v \,.$$

Again, such substitutions represent "extra context" compared with the name-preserving forms actually stated in the table, and are therefore redundant.

The following definition removes the redundancies we have discussed.

Definition 8.6 (trim transition) In each of $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$, $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\mathrm{a}\pi}$ we say that a transition $p \xrightarrow{L} p'$ is *trim* if it is in EP and its label L has one of the following forms:

$$\begin{gathered}(x(z).s \lhd id) \lhd id_x \\ (\overline{x}y \lhd id) \lhd id_x \\ id \\ id \lhd {}^{x}/_{xu} \,.\end{gathered}$$

We denote by TP the transition system consisting of the trim transitions over process bigraphs. ■

From this definition together with Lemma 8.4 the characterization of TP is immediate:

Lemma 8.7 The trim transitions $p \xrightarrow{L} p'$ in $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$, $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\mathrm{a}\pi}$ are exactly those of the forms given by the cases in Table 8.2. ■

The following lemma establishes the soundness of working with trim transitions.

Lemma 8.8 In each of $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$, $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\mathrm{a}\pi}$

(1) TP is adequate; that is, $\sim^{\mathrm{TP}} = \sim$ on process bigraphs;

(2) TP is definite; hence, $\sim_{\mathrm{TP}} = \sim$ on process bigraphs.

Proof

(1) The inclusion $\sim \subseteq \sim^{\mathrm{TP}}$ on process bigraphs is immediate, since the trim transitions are a subset of the standard active transitions. For the reverse inclusion we prove that $\sim^{\mathrm{TP}}_{\mathrm{EP}}$ is an EP-bisimulation up to context and active bisimilarity; this will suffice because of Lemma 8.5(2). Suppose $p \sim^{\mathrm{TP}}_{\mathrm{EP}} q$ and let $p \xrightarrow{L} p'$ in EP; we must find a transition $q \xrightarrow{L} q'$ for some q' such that $p' \, (\sim^{\mathrm{TP}}_{\mathrm{EP}})^{\mathrm{C},\sim} \, q'$. If the transition of p is trim, then the matching transition of q follows immediately. Suppose instead the transition of p is not trim. Then one

Case	p	L	p'
(a1)	$\nu Z\,(\overline{x}y.q \mid t)$	$(x(z).s \lhd id) \lhd id_x$	$\nu Z\,(q \mid (y/z)\,s \mid t)$
(a2)	$\nu Z\,(x(z).s \mid t)$	$(\overline{x}y \lhd id) \lhd id_x$	$\nu Z\,((y/z)\,s \mid t)$
(a3)	$\nu Z\,(\overline{x}y.q \mid x(z).s \mid t)$	id	$\nu Z\,(q \mid (y/yz)\,s \mid t)$
(a4)	$\nu Z\,(\overline{x}y.q \mid u(z).s \mid t)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,(q \mid (y/yz)\,s \mid t)$
(b1)	$\nu Z\,((\overline{x}y.q + m) \mid g)$	$(x(z).s \lhd id) \lhd id_x$	$\nu Z\,(q \mid (y/z)\,s \mid g)$
(b2)	$\nu Z\,((x(z).s + n) \mid g)$	$(\overline{x}y \lhd id) \lhd id_x$	$\nu Z\,(\mathbf{0} \mid (y/z)\,s \mid g)$
(b3)	$\nu Z\,((\overline{x}y.q + m) \mid (x(z).s + n) \mid g)$	id	$\nu Z\,(q \mid (y/yz)\,s \mid g)$
(b4)	$\nu Z\,((\overline{x}y.q + m) \mid (u(z).s + n) \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,(q \mid (y/yz)\,s \mid g)$
(b5)	$\nu Z\,(!x(z).s \mid g)$	$(\overline{x}y \lhd id) \lhd id_x$	$\nu Z\,(\mathbf{0} \mid (y/z)\,s \mid !x(z).s \mid g)$
(b6)	$\nu Z\,((\overline{x}y.q + m) \mid !x(z).s \mid g)$	id	$\nu Z\,(q \mid (y/z)\,s \mid !x(z).s \mid g)$
(b7)	$\nu Z\,((\overline{x}y.q + m) \mid !u(z).s \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,(q \mid (y/yz)\,s \mid !x(z).s \mid g)$
(c1)	$\nu Z\,(\overline{x}y \mid g)$	$(x(z).s \lhd id) \lhd id_x$	$\nu Z\,((y/z)\,s \mid g)$
(c2)	$\nu Z\,((x(z).s + n) \mid g)$	$(\overline{x}y \lhd id) \lhd id_x$	$\nu Z\,((y/z)\,s \mid g)$
(c3)	$\nu Z\,(\overline{x}y \mid x(z).s + n) \mid g)$	id	$\nu Z\,((y/yz)\,s \mid g)$
(c4)	$\nu Z\,(\overline{x}y \mid (u(z).s + n) \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,((y/yz)\,s \mid g)$
(c5)	$\nu Z\,(!x(z).s \mid g)$	$(\overline{x}y \lhd id) \lhd id_x$	$\nu Z\,((y/z)\,s \mid !x(z).s \mid g)$
(c6)	$\nu Z\,(\overline{x}y \mid !x(z).s \mid g)$	id	$\nu Z\,((y/z)\,s \mid !x(z).s \mid g)$
(c7)	$\nu Z\,(\overline{x}y \mid !u(z).s \mid g)$	$id \lhd x/xu$	$(x/xu)\,\nu Z\,((y/yz)\,s \mid !x(z).s \mid g)$

Table 8.2 Forms of a trim transition $p \xrightarrow{L} p'$.
Cases (a1–4) apply to $\mathbf{BBG}_{\mathrm{sf}\pi}$; cases (b1–4) apply to $\mathbf{BBG}_{\mathrm{f}\pi}$; cases (b1–7) apply to $\mathbf{BBG}_{\pi}$. cases (c1–7) apply to $\mathbf{BBG}_{\mathrm{a}\pi}$. In all cases the label $L = L_0 \lhd \sigma$ must have L_0 robustly discrete with names not among Z.

of the cases (b1–8) of Table 8.1 applies. The proof is similar for all cases; we treat just case (b2), where we have

$$\begin{aligned} p &= \boldsymbol{\nu} Z\,((x(z).p_0 + l) \mid g) \\ L &= ((\overline{x}y.s + m) \lhd id) \lhd id_x \\ p' &= \boldsymbol{\nu} Z\,(s \mid (y/z)\,p_0 \mid g)\,, \end{aligned}$$

such that L has no names among Z. Then we can infer another transition $p \xrightarrow{M} p'_0$ in EP, where $M \stackrel{\text{def}}{=} (\overline{x}y \lhd id) \lhd id_x$ and $p'_0 \stackrel{\text{def}}{=} \boldsymbol{\nu} Z\,(\mathbf{0} \mid y/z\,p_0 \mid g)$. This transition is trim, so by the assumption that $p \sim^{\text{TP}}_{\text{EP}} q$ there is also a transition $q \xrightarrow{M} q'_0$ in EP for some q'_0 such that $p'_0 \sim^{\text{TP}}_{\text{EP}} q'_0$. Using Table 8.1 again, we find that q has the form $\boldsymbol{\nu} W\,((x(z).q_0 + n) \mid h)$ with W disjoint from the names of L, and $q'_0 = \boldsymbol{\nu} W\,(\mathbf{0} \mid y/z\,q_0 \mid h)$. Then we can infer another EP-transition $q \xrightarrow{L} q' \stackrel{\text{def}}{=} \boldsymbol{\nu} W\,(s \mid (y/z)\,q_0 \mid h)$. Note that $p' \mid \mathbf{0} = p'_0 \mid s$ and $q' \mid \mathbf{0} = q'_0 \mid s$, since s is part of L and therefore has no names among Z or W; hence $p' \sim p'_0 \mid s$ and $q' \sim q'_0 \mid s$, and it follows that $p'\ (\sim^{\text{TP}}_{\text{EP}})^{\mathsf{C},\sim}\ q'$, as required.

(2) Immediate from definiteness of EP and the definition of TP. ■

Thus, we have further optimized the proof technique for bisimilarity in our BRSs for π-calculus: A trim bisimulation is sufficient to show bisimilarity.

8.4 Comparison with raw bisimilarities

For the comparison of bisimilarities we shall map early and ground bisimilarity from the π-calculus into the BRSs and do the comparison there. An alternative to this approach would be to reflect the bigraphical bisimilarity back into the calculus; but this is technically less convenient, because the translation of process bigraphs back into the π-calculus loses information about idle names, and hence about the exact contents of interfaces, which makes it awkward to represent the bigraphical transitions. The following definition therefore sets up the raw transition systems on process bigraphs that arise as the images of early and late transitions in the π-calculus.

Definition 8.9 In each of $\mathbf{BBG}_{\text{sf}\pi}$, $\mathbf{BBG}_{\text{f}\pi}$, $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\text{a}\pi}$ the *early transition system* has process bigraphs as agents, and its transition relation is the least such that $\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] \xrightarrow{\ell} \mathsf{P}_{\mathsf{pr}(X)}[\![P']\!]$ for every early transition $P \xrightarrow{\ell} P'$ and every $X \supseteq \text{fn}(P, Q)$ (and similarly for the subcalculi). The *late transition system* is similarly defined as the image of the late transition relation in the calculus. We denote by $\overset{\text{e}}{\sim}_\pi$ and $\overset{\text{g}}{\sim}_\pi$ the bisimilarities for early and late transitions, respectively; by $\sim_\pi$ the relation such that $p \sim_\pi q$ iff $(\sigma)\,p \overset{\text{e}}{\sim} (\sigma)\,q$ for every substitution σ; and by $\overset{\text{a}}{\sim}_\pi$ the asynchronous bisimilarity on late transitions, where 'asynchronous' refers to the weakened input matching described in Section 6.5. ■

We call the early and late transitions, collectively, *raw transitions*, and we shall also use 'raw' as a collective qualification for the equivalences defined in terms of raw transitions. In contrast, we shall use the term 'contextual' to refer to trim transitions and their associated bisimilarity, which, as we have seen, coincides with active bisimilarity on process bigraphs.

Note that we defined the raw equivalences by co-induction in the BRSs, not as images of the corresponding equivalences in the π-calculus; but using Proposition 6.3 it is easy to demonstrate that the raw equivalences do indeed coincide with these images:

Proposition 8.10 $P \overset{e}{\sim} Q$ iff $\mathsf{P}_{\mathsf{pr}(X)}[\![P]\!] \overset{e}{\sim}_\pi \mathsf{P}_{\mathsf{pr}(X)}[\![Q]\!]$, and similarly for the subcalculi. The analogous statement is true also of $\overset{g}{\sim}$, $\sim$ and $\overset{a}{\sim}$. ∎

The comparison of equivalences will rely on the following lemma, which characterizes raw transitions in bigraphs.

Lemma 8.11 The raw transitions $p \xrightarrow{\ell} p'$ in $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$, $\mathbf{BBG}_{\pi}$ and $\mathbf{BBG}_{\mathrm{a}\pi}$ are exactly those of the forms given by the cases in Table 8.3.

Proof Via the bijection between processes (up to structural congruence) and process bigraphs, the present characterization is equivalent to a standard characterization of transitions in the π-calculus. The latter is proved, in one direction, by exhibiting transition inferences for each form of transition stated, and in the other direction, by showing inductively on the depth of transition inference that each transition has one of the forms stated. ∎

Comparing the characterizations of trim transitions (Table 8.2) and raw transitions (Table 8.3) we find that, apart from the somewhat laborious notation employed for the contextual labels of the former, they agree to a large extent but also show important differences. In preparation for the comparison of the associated equivalences, we discuss the relationship between trim and raw transitions informally.

On *silent transitions* (cases (a3, b3, b6, c3, c6) in Table 8.2 and cases (a5, b5, b8, c3–4) in Table 8.3) the two systems agree completely; thus, as expected, the raw label τ corresponds to the identity contextual label.

The contextual *input transitions* (cases (a2, b2, b5, c2, c5) in Table 8.2) essentially agree with raw bound input transitions (cases (a4, b4, b7) in Table 8.3), the only difference being the (insignificant) extra occurrence of $\mathbf{0}$ in cases (b2, b5) of Table 8.2. The apparent difference that the contextual transitions result in a substitution y/z in p', where bound inputs do not, is only superficial, as y is forced to be chosen fresh; the substitution is therefore injective, and it can be mimicked by an alpha-conversion in the agents of a raw transition. On the other hand, the substitution y/yz that arises in a raw free input transition (cases (a3, b3, b6) in Table 8.3) is not in general injective; free input transitions therefore provide greater observational power than the contextual input transitions.

Case	p	ℓ	p'	Conditions
(a1)	$\nu Z\,(\overline{x}y.q \mid g)$	$\overline{x}y$	$\nu Z\,(q \mid g)$	$x, y \notin Z$
(a2)	$\nu Zz\,(\overline{x}z.q \mid g)$	$\overline{x}(z)$	$\nu Z\,(q \mid g)$	$x \notin Zz$
(a3)	$\nu Z\,(x(z).s \mid g)$	xy	$\nu Z\,((y/yz)\,s \mid g)$	$x, y \notin Z$
(a4)	$\nu Z\,(x(z).s \mid g)$	$x(z)$	$\nu Z\,(s \mid g)$	$x, z \notin Z$
(a5)	$\nu Z\,(\overline{x}y.q \mid x(z).s \mid g)$	τ	$\nu Z\,(q \mid (y/yz)\,s \mid g)$	
(b1)	$\nu Z\,((\overline{x}y.q + m) \mid g)$	$\overline{x}y$	$\nu Z\,(q \mid g)$	$x, y \notin Z$
(b2)	$\nu Zz\,((\overline{x}z.q + m) \mid g)$	$\overline{x}(z)$	$\nu Z\,(q \mid g)$	$x \notin Zz$
(b3)	$\nu Z\,((x(z).s + n) \mid g)$	xy	$\nu Z\,((y/yz)\,s \mid g)$	$x, y \notin Z$
(b4)	$\nu Z\,((x(z).s + n) \mid g)$	$x(z)$	$\nu Z\,(s \mid g)$	$x, z \notin Z$
(b5)	$\nu Z\,((\overline{x}y.q + m) \mid (x(z).s + n) \mid g)$	τ	$\nu Z\,(q \mid (y/yz)\,s \mid g)$	
(b6)	$\nu Z\,(!x(z).s \mid g)$	xy	$\nu Z\,((y/yz)\,s \mid !x(z).s \mid g)$	$x, y \notin Z$
(b7)	$\nu Z\,(!x(z).s \mid g)$	$x(z)$	$\nu Z\,(s \mid !x(z).s \mid g)$	$x, z \notin Z$
(b8)	$\nu Z\,((\overline{x}y.q + m) \mid !x(z).s \mid g)$	τ	$\nu Z\,(q \mid (y/yz)\,s \mid !x(z).s \mid g)$	
(c1)	$\overline{x}y \mid g$	$\overline{x}y$	$\mathbf{0} \mid g$	
(c2)	$\nu z\,(\overline{x}z \mid g)$	$\overline{x}(z)$	$\mathbf{0} \mid g$	$x \neq z$
(c3)	$\nu Z\,(\overline{x}y \mid (x(z).s + n) \mid g)$	τ	$\nu Z\,((y/yz)\,s \mid g)$	
(c4)	$\nu Z\,(\overline{x}y \mid !x(z).s \mid g)$	τ	$\nu Z\,((y/yz)\,s \mid !x(z).s \mid g)$	

Table 8.3 Forms of a raw transition $p \xrightarrow{\ell} p'$.
Cases (a1–5) apply to $\mathbf{BBG}_{\mathrm{sf}\pi}$; cases (b1–5) apply to $\mathbf{BBG}_{\mathrm{f}\pi}$; cases (b1–8) apply to $\mathbf{BBG}_{\pi}$; cases (b3–4, b6–7, c1–4) apply to $\mathbf{BBG}_{\mathrm{a}\pi}$.

Output transitions have only one basic form in the contextual transition system (cases (a1, b1, c1) in Table 8.2); this does the job of both free and bound output transitions in the raw system (cases (a1–2, b1–2, c1–2) in Table 8.3). (Recall that, unlike free and bound input, both forms of output are used in both the early and late transition systems.) The contextual label for output contains an arbitrary process bigraph s that also appears in the result p' of the transition; this is an undesirable feature, since it makes bisimulation proofs more difficult. Having two forms of output in the raw system can be seen as the price of avoiding such an arbitrary process parameter in the label. The contextual transition can represent either free or bound output, depending on whether the datum y is among the restricted names Z. Where a bound output transition (cases (a2, b2, c2) in Table 8.3) removes the restriction on z in p', the contextual transition retains the restriction but inserts in its scope the arbitrary parameter s. We might say that "extrusion of names" is replaced by "intrusion of agents".

The *substitution transitions* in the contextual transition system (cases (a4, b4, b7, c4, c7) in Table 8.2) have no counterpart in the raw system. Such a transition arises when the only contribution of the context is a substitution that connects the channels of the output and input prefixes. We can see these transitions as recovering the observational power "lost" in contextual input transitions relative to free input. Apart from this, it is clear that in the absence of this class of transitions, contextual bisimilarity would not be closed under substitution of names, and hence not a congruence with respect to all bigraphical contexts.

From these observations it seems one could reasonably hope for contextual bisimilarity to recover early congruence (or some other existing behavioural congruence on the π-calculus). As we shall see, however, it is stronger than existing congruences in the synchronous calculi—the substitution transitions provide stronger observational power than that afforded by any conventional approaches.

In the synchronous calculi it is of some interest, therefore, to study also the version of contextual bisimilarity that arises from simply discarding the substitution transitions. Apart from its independent interest, it will also serve as a useful step in the analysis of full contextual bisimilarity itself. We call the transition system without substitutions *mono*, because its labels, unlike the substitutions, are monos in the underlying category.

Definition 8.12 (mono transitions) In each BRS, let the *mono transition system*, denoted $^{\cdot}\mathrm{TP}$ consist of all trim transitions except those labelled with a substitution. We write $\dot{\sim}$ for the associated bisimilarity called *(strong) mono bisimilarity*. ■

Lemma 8.13 In either of $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_{\pi}$, let $p, q : (X)$, and suppose $\nu w\,(\overline{u}w \mid p) \dot{\sim} \nu w\,(\overline{u}w \mid q)$ for some u and w with $u \notin X$. Then $p \dot{\sim} q$.

Proof See Appendix A. ■

The property in the lemma is not true of full contextual bisimilarity, because substitution transitions cannot be mimicked "through" the restriction in

the same way that we do for mono transitions in the proof. For example, the processes

$$p = \overline{x} \mid y$$
$$q = \overline{x}.y + y.\overline{x}\,,$$

which we have used in examples before, are bisimilar under $\dot{\sim}$, but not under $\sim$, because only p has a substitution transition; but, as the reader may easily verify, it holds that

$$\nu y\,(\overline{u}y \mid p) \sim \nu y\,(\overline{u}y \mid q)\,.$$

The lemma does not hold in $\mathbf{BBG}_{\mathrm{a}\pi}$. The reason for this is directly related to the preceding discussion, for as we shall see later, mono bisimilarity does coincide with contextual bisimilarity in $\mathbf{BBG}_{\mathrm{a}\pi}$.

Using the lemma we now establish the main result about mono bisimilarity:

Theorem 8.14 (mono bisimilarity) In $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_{\pi}$ mono bisimilarity lies between early and ground bisimilarity:

$$\overset{\mathrm{e}}{\sim}_{\pi} \subseteq \dot{\sim} \subseteq \overset{\mathrm{g}}{\sim}_{\pi}\,.$$

Moreover, in $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_{\pi}$ both inclusions are strict.

Proof See Appendix A. ∎

We have not settled the question whether $\dot{\sim}$ coincides with either of $\overset{\mathrm{e}}{\sim}_{\pi}$ and $\overset{\mathrm{g}}{\sim}_{\pi}$ in $\mathbf{BBG}_{\mathrm{sf}\pi}$; indeed, it appears to be an open question whether even $\overset{\mathrm{e}}{\sim}$ and $\overset{\mathrm{g}}{\sim}$ differ in $\mathrm{sf}\pi$.[1]

It is clear from the examples that $\dot{\sim}$, like $\overset{\mathrm{g}}{\sim}_{\pi}$, is not always preserved by parallel composition. For this reason it is of little interest from a practical point of view. However, using the result that it is included in ground bisimilarity, it is now an easy step to show that our main equivalence, contextual bisimilarity, is included in early congruence:

Theorem 8.15 (contextual bisimilarity) In $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_{\pi}$ contextual bisimilarity implies early congruence:

$$\sim \subseteq \sim_{\pi}\,.$$

Moreover, in $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_{\pi}$ the inclusion is strict.

Proof Since $\dot{}\mathrm{TP} \preccurlyeq \mathrm{TP}$ we have $\sim \subseteq \dot{\sim}$, and hence by the preceding theorem also $\sim \subseteq \overset{\mathrm{g}}{\sim}_{\pi}$. But $\sim$ is a process congruence (Theorem 8.2), and $\sim_{\pi}$ is the largest process congruence included in $\overset{\mathrm{g}}{\sim}_{\pi}$; hence $\sim \subseteq \sim_{\pi}$.

[1] The question is stated in [38] to be open; to the author's knowledge it has not been settled since the publication of that book.

For strictness, consider the following processes

$$p = \nu z\,((\overline{x} + \overline{z}) \mid (y + z))$$
$$q = \nu z\,((\overline{x}.y + y.\overline{x} + z) \mid \overline{z})\,.$$

It is easy to verify that $p \overset{e}{\sim}_\pi q$, and also that $(x/xy)\,p \overset{e}{\sim}_\pi (x/xy)\,q$; hence $p \sim_\pi q$. But only p has a substitution transition, so $p \not\sim q$. ∎

The example used for strictness is due to Milner, who uses it in [26] to show the similar result for CCS. Again, the example uses '+', and so does not apply to $\mathbf{BBG}_{\mathrm{sf}\pi}$, where we leave the strictness question open.

The strongest congruence that is conventionally applied in the π-calculus is Sangiorgi's *open bisimilarity* [36, 38]. Its main motivation is very similar to that of our contextual notion of bisimilarity, namely to provide a co-inductively defined process congruence. It does not use substitution transitions; instead it employs substitution in the transition matching requirement as follows:

$$\sigma P \xrightarrow{\ell} P' \quad \text{implies} \quad \sigma Q \xrightarrow{\ell} Q'\,;$$

the effect is that an open bisimulation is forced to be substitution-closed. The substitution σ is not arbitrary, as provision is made for not coalescing names that have earlier appeared in bound outputs; the full definition is therefore somewhat involved, and we omit the details. It is clear, however, that even open bisimilarity does not distinguish the processes p and q that we used in the above proof. Thus, in $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_\pi$ contextual bisimilarity differs also from open bisimilarity. (We conjecture that it is included in open bisimilarity, but do not pursue the issue.)

Thus, we cannot claim that the equivalence recovered by contextual bisimilarity in $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_\pi$ is a well-known equivalence already proven to be useful for a particular purpose. From the examples we have given of the extra distinctions it makes, it is not clear that the difference is more than pathological; it might well be insignificant for many applications. A more detailed study, involving practical applications of the π-calculus, seems to be required to settle the question.

It still remains to make the comparison between contextual bisimilarity and existing equivalences in the asynchronous calculus. Here, as may be recalled from Section 6.5, the situation is simplified by the fact that early and ground bisimilarity coincide and form a congruence, which we shall denote here by $\sim_\pi$. On the other hand, it is complicated by the existence of asynchronous bisimilarity $\overset{a}{\sim}_\pi$, a strictly weaker congruence.

Let us investigate how contextual bisimilarity compares with $\sim_\pi$ and $\overset{a}{\sim}_\pi$ on a few examples. Recall that for

$$p = x(z).\overline{x}z + \tau$$
$$q = \tau\,.$$

we have $p \overset{\text{a}}{\sim}_\pi q$ but $p \not\sim_\pi q$. Clearly, we also have $p \not\sim q$, as there is a contextual transition

$$p \xrightarrow{\overline{x}v \triangleleft id_x} \overline{x}v \ ,$$

whereas q has no non-identity transitions. Thus, contextual bisimilarity does not, in general, admit the weakened form of input-matching employed in asynchronous bisimilarity. But now consider the processes

$$p_1 \overset{\text{def}}{=} \nu x\,(\overline{u}x \mid p)$$
$$q_1 \overset{\text{def}}{=} \nu x\,(\overline{u}x \mid q) \ .$$

By congruence, $p_1 \overset{\text{a}}{\sim}_\pi q_1$; but $p_1 \not\sim_\pi q_1$, as the raw transition sequence

$$p_1 \xrightarrow{\overline{u}(x)} p \xrightarrow{x(z)}$$

cannot be matched by q_1. As we noted above, contextual transitions employ "agent intrusion" rather than "name extrusion"; this is witnessed by the transitions

$$p_1 \xrightarrow{u(x).s \triangleleft id_u} \nu x\,(s \mid p)$$
$$q_1 \xrightarrow{u(x).s \triangleleft id_u} \nu x\,(s \mid q) \ .$$

Is there a choice of s that will distinguish between the two processes? If there were, it would necessarily involve an interaction with p that could not be mimicked by q; thus s would have to be of the form $\overline{x}v \mid s_0$ (or of a similar form with v restricted, in which case the argument is similar). The interaction between s and p is represented by the transition

$$\nu x\,(s \mid p) \xrightarrow{id} \nu x\,(s_0 \mid \overline{x}v) \ ;$$

but this transition can be matched by $\nu x\,(s \mid q)$, because internal action in q leads to the transition

$$\nu x\,(s \mid q) \xrightarrow{id} \nu x\,(s \mid \mathbf{0}) = \nu x\,(s_0 \mid \overline{x}v \mid \mathbf{0}) \ .$$

Thus, apart from the insignificant extra occurrence of $\mathbf{0}$, the process q_1 does succeed in matching the transitions of p_1, and it does so exactly because the "intruder" s has no more distinguishing power than that afforded by asynchronous bisimilarity. Thus, even though contextual bisimilarity does not admit the weakened form of input matching in general, it seems to do so when the channel of the input is a private channel that the agent has previously made available to the environment. From these observations, it looks as if contextual bisimilarity falls strictly between $\sim_\pi$ and $\overset{\text{a}}{\sim}_\pi$. To show that this is indeed the case, we shall characterize contextual bisimilarity in terms of the following raw equivalences (defined here in $\mathbf{BBG}_{\text{a}\pi}$, but the definition can be made analogously in aπ itself.)

Definition 8.16 (restriction-asynchronous bisimilarity) A *(strong) restriction-asynchronous simulation* is a family $\mathcal{S} = \{\mathcal{S}_Z\}$ of relations indexed by name sets Z, where each $\mathcal{S}_Z$ is defined over the homsets $(\epsilon, \langle pr(X)\rangle$ with $X \cap Z = \emptyset$; whenever $(p,q) \in \mathcal{S}_Z$ and $p \xrightarrow{\ell} p'$ we require

(1) if ℓ is a free output $\overline{x}y$ or the silent action τ, then $q \xrightarrow{\ell} q'$ for some q' such that $(p',q') \in \mathcal{S}_Z$;

(2) if ℓ is an input $x(z)$ then either

(a) $q \xrightarrow{x(z)} q'$ for some q' such that $(p',q') \in \mathcal{S}_Z$; or

(b) $x \in Z$ and $q \xrightarrow{\tau} q'$ for some q' such that $(p', q' \mid \overline{x}z) \in \mathcal{S}_Z$;

(3) if ℓ is a bound output $\overline{x}(z)$ then $q \xrightarrow{\overline{x}(z)} q'$ for some q' such that $(p',q') \in \mathcal{S}_{Z\cup\{z\}}$.

$\mathcal{S}$ is a *(strong) restriction-asynchronous bisimulation* if also its inverse, the family consisting of all $(\mathcal{S}_Z)^{-1}$, is a strong restriction-asynchronous simulation. We write $\overset{\text{ra}}{\sim}\pi = \{\overset{\text{ra}}{\sim}_Z \pi\}$ for the largest strong restriction-asynchronous bisimulation. *Strong restriction-asynchronous bisimilarity* is the relation $\overset{\text{ra}}{\sim}_\emptyset \pi$, which by a slight abuse of notation we also denote by $\overset{\text{ra}}{\sim}\pi$. ■

In this definition we use the set Z to remember which private names have been output from the agents (clause (3)) and we allow the additional input matching only when the input channel x lies in Z (clause (2b)). Thus, the requirement is stronger than in asynchronous bisimilarity, but weaker than in ordinary bisimilarity. We therefore have inclusions as stated in the following proposition, and the inclusions are easily seen to be strict by considering the examples discussed above.

Proposition 8.17 The following inclusions hold in $\mathbf{BBG}_{a\pi}$ and are strict:

$$\sim_\pi \subseteq \overset{\text{ra}}{\sim}_\pi \subseteq \overset{\text{a}}{\sim}_\pi .$$

■

Theorem 8.18 In $\mathbf{BBG}_{a\pi}$ contextual bisimilarity and restriction-asynchronous bisimilarity coincide:

$$\sim = \overset{\text{ra}}{\sim}_\pi .$$

Proof See Appendix A. ■

In conclusion, the situation for aπ is similar to that for the full π-calculus, namely that contextual bisimilarity recovers a congruence that is close to pre-existing equivalences but does not coincide with any of them. Our overall purpose with this study—to determine whether the general behavioural theory of bigraphs, when instantiated to the π-calculus, yields satisfactory results

compared to the existing behavioural theory for π-calculus—is therefore only partially achieved: We can conclude that considerable agreement exists among the two theories, but also that there are small differences whose importance can only be judged from a certain amount of practical experience with the theories.

Chapter 9

Mobile Ambients

In this chapter we repeat the exercise of the previous chapters and model another calculus of mobile processes, namely the calculus of *mobile ambients* of Cardelli and Gordon [4]. This calculus represents a different view of mobility from that embodied by the π-calculus. In the π-calculus mobility is represented by movement of communication channels, and we naturally modelled that as movement of links in the bigraph model. In mobile ambients, on the other hand, an ambient acts both as an entity that can be moved and as a location that may contain sub-ambients; thus, the operational semantics of the calculus deals with movement of (sub-)terms, unlike that of the π-calculus which deals only with movement of names.

One purpose of making this second application of bigraphs is to help illustrate the range of applicability of the framework, and to provide evidence that bigraphs and their behavioural theory are not overly influenced by the specifics of the π-calculus. Moreover, this application takes us into territory where a reaction semantics forms the primary (and indeed original) definition of dynamics; it is fair to say that no widely accepted notion of labelled transitions exists a priori for mobile ambients, although the question has been the subject of some relatively recent work [20, 21]; we comment in some detail on the relationship with that work at the end of the chapter.

In comparison with the material on the π-calculus model, our account in this chapter of the mobile ambients model is much shorter. Partly, this is because detailed explanation of similar concepts and results is not repeated, and because we omit proofs, which are all straightforwardly adapted from the previous ones. We also put less emphasis on completeness, modelling just the fragment of the calculus concerned with mobility, and leaving out communication primitives and replication. Thus, we concentrate on the features that are substantially new compared with the π-calculus, and although we shall not attempt to justify the conjecture, we suggest that a model of the full ambient calculus could be obtained essentially as a combination of the two models without major difficulty.

We start by giving in the first section a brief review of mobile ambients. In

Section 9.2 we build the bigraph model and establish structural and dynamic correspondence. Section 9.3 studies the behavioural theory induced on mobile ambients by the model and compares it with other work on transition semantics for the calculus.

9.1 Syntax and semantics

The *processes* of (our fragment of) mobile ambients are given by the following abstract syntax:

$$P, Q \ ::= \ n[P] \ \mid \ P \mid Q \ \mid \ \nu n\, P \ \mid \ \mathbf{0} \ \mid \ \mathsf{in}\, n.P \ \mid \ \mathsf{out}\, n.P \ \mid \ \mathsf{open}\, n.P\,,$$

where n denotes a *name*, drawn from an infinite set and ranged over in this chapter by l, m, n. The expression $n[P]$ denotes an *ambient* named n and containing the process P; parallel composition '$\mid$', restriction 'ν', and inaction '$\mathbf{0}$', are as in the π-calculus. In the last three process expressions the prefixes denote *capabilities*:

- $\mathsf{in}\, n$ is the capability to enter into ambient n;
- $\mathsf{out}\, n$ is the capability to exit out of ambient n;
- $\mathsf{open}\, n$ is the capability to open ambient n.

We use π to range over prefixes.

Bound and free names, alpha-conversion, and substitution of free names, are as in the π-calculus, except that in mobile ambients the only binding construct is restriction.

In applications it is usual to model entities by processes in which every prefix occurs inside an ambient. We shall refer to such processes as *systems*. Systems are easily seen to be generated by the abstract syntax

$$M, N \ ::= \ n[P] \ \mid \ M \mid N \ \mid \ \nu n\, M \ \mid \ \mathbf{0}$$

Structural congruence $\equiv$ is the smallest congruence over process terms that includes alpha-equivalence and satisfies the following axioms:

$$\begin{aligned}
(P \mid Q) \mid R &\equiv P \mid (Q \mid R) \\
P \mid Q &\equiv Q \mid P \\
P \mid \mathbf{0} &\equiv P \\
\\
\nu m\, \nu n\, P &\equiv \nu n\, \nu m\, P \\
\nu m\, n[P] &\equiv n[\nu m\, P] && \text{if } m \neq n \\
\nu m\, (P \mid Q) &\equiv P \mid \nu m\, Q && \text{if } m \notin \mathrm{fn}(P) \\
\nu m\, \pi.P &\equiv \pi.\nu m\, P && \text{if } m \notin \mathrm{names}(\pi) \\
\nu m\, \mathbf{0} &\equiv \mathbf{0}\,.
\end{aligned}$$

Note that the property of being a system is invariant under structural congruence.

The penultimate axiom that allows permutation of restriction and prefix is not normally included. The situation is similar to that in the π-calculus: Including the axiom makes for a natural graphical model of structural congruence, and at the same time does not affect the dynamics.

The behaviour of processes is given as a reaction relation in the same style as in the π-calculus. There are three reaction rules:

$$\begin{aligned} n[\mathsf{in}\, m.P \mid Q] \mid m[R] &\to m[n[P \mid Q] \mid R] && \text{(enter)} \\ m[n[\mathsf{out}\, m.P \mid Q] \mid R] &\to n[P \mid Q] \mid m[R] && \text{(exit)} \\ \mathsf{open}\, n.P \mid n[Q] &\to P \mid Q && \text{(open)}\,. \end{aligned}$$

The 'enter' rule expresses the movement of ambient n into another ambient m; the 'exit' rule expresses the movement of ambient n out of the immediately surrounding ambient m; and the 'open' rule expresses the opening (or lifting) of the ambient n.

The reaction relation $\to$ is the least relation over processes which admits the three reaction rules, is preserved by all non-prefix contexts, and is closed under structural congruence. (By a *non-prefix context* we mean one in which the hole is not underneath any prefix.)

9.2 The bigraph model

We model mobile ambients using the BRS

$$\mathbf{BBG}_{\mathrm{ma}} = \mathbf{BBG}(\Sigma_{\mathrm{ma}}, \mathcal{R}_{\mathrm{ma}})\,,$$

where the signature Σ_{ma} is given by the following definition. (We give the reaction rules $\mathcal{R}_{\mathrm{ma}}$ later.)

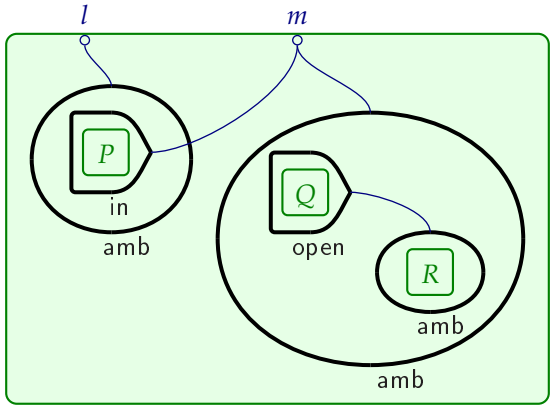
Definition 9.1 The signature Σ_{ma} is binding and has activeness; its controls are

$$\begin{aligned} \mathsf{in}, \mathsf{out}, \mathsf{open} &: 0 \to 1 \quad \text{(passive)} \\ \mathsf{amb} &: 0 \to 1 \quad \text{(active)}\,. \end{aligned}$$

It has sorts pr and sys. The sorting condition stipulates: A sys-root may only have amb-nodes and sys-sites among its children. ■

Figure 9.1 shows a typical process of mobile ambients modelled as a bigraph. The ambient construction is modelled using amb-nodes. The signature declares amb-nodes active to model the fact that ambients preserve reaction. Thus, the only function of an amb-node is to group the collection of nodes it contains, and to provide a name for this group using linking. The prefix constructions in, out and open are modelled by passive nodes in a similar manner to the prefixes in the π-calculus.

Figure 9.2 shows an ion for each of the controls of Σ_{ma}. The sorting condition allows several variants of these ions: all could equally be given sort sys at



$l[\mathsf{in}\, m.P] \mid \nu n\, m[\mathsf{open}\, n.Q \mid n[R]]$

Figure 9.1 Example of a bigraph for an ambient process

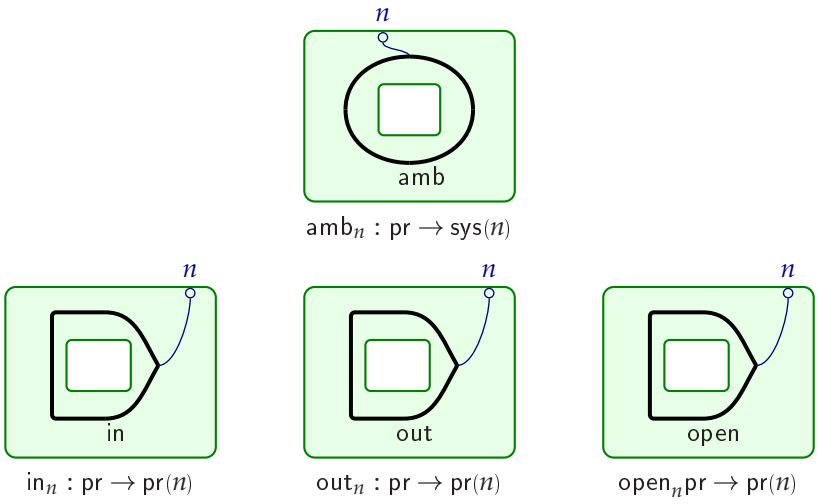


Figure 9.2 Ions in $\mathbf{BBG}_{\mathrm{ma}}$

the inner face, and the amb-ion could be given sort pr at the outer face. The ions shown are canonical in the sense that the others can be obtained from them by pre- or post-composing suitable resortings, and in the following, whenever we write an ion expression we shall always take it to denote the sorting instance shown here. The only variant we shall make use of is the ion

$$^{\mathsf{pr}}/_{\mathsf{sys}}\,\mathsf{amb}_n : \mathsf{pr} \to \mathsf{pr}(n)$$

obtained by resorting amb_n at its outer face.

The ground local primes of $\mathbf{BBG}_{\mathrm{ma}}$ fall in two classes, depending on the sort at their outer face; those with sort pr we call *process bigraphs*, and those with sort sys we call *system bigraphs*. We let p, q, ... range over process bigraphs, and a, b, ... over system bigraphs. In the case of molecules, we shall similarly talk of *process molecules* and *system molecules*, depending on their outer sort.

Lemma 9.2 In $\mathbf{BBG}(\Sigma_{\mathrm{ma}})$ every bigraph with names (X) belonging to one of the following classes can be expressed in the forms given:

System bigraphs:	$(/Z)\,a$	$a : \mathsf{sys}(XZ)$ open
Open system bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n \geq 0$, each $\mu_i : \mathsf{sys}(X)$ open
Open system molecules:	$\mathsf{amb}_n\, p$	$p : \mathsf{pr}(X)$ open
Process bigraphs:	$(/Z)\,p$	$p : \mathsf{pr}(XZ)$ open
Open process bigraphs:	$\mu_1 \mid \cdots \mid \mu_n$	$n \geq 0$, each $\mu_i : \mathsf{pr}(X)$ open
Open process molecules:	$\begin{cases} ^{\mathsf{pr}}/_{\mathsf{sys}}\,\mathsf{amb}_n\, p \\ \mathsf{in}_n\, p \\ \mathsf{out}_n\, p \\ \mathsf{open}_n\, p \end{cases}$	$p : \mathsf{pr}(X)$ open .

Moreover, these forms are unique up to the choice of fresh names Z and the ordering of the μ_i. ∎

We translate ambient calculus processes into bigraphs as follows:

Definition 9.3 The translation $\mathsf{M}_{\mathsf{pr}(X)}[\![-]\!]$ maps every process P such that $\mathrm{fn}(P) \subseteq X$ into the homset $(\epsilon, \mathsf{pr}(X))$ of $\mathbf{BBG}(\Sigma_{\mathrm{ma}})$ as follows:

$$\begin{aligned}
\mathsf{M}_{\mathsf{pr}(X)}[\![n[P]]\!] &= {}^{\mathsf{pr}}/_{\mathsf{sys}}\,(\mathsf{amb}_n\, \mathsf{M}_{\mathsf{pr}(X)}[\![P]\!]) \\
\mathsf{M}_{\mathsf{pr}(X)}[\![P \mid Q]\!] &= \mathsf{M}_{\mathsf{pr}(X)}[\![P]\!] \mid \mathsf{M}_{\mathsf{pr}(X)}[\![Q]\!] \\
\mathsf{M}_{\mathsf{pr}(X)}[\![\nu n\, P]\!] &= (/m)\, \mathsf{M}_{\mathsf{pr}(Xm)}[\![\{^m/_n\}P]\!] \\
\mathsf{M}_{\mathsf{pr}(X)}[\![\mathbf{0}]\!] &= \mathsf{pr}(X) \\
\mathsf{M}_{\mathsf{pr}(X)}[\![\mathsf{in}\, n.P]\!] &= \mathsf{in}_n\, \mathsf{M}_{\mathsf{pr}(X)}[\![P]\!] \\
\mathsf{M}_{\mathsf{pr}(X)}[\![\mathsf{out}\, n.P]\!] &= \mathsf{out}_n\, \mathsf{M}_{\mathsf{pr}(X)}[\![P]\!] \\
\mathsf{M}_{\mathsf{pr}(X)}[\![\mathsf{open}\, n.P]\!] &= \mathsf{open}_n\, \mathsf{M}_{\mathsf{pr}(X)}[\![P]\!] \,.
\end{aligned}$$

The translation $\mathsf{M}_{\mathsf{sys}(X)}[\![-]\!]$ maps every system M with $\mathrm{fn}(M) \subseteq X$ into the homset $(\epsilon, \mathsf{sys}(X))$ of $\mathbf{BBG}(\Sigma_{\mathrm{ma}})$ as follows:

$$\begin{aligned}
\mathsf{M}_{\mathsf{sys}(X)}[\![n[P]]\!] &= \mathsf{amb}_n \, \mathsf{M}_{\mathsf{pr}(X)}[\![P]]\!] \\
\mathsf{M}_{\mathsf{sys}(X)}[\![M \mid N]\!] &= \mathsf{M}_{\mathsf{sys}(X)}[\![M]\!] \mid \mathsf{M}_{\mathsf{sys}(X)}[\![N]\!] \\
\mathsf{M}_{\mathsf{sys}(X)}[\![\nu n \, M]\!] &= (/m) \, \mathsf{M}_{\mathsf{sys}(Xm)}[\![\{m/n\} M]\!] \\
\mathsf{M}_{\mathsf{sys}(X)}[\![\mathbf{0}]\!] &= \mathsf{sys}(X) \, .
\end{aligned}$$

■

As in the π-calculus we model parallel composition by prime product. We model restriction by name closure; this is sufficient in the absence of replication.

The following theorem states the correctness of our model with respect to structure, namely that every process bigraph represents a process of mobile ambients, and that the model recovers structural congruence.

Theorem 9.4 (structural correspondence)

(1) The maps $\mathsf{M}_{\mathsf{pr}(X)}[\![-]\!]$ and $\mathsf{M}_{\mathsf{sys}(X)}[\![-]\!]$ are surjective onto the homsets $(\epsilon, \mathsf{pr}(X))$ and $(\epsilon, \mathsf{sys}(X))$, respectively, of $\mathbf{BBG}(\Sigma_{\mathrm{ma}})$;

(2) $P \equiv Q$ iff $\mathsf{M}_{\mathsf{pr}(X)}[\![P]\!] = \mathsf{M}_{\mathsf{pr}(X)}[\![Q]\!]$. ■

We now turn to the dynamics of our model:

Definition 9.5 The rule set $\mathcal{R}_{\mathrm{ma}}$ consists of three linear reaction rules

$$(R_{\mathrm{enter}}, R'_{\mathrm{enter}}, Id) \qquad (R_{\mathrm{exit}}, R'_{\mathrm{exit}}, Id) \qquad (R_{\mathrm{open}}, R'_{\mathrm{open}}, Id) \, ,$$

where

$$\begin{aligned}
R_{\mathrm{enter}} &= \mathsf{amb}_n \, (\mathsf{in}_m \mid id_1) \mid \mathsf{amb}_m : \langle \mathsf{pr}, \mathsf{pr}, \mathsf{pr} \rangle \to \mathsf{sys}(mn) \\
R'_{\mathrm{enter}} &= \mathsf{amb}_m \, (\mathsf{amb}_n.(id_1 \mid id_1) \mid id_1) : \langle \mathsf{pr}, \mathsf{pr}, \mathsf{pr} \rangle \to \mathsf{sys}(mn) \\
R_{\mathrm{exit}} &= \mathsf{amb}_m \, (\mathsf{amb}_n.(\mathsf{out}_m \mid id_1) \mid id_1) : \langle \mathsf{pr}, \mathsf{pr}, \mathsf{pr} \rangle \to \mathsf{sys}(mn) \\
R'_{\mathrm{exit}} &= \mathsf{amb}_n \, (id_1 \mid id_1) \mid \mathsf{amb}_m : \langle \mathsf{pr}, \mathsf{pr}, \mathsf{pr} \rangle \to \mathsf{sys}(mn) \\
R_{\mathrm{open}} &= \mathsf{open}_n \mid \mathsf{amb}_n : \langle \mathsf{pr}, \mathsf{pr} \rangle \to \mathsf{pr}(n) \\
R'_{\mathrm{open}} &= id_1 \mid id_1 : \langle \mathsf{pr}, \mathsf{pr} \rangle \to \mathsf{pr}(n) \, .
\end{aligned}$$

■

The reaction rules are illustrated in Figure 9.3.

The following lemma characterizes the active bigraphs in $\mathbf{BBG}_{\mathrm{ma}}$. (Recall that a bigraph is active if no site is a descendant of a passive node.) The main observation is that in $\mathbf{BBG}_{\mathrm{ma}}$ only the amb-control is active, and so a site can be nested only inside amb-nodes.

Lemma 9.6 A bigraph $S : \mathsf{pr}(XZ) \to \mathsf{pr}(X)$ is active iff it can be expressed in the form

$$S = (/Z) \, ((A_1 \circ \cdots \circ A_k) \mid p)$$

for some $k \geq 0$, where each A_i is of the form

$$A_i = \mathsf{amb}_n \, (id_{\mathsf{pr}(XZ)} \mid p) \, .$$

■

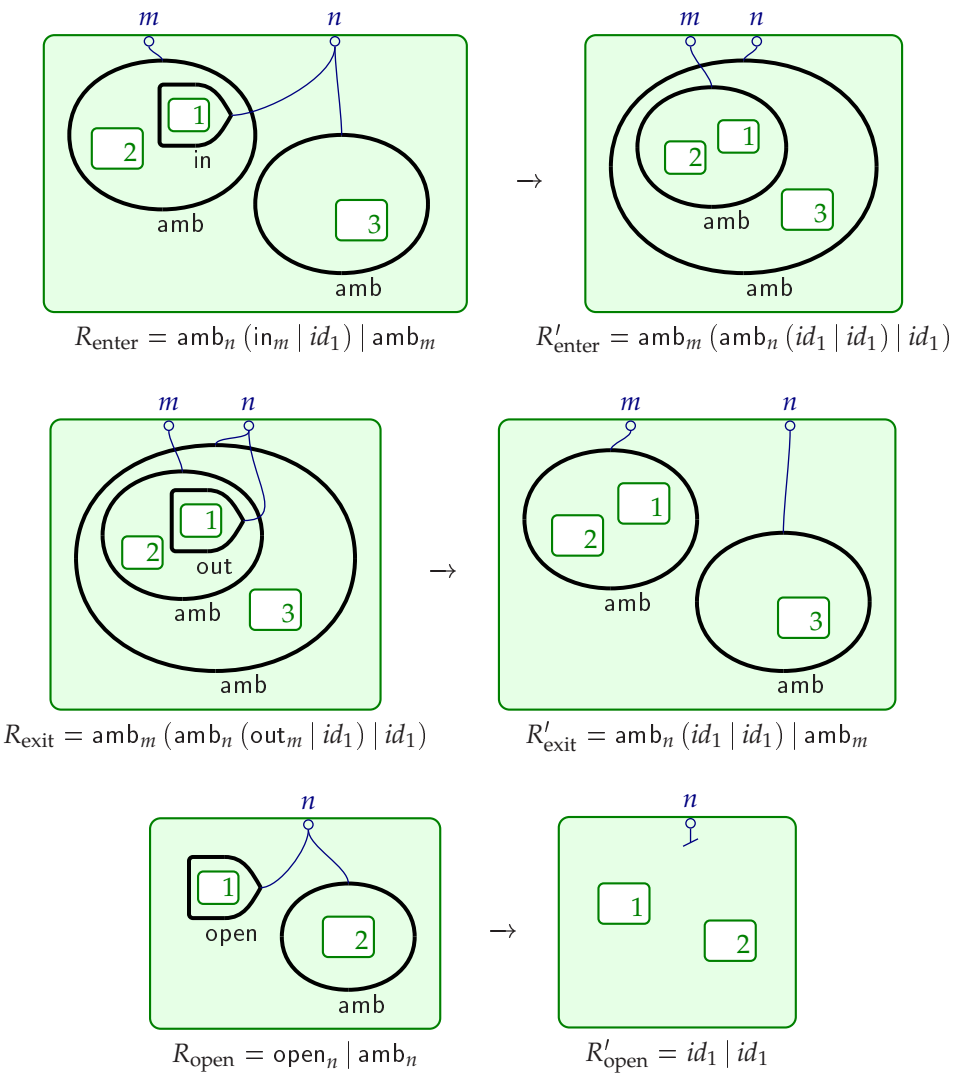


Figure 9.3 The reaction rules in $\mathcal{R}_{\text{ma}}$

Henceforth, we let S range over such bigraphs. They correspond exactly to the non-prefix contexts of the calculus, i.e. those that preserve reaction:

Lemma 9.7 For every non-prefix context C of mobile ambients and any names $X \supseteq \mathrm{fn}(C)$ there exists an active $S : \mathsf{pr}(XZ) \to \mathsf{pr}(X)$ such that for any P with $\mathrm{fn}(P) \subseteq XZ$ we have

$$\mathsf{M}_{\mathsf{pr}(X)}[\![C[P]]\!] = S \circ \mathsf{M}_{\mathsf{pr}(XZ)}[\![P]\!] \,. \qquad \blacksquare$$

Using this, it is straightforward to establish that the model recovers reaction exactly:

Theorem 9.8 (dynamic correspondence) $P \to P'$ iff $\mathsf{M}_{\mathsf{pr}(X)}[\![P]\!] \to \mathsf{M}_{\mathsf{pr}(X)}[\![P']\!]$. $\blacksquare$

9.3 Contextual transitions and bisimilarity

The abstract BRS $\mathbf{BBG}_{\mathrm{ma}}$ is equipped with the standard active transition system and the associated strong and weak active bisimilarities $\sim$ and $\approx$, as given by Definition 5.19. The following property states that bisimilarity is the same in all homsets into which a pair of processes are mapped.

Proposition 9.9 Let $X, Y \supseteq \mathrm{fn}(P, Q)$. Then

$$\mathsf{M}_{\mathsf{pr}(X)}[\![P]\!] \sim \mathsf{M}_{\mathsf{pr}(X)}[\![Q]\!] \quad \text{iff} \quad \mathsf{M}_{\mathsf{pr}(Y)}[\![P]\!] \sim \mathsf{M}_{\mathsf{pr}(Y)}[\![Q]\!] \,.$$

The analogous result holds also for $\approx$. $\blacksquare$

The congruence result for active bisimilarity (Theorem 5.21) ensures immediately that bisimilarity is also a congruence for mobile ambients:

Theorem 9.10 (process congruence) Strong and weak active bisimilarity $\sim$ and $\approx$ are process congruences; that is, for any context C of mobile ambients and any name set $Y \supseteq \mathrm{fn}(C[P], C[Q])$,

$$\text{if } \mathsf{M}_{\mathsf{pr}(X)}[\![P]\!] \sim \mathsf{M}_{\mathsf{pr}(X)}[\![Q]\!] \text{ then } \mathsf{M}_{\mathsf{pr}(Y)}[\![C[P]]\!] \sim \mathsf{M}_{\mathsf{pr}(Y)}[\![C[Q]]\!] \,,$$

and similarly for $\approx$. $\blacksquare$

We wish to obtain a transition system for system bigraphs, and in order to reduce the size of the transition system we shall apply the theory of engaged transitions. Thus, we shall be interested in the following transition system:

Definition 9.11 In $\mathbf{BBG}_{\mathrm{ma}}$ let ES denote the transition system that consists of the engaged transitions over system bigraphs. $\blacksquare$

Engaged transitions over system bigraphs are characterized by the following lemma. (In Table 9.1 and henceforth we use ambient calculus notation for system bigraphs.)

Case	a	L	a'
(a1)	$\nu Z\,(m[\mathsf{in}\,n.p \mid q] \mid b)$	$(n[s] \mid id) \lhd id_n$	$\nu Z\,(n[m[p \mid q] \mid s] \mid b)$
(a2)	$\nu Z\,(n[s] \mid b)$	$(m[\mathsf{in}\,n.p \mid q] \mid id) \lhd id_n$	$\nu Z\,(n[m[p \mid q] \mid s] \mid b)$
(a3)	$S[m[\mathsf{in}\,n.p \mid q] \mid n[s]]$	id	$S[n[m[p \mid q] \mid s]]$
(a4)	$S[m[\mathsf{in}\,n.p \mid q] \mid l[s]]$	$id \lhd {}^{n}\!/_{nl}$	$({}^{n}\!/_{nl})\,S[n[m[p \mid q] \mid s]]$
(b1)	$\nu Z\,(m[\mathsf{out}\,n.p \mid q] \mid b)$	$n[s \mid id] \lhd id_n$	$\nu Z\,(m[p \mid q] \mid n[s \mid b])$
(b2)	$S[n[m[\mathsf{out}\,n.p \mid q] \mid s]]$	id	$S[m[p \mid q] \mid n[s]]$
(b3)	$S[l[m[\mathsf{out}\,n.p \mid q] \mid s]]$	$id \lhd {}^{n}\!/_{nl}$	$({}^{n}\!/_{nl})\,S[m[p \mid q] \mid n[s]]$
(c1)	$\nu Z\,(n[s] \mid b)$	$m[\mathsf{open}\,n.p \mid q \mid id] \lhd id_n$	$\nu Z\,m[p \mid q \mid s \mid b]$
(c2)	$S[m[\mathsf{open}\,n.p \mid q \mid n[s]]]$	id	$S[m[p \mid q \mid s]]$
(c3)	$S[m[\mathsf{open}\,n.p \mid q \mid l[s]]]$	$id \lhd {}^{n}\!/_{nl}$	$({}^{n}\!/_{nl})\,S[m[p \mid q \mid s]]$

Table 9.1 Forms of an engaged transition $a \xrightarrow{L} a'$.
In all cases the label $L = L_0 \lhd \sigma$ must have L_0 robustly discrete with names not among Z.

Lemma 9.12 The engaged transitions $a \xrightarrow{L} a'$ in $\mathbf{BBG}_{\mathrm{ma}}$ are exactly those of the forms given by the cases in Table 9.1, up to a bijection on the outer names of L and p'. ∎

The redexes of $\mathbf{BBG}_{\mathrm{ma}}$ are all simple, and the following adequacy and definiteness properties follow easily. (We use $\approx_{\mathrm{ES}}$ to denote the bisimilarity in which every L-transition in ES is matched by a sequence of ES-transitions with composite label L.)

Lemma 9.13

(1) ES is adequate for strong and weak active bisimilarity; that is $\sim^{\mathrm{ES}} = \sim$ and $\approx^{\mathrm{ES}} = \approx$ on system bigraphs;

(2) ES is definite; hence, $\sim_{\mathrm{ES}} = \sim$ and $\approx_{\mathrm{ES}} = \approx$ on system bigraphs. ∎

Merro and Nardelli [21] give a transition system for mobile ambients which provides a co-inductive characterization of barbed congruence. This is a *raw* transition system, in the sense we have been using the term earlier: it is tailored to represent, as succinctly as possible, the potential of agents for interaction with the environment. The transition system is defined via a somewhat complicated set of inference rules involving several auxiliary transition relations; we do not repeat the rules here, but give an overview of the resulting transitions in Table 9.2.

Case	a	ℓ	a'	Conditions
(a1)	$\nu Z\,(m[\mathsf{in}\,n.p \mid q] \mid b)$	$m\,\mathsf{enter}\,n$	$\nu Z\,(n[m[p \mid q] \mid o] \mid b)$	$m, n \notin Z$
(a1$*$)	$\nu Zm\,(m[\mathsf{in}\,n.p \mid q] \mid b)$	$*\,\mathsf{enter}\,n$	$\nu Zm\,(n[m[p \mid q] \mid o] \mid b)$	$n \notin Zm$
(a2)	$\nu Z\,(n[s] \mid b)$	$n\,\overline{\mathsf{enter}}\,m$	$\nu Z\,(n[m[o] \mid s] \mid b)$	$m, n \notin Z$
(a3)	$S[m[\mathsf{in}\,n.p \mid q] \mid n[s]]$	τ	$S[n[m[p \mid q] \mid s]]$	
(b1)	$\nu Z\,(m[\mathsf{out}\,n.p \mid q] \mid b)$	$m\,\mathsf{exit}\,n$	$\nu Z\,(m[p \mid q] \mid n[o \mid b])$	$m, n \notin Z$
(b1$*$)	$\nu Zm\,(m[\mathsf{out}\,n.p \mid q] \mid b)$	$*\,\mathsf{exit}\,n$	$\nu Zm\,(m[p \mid q] \mid n[o \mid b])$	$n \notin Zm$
(b2)	$S[n[m[\mathsf{out}\,n.p \mid q] \mid s]]$	τ	$S[m[p \mid q] \mid n[s]]$	
(c1)	$\nu Z\,(n[s] \mid b)$	$m\,\mathsf{open}\,n$	$\nu Z\,m[o \mid s \mid b]$	$m, n \notin Z$
(c2)	$S[m[\mathsf{open}\,n.p \mid q \mid n[s]]]$	τ	$S[m[p \mid q \mid s]]$	

Table 9.2 Forms of a raw transition $a \xrightarrow{\ell} a'$.

As can be seen from the table, four different labels (apart from τ) are employed; they have the following intuitive meanings:

$m\,\mathsf{enter}\,n$: m enters n;

$n\,\overline{\mathsf{enter}}\,m$: n is entered by m;

$m\,\mathsf{exit}\,n$: m exits n;

$m\,\mathsf{open}\,n$: n is opened by m.

In the first and third case there are also special forms with m replaced by $*$, which indicates that the name m of the ambient performing the action is restricted.

The simplicity of the raw labels is helped by the use of a special convention: in each case the result a' of a raw transition contains the special process o that plays the role of place-holder for an arbitrary process. The place-holder is used in the definition of bisimulation, which states that, for related a and b, every transition $a \xrightarrow{\ell} a'$ must be matched by a transition $b \xrightarrow{\ell} b'$ such that $\{p/o\}a'$ and $\{p/o\}b'$ are related for every process p. Thus, arbitrary parameters (as they appear in our contextual labels) are avoided in the raw labels, but only at the cost of introducing them in the definition of bisimilarity.

With the special meaning of o in mind, we find a close correspondence between the contextual and raw transition systems by comparing Tables 9.1 and 9.2:

- enter-transitions in cases (a1) and (a1$*$) of Table 9.2: these correspond to the contextual transition in case (a1) of Table 9.1;

- $\overline{\text{enter}}$-transitions in case (a2) of Table 9.2: these correspond to the contextual transition in case (a2) of Table 9.1;
- exit-transitions in cases (b1) and (b1$*$) of Table 9.2: these correspond to the contextual transition in case (b1) of Table 9.1;
- open-transitions in case (c1) of Table 9.2: these correspond to the contextual transition in case (c1) of Table 9.1;
- τ-transitions in case (a3, b2, c2) of Table 9.2: these correspond to the identity-transitions (cases (a3, b2, c2) of Table 9.1);
- The contextual substitution transitions in cases (a4, b3, c3) of Table 9.1 have no corresponding raw transitions.

Thus, if we disregard substitution transitions (as we considered also for the π-calculus in the previous chapter) we arrive at a transition system essentially identical to that of Merro and Nardelli. However, the bisimilarity for this transition system is strictly stronger than the bisimilarity that Merro and Nardelli define, and hence also stronger than barbed congruence. The reason for the discrepancy is that transitions with the special $*$-forms of enter- and exit-labels provide "too much" observational power; that is, unless they are treated specially, the resulting bisimilarity is strictly stronger than barbed congruence. The situation is similar to that for the asynchronous π-calculus, where (as previously discussed) a weakened form of input matching is employed in order to recover barbed congruence. Merro and Nardelli, consequently, use a similarly weakened form of matching for their $*$-labels. Without this, their bisimilarity would not, for example, validate the equivalence

$$\nu n\, n[p] \approx \mathbf{0} \quad (n \notin \mathrm{fn}(p))$$

known as the "perfect firewall" equation. Indeed, our contextual bisimilarity does not validate this equation, as the left-hand side generally has non-identity contextual transitions; for example, if p is of the form $\mathsf{in}\, m$ then there is a transition of the form in case (a1) of Table 9.1.

Thus, as was the case for our model of the π-calculus, the bisimilarity induced by the bigraph model is strictly stronger than barbed congruence. Again, it is unclear without substantial studies of applications of the calculus to what extent the induced notion of bisimilarity might be useful in practice. In any case it is encouraging—from a bigraph perspective—that the labelled transition system derived from the bigraph model so closely resembles the system arrived at independently by other researchers.

Chapter 10

Conclusions

In this short closing chapter we summarize the results and conclusions given in the main chapters of the dissertation, and we discuss some directions for future work.

The material in Part I of the dissertation is partly a review of the existing theory of reactive systems and of pure and binding bigraphs, and partly three main developments of these theories: First, a theory of weak bisimilarity for reactive systems, based on composition of reaction rules. Secondly, a notion of place-sorting for bigraphs, for which we obtain results that parallel the main results for unsorted bigraphs, namely (1) the existence of IPOs and the consequent congruence of bisimilarity, and (2) the adequacy of engaged transitions. Thirdly, an adequacy theorem for engaged transitions in simple BRSs, allowing, in particular, for BRSs with replication.

Thus, we have succeeded in extending the applicability of reactive systems and bigraphs. The extensions are necessary for the applications we have given in Part II, but at the same time they are general enough that we can expect them to be useful also in future applications. A side-effect of achieving the extensions is to further provide confidence in the theory that existed previously: if natural extensions had turned out to be problematic, one would reasonably suspect that the existing theory had built-in inadequacies.

Part II provides a challenge to the theory from Part I, namely to test that it can handle the substantial applications provided by existing calculi for mobile processes. It passes the test, at least in the sense that the structure (structural congruence) and behaviour (reaction) can be represented faithfully and naturally in bigraphs. As for the transition systems and behavioural equivalences that the encodings induce, the picture is less clear: We find a substantial, but not complete, agreement between engaged transitions and the "raw" transition systems for the calculi. In all cases, the small discrepancies in the transition systems lead to behavioural equivalences that differ slightly from any of the established equivalences in the calculi. As we have argued, it is not clear that the differences are important for applications of the calculi; to settle this question would require detailed study of a range of such applications, something

that represents one line of future work.

As explained in the introduction, the work in this dissertation has concentrated on establishing ties with process calculus, and in particular via the notions of labelled transitions and bisimilarity. Clearly, this is just one area of many that must be addressed before bigraphs can be convincingly proposed as a suitable framework for modelling mobile computation. Several natural extensions to the framework should be explored, such as features for modelling continuous phenomena (e.g. time) or for specifying probabilistic behaviour. Moreover, semantic notions other than bisimilarity should be developed, including other transition-based equivalences and preorders, logics, and notions based on reaction, such as reachability or confluence.

For bigraphs to be accepted as a practical tool, perhaps the most important future work is simply to extend the range of applications substantially, and to refine the framework according to the feedback this will provide. It is not realistic, however, to expect a large number of substantial applications without some form of practical support in the form of programming languages and development tools. Fortunately, work is underway on a bigraphical programming language at Copenhagen's IT University.

Whether or not bigraphs will, in the longer run, turn out to be among the best tools for constructing and understanding mobile systems remains to be seen. In any case, such tools are increasingly necessary, and the struggle to provide them must continue. One can at least say that bigraphs have been established as a strong candidate in this endeavour; it is to be hoped, therefore, that they will attract much further research, both at a theoretical and a practical level.

Appendix A

Proofs for Section 8.4

Lemma 8.13 In either of $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_{\pi}$, let $p, q : (X)$, and suppose $\nu w\,(\overline{u}w \mid p) \dot{\sim} \nu w\,(\overline{u}w \mid q)$ for some u and w with $u \notin X$. Then $p \dot{\sim} q$.

Proof Let $\mathcal{S}$ consist of all pairs $(p, q : (X))$ satisfying the condition. We show that $\mathcal{S}$ is a mono bisimulation up to $\dot{\sim}$. Suppose $(p, q) \in \mathcal{S}$ and $p \xrightarrow{L} p'$. We must consider all possible forms of the transition according to Table 8.2 cases (a1–b7). The argument is analogous in each case; we give it just for case (b1). In this case we have the forms

$$\begin{aligned} p &= \nu Z\,((\overline{x}y.p_0 + m) \mid g) \\ L &= (x(z).s \triangleleft id) \triangleleft id_x \\ p' &= \nu Z\,(p_0 \mid (y/z)\,s \mid g)\,. \end{aligned}$$

The non-trivial case is when $x = w$. Let $M = (u(x).x(z).(\overline{v}x \mid s) \triangleleft id) \triangleleft id_u$. Then

$$\begin{aligned} \nu w\,(\overline{u}w \mid p) &\xrightarrow{M} \nu w\,(\mathbf{0} \mid w(z).(\overline{v}w \mid s) \mid p) \\ &\xrightarrow{id} \nu w\,(\mathbf{0} \mid \overline{v}w \mid p') \\ &\dot{\sim}\ \nu w\,(\overline{v}w \mid p')\,. \end{aligned}$$

By the assumption $(p, q) \in \mathcal{S}$ there is also a transition sequence

$$\nu w\,(\overline{u}w \mid q) \xrightarrow{M}\xrightarrow{id} q_1'$$

for some q_1' such that $\nu w\,(\overline{v}w \mid p') \dot{\sim} q_1'$. Because of the form of M, this requires $q \xrightarrow{L} q'$ for some q' such that $q_1' = \nu w\,(\mathbf{0} \mid \overline{v}w \mid q') \dot{\sim} \nu w\,(\overline{v}w \mid q')$. Then, using the easily verified property that $t \mid \mathbf{0} \dot{\sim} t$, we have $(p', q') \in \mathcal{S}^{\dot{\sim}}$, and we are done. ∎

Theorem 8.14 (mono bisimilarity) In $\mathbf{BBG}_{\mathrm{sf}\pi}$, $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_{\pi}$ mono bisimilarity lies between early and ground bisimilarity:

$$\overset{\mathrm{e}}{\sim}_{\pi} \subseteq \dot{\sim} \subseteq \overset{\mathrm{g}}{\sim}_{\pi}\,.$$

Moreover, in $\mathbf{BBG}_{\mathrm{f}\pi}$ and $\mathbf{BBG}_{\pi}$ both inclusions are strict.

Proof For the first inclusion we show that $\overset{\mathrm{e}}{\sim}_{\pi}$ is a mono bisimulation. Suppose $p \overset{\mathrm{e}}{\sim}_{\pi} q$, and let $p \xrightarrow{L} p'$ be a mono transition; we must find a transition $q \xrightarrow{L} q'$ such that $p' \overset{\mathrm{e}}{\sim}_{\pi} q'$. Table 8.2 cases (a1–3, b1–3, b5–6) list the possible forms for the transition of p. Let us consider case (b1), where we have the forms

$$\begin{aligned} p &= \boldsymbol{\nu} Z\,((\overline{x}y.p_0 + m) \mid g) \\ L &= (x(z).s \triangleleft id) \triangleleft id_x \\ p' &= \boldsymbol{\nu} Z\,(p_0 \mid (y/z)\,s \mid g)\,. \end{aligned}$$

There are two subcases, according to whether y is among Z or not. We assume $y \in Z$; the proof in the other subcase is similar. For p in the form stated we infer from Table 8.3 an early transition

$$p \xrightarrow{\overline{x}(y)} p_0' \overset{\text{def}}{=} \boldsymbol{\nu} Z'\,(p_0 \mid g)\,,$$

where $Z' = Z \setminus \{y\}$. Then, since $p \overset{\mathrm{e}}{\sim}_{\pi} q$, there is an early transition $q \xrightarrow{\overline{x}(y)} q_0'$ for some q_0' such that $p_0' \overset{\mathrm{g}}{\sim}_{\pi} q_0'$. Using Table 8.3 again, we find that q is of the form $\boldsymbol{\nu} W y\,(\overline{x}y.q_0 \mid h)$, where we can assume the names W to be fresh, and $q_0' = \boldsymbol{\nu} W\,(q_0 \mid h)$. From Table 8.2 we then infer a mono transition

$$q \xrightarrow{L} q' \overset{\text{def}}{=} \boldsymbol{\nu} W y\,(q_0 \mid (y/z)\,s \mid h)\,.$$

Since s is part of L and therefore has no names among Z' or W, we can write p' and q' in the forms

$$\begin{aligned} p' &= \boldsymbol{\nu} y\,((y/z)\,s \mid p_0') \\ q' &= \boldsymbol{\nu} y\,((y/z)\,s \mid q_0')\,. \end{aligned}$$

Hence, using the fact that $\overset{\mathrm{e}}{\sim}_{\pi}$ is a non-input congruence, we obtain $p' \overset{\mathrm{e}}{\sim}_{\pi} q'$ as required. (It may be noted that this last step would not go through if one attempted a similar proof with $\overset{\mathrm{g}}{\sim}_{\pi}$ in place of $\overset{\mathrm{e}}{\sim}_{\pi}$.)

The arguments for all other applicable cases are similar.

For the second inclusion we show that $\dot{\sim}$ is a ground bisimulation. Suppose $p \dot{\sim} q$ and let $p \xrightarrow{\ell} p'$ be a late transition; we must find a late transition $q \xrightarrow{\ell} q'$ such that $p' \dot{\sim} q'$. Table 8.3 cases (a1–b8) minus cases (a1, b3, b6) list the possible forms for the transition of p. Let us consider first case (b2) (bound output). In this case the label ℓ is $\overline{x}(z)$, and p and p' have the forms

$$\begin{aligned} p &= \boldsymbol{\nu} Z z\,((\overline{x}z.p_0 + m) \mid g) \\ p' &= \boldsymbol{\nu} Z\,(p_0 \mid g)\,. \end{aligned}$$

We define two contextual labels:

$$\begin{aligned} L_1 &= (x(z).\overline{z}v.\overline{u}z \triangleleft id) \triangleleft id_x \\ L_2 &= (x(z).\overline{u}z \triangleleft id) \triangleleft id_x\,, \end{aligned}$$

where u and v are fresh. Using Table 8.2 we infer two trim transitions

$$p \xrightarrow{L_1} p_1' \stackrel{\text{def}}{=} \nu Z z\,(p_0 \mid \overline{z}v.\overline{u}z \mid g)$$
$$p \xrightarrow{L_2} p_2' \stackrel{\text{def}}{=} \nu Z z\,(p_0 \mid \overline{u}z \mid g)\,.$$

Then, since $p \nsim q$, there must be trim transitions $q \xrightarrow{L_1} q_1'$ and $q \xrightarrow{L_2} q_2'$ for some q_1' and q_2' such that $p_1' \nsim q_1'$ and $p_2' \nsim q_2'$. Using Table 8.2 again, we find that q, q_1' and q_2' have the forms

$$q = \nu W\,((\overline{x}y.q_0 + n) \mid h)$$
$$q_1' = \nu W\,(q_0 \mid \overline{z}v.\overline{u}z \mid h)$$
$$q_2' = \nu W\,(q_0 \mid \overline{u}z \mid h)\,,$$

where we can assume the names W to be fresh. We claim that $z \in W$; for otherwise $q_1' \xrightarrow{L_z} \xrightarrow{L_u}$, where $L_z = (z(w) \triangleleft id) \triangleleft id_z$ and $L_u = (u(z) \triangleleft id) \triangleleft id_u$, and p_1' cannot match this transition sequence, a contradiction with $p_1' \nsim q_1'$. From Table 8.3 we then infer a late transition

$$q \xrightarrow{\overline{x}(z)} q' \stackrel{\text{def}}{=} \nu W'(q_0 \mid h)\,,$$

where $W' = W \setminus \{z\}$. We can write p_2' and q_2' in the forms

$$p_2' = \nu z\,(\overline{u}z \mid p')$$
$$q_2' = \nu z\,(\overline{u}z \mid q')\,.$$

As $p_2' \nsim q_2'$, the required relationship $p' \nsim q'$ follows from Lemma 8.13.

For case (b1)—a free output transition $p \xrightarrow{\overline{x}y} p'$—the argument is similar. We use L_1 again, this time to establish that y is open in q; the main argument then goes through using the simpler $L_3 = (x(z) \triangleleft id) \triangleleft id_x$ in place of L_2, and using only $t \mid \mathbf{0} \nsim t$ in place of Lemma 8.13.

Case (b4)—a bound input transition $p \xrightarrow{x(z)} p'$—is simpler, corresponding essentially to a contextual transition with label $(\overline{x}z \triangleleft id) \triangleleft id_x$; again we need only to use $t \mid \mathbf{0} \nsim t$. Case (b5)—a silent transition $p \xrightarrow{\tau} p'$—is simpler still, as τ- and id-transitions coincide exactly.

Cases (b6–7), which apply only in $\mathbf{BBG}_\pi$, are analogous to cases (b4–5). Cases (a1–2, a4–5), which apply only in $\mathbf{BBG}_{\mathrm{sf}\pi}$, are analogous to cases (b1–2, b4–5), but simplified by the fact that the equality $t \mid \mathbf{0} = t$ holds in $\mathbf{BBG}_{\mathrm{sf}\pi}$. This completes the proof of the second inclusion.

For strictness of the inclusions, consider the process bigraphs

$$p = x(y).(\overline{x} \mid y)$$
$$q = x(y).(\overline{x}.y + y.\overline{x}).$$

For these we claim

$$p \quad \overset{\mathrm{e}}{\nsim}_\pi \quad q \qquad\qquad \overline{u}x \mid p \quad \overset{\dot{\nsim}}{\underset{\mathrm{g}}{\nsim}}_\pi \quad \overline{u}x \mid q\,.$$

It is easy to verify that $p \mathrel{\dot{\sim}} q$; the crux is that the label L of a transition of p or q will have the form $(\overline{x}v \triangleleft id) \triangleleft id_x$ with v fresh, and hence the resulting substitution cannot coalesce x and y. But $p \overset{\mathrm{e}}{\nsim}_\pi q$, because there are free input transitions which do coalesce x and y. The addition of the parallel component $\overline{u}x$ makes no difference for $\overset{\mathrm{g}}{\nsim}_\pi$, as the only new late transitions (labelled $\overline{u}x$) lead to p and q. But taking $L = (u(z).\overline{z}z \triangleleft id) \triangleleft id_u$ we have

$$\begin{aligned} p &\xrightarrow{L} \overline{x}x \mid p \xrightarrow{id} \overline{x} \mid x \\ q &\xrightarrow{L} \overline{x}x \mid q \xrightarrow{id} \overline{x}.x + x.\overline{x}\,, \end{aligned}$$

whence $p \mathrel{\dot{\nsim}} q$. ∎

Theorem 8.18 (asynchronous contextual bisimilarity) In $\mathbf{BBG}_{\mathrm{a}\pi}$ contextual bisimilarity and restriction-asynchronous bisimilarity coincide:

$$\sim \;=\; \overset{\mathrm{ra}}{\sim}_\pi\,.$$

Proof For each pair of disjoint finite name sets X and Z, we fix a family $\{u_z\}_{z\in Z}$ of names disjoint from X and Z, and we define the process

$$ext_{X,Z} \overset{\text{def}}{=} \prod_{z\in Z} \overline{u_z}z\,.$$

For the inclusion $\sim \;\subseteq\; \overset{\mathrm{ra}}{\sim}_\pi$ we show that $\{\mathcal{S}_Z\}$ is a restriction-asynchronous bisimulation, where

$$\mathcal{S}_Z \overset{\text{def}}{=} \{(p,q : (X)) \mid \boldsymbol{\nu} Z\,(ext_{X,Z} \mid p) \sim \boldsymbol{\nu} Z\,(ext_{X,Z} \mid q)\}\,.$$

Suppose $(p,q) \in \mathcal{S}_Z$, and let $p \xrightarrow{\ell} p'$; we must find a matching transition $q \xrightarrow{\ell} q'$ according to Definition 8.16. Table 8.3 cases (b4, b7, c1–4) list the possible forms for the transition of p. We treat cases (b4) and (c2); the others are easier.

Case (b4) (input). The label ℓ is $x(z)$, and p and p' have the forms

$$\begin{aligned} p &= \boldsymbol{\nu} W\,((x(z).p_0 + p_1) \mid g) \\ p' &= \boldsymbol{\nu} W\,(p_0 \mid g)\,. \end{aligned}$$

There are two subcases according to whether x is among Z or not. We treat the case $x \in Z$; the other is straightforward.

Define a contextual label

$$L = ((u_x(x).(\overline{u_x}x \mid \overline{x}z)) \triangleleft id) \triangleleft id_u\,;$$

then by Table 8.2 we infer trim transitions

$$\begin{aligned}\boldsymbol{\nu} Z\,(ext_{X,Z} \mid p) &\xrightarrow{L} p_1' \stackrel{\text{def}}{=} \boldsymbol{\nu} Z\,(ext_{X,Z} \mid \overline{x}z \mid p)\\ &\xrightarrow{id} p_2' \stackrel{\text{def}}{=} \boldsymbol{\nu} Z\,(ext_{X,Z} \mid p')\,.\end{aligned}$$

Then, since $\boldsymbol{\nu} Z\,(ext_{X,Z} \mid p) \sim \boldsymbol{\nu} Z\,(ext_{X,Z} \mid q)$, there must be trim transitions $\boldsymbol{\nu} Z\,(ext_{X,Z} \mid q) \xrightarrow{L} q_1' \xrightarrow{id} q_2'$ such that $p_1' \sim q_1'$ and $p_2' \sim q_2'$. Using Table 8.2 again, we find from the first transition that q_1' has the form $\boldsymbol{\nu} Z\,(ext_{X,Z} \mid \overline{x}z \mid q)$; from the second transition we find that one of the following is the case:

(i) q and q_2' have the forms

$$\begin{aligned}q &= \boldsymbol{\nu} V\,((x(z).q_0 + q_1) \mid h)\\ q_2' &= \boldsymbol{\nu} Z\,(ext_{X,Z} \mid q')\,,\end{aligned}$$

where $q' = \boldsymbol{\nu} V\,(q_0 \mid h)$.

(ii) $q \xrightarrow{id} q'$ for some q' such that $q_2' = \boldsymbol{\nu} Z\,(ext_{X,Z} \mid \overline{x}z \mid q')$.

In case (i) we find by Table 8.3 a raw transition $q \xrightarrow{x(z)} q'$, and from the bisimilarity $p_2' \sim q_2'$ it follows that $(p', q') \in \mathcal{S}_Z$; thus we have a matching transition according to case (2a) of Definition 8.16.

In case (ii) we infer readily a raw transition $q \xrightarrow{\tau} q'$, and from the forms of p_2' and q_2' it follows that $(p_1', \overline{x}v \mid q') \in \mathcal{S}_Z$; thus we have a matching transition according to case (2b) of Definition 8.16. This concludes the proof for case (b4).

Case (c2) (bound output). The label ℓ is $\overline{x}(z)$, and p and p' have the forms

$$\begin{aligned}p &= \boldsymbol{\nu} z\,(\overline{x}z \mid g)\\ p' &= \mathbf{0} \mid g\,.\end{aligned}$$

Again there are two subcases according to whether x is among Z or not, and we treat only the most interesting case, namely $x \in Z$.

Define a contextual label

$$L = ((u_x(x).x(z).(\overline{u_x}x \mid \overline{u_z}z)) \triangleleft id) \triangleleft id_u\,;$$

then by Table 8.2 we infer trim transitions

$$\begin{aligned}\boldsymbol{\nu} Z\,(ext_{X,Z} \mid p) &\xrightarrow{L} p_1' \stackrel{\text{def}}{=} \boldsymbol{\nu} Z\,(ext_{X,Z\setminus\{x\}} \mid x(z).(\overline{u_x}x \mid \overline{u_z}z) \mid p)\\ &\xrightarrow{id} p_2' \stackrel{\text{def}}{=} \boldsymbol{\nu} Zz\,(ext_{X,Zz} \mid p')\,.\end{aligned}$$

Then, since $\boldsymbol{\nu} Z\,(ext_{X,Z} \mid p) \sim \boldsymbol{\nu} Z\,(ext_{X,Z} \mid q)$, there must be trim transitions $\boldsymbol{\nu} Z\,(ext_{X,Z} \mid q) \xrightarrow{L} q_1' \xrightarrow{id} q_2'$ such that $p_1' \sim q_1'$ and $p_2' \sim q_2'$. Using Table 8.2 again, we find from the first transition that q_1' has the form

$$q_1' = \boldsymbol{\nu} Z\,(ext_{X,Z\setminus\{x\}} \mid x(z).(\overline{u_x}x \mid \overline{u_z}z) \mid q)\,,$$

and from the second transition that q and q_2' have the forms

$$q = \boldsymbol{\nu} z\,(\overline{x}z \mid h)$$
$$q_2' = \boldsymbol{\nu} Zz\,(ext_{X,Zz} \mid q')\,,$$

where $q' = \mathbf{0} \mid h$. From these forms we inter by Table 8.3 a raw transition $q \xrightarrow{\overline{x}(z)} q'$, and from the bisimilarity $p_2' \sim q_2'$ it follows that $(p', q') \in \mathcal{S}_{Zz}$; thus we have a matching transition according to case (3) of Definition 8.16. This concludes the proof of case (c2), and we have established the inclusion $\sim \subseteq \overset{\text{ra}}{\sim}_\pi$.

For the opposite inclusion we show that

$$\mathcal{S} \stackrel{\text{def}}{=} \{(\boldsymbol{\nu} Z\,(p \mid g)\,,\, \boldsymbol{\nu} Z\,(q \mid g)) \mid p \overset{\text{ra}}{\underset{Z}{\sim}}_\pi q\}\,.$$

is a trim bisimulation up to context. Suppose $p \overset{\text{ra}}{\underset{Z}{\sim}}_\pi q$ and let $\boldsymbol{\nu} Z\,(p \mid g) \xrightarrow{L} p'$ be a trim transition; we must find a trim transition $\boldsymbol{\nu} Z\,(q \mid g) \xrightarrow{L} q'$ for some q' such that $(p', q') \in \mathcal{S}^{\mathrm{C}}$. Table 8.2 lists the possible forms for the transition of $\boldsymbol{\nu} Z\,(p \mid g)$. The proof proceeds by a large number of cases, each arising from a particular case from the table combined with a particular distribution among p and g of the subterms that are part of the redex of the transition. All cases are similar; we treat just two among the most interesting.

Case p, g, L and p' have the forms

$$p = \boldsymbol{\nu} y\,(\overline{x}y \mid p_0)$$
$$g = \boldsymbol{\nu} W\,(x(z).g_0 \mid g_1)$$
$$L = id$$
$$p' = \boldsymbol{\nu} Zy\,(p_0 \mid \boldsymbol{\nu} W\,((y/yz)\,g_0 \mid g_1))\,.$$

In this case we infer from Table 8.3 a raw transition $p \xrightarrow{\overline{x}(y)} p_0$. Then, since $p \overset{\text{ra}}{\underset{Z}{\sim}}_\pi q$, there must be a transition $q \xrightarrow{\overline{x}(y)} q_0$ for some q_0 such that $p_0 \overset{\text{ra}}{\underset{Zy}{\sim}}_\pi q_0$ (case (3) of Definition 8.16). Using Table 8.3 again we find that q must have the form $\boldsymbol{\nu} y\,(\overline{x}y \mid q_0)$. Then, using Table 8.2, we infer a trim transition

$$\boldsymbol{\nu} Z\,(q \mid g) \xrightarrow{id} q' \stackrel{\text{def}}{=} \boldsymbol{\nu} Zy\,(q_0 \mid \boldsymbol{\nu} W\,((y/yz)\,g_0 \mid g_1))\,.$$

From the forms of p' and q' and the bisimilarity $p_0 \overset{\text{ra}}{\underset{Zy}{\sim}}_\pi q_0$ it follows that $(p', q') \in \mathcal{S}$.

Case p, g, L and p' have the forms

$$p = \boldsymbol{\nu} W\,(x(z).p_0 \mid p_1)$$
$$g = \boldsymbol{\nu} y\,(\overline{x}y \mid g_0)$$
$$L = id$$
$$p' = \boldsymbol{\nu} Zy\,(\boldsymbol{\nu} W\,((y/yz)\,p_0 \mid p_1) \mid g_0)\,.$$

In this case we infer from Table 8.3 a raw transition

$$p \xrightarrow{x(z)} p'_0 \stackrel{\text{def}}{=} \nu W\,(p_0 \mid p_1)\,.$$

Then, since $p \overset{\text{ra}}{\underset{Z}{\approx}}_\pi q$, one of the following must hold:

(i) $q \xrightarrow{x(z)} q'_0$ for some q'_0 such that $p'_0 \overset{\text{ra}}{\underset{Z}{\approx}}_\pi q'_0$ (case (2a) of Definition 8.16);

(ii) $q \xrightarrow{\tau} q'_0$ for some q'_0 such that $p'_0 \overset{\text{ra}}{\underset{Z}{\approx}}_\pi \overline{x}z \mid q'_0$ (case (2b) of Definition 8.16).

In case (i) we find from Table 8.3 that q and q'_0 must have the forms

$$\begin{aligned} q &= \nu V\,(x(z).q_0 \mid q_1) \\ q'_0 &= \nu V\,(q_0 \mid q_1)\,, \end{aligned}$$

and using Table 8.2 we then infer a trim transition

$$\nu Z\,(q \mid g) \xrightarrow{id} q' \stackrel{\text{def}}{=} \nu Z y\,(\nu V\,((y/yz)\,q_0 \mid q_1) \mid g_0)\,.$$

Without loss of generality we can assume the names z, W and Z to be all chosen fresh, and therefore p' and q' can be written in the forms

$$\begin{aligned} p' &= \nu y\,(y/yz)\,\nu Z\,(p'_0 \mid g_0) \\ q' &= \nu y\,(y/yz)\,\nu Z\,(q'_0 \mid g_0)\,, \end{aligned}$$

and it follows from the bisimilarity $p'_0 \overset{\text{ra}}{\underset{Z}{\approx}}_\pi q'_0$ that $(p', q') \in \mathcal{S}^{\mathrm{C}}$.

In case (ii) we readily infer a trim transition

$$\begin{aligned} \nu Z\,(q \mid g) \xrightarrow{id} q' &\stackrel{\text{def}}{=} \nu Z\,(q'_0 \mid g) \\ &= \nu y\,\nu Z\,(q'_0 \mid \overline{x}y \mid g_0) \\ &= \nu y\,(y/yz)\,\nu Z\,(q'_0 \mid \overline{x}z \mid g_0)\,, \end{aligned}$$

and it follows from the bisimilarity $p'_0 \overset{\text{ra}}{\underset{Z}{\approx}}_\pi \overline{x}z \mid q'_0$ that $(p', q') \in \mathcal{S}^{\mathrm{C}}$.

This concludes the proof of the theorem. ∎

Bibliography

[1] Andrea Asperti and Nadia Busi. Mobile Petri nets. Technical Report UBLCS-96-10, University of Bologna, 1996.

[2] Gérard Berry and Gérard Boudol. The chemical abstract machine. *Theoretical Computer Science*, 98:217–248, 1992.

[3] Maria Buscemi and Vladimiro Sassone. High-level Petri nets as type theories in the join calculus. In *Proceedings of FoSSaCS 2001*. Springer LNCS 2030, 2001.

[4] Luca Cardelli and Andrew D. Gordon. Mobile ambients. In *Proceedings of FoSSaCS 1998*, pages 140–155. Springer LNCS 1378, 1998.

[5] Rocco de Nicola, Gian Luigi Ferrari, and Rosario Pugliese. KLAIM: a kernel language for agents interaction and mobility. *IEEE Transactions on Software Engineering*, 24(5), 1998.

[6] Rocco de Nicola and Matthew Hennessy. Testing equivalences for processes. *Theoretical Computer Science*, 34:83–134, 1984.

[7] Cédric Fournet and Georges Gonthier. The reflexive CHAM and the join-calculus. In *Proceedings of POPL 1996*, pages 372–385. ACM Press, 1996.

[8] Fabio Gadducci and Ugo Montanari. The tile model. In Gordon Plotkin, Colin Stirling, and Mads Tofte, editors, *Proof, Language and Interaction: Essays in Honour of Robin Milner*. MIT Press, 1998.

[9] Jan Friso Groote and Frits Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, 100(2):202–260, 1992.

[10] Matthew Hennessy and James Riely. Resource access control in systems of mobile agents. In *High-Level Concurrent Languages*, volume 16, pages 3–17. Elsevier Science Publishers, 1998.

[11] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[12] Ole Høgh Jensen and Robin Milner. Bigraphs and transitions. In *Proceedings of POPL 2003*, pages 38–49. ACM Press, 2003.

[13] Ole Høgh Jensen and Robin Milner. Bigraphs and mobile processes. Technical Report UCAM-CL-TR-580, Cambridge University Computer Lab, 2004.

[14] Kim G. Larsen. *Context Dependent Bisimulation between Processes*. PhD thesis, University of Edinburgh, 1986.

[15] Kim G. Larsen and Liu Xinxin. Compositionality through an operational semantics of contexts. *Journal of Logic and Computation*, 1(6):761–795, 1991.

[16] James Leifer. *Operational Congruences for Reactive Systems*. PhD thesis, University of Cambridge, 2001.

[17] James Leifer and Robin Milner. Transition systems, link graphs and Petri nets. Technical Report UCAM-CL-TR-598, Cambridge University Computer Lab, 2004.

[18] James J. Leifer and Robin Milner. Deriving bisimulation congruences for reactive systems. In *Proceedings of CONCUR 2000*, pages 243–258. Springer LNCS 1877, 2000.

[19] Narciso Martí-Oliet and José Meseguer. Rewriting logic: Roadmap and bibliography. *Theoretical Computer Science*, 2001.

[20] Massimo Merro and Matthew Hennessy. Bisimulation congruences in safe ambients. In *Proceedings of POPL 2002*, pages 71–80. ACM Press, 2002.

[21] Massimo Merro and Francesco Zappa Nardelli. Bisimulation proof methods for mobile ambients. In *Proceedings of ICALP 2003*. Springer LNCS 2719, 2003.

[22] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.

[23] Robin Milner. The polyadic π-calculus: a tutorial. In F. L. Bauer, W. Brauer, and H. Schwichtenberg, editors, *Logic and Algebra of Specification*. Springer-Verlag, 1993.

[24] Robin Milner. *Communicating and Mobile Systems: The π-Calculus*. Cambridge University Press, 1999.

[25] Robin Milner. Bigraphical reactive systems. In *Proceedings of CONCUR 2001*, pages 16–35. Springer LNCS 2154, 2001.

[26] Robin Milner. Pure bigraphs: Structure and dynamics. *Information and Computation*, 204(1):60–122, 2006.

[27] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes, parts I and II. *Information and Computation*, 100:1–77, 1992.

[28] Robin Milner and Davide Sangiorgi. Barbed bisimulation. In *Proceedings of ICALP 1992*. Springer LNCS 623, 1992.

[29] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structures and domains, part i. *Theoretical Computer Science*, 13(1):85–108, 1981.

[30] David Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science*. Springer LNCS 104, 1981.

[31] Carl Adam Petri. *Kommunikation mit Automaten*. PhD thesis, University of Bonn, 1962.

[32] Benjamin C. Pierce and David N. Turner. Concurrent objects in a process calculus. In *Proceedings of Theory and Practice of Parallel Programming 1995*, pages 187–215. Springer LNCS 907, 1995.

[33] Aviv Regev, William Silverman, and Ehud Shapiro. Representation and simulation of biochemical processes using the π-calculus process algebra. In *Proceedings of Pacific Symposium on Biocomputing 2001*, pages 459–470. World Scientific Press, 2001.

[34] Grzegorz Rozenberg, editor. *Handbook of Graph Grammars and Computing by Graph Transformations*. World Scientific, 1997.

[35] Davide Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher Order Paradigms*. PhD thesis, University of Edinburgh, 1992.

[36] Davide Sangiorgi. A theory of bisimulation for the π-calculus. *Acta Informatica*, 33:69–97, 1996.

[37] Davide Sangiorgi. On the bisimulation proof method. *Mathematical Structures in Computer Science*, 6(5):447–479, 1998.

[38] Davide Sangiorgi and David Walker. *The π-Calculus—A Theory of Mobile Processes*. Cambridge University Press, 2001.

[39] Vladimiro Sassone and Pawel Sobociński. Deriving bisimulation congruences: 2-categories vs. precategories. In *Proceedings of FoSSaCS 2003*, pages 409–424. Springer LNCS 2003, 2003.

[40] Peter Sewell. From rewrite rules to bisimulation congruences. *Theoretical Computer Science*, 274(1–2):183–230, 2002.

[41] Howard Smith and Peter Fingar. *Business Process Mangament—the third wave*. Meghan-Kiffer Press, 2003.

[42] Pawel Sobociński. *Deriving Process Congruences from Reaction Rules*. PhD thesis, University of Aarhus, 2004.

[43] Terese. *Term rewriting systems*, volume 55 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge University Press, 2003.

[44] Glynn Winskel. Petri nets, algebras, morphisms, and compositionality. *Information and Computation*, 72(3):197–238, 1987.

Index