BRICS DS-99-1 G. L. Cattani: Presheaf Models for Concurrency

BRICS

Basic Research in Computer Science

Presheaf Models for Concurrency

(Unrevised)

Gian Luca Cattani

BRICS Dissertation Series **DS-99-1**

ISSN 1396-7002 **April 1999**

See back inner page for a list of recent BRICS Dissertation Series publications. Copies may be obtained by contacting:

BRICS
Department of Computer Science
University of Aarhus
Ny Munkegade, building 540
DK–8000 Aarhus C
Denmark

Telephone: +45 8942 3360
Telefax: +45 8942 3255
Internet: BRICS@brics.dk

BRICS publications are in general accessible through the World Wide Web and anonymous FTP through these URLs:

`http://www.brics.dk`
`ftp://ftp.brics.dk`
This document in subdirectory `DS/99/1/`

Presheaf Models for Concurrency

Gian Luca Cattani

Ph.D. Dissertation

Department of Computer Science
University of Aarhus
Denmark

Presheaf Models for Concurrency

Dissertation
presented to the Faculty of Science
of the University of Aarhus
in partial fulfillment of the requirements for the
Ph.D. degree

by
Gian Luca Cattani
October 5, 2000

To my parents,
Dora and Gian Carlo

Abstract

In this dissertation we investigate presheaf models for concurrent computation. Our aim is to provide a systematic treatment of bisimulation for a wide range of concurrent process calculi. Bisimilarity is defined abstractly in terms of *open maps* as in the work of Joyal, Nielsen and Winskel. Their work inspired this thesis by suggesting that presheaf categories could provide abstract models for concurrency with a built-in notion of bisimulation.

We show how presheaf categories, in which traditional models of concurrency are embedded, can be used to deduce congruence properties of bisimulation for the traditional models. A key result is given here; it is shown that the homomorphisms between presheaf categories, i.e., colimit preserving functors, preserve open map bisimulation.

We follow up by observing that presheaf categories and colimit preserving functors organise in what can be considered as a category of non-deterministic domains. Presheaf models can be obtained as solutions to recursive domain equations. We investigate properties of models given for a range of concurrent process calculi, including **CCS**, **CCS** with value-passing, π-calculus and a form of **CCS** with linear process passing. Open map bisimilarity is shown to be a congruence for each calculus. These are consequences of general mathematical results like the preservation of open map bisimulation by colimit preserving functors. In all but the case of the higher order calculus, open map bisimulation is proved to coincide with traditional notions of bisimulation for the process terms. In the case of higher order processes, we obtain a finer equivalence than the one one would normally expect, but this helps reveal interesting aspects of the relationship between the presheaf and the operational semantics. For a fragment of the language, corresponding to a form of λ-calculus, open map bisimulation coincides with applicative bisimulation.

In developing a suitable general theory of domains, we extend results and notions, such as the limit-colimit coincidence theorem of Smyth and Plotkin, from the order-enriched case to a "fully" 2-categorical situation. Moreover we provide a domain theoretical analysis of (open map) bisimulation in presheaf categories. We present, in fact, induction and coinduction principles for recursive domains as in the works of Pitts and of Hermida and Jacobs and use them to derive a coinduction property based on bisimulation.

Acknowledgements

Personal debts can never be adequately acknowledged.

I am especially grateful to my supervisor Glynn Winskel. Not only has he taught me how to do research, but he also transmitted his enthusiasm for it. It has always been a pleasure and a source of learning to discuss ideas with him and this thesis owes much to his stimulating guidance. While leaving me the freedom of choosing the problems I wished to work on, he has always been very involved in what I was doing to the point that this thesis can, in fact, be regarded as the result of four years of joint work. I shall also heartily thank him for his friendship.

Pino Rosolini gave unstinting support in more ways than one. He guided my first steps as a researcher when I was working on my 'tesi di laurea'. Later, when I decided to go on with postgraduate studies, he put me in contact and warmly suggested that I should study with Glynn. Ever since then he discreetly followed my progresses as a PhD student while always being available whenever I needed his help or advice.

Thanks are due to Vladimiro Sassone, Ian Stark and Marcelo Fiore. They all showed me friendship and stimulated my research. Marcelo in particular has been very influential in the development of an important part of this thesis, Chapter 6.

Many other people have influenced my work, taught me things or given advice in the last four years. It is hard to list everyone, but at least I wish to mention Mogens Nielsen, John Power, Jaap van Oosten, Alex Simpson, Anders Kock, Carsten Butz and Prakash Panangaden.

My PhD studies had been funded by BRICS and the BRICS PhD School. I wish to thank the BRICS management for having given me the chance of studying in Aarhus and more generally I wish to thank all the BRICS and DAIMI people who have created a perfect environment for foreign students like me.

Finally, my greatest thanks go to my wife Alida, for her love and dedication over the last ten years, and to my parents Dora and Gian Carlo, to whom this thesis is dedicated, for all their love and support throughout my life.

Contents

Introduction

This thesis aims to provide a systematic treatment of bisimulation for a wide range of concurrent process languages. We shall investigate so-called presheaf categories, as models for concurrency with a built-in notion of bisimulation, for the purpose of giving denotational semantics to concurrent process languages. This work is an offspring of the work on **models for concurrency** (see [141]) and on **open map bisimulation** [64]. In [141], the operations involved in the semantics of process languages across a range of different models were unified as instances of the same categorical constructions. In [64] an abstract notion of bisimulation, parametrised by a notion of observation shape or computation path, was introduced to accompany the models.

As we shall see, also in presheaf models the semantics to process languages can be given uniformly and this allows us to prove, independently from the specific models chosen for every particular language, some key properties of bisimilarity, such as congruence properties.

To better understand the significance and possible impact of such an effort towards unification, in this introduction, we recall briefly what kind of computational issues we are addressing, what are the main approaches to the semantics of concurrent process languages and what kind of problems one expects to face when following them. We shall then motivate the 'bisimulation from open maps' approach and show how this naturally leads to consider presheaf categories as models for concurrency. We will then highlight our main results together with a brief summary of the content of this thesis.

Background

The theory of concurrency aims to model and analyse the behaviour of systems made of many agents, simultaneously active and able to communicate with each other. The difficulties one encounters can be summarised in two major points: Firstly, the presence of different threads of control may lead to subtle, nondeterministic, and often unanticipated interactions among the various components of a system; second many concurrent systems, e.g., operating systems or distributed databases, have a behaviour that can be described as 'reactive' in the sense that they are designed to engage in a possibly endless series of interactions with the environment.

It becomes impossible to set up a semantic theory for these systems based on the input-output paradigm typical of sequential computations. In fact, unlike sequential computation, for concurrent computation there seems to be no general agreement of what its models should be. A different notion of behaviour, no longer based on the functional paradigm is needed. There are several reasons for this. Because concurrent systems are often designed to be non-terminating and continuously interacting with the environment their semantics should be based on their stimulus/response patterns, varying over time. These patterns rest often on recognising the existence of certain atomic elements of behaviour associated to the level of abstraction the description of the system. Central to the various approaches is the need for a satisfactory notion of behavioural equivalence between systems to replace the usual extensional equality of functions. Many equivalences are based on **bisimulation** [96, 80] which is roughly a relation between systems, matching the patterns of actions of one by those of the other.

But, how does one give semantics to concurrent processes languages and how does one find the appropriate bisimulation relation? There are, at least, two major approaches.

Operational semantics

A common way to give semantics of programming languages is in terms of transition relations which specify, for a term of the language at a certain state, which computation steps it can make. Such *operational semantics* is often given in a syntax-directed way [102].

Consider, as an example, the simple process language described by the following syntax:

$$P ::= \mathbf{Nil} \;\mid\; a.P \;\mid\; \overline{a}.P \;\mid\; (P \mid P) \;\mid\; (P + P)\ ,$$

where $a \in L$, with L a set of labels that stands for names of communication channels. A process term P can denote:

- A deadlocked process, **Nil**.
- A process, $a.P$, which can perform an input action along a channel a before becoming the process P.
- A process, $\overline{a}.P$, which can perform an output action along a channel a before becoming the process P.
- The parallel composition of two processes, $P_1 \mid P_2$.
- A process $P_1 + P_2$ which, depending on the environment, may behave like P_1 or P_2.

In order to be able to denote infinite (i.e., non-terminating) processes, we could have added the possibility of recursive definitions of process terms. For simplicity we leave this out here. The operational semantics of the language can be given by the following set of rules:

$$\frac{}{a.P \xrightarrow{a} P} \qquad \frac{}{\overline{a}.P \xrightarrow{\overline{a}} P}$$

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

$$\frac{P \xrightarrow{\alpha} P'}{P + Q \xrightarrow{\alpha} P'} \qquad \frac{Q \xrightarrow{\alpha} Q'}{P + Q \xrightarrow{\alpha} Q'}$$

$$\frac{P \xrightarrow{a} P' \qquad Q \xrightarrow{\overline{a}} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'} \qquad \frac{P \xrightarrow{\overline{a}} P' \qquad Q \xrightarrow{a} Q'}{P \mid Q \xrightarrow{\tau} P' \mid Q'}$$

where $\alpha \in L \cup \{\overline{a} \mid a \in L\} \cup \{\tau\}$. Note that we have introduced a "new" action symbol, 'τ', to mark the communication along channels of processes emitting outputs with processes willing to receive inputs.

A symmetric relation, $\mathcal{R}$, between process terms is defined to be a bisimulation if it satisfies the following property:

> If $P\,\mathcal{R}\,Q$ and $P \xrightarrow{\alpha} P'$ then there exists a process term Q' such that $Q \xrightarrow{\alpha} Q'$ and $P'\,\mathcal{R}\,Q'$, where $\alpha \in \{\tau\} \cup L \cup \{\overline{a} \mid a \in L\}$.

Two process terms will be said to be bisimilar if there exists a bisimulation which relates them.

Having defined bisimilarity, the next step usually taken is that of proving that it is a *congruence* with respect to the operators of the language.

This is the pattern usually followed. Unfortunately, things are not always so smooth, since process terms can be much more complicated that the ones proposed in the example above. They can, for instance, feature the possibility of passing channel names or other form of values, or even process terms. Sometimes new computational paradigms, other than simple input and output, might be built in to the language, e.g., the "firewalls" in Cardelli and Gordon's Ambient Calculus [20]. All this can considerably increase the complexity of the operational semantics and consequently of bisimulation. In turn, this makes it very hard to prove the basic facts one wants for bisimilarity, such as its congruence properties. In fact one often starts with a definition of operational semantics and bisimulation, aiming at the congruence result, but is always ready to tune them to each other in order to be able to prove it. Much of the problem here, relies on the fact that many of the choices, for both the operational semantics and the bisimilarity, are taken in an *ad hoc* fashion.

Another drawback of semantics based on syntactic models arises when one tries to relate and formally compare different models. The comparison has to go via syntactic translations which are often very difficult to find, justify and understand.

Denotational semantics

A *denotational semantics* for concurrent languages can often be obtained by enhancing the usual domain theoretic tools, used for the semantics of sequential programming languages [137, 104, 121], with constructions able to capture non-deterministic behaviour. These are often given as *powerdomain* constructions and provide for any suitable domain D, a new domain $\mathcal{P}(D)$ whose elements are roughly subsets of D (see [101, 104, 3]). If a sequential program denoted a function

$$f : D \longrightarrow D \ ,$$

for some suitable domain D (of states for instance), a non-deterministic counterpart would denote a function

$$f : D \longrightarrow \mathcal{P}(D) \ .$$

For the small process language of the previous section, a suitable domain could be given by solving the following recursive domain equation [3]:

$$D = \mathcal{P}(D_\perp + \sum_{a \in L} D_\perp + \sum_{a \in L} D_\perp) \ ,$$

where $D_\perp$ is the construction which adds a bottom element to a domain and $\sum_{i \in I} D_i$ is the sum of domains.

Term constructors and constants such as $a.-$, $(- \mid -)$, **Nil**, … will denote endofunctions of suitable arity on the domain D. Terms in the language are equated if they denote the same element of the domain. But there is no guarantee that this relation will in fact coincide with bisimilarity. On the contrary there are reasons to believe that in general this will not be the case. Recursively defined terms which have equal denotations at each finite stage in their least fixed point definitions, will also have equal denotations. But it is known that the coinductive definition of bisimilarity does not in general *close at* ω in this way. By restricting the degree of branching allowed, one can sometime overcome this deficiency [51, 3, 126, 32, 50].

A possible way to systematise the definition of bisimulation and overcome the limitations of traditional domain theory came out of work on models for concurrency.

Models for concurrency

Many different kinds of models for concurrent/distributed computation have been studied. Work such as [141, 119, 135, 136] has concentrated on understanding their structure and mutual relationships. The natural tools for carrying out this task were provided by **category theory** [76]. The idea was to turn classes of models into categories by adding suitable notions of morphisms between models to account for the possibility of a model simulating the behaviour of another. There were two main outcomes to this approach. First, semantics given to process languages across different categories of models were "unified" in the sense that it was shown that the operations denoted by the term constructors had often the same categorical status. Hence a "general" language for so-called **CCS**-like process calculi was devised, together with specifications of the minimal required properties of a category to be considered a category of models for the language. By means of this it became possible to give semantics uniformly to a number of different process calculi such as **CCS** [82], **CSP** [55], **SCCS** [81] and **ACP** [6]. Second, there was the possibility of formally relating categories of models by means of adjunctions or coreflections (adjoint pairs whose left adjoint is a full embedding) of one category onto another. This accounted for the informal understanding that certain models were more or less expressive or abstract than others [119]. We can exemplify this using two well-known classes of models, transitions systems and synchronisation trees. A *transition system*, T, over a set of action, **Act**, is a triple

$$(S, i, \mathit{tran})$$

where

- S is a set of states.
- $i \in S$ is a distinguished *initial* state.
- $\mathit{tran} \subseteq S \times \mathbf{Act} \times S$ is a *transition relation.*

Triples (s, a, s') in *tran* are usually written $s \xrightarrow{a} s'$. The idea is that the action are the observable parts of the computation steps of a system, that starting at i can evolve according to the transition rules. If a transition $s \xrightarrow{a} s'$ is a step of computation, then a morphism must tell us how to simulate it in a different system. A function $f : S_1 \to S_2$ is a transition system morphism, $f : T_1 \to T_2$, where $T_1 = (S_1, i_1, \mathit{tran}_1)$ and $T_2 = (S_2, i_2, \mathit{tran}_2)$ are two transition systems if

1. $f(i_1) = i_2$ and
2. $s \xrightarrow{a} s' \in \mathit{tran}_1$ implies $f(s) \xrightarrow{a} f(s') \in \mathit{tran}_2$.

The resulting category we write as $\mathcal{TS}_{\mathbf{Act}}$.

Transition systems may loop, i.e., there might be sequences of transitions

$$s \xrightarrow{a_1} s_1 \xrightarrow{a_2} s_2 \xrightarrow{a_3} s_3 \longrightarrow \cdots \xrightarrow{a_n} s_n = s \ ,$$

or some state can be reached with different computations,

$$\begin{tikzcd}
& s_1 \arrow[dr, "b"] & \\
s \arrow[ur, "a"] \arrow[dr, "c"'] & & s' \\
& s_2 \arrow[ur, "d"'] \ . &
\end{tikzcd}$$

Transition systems can then be *unfolded* by unwinding the loops and duplicating states that are reachable in different ways. This *unfolding* operation results in a labelled tree. Labelled trees (or *synchronisation trees* as they are often called and as we will always call them) are special transition systems whose transition graph is a tree. Since any synchronisation tree is a transition system there is an embedding

$$e : \mathcal{ST}_{\mathbf{Act}} \hookrightarrow \mathcal{TS}_{\mathbf{Act}}$$

of the category $\mathcal{ST}_{\mathbf{Act}}$ of synchronisation trees over **Act** in the category of transition systems over **Act**. The unfolding operation results in a *right adjoint* $\mathcal{U}$ to e that moreover is also a left inverse to e, i.e., the unfolding of a tree returns the tree itself. It is these kinds of formally precise relationship that one is seeking with the hope of being able to use them to deduce general properties of models as well as uniformity in the compositional semantics of languages across different models. This theoretical work can also have an impact on practical issues. For example, an unfolding result, analogous to the one sketched here, for Petri Nets and Event Structures studied in [93] was used for the purpose of model checking algorithms for Petri Nets [79].

This analysis accounted for the structure of process terms. What about their behavioural equivalences? Is there a way of representing bisimilarity abstractly in the models? A proposed answer came by adopting the notion of open map, originally developed by Joyal and Moerdijk for topos theoretic purposes [62, 63], in the context of models for concurrency.

Open map bisimulation

Morphisms in a category of models are understood as kinds of functional simulations. It is natural to ask oneself whether it is possible to distinguish among all such *functional* simulations those that in fact are bisimulations. Let's do this with an example. Consider two transition systems over the same set of labels, **Act**, $T_1 = (S_1, i_1, tran_1)$ and $T_2 = (S_2, i_2, tran_2)$. Let $f : T_1 \to T_2$ be a morphism from the first one to the second one. In other words, as we said, let $f : S_1 \to S_2$ be a function between the two sets of states such that $f(i_1) = i_2$ and such that if $s \xrightarrow{a} s' \in tran_1$, then $f(s) \xrightarrow{a} f(s') \in tran_2$.

The morphism f is said to be a *zig-zag morphism* [12] if it further satisfies the following property, for every reachable state $s \in S$ and every label $a \in \mathbf{Act}$:

$$f(s) \xrightarrow{a} t' \in tran_2 \Longrightarrow \exists\, t \in S \;\; s \xrightarrow{a} t \in tran_1 \,\wedge\, f(t) = t' \ .$$

In other words, f not only preserves but also *reflects* reachable transitions. It takes a moments reflection to see that if f is a zig-zag morphism, then its graph is a bisimulation between the two transition systems.

Suppose we are given a sequence of transitions in T,

$$\sigma = i \xrightarrow{a_o} s_o \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s \ .$$

If $f : T \to T'$ is a morphism as above, it induces a 'simulating' sequence in T':

$$\sigma' = i' \xrightarrow{a_o} f(s_o) \xrightarrow{a_1}{}' f(s_1) \xrightarrow{a_2}{}' \dots \xrightarrow{a_n}{}' f(s) \ .$$

If f is zig-zag, then, if τ' is the extension of σ' with an extra transition $f(s) \xrightarrow{a}{}' t'$, one has that also σ can be extended to a sequence

$$\tau = i \xrightarrow{a_o} s_o \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s \xrightarrow{a} t$$

and moreover t can be chosen such that $f(t) = t'$.

Abstracting away, finite sequences of transitions can be thought of computation shapes which one can observe of a system, and a morphism

$$p : \sigma \to T \ ,$$

from one such computation shape to a transition system as a computation of shape σ observed in T. This motivates the following definition.

> Let $\mathcal{M}$ be a category and let $\mathbb{P}$ be a subcategory of $\mathcal{M}$:
>
> $$\mathbb{P} \hookrightarrow \mathcal{M} \ .$$
>
> A morphism $f : M_1 \to M_2$ in $\mathcal{M}$ is a $\mathbb{P}$-open map if it satisfies the following *path-lifting* property: For every commuting square
>
> $$\begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & M_1 \arrow[d, "f"] \\ Q \arrow[r, "q"'] & M_2 \end{tikzcd}$$
>
> with $m : P \to Q$ a morphism in $\mathbb{P}$, there exists a morphism
>
> $$r : Q \longrightarrow M_1 \ ,$$
>
> splitting the square in two commutative triangles:
>
> $$\begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & M_1 \arrow[d, "f"] \\ Q \arrow[r, "q"'] \arrow[ur, dotted, "r"] & M_2 \ . \end{tikzcd}$$

The definition can be understood intuitively as follows. Recall that $f : M_1 \to M_2$ gives a way for M_2 to simulate the behaviour of M_1. If P is a computation shape, $p : P \to M_1$ describes a computation of shape P in M_1. Via f, this is simulated in M_2 with fp. On the other hand we have Q that extends P (via $m : P \to Q$) and $q : Q \to M_2$ is a computation of shape Q in M_2 that extends pf, since the square commutes ($qm = fp$). What the condition tells us is that if f is $\mathbb{P}$-open, then we should be able to match the extension m already at the level of M_1 (the reason why the upper triangle commutes) and this should be consistent with the computation q and the simulation f (the reason why the lower triangle commutes).

The name "open map" is inherited from the work of Joyal and Moerdijk [62]. There they define axiomatically classes of open maps in toposes [77]. As we shall see in the next section, a main example of such a class (Example 1.1 in their paper) can be given in terms of the *path-lifting* property of the definition above.

Open maps account for functional bisimulation, but since one wants to express all possible bisimulations, one defines two models M_1 and M_2 to be $\mathbb{P}$-open map bisimilar if they are connected by a span of $\mathbb{P}$-open maps, f and g:

$$\begin{tikzcd} & M_3 \arrow[dl, "f"'] \arrow[dr, "g"] & \\ M_1 & & M_2 \end{tikzcd} .$$

For several well studied categories of models, like those of transition systems, synchronisation trees and event structures the choice of the corresponding path categories is natural and as it was shown in [64], the notion of bisimulation one gets coincides with already existing ones. For instance, in our example with transition systems, $\mathcal{M}$ would be $\mathcal{TS}_{\mathbf{Act}}$ and $\mathbb{P}$ its full subcategory of finite transition systems whose transition graph has only one branch which is non-looping, i.e., $\mathbb{P}$ is equivalent to $\mathbf{Act}^*$ the partial order of finite words over **Act** regarded as a category.

So for categories of models we have a notion of bisimulation parameterised by the choice of a suitable category of computation-path shapes, a *path category* as we shall call it in this thesis. There is much freedom in the choice of the path category, $\mathbb{P}$, which, in principle, bears no particular relationship with the category $\mathcal{M}$ other than being one of its subcategories. As extreme examples, by choosing $\mathbb{P}$ to be the empty category one obtains that any two objects of $\mathcal{M}$ are bisimilar, while by choosing it to be $\mathcal{M}$ itself, only isomorphic objects will be related. For some concrete examples, such as transition systems, certain path categories immediately suggest themselves as "natural" choices (see [64]) but, in general we cannot expect this to always happen. Moreover, this problem is related to that of proving abstractly properties of bisimilarity, notably that it is a congruence with respect to the term constructors of the language. To prove such results, it seems that a more disciplined way of providing models out of path categories is needed.

Presheaf models

There is an important class of categories which are equipped with a canonical choice of path category. These are the so-called **presheaf categories**. Given a (small) category $\mathbb{C}$, the category of presheaves over $\mathbb{C}$, $\widehat{\mathbb{C}}$, is the category of contravariant functors $F : \mathbb{C}^{\mathrm{op}} \to \mathbf{Set}$ (where **Set** is the category of sets and functions) and natural transformations. Presheaves are a central concept in category theory and especially topos theory [59, 77]. What is crucial for us is that the category $\widehat{\mathbb{C}}$ is also a concrete representation of the *free colimit completion of* $\mathbb{C}$. That is, $\widehat{\mathbb{C}}$ extends $\mathbb{C}$ and any colimit preserving functor from $\widehat{\mathbb{C}}$ to any category with colimits is uniquely (up to isomorphism) determined by its action on $\mathbb{C}$. This means that for any cocomplete category[1] $\mathcal{E}$, and functor $F : \mathbb{C} \to \mathcal{E}$, there exists a unique (up to a natural isomorphism) functor $F_! : \widehat{\mathbb{C}} \to \mathcal{E}$ that preserves colimits and makes the triangle

$$\begin{tikzcd} \mathbb{C} \arrow[r, "\mathrm{y}_{\mathbb{C}}"] \arrow[dr, "F"'] & \widehat{\mathbb{C}} \arrow[d, "F_!"] \\ & \mathcal{E} \end{tikzcd}$$

commute, where $\mathrm{y}_{\mathbb{C}}$ is the well known Yoneda embedding [76]. We shall use this property repeatedly throughout the thesis. The "inclusion" of $\mathbb{C}$ into $\widehat{\mathbb{C}}$, provided by the *Yoneda embedding functor*, from the open maps point view equips $\widehat{\mathbb{C}}$ with a canonical choice of path category.

This was an important example (Example 1.1) of a class of open maps in the sense of [62]. There they declared a map $f : X \to Y$ in $\widehat{\mathbb{C}}$ to be open if for every arrow $m : C \to D$ in $\mathbb{C}$ the associated naturality square

$$\begin{tikzcd} X(D) \arrow[r, "X(m)"] \arrow[d, "f_D"'] & X(C) \arrow[d, "f_C"] \\ Y(D) \arrow[r, "Y(m)"'] & Y(C) \end{tikzcd}$$

was a quasi-pullback. As we shall recall in Chapter 2, this condition, via the Yoneda lemma, is easily seen to be equivalent to the path-lifting property of the definition of the previous section.

As observed in [64] presheaf categories over the appropriate path categories, fully and faithfully embed (and sometimes are equivalent to) traditional categories of models and the embeddings preserve and sometimes reflect open map bisimulation. That paper suggested that one should look into presheaves as "abstract" categorical models of concurrent computation with a built-in notion of bisimulation.

There are also intuitive reasons for considering presheaf categories as models for concurrency. If the objects of a small category $\mathbb{P}$ are to be thought as finite deterministic computations, how does one generate non-deterministic, possibly infinite processes out of them? One wants to add the possibility of choosing between different possible computations (i.e., one adds coproducts to the category) and joining computations together when they are supposed to agree for a while before a choice (a branching) is taken (and this corresponds to adding coequalisers). Altogether then one is adding, by a known result of category theory [76], colimits of the size of the added coproducts (the non deterministic choices) and if these are unbounded, one is adding *all* colimits.

Given the possibility of embedding traditional models in presheaf categories, one hope was to be able to use the rich categorical structure of presheaves to prove properties

[1] I.e., with colimits of all small diagrams.

of open map bisimulation that could then be transferred to the traditional models. It was later realised [138] that the category (more correctly speaking the 2-category) of presheaf categories and colimit preserving functors possesses features that that lead to it being considered as a category of non-deterministic domains, where elements are replaced by presheaves and the usual information order is replaced by more detailed natural transformations between presheaves. A way of building presheaf models for concurrent process calculi along domain theoretical lines was viable. The idea is that presheaf models for concurrent process calculi can be obtained by solving appropriate domain equations for the path category. For instance one can use a lifting construction $\mathbb{P}_\perp$, meaning that a strict new initial object, $\perp$, was added to $\mathbb{P}$, to represent the requirement the a certain action, represented by $\perp$ is to be observed before any further observation in $\mathbb{P}$ can be made. The connection with *prefixing* is evident and, for instance, the process language above is modelled by synchronisation trees over the set

$$\mathbf{Act} \stackrel{\text{def}}{=} \{\tau\} \cup L \cup \{\overline{a} \mid a \in L\}$$

and they are the presheaf model, $\widehat{\mathbb{P}}$, where $\mathbb{P}$ is a solution to

$$\mathbb{P} = \mathbb{P}_\perp + \sum_{a \in L} \mathbb{P}_\perp + \sum_{a \in L} \mathbb{P}_\perp = \sum_{a \in \mathbf{Act}} \mathbb{P}_\perp \ .$$

The solution to this equation, in fact is provided by the partial order, $\mathbf{Act}^+$, of non-empty finite words of **Act** regarded as a category and there is an equivalence of categories:

$$\mathcal{ST}_{\mathbf{Act}} \simeq \widehat{\mathbf{Act}^+} \ .$$

Open map bisimulation in this case corresponds to Park-Milner bisimulation.

For these intuitions to make precise sense one needs to develop a suitably general theory of domains. The axiomatic approach [105, 30, 35] to the theory of domains paves the way. There one defines axiomatically classes of categories that can be thought of as *categories of domains* (generally order-enriched ones) and that by the axioms are guaranteed to provide uniform solution to recursive domain equations [125, 30]. From this perspective presheaf categories can be thought of as categories of non-deterministic domains and the operation of forming the presheaf category as analogous to a power-domain construction.

This thesis builds on the above hopes and intuitions about presheaves. We analyse properties of presheaf categories and colimit preserving functors. We prove that open map bisimulation is preserved by these kinds of functors; this leads to abstract congruence results of bisimilarity for the semantics of **CCS**-like process languages. Further, we develop a general theory of domains appropriate to our needs and test our method against non-trivial examples of process languages, ranging from **CCS**, to the π-calculus, to a process passing calculus. We use the theory developed to study open map bisimulation from a domain theoretical perspective and provide the first, to our knowledge, domain theoretic characterisation of bisimulation for *arbitrary* trees. In this thesis we aim to reconcile the semantics of concurrency with domain theory. Our models possess an abstract bisimulation and come automatically equipped with congruence properties.

Synopsis

The first two chapters provide some background material. **Chapter 1**, assuming some basic knowledge of category theory [76], say up to the level of adjunctions and limits, introduces the key categorical concepts that will be used in this thesis. **Appendix A** is a companion to this chapter and provides some basic definitions of enriched category theory.

Chapter 2 essentially summarises the definitions and results of [64], where the notion of open map bisimulation is introduced. The definition of some well-known categories of models for concurrency, including transition systems, synchronisation trees and event structures are also recalled.

The original contributions of this thesis start at **Chapter 3**. There we refine the axiomatisation of categorical models for **Proc**, a general **CCS**-like language, as given implicitly in [141], with extra logical assumptions. We then define presheaf models for **Proc** and show that they satisfy the axiomatisation. The main highlight of the chapter is Proposition 3.2.5 (though a proof of it is postponed to the following chapter) which asserts that the colimit preserving functors between presheaf categories preserve open maps. For this reason it is possible to prove for presheaf models that open map bisimulation is a congruence with respect to the interpretation of the operators of **Proc**. We use this result to (re)prove that strong history preserving bisimulation [41, 113] for event structures is a congruence. Further we show that a refinement functor on the particular presheaf model that extends event structures obtained as a colimit preserving functor coincides on event structures with a refinement proposed in [41] and again this entails that such a strong history preserving bisimulation is preserved by this refinement.

The results of this chapter were announced in a joint paper with Glynn Winskel [26] that appeared in the proceedings of CSL '96.

Chapter 4 is devoted to the study of the bicategory **Prof** of profunctors or equivalently the 2-category **Cocont** of presheaf categories, colimit preserving functors and natural transformations. Here we make explicit the categorical folklore about **Prof** failing to be a compact closed category just because it fails to be a category [66]. It is known that compact closed categories can provide (degenerate) models of classical linear logic [38, 39]. An exponentiation (pseudo) functor is provided for this purpose. In particular the way the exponential is built suggests analogies between presheaves and powerdomains [101] and hence between **Prof** and categories of non-deterministic domains [51]. The technically simple but very important notion of *lifting* of a category is introduced and this allows us to represent in **Prof** *connected colimit* preserving functors between presheaf categories. This is the largest class of (non trivial) functors between presheaf categories that we have proved to preserve open map bisimulation. Moreover this class seems to include all the functors needed in modelling linear process calculi. To summarise the investigation on the structure of **Prof** we define a linear type theory extended with lifting that we interpret in it. This will be further extended in Chapter 6 with recursive types. Finally we prove in full detail that the horizontal composition of (epimorphic) open 2-cells is (epimorphic) open. This entails immediately as corollaries that both colimit preserving functors and connected colimit preserving functors preserve open map bisimulation as announced earlier.

Chapter 5 gives the reader a break from categorical issues and provides two examples of how we use **Prof** as a category of domains to deduce presheaf models of process calculi. We first rework the usual synchronisation tree semantics of **CCS** in this new setting. Recall that as we said before the category of synchronisation trees over a fixed set of labels, **Act**, is in fact equivalent to a presheaf category, more precisely to the category of presheaves over $\mathbf{Act}^+$, the partial order (regarded as a category) of finite and non-empty words over **Act**, with the prefix ordering. The second example we borrow from [138] and so provide a presheaf model for a form of **CCS** with value passing, with both late and early semantics.

In **Chapter 6** we generalise classical results on the solution of recursive domain equation from the order enriched case [125] to a more general class of 2-categories. The generalisation proceeds in three directions:

1. We consider *adjoint pairs* rather than *embedding-projection pairs*. This follows established categorical folklore [58, 49, 129, 124].
2. We move up from order enriched categories to 2-categories whose *hom categories* have colimits of ω-chains.
3. We consider *pseudo* limits (elsewhere called bilimits [127]) instead of enriched ones.

The pay off for this effort is a general theory of domains that specialises in the order enriched case to the usual one.
A "pseudo" version of the *Basic Lemma* [125] is considered and it allows the construction of solutions to recursive domain equations as colimits of the "standard" chain of iterations. Having an axiomatic treatment in mind [105, 30, 35] a pseudo version of Freyd's notions of *algebraic completeness and compactness* [35] is also developed and building on the thesis work of Fiore [30] a class of 2-categories that are axiomatically provable to be pseudo algebraically compact is devised. Not surprisingly, **Cocont** belongs to such class. This allows us to extend the type theory of Chapter 4 with recursive types and so formalising our intuition about **Cocont** being a category of domains.
Further we use these results to provide a domain theoretical understanding of open map bisimulation by means of relational structures [94, 100] and induction/coinduction principles for recursively defined domains as in [99, 31]. In particular we define the notion of *intensional relation* in **Cocont** and give a domain theoretical characterisation of strong bisimulation for *arbitrary* trees.

The results presented in this chapter are part of a joint paper with Marcelo Fiore and Glynn Winskel [21] that appeared in the proceedings of LICS '98.

Chapter 7 tackles the task of providing presheaf models for name-passing calculi. Our example is the π-calculus [87, 88]. The bicategory **Prof** is indexed with a category of name sets, $\mathcal{I}$, as in [126, 32, 50]. A model for the *late* π-calculus is given and it is proved that for processes with free names within a certain set s, open map bisimilarity at the fibre over s coincide with late bisimulation, while open map bisimilarity for each substitution of the free names coincide with the largest congruence included in late bisimilarity. A model for the π-calculus with *early* bisimulation is sketched along with an arrow in $\mathbf{Prof}^{\mathcal{I}}$ that maps the late interpretation onto the early one.

This chapter is based on a joint paper with Ian Stark and Glynn Winskel [25] that appeared in the proceedings of CTCS '97.

Chapter 8 shows the state of our knowledge as far as the modelling of higher order process calculi [130, 116, 118] with presheaves is concerned and is part of ongoing research. A denotational model for a *linear* higher order process language over a fixed set of channels is deducible from the work of the previous chapters. It comes equipped with a notion of bisimulation and a proof that bisimulation is a congruence with respect to the term constructors. The difficulty is in reading off an operational understanding of the bisimulation. Elements of presheaves, i.e., elements of the sets $X(P)$, for $X : \mathbb{P}^{\mathrm{op}} \to \mathbf{Set}$ a presheaf, play an essential role in characterising the bisimulation for higher order processes, e.g., abstractions. Bisimulation between abstractions is characterised not only by its pointwise behaviour, i.e., by the fact that bisimilar inputs are mapped to bisimilar outputs, but requires also a uniformity constraint with respect to the input. It is here that it seems essential to trace in a presheaf when two elements are related instances of the same observation, for different inputs. Although we lack a proof, elements correspond to derivation trees in the operational semantics. Hence to characterise bisimulation operationally one is led to decorate the transition arrows with expressions that account for the derivation tree that allowed the transition. Unfortunately we do not yet see how to read off the functorial action of presheaf denotations from a notation for derivations. Still we prove results like a Substitution Lemma asserting that substitution in the language amounts to application in the model and a stability-like property [13] of open terms which helps simplify the characterisation of open map bisimulation. A sub-calculus of the higher order calculus corresponds to a λ-calculus and, in absence of non-determinism the above distinctions on derivation trees become vacuous. In this case open map bisimulation corresponds to *applicative bisimulation* [1].

Chapter 9 concludes this thesis and gives pointers to related and possible future work.

Chapter 1

Categorical Background

We review in this chapter some concepts of category theory that we will need in the sequel and that are either not covered in [76] or are so important in our development that needs to be recalled explicitly anyway. This also in order to fix some notation for the subsequent development. We will deal in this chapter (in quite a scattered manner) with presheaf categories, Kan extensions and fibrations. Bicategories and 2-categories shall play an important role as well, starting from Chapter 4. A proper introduction to these notions would make the size of this introductory chapter growing too much. For the knowledgeable reader we shall say a few things concerning the terminology that we have decided to adopt here and instead refer to the literature for most of the definitions and concepts. As a primer guide we have reported the most basic definitions in Appendix A.

1.1 Notation

We shall write $\mathbb{C}, \mathbb{D}, \mathbb{E}, \ldots$ or $\mathbb{P}, \mathbb{Q}, \mathbb{R}, \ldots$ to indicate *small* categories, i.e., categories for which the collection of arrows is a set. Instead we shall write $\mathcal{B}, \mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{K} \ldots$ for larger categories. In particular we will consider in this thesis mainly *locally small* categories. By this we mean categories, $\mathcal{C}$, whose collection of objects, $|\mathcal{C}|$, is a class and for which, given any two objects, $c, c' \in |\mathcal{C}|$, the corresponding *hom-collection*, $\mathcal{C}(c, c')$ is a set. We shall write **Set** for the (locally small) category of sets and functions and **Cat** for the (locally small) category of small categories and functors. If $\mathcal{C}$ and $\mathcal{D}$ are two categories, we write $\mathbf{CAT}(\mathcal{C}, \mathcal{D})$ for the category of functors between $\mathcal{C}$ and $\mathcal{D}$ (we shall write **Cat** instead of **CAT**, if $\mathcal{C}$ and $\mathcal{D}$ are known to be small) and natural transformations between such functors. Generally we shall also write **CAT** for the category of (locally small) categories and functors.

If $F, G : \mathcal{C} \to \mathcal{D}$ are two functors, a natural transformation α from F to G will be indicated in general with the *dotted* arrow notation $\alpha : F \stackrel{\cdot}{\longrightarrow} G$. When considering categories of functors and natural transformations, e.g., $\mathbf{CAT}(\mathbb{C}^{\mathrm{op}}, \mathbf{Set})$, the dot will disappear. When natural transformations will play the role of 2-cells in a 2-category they will be indicated with the double arrow notation, $\alpha : F \Longrightarrow G$.

An important category for us will be $\mathbf{Set}_*$ the category of sets and partial maps. Arrows of $\mathbf{Set}_*$ will normally be indicated with the notation, $f : X \rightharpoonup Y$. To avoid confusion the product of sets, say X and Y, as objects of $\mathbf{Set}_*$ will be indicated as $(X \times_* Y)$.

When 2-categories and bicategories will be used we shall employ the same notation for both a 2-category and its underlying category of objects and 1-cells, e.g., **Cat** will stand for both the 2-category of small categories functors and natural transformations and for the category of small categories and functors. The context should always disambiguate the possible confusion.

1.2 Presheaf categories

Presheaves are the central notion around which this thesis is built.

Definition 1.2.1 (Presheaf categories) *If $\mathbb{C}$ is a small category, define the (locally small) category of* presheaves over $\mathbb{C}$, $\widehat{\mathbb{C}}$, *to be the category of contravariant endofunctors from $\mathbb{C}$ to* **Set** *and natural transformations. That is, objects of $\widehat{\mathbb{C}}$ are functors, $X : \mathbb{C}^{\mathrm{op}} \to$* **Set** *while arrows $f : X \to Y$ are natural transformations between such functors. If $\mathbb{C}$ has initial object, we say that a presheaf, X, is* rooted *if $X(0)$ is a singleton set, for any initial object 0 of $\mathbb{C}$. We write $\overline{\mathbb{C}}$ for the full subcategory of $\widehat{\mathbb{C}}$ of rooted presheaves.*

Observe that the category of rooted presheaves is equivalent to the category of presheaves over the full subcategory of $\mathbb{C}$ consisting of all but the initial objects.

Presheaf categories are important in category theory for at least two reasons. On the one hand they provide examples of Grothendieck toposes (for trivial topologies); even better, every Grothendieck topos is a reflective subcategory of a presheaf category [77]. On the other hand, the presheaf constructions yields an explicit description of the free completion of a small category under all small colimits. We will be mainly concerned with this second way of looking at presheaf categories, even if we will occasionally refer to topos theoretic concepts and terminology.

Definition/Proposition 1.2.2 (Yoneda embedding) *If $\mathbb{C}$ is a small category, then define the* Yoneda embedding *of $\mathbb{C}$ to be the following functor:*

$$\begin{array}{ccc}
\mathbb{C} & \xrightarrow{\mathrm{y}_{\mathbb{C}}} & \widehat{\mathbb{C}} \\
c & & \mathbb{C}(-,c) \\
f\downarrow & \mapsto & \downarrow f\circ - \\
c' & & \mathbb{C}(-,c')
\end{array} .$$

Presheaves isomorphic to $\mathrm{y}_{\mathbb{C}}(c)$ for some $c \in |\mathbb{C}|$ are called representables.

The Yoneda embedding is a full and faithful functor.

Proposition 1.2.3 (Yoneda lemma) *Let $\mathbb{C}$ be a small category. Let X be a presheaf over $\mathbb{C}$. Let $c \in |\mathbb{C}|$. Then there exists a bijection*

$$X(c) \overset{b_{c,X}}{\cong} \widehat{\mathbb{C}}(\mathrm{y}_{\mathbb{C}}(c), X)$$

natural in c and X. Hence the functor X is naturally isomorphic to the functor $\widehat{\mathbb{C}}(\mathrm{y}_{\mathbb{C}}(-), X)$.

Because of the Yoneda lemma we can always identify the elements $x \in X(c)$ of a presheaf X at c, with the corresponding natural transformation $x : \mathrm{y}_{\mathbb{C}}(c) \stackrel{\cdot}{\longrightarrow} X$. We can now state the property asserting that $\widehat{\mathbb{C}}$ is the free colimit completion of $\mathbb{C}$.

Proposition 1.2.4 *Let $\mathbb{C}$ be a small category. Let $\mathcal{E}$ be a cocomplete category (i.e., a category with all small colimits) and let $F : \mathbb{C} \to \mathcal{E}$ be a functor. Then $\widehat{\mathbb{C}}$ is cocomplete and there exists a unique (up to a natural isomorphism) colimit preserving functor $F_! : \widehat{\mathbb{C}} \to \mathcal{E}$ such that $F = F_! \mathrm{y}_{\mathbb{C}}$:*

$$\begin{tikzcd} \mathbb{C} \arrow[r, "\mathrm{y}_{\mathbb{C}}"] \arrow[dr, "F"'] & \widehat{\mathbb{C}} \arrow[d, dotted, "\exists! F_!"] \\ & \mathcal{E} \end{tikzcd} .$$

Proposition 1.2.4 is one of the most important results of category theory. The reader might look at [65] for a proof of it.

NOTATION:

- We shall indicate with the empty set symbol, $\emptyset$, the initial presheaf in any presheaf category, i.e., the presheaf that maps every object to the empty set.
- If $F : \widehat{\mathbb{C}} \to \widehat{\mathbb{D}}$ is a colimit preserving functor, we say that F is *cocontinuous* [65, 98]. We write $\mathbf{Cocont}(\mathbb{C}, \mathbb{D})$ for the category of cocontinuous functors between $\widehat{\mathbb{C}}$ and $\widehat{\mathbb{D}}$ and natural transformations.

The freeness property described above yields the following proposition.

Proposition 1.2.5 *The category $\mathbf{CAT}(\mathbb{C}, \widehat{\mathbb{D}})$ is equivalent to $\mathbf{Cocont}(\mathbb{C}, \mathbb{D})$ for any two small categories $\mathbb{C}$ and $\mathbb{D}$.*

Presheaf categories are not just cocomplete categories but they also provide examples of Grothendieck toposes [77].

Proposition 1.2.6 *If $\mathbb{C}$ is a small category, the category $\widehat{\mathbb{C}}$ is a Grothendieck topos, that is, $\widehat{\mathbb{C}}$ is cartesian closed, complete and cocomplete and it has a sub-object classifier.*

As far as we are concerned in this thesis the main properties of $\widehat{\mathbb{C}}$ that we shall need are associated with its (co)completeness.

Going back to the freeness property, a calculation (see for instance [77], pages 40–44 where things are presented in an inverse order to here) shows that the functor $F_!$ of Proposition 1.2.4 has a right adjoint F^* given by restricting the hom sets of $\mathcal{E}$ to range over the images of objects of $\mathbb{C}$ under F, i.e., for every $E \in \mid \mathcal{E} \mid$, $F^*(E) = \mathcal{E}(F(-), E)$. In Chapter 3 we will make an extensive use of instances of the functor F^* above that therefore deserve special terminology.

Definition 1.2.7 (Canonical functors) *Let $\mathcal{M}$ be any locally small category. Let $F : \mathbb{C} \to \mathcal{M}$, with $\mathbb{C}$ small. Define the* canonical functor from $\mathcal{M}$ to $\widehat{\mathbb{C}}$ according to F,

$c_F : \mathcal{M} \to \widehat{\mathbb{C}}$ to be:

$$\begin{array}{ccc} \mathcal{M} & \xrightarrow{c_F} & \widehat{\mathbb{C}} \\ M & & \mathcal{M}(F(-), M) \\ f\downarrow & \mapsto & \downarrow f \circ - \\ M' & & \mathcal{M}(F(-), M') \ . \end{array}$$

If F is an embedding $\mathbb{C} \hookrightarrow \mathcal{M}$, we write $c_{\mathcal{M}}$ for c_F.

Recall that a functor $F : \mathcal{C} \to \mathcal{D}$ is *dense* if every object, D, of $\mathcal{D}$ is a 'canonical' colimit of objects of $\mathcal{C}$ according to F, i.e.,

$$D \cong colim(F/D \to \mathcal{C} \xrightarrow{F} \mathcal{D}) \ .$$[1]

Proposition 1.2.8 *In the situation of Definition 1.2.7, c_F is full and faithful if and only if F is dense.*

Often in this case the functor F will just be an inclusion of categories and we will talk of the canonical embedding, *$c_{\mathcal{M}}$.*

The functor c_F is known to preserve all limits that exist in $\mathcal{M}$; on the contrary c_F does not, in general, preserves colimits in $\mathcal{M}$. Still colimits of certain diagrams are preserved as the following proposition that we shall need in Section 3.5 shows.

Proposition 1.2.9 *Let $\mathcal{M}$ be a locally small category, $\mathbb{P}$ be a small category and let $F : \mathbb{P} \to \mathcal{M}$ be a functor. Let $\Delta : \mathbb{D} \to \mathcal{M}$ be another functor from a small category $\mathbb{D}$ satisfying the following property of "density with respect to F": If $(M, \delta_D : \Delta(D) \to M)$ is a colimiting cone for Δ, then for any $P \in \mid \mathbb{P} \mid$ and $p : F(P) \to M$, there exists a $D \in \mid \mathbb{D} \mid$ and $d : F(p) \to \Delta(d)$ such that:*

- *$p = \delta_D d$.*
- *For any other factorisation*

$$\begin{tikzcd} F(P) \arrow[rr, "p"] \arrow[dr, "d'"'] & & M \\ & \Delta(D') \arrow[ur, "\delta_{D'}"'] & \end{tikzcd} ,$$

there exists $m : D \to D'$ such that

$$\Delta(m)d = d' \qquad \text{and} \qquad \delta_{D'}\Delta(m) = \delta_D \ .$$

Then $c_F(M) \cong$ colim *$c_F\Delta$.*

Remark: Proposition 1.2.9 above can be made into an "if and only if" statement if we replace the condition on m by saying that any two factorisations are connected by a chain of spans

$$D = D_0 \xleftarrow{m_1} D_1 \xrightarrow{m_2} D_2 \leftarrow \cdots \xrightarrow{m_n} D_n = D'$$

[1]More details about density and this proposition can be found in [76], pages 241–243.

in $\mathbb{D}$ with $p_i : F(P_i) \to \Delta(D_i)$ such that:

$$\begin{aligned} \delta_{D_{i-1}} m_i &= \delta_{D_i} \text{ (for } i \text{ odd)} \\ \delta_{D_i} m_i &= \delta_{D_{i-1}} \text{ (for } i \text{ even)} \\ m_i p_i &= p_{i-1} \text{ (for } i \text{ odd)} \\ m_i P_{i-1} &= p_i \text{ (for } i \text{ even)} . \end{aligned}$$

In the case of a functor $F : \mathbb{C} \to \mathbb{D}$ between small categories, one does not only have an extension $F_! : \widehat{\mathbb{C}} \to \widehat{\mathbb{D}}$ of the functor $\mathrm{y}_{\mathbb{D}}F$ with a right adjoint F^*, but a right adjoint to F^* as well. Because of property of left adjoints, this imply that F^* preserves colimits and hence can be seen as the extension of c_F. Such a triple of adjoint functors, $F_! \dashv F^* \dashv F_*$ between presheaf categories is an example of what is called an *essential geometric morphism* in the topos theoretic jargon. A natural question is whether any essential geometric morphism between presheaf categories arises from some functor between the underlying base categories. The answer is yes, modulo Morita equivalence.

Definition 1.2.10 (Morita equivalence) *Two small categories are said to be* Morita equivalent *if they give rise to equivalent presheaf categories.*

Morita equivalent small categories have equivalent Cauchy completions as well.

Definition 1.2.11 (Cauchy complete categories) *A category $\mathcal{C}$ is said to be Cauchy complete if every idempotent arrow splits, i.e., if for every $e : c \to c$ in $\mathcal{C}$ such that $ee = e$, there exists $p : c \to c'$ and $i : c' \to c$ in $\mathcal{C}$ such that*

$$e = pi \qquad \text{and} \qquad pi = 1_c \ .$$

When this is the case one says that c' is a retract *of c.*

Example 1.2.12

1. **Set** *is Cauchy complete.*
2. *Any presheaf category is Cauchy complete.*
3. *Concerning small categories, it is easy to see that every partial order category is Cauchy complete. Another example is given by the category* $\mathbf{Pom}_L$ *of pomsets over a set of labels L (cf. Chapter 2).*

Proposition 1.2.13 *Every small category, $\mathbb{C}$, can be completed to a Cauchy complete one, $\mathbb{C}^c$, that is still small.*

Proof: We simply indicate where the Cauchy completion of a category $\mathbb{C}$ has to be found.

Given a small category $\mathbb{C}$, consider the full subcategory of $\widehat{\mathbb{C}}$ of retracts of representables. This is the Cauchy completion of $\mathbb{C}$. □

As we said small categories with equivalent Cauchy completion are Morita equivalent.

Proposition 1.2.14 *Every small category $\mathbb{C}$ is Morita equivalent to its Cauchy completion.*

We can now go back to our original problem of characterising essential geometric morphisms between presheaf categories.

Proposition 1.2.15 *Let $\mathbb{C}$ and $\mathbb{D}$ be two categories. There is an equivalence of categories between* $\mathbf{Cat}(\mathbb{C}^c, \mathbb{D}^c)$ *and* $\mathbf{EGeom}(\widehat{\mathbb{C}}, \widehat{\mathbb{D}})$ *where the latter has essential geometric morphisms as objects and natural transformations between the leftmost adjoints as arrows.*

We shall employ this proposition in Chapter 4 and 6. We conclude this section by recalling the *category of elements of a presheaf* construction which accounts also for the density of the Yoneda embedding.

Definition 1.2.16 (Category of elements) *Let $\mathbb{C}$ be a small category. Let $X \in |\widehat{\mathbb{C}}|$. Define the* category of elements *of X, $\mathcal{E}l(X)$ to consists of:*

- **Objects:** *Pairs $\langle c, x\rangle \in |\mathbb{C}| \times X(c)$*
- **Arrows:** *$f : \langle c, x\rangle \to \langle c'x'\rangle$ is an arrow of $\mathcal{E}l(X)$ if $f : c \to c'$ is an arrow of $\mathbb{C}$ and $X(f)(x') = x$.*

There is an obvious projection functor $\pi : \mathcal{E}l(X) \to \mathbb{C}$ that takes any pair to its first component and it is the identity on arrows.

Observe that after the Yoneda lemma, the category of elements of X is equivalent to the category $\mathrm{y}_{\mathbb{C}}/X$ of objects the arrows $x : \mathrm{y}_{\mathbb{C}}(c) \to X$ of $\widehat{\mathbb{C}}$ and arrows, $f : x \to x'$, arrows $f : c \to c'$ of $\mathbb{C}$ such that $x = x'\mathrm{y}_{\mathbb{C}}(f)$ (cf. Proposition 1.2.8).

Proposition 1.2.17 (Density of Yoneda) *Let $\mathbb{C}$ be a small category. Let $X \in |\widehat{\mathbb{C}}|$, then X is naturally isomorphic to*

$$\mathrm{colim}\ (\mathcal{E}l(X) \xrightarrow{\pi} \mathbb{C} \xrightarrow{\mathrm{y}_{\mathbb{C}}} \widehat{\mathbb{C}})\ .$$

1.3 Kan extensions

The situation described in Proposition 1.2.4 about a functor $F_! : \widehat{\mathbb{C}} \to \mathcal{E}$ extending another one $F : \mathbb{C} \to \mathcal{E}$ along a third $\mathrm{y}_{\mathbb{C}}$ is an instance of the more general notion of (left) Kan extension. We will make use of Kan extensions at several places (though mainly when dealing with presheaf categories) so we give here a brief introduction to the notion.

Definition 1.3.1 (Kan Extensions) *If $\mathcal{C} \xleftarrow{G} \mathcal{A} \xrightarrow{F} \mathcal{B}$ is a span of functors, one says that a pair (H, α) is a* left Kan extension *of G along F if*

- *$H : \mathcal{B} \to \mathcal{C}$ is a functor*
- *$\alpha : G \stackrel{\cdot}{\longrightarrow} HF$ is a natural transformation satisfying the following universal property:*
  for every other pair (K, β) with $\beta : G \stackrel{\cdot}{\longrightarrow} KF$ there exists a unique $\gamma : H \stackrel{\cdot}{\longrightarrow} K$ such that $\beta = \gamma_F \cdot \alpha$.

By the usual abuse of language we will often address the functor H as the left Kan extension of G along F and write $\mathrm{Lan}_F(G)$ to indicate it.

Note that the triangle

$$
\begin{tikzcd}
\mathcal{A} \arrow[r, "F"] \arrow[dr, "G"'] & \mathcal{B} \arrow[d, "\mathrm{Lan}_F(G)"] \\
& \mathcal{C}
\end{tikzcd}
$$

need not commute, not even up to natural isomorphism. Still, this happens in many cases of interest.

Proposition 1.3.2 *If F is full and faithful and $(\mathrm{Lan}_F(G), \alpha)$ exists then α is a natural isomorphism.*

If $\mathcal{C}$ is cocomplete and $\mathcal{A}$ is small, then $\mathrm{Lan}_F(G)$ always exists for any F and G and can be computed "pointwise" (see [17], Vol. 1). On objects it is given by the following colimit

$$\mathrm{Lan}_F(G)(b) = \mathrm{colim}\ (\mathcal{E}l(\mathcal{B}(F(-), b)) \xrightarrow{\pi} \mathcal{A} \xrightarrow{G} \mathcal{C})\ .$$

We will return to this in Chapter 4.

Example 1.3.3 *In the previous section we saw that any functor F from a small category $\mathbb{C}$ to a cocomplete category $\mathcal{E}$ could be extended to a colimit preserving functor, $G : \widehat{\mathbb{C}} \to \mathcal{E}$. The functor G is the left Kan extension of F along the Yoneda embedding $\mathrm{y}_{\mathbb{C}}$.*

If $F : \mathbb{C} \to \mathbb{D}$, then $F_! = \mathrm{Lan}_{\mathrm{y}_{\mathbb{C}}}(\mathrm{y}_{\mathbb{D}} F)$ and $F^* = \mathrm{Lan}_{\mathrm{y}_{\mathbb{D}}}(c_F) = \mathrm{Lan}_{\mathrm{y}_{\mathbb{D}}}(F^* \mathrm{y}_{\mathbb{D}})$.

Left Kan extensions compose:

Proposition 1.3.4 *If (H, α) is the left Kan extension of G along F for $\mathcal{C} \xleftarrow{G} \mathcal{A} \xrightarrow{F} \mathcal{B}$ and (K, β) is the left Kan extension of H along F', for $\mathcal{C} \xleftarrow{H} \mathcal{B} \xrightarrow{F'} \mathcal{D}$, then $(K, \beta_F \cdot \alpha)$ is the left Kan extension of G along $F'F$:*

$$\mathrm{Lan}_{F'F}(G) = \mathrm{Lan}_{F'}(\mathrm{Lan}_F(G))\ .$$

For sake of completeness we must mention the existence of the dual notion of *right Kan extension*, whose definition is like Definition 1.3.1 where the directions of all natural transformations involved are reversed. We will not deal particularly with right Kan extensions in this thesis.

1.4 Fibrations

Indexing structures play a fundamental role in the categorical analysis of models for concurrency [141, 136]. In the context of categorical models for **CCS**-like languages we will consider presheaf categories indexed by a category of labelling sets. There is a tight correspondence between indexed categories and fibrations, the former represent the class of fibrations for which a definite (coherent) choice of a cleavage has been made. We introduce therefore in this section the basic terminology of fibred category theory together with pointers to the related notion of *elementary existential doctrine* [71] of which the presheaf models of Chapter 3 will be an example.

Definition 1.4.1 (Cartesian arrows) *Let $\pi : \mathcal{E} \to \mathcal{B}$ be a functor. An arrow in $\mathcal{E}$, $f : e' \to e$ is* cartesian *(with respect to π) if for every other arrow $g : e'' \to e$ such that $\pi(g) = \beta\alpha$ with $\beta = \pi(f)$, there exists a unique $h : e'' \to e'$ with $g = fh$ and $\pi(h) = \alpha$:*

$$\begin{tikzcd} e'' \arrow[dr, dotted, "\exists! h"'] \arrow[drr, bend left, "g"] & & \\ & e' \arrow[r, "f"'] & e \end{tikzcd}$$

$$\pi(e'') \xrightarrow{\alpha} \pi(e') \xrightarrow{\beta} \pi(e) \qquad .$$

Definition 1.4.2 (Fibrations) *A functor $\pi : \mathcal{E} \to \mathcal{B}$ is a* fibration *if for every $\beta : b' \to b$ in $\mathcal{B}$ and $e \in |\mathcal{E}|$ such that $\pi(e) = b$, there exists a cartesian arrow, f, of codomain e such that $\pi(f) = \beta$. The arrow f is called a* cartesian lifting of e with respect to β.

Definition 1.4.3 *If $\pi : \mathcal{E} \to \mathcal{B}$ is a functor, an arrow $f : e' \to e$ of $\mathcal{E}$ is said to be* vertical *if $\pi(f) = 1_e$, and is said to be* horizontal, *otherwise.*

Definition 1.4.4 *If $\pi : \mathcal{E} \to \mathcal{B}$ is a functor and b an object of $\mathcal{B}$. Define the fibre over b (with respect to π) to be the subcategory $\mathcal{E}_b$ of $\mathcal{E}$ of those objects e and arrows f such that $\pi(e) = b$ and $\pi(f) = id_b$.*

Example 1.4.5 *For any presheaf X in a presheaf category $\widehat{\mathbb{C}}$, the projection functor $\mathcal{E}l(X) \xrightarrow{\pi} \mathbb{C}$ is a fibration. In fact is a so-called* discrete *fibration, since for any object $c \in |\mathbb{C}|$ the fibre over c is a discrete category, i.e., a category whose only arrows are the identity ones.*

If π is a fibration then a choice of cartesian arrows induces *cartesian lifting functors* between the fibres:

Proposition 1.4.6 (Cartesian lifting functors) *Let $\pi : \mathcal{E} \to \mathcal{B}$ a fibration. Let $\beta : b' \to b$ be an arrow in $\mathcal{B}$. For every object $e \in |\mathcal{E}_b|$ let $\beta^*_e : \beta^*(e) \to e$ be a chosen cartesian lifting of e with respect to β. This choice induces the following cartesian lifting functors $\beta^* : \mathcal{E}_b \to \mathcal{E}_{b'}$:*

- **On objects** $e \mapsto \beta^*(e)$ *as chosen above*
- **On arrows** $(f : \overline{e} \to e) \mapsto \beta^*(f)$ *that is defined to be the unique arrow such that the following square commutes:*

$$\begin{tikzcd} \beta^*(\overline{e}) \arrow[r, "\beta^*_{\overline{e}}"] \arrow[d, "\beta^*(f)"'] & \overline{e} \arrow[d, "f"] \\ \beta^*(e) \arrow[r, "\beta^*_e"'] & e. \end{tikzcd}$$

A choice of cartesian arrows for a fibration is called a *cleavage* and a fibration with a chosen cleavage is called a *cloven* fibration. If the choice of the cleavage is functorial, i.e., $(1_b)^*_e = 1_e$ and for $b'' \xrightarrow{\alpha} b' \xrightarrow{\beta} b$, $(\beta\alpha)^*_e = \alpha^*_{\beta^*_e}$, then the fibration is said to be *split*.

We will make extensive use of the dual notion of cofibration:

Definition 1.4.7 (Cofibrations and Bifibrations) *A functor $\pi : \mathcal{E} \to \mathcal{B}$ is a* cofibration *if the dual functor $\pi^{\mathrm{op}} : \mathcal{E}^{\mathrm{op}} \to \mathcal{B}^{\mathrm{op}}$ is a fibration. A functor that is both a fibration and a cofibration is called a* bifibration.

Dually one talks of cocartesian arrows, cocartesian liftings and cocartesian lifting functors.

1.4.1 The Grothendieck construction

Cloven fibrations are equivalent to indexed categories.[2] In fact any indexed category gives rise (via a construction due to Grothendieck [48]) to a cloven fibration and vice versa any cloven fibration induces an indexed category.

Definition 1.4.8 (Indexed categories) *Let $\mathcal{B}$ be any category. A $\mathcal{B}$-indexed category in* **CAT** *is given by a pseudo-functor $F : \mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$, that is F associates to each object of b, a category $F(b)$, to any arrow $\beta : b' \to b$ a functor $\beta^* : F(b) \to F(b')$ with natural isomorphisms, $\varphi_b : 1_{F(b)} \xrightarrow{\simeq} (1_b)^*$ and $\varphi_{(\alpha,\beta)} : \alpha^*\beta^* \xrightarrow{\simeq} (\beta\alpha)^*$ for any $b \in \mid \mathcal{B} \mid$ and for any two arrows $b'' \xrightarrow{\alpha} b' \xrightarrow{\beta} b$ of $\mathcal{B}$ satisfying the coherence conditions given by commutativity of the following diagrams:*

$$\begin{tikzcd}
\alpha^*(1_b)^* \arrow[dr, "\varphi_{\alpha,1_b}"'] & \alpha^* \arrow[l, "\alpha^*\varphi_b"'] \arrow[r, "(\varphi_b)_{\alpha^*}"] \arrow[d, "1_{\alpha^*}"] & (1_{b'})^*\alpha^* \arrow[dl, "\varphi_{1_{b'},\alpha}"] \\
& \alpha^* &
\end{tikzcd}$$

$$\begin{tikzcd}
\alpha^*\beta^*\gamma^* \arrow[r, "(\varphi_{\alpha,\beta})_{\gamma^*}"] \arrow[d, "\alpha^*\varphi_{\beta,\gamma}"'] & (\beta\alpha)^*\gamma^* \arrow[d, "\varphi_{\beta\alpha,\gamma}"] \\
\alpha^*(\gamma\beta)^* \arrow[r, "\varphi_{\alpha,\gamma\beta}"'] & (\gamma\beta\alpha)^*
\end{tikzcd} ,$$

with γ being another arrow of $\mathcal{B}$, $\gamma : b \to b'''$. [3]

Definition/Proposition 1.4.9 (Grothendieck construction) *Given a $\mathcal{B}$-indexed category $F : \mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$, define the following category $Groth(F)$:*

- **Objects:** *Pairs $\langle c, b\rangle$, with $c \in \mid F(b) \mid$ and $b \in \mid \mathcal{B} \mid$.*
- **Arrows:** *A pair $\langle f, \beta\rangle$ is an arrow from $\langle c', b'\rangle$ to $\langle c, b\rangle$ if $\beta : b' \to b$ is an arrow in $\mathcal{B}$ and $f : c' \to F(\beta)(c)$ is an arrow in $F(b')$. If $\langle f, \alpha\rangle : \langle c'', b''\rangle \to \langle c', b'\rangle$*

[2] See for example [52] for a precise account of this statement.

[3] Later in this chapter, we will summarise some basic notions of 2-category theory. This will include a general definition of pseudo-functor between 2-categories.

and $\langle g, \beta \rangle : \langle c', b' \rangle \to \langle c, b \rangle$ *then their composite is the pair* $\langle h, \beta\alpha \rangle$ *where* h *is the following arrow*

$$c \xrightarrow{f} F(\alpha)c' \xrightarrow{F(\alpha)g} F(\alpha)F(\beta)c'' \xrightarrow{\varphi_{\alpha,\beta}} F(\beta\alpha)c'' \ .$$

The coherence conditions of Definition 1.4.8 ensure associativity of composition.

The obvious projection $\pi : Groth(F) \to \mathcal{B}$ *that projects any pair onto its the second component is a fibration. A cartesian lifting for* $\langle c, b \rangle$ *with respect to* $\beta : b' \to b$ *is given by the pair* $\langle 1_{F(\beta)c}, \beta \rangle$.

The category of elements construction of Definition 1.2.16 is an example of application of the Grothendieck construction (cf. Example 1.4.5).

Our main example of a bifibration will be given by a Lawvere's elementary existential doctrine [71] whose categories of attributes will be presheaf categories. We will naturally consider the following two conditions on top of our fibrations.

Definition 1.4.10 *Let* $\mathcal{P} : \mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$ *be a pseudo-functor. If* $\beta : b' \to b$ *is an arrow in* $\mathcal{B}$*, we write* β^* *for* $\mathcal{P}(\beta)$*. Suppose that for any* β*,* β^* *has a left adjoint* $\beta_!$.

- **Beck-Chevalley Condition:** *Say that* $\mathcal{P}$ *satisfy the* Beck-Chevalley condition *if for every pullback square in* $\mathcal{B}$

$$\begin{tikzcd} b''' \arrow[r, "\overline{\beta}"] \arrow[d, tail, "\overline{\alpha}"'] & b' \arrow[d, tail, "\alpha"] \\ b'' \arrow[r, "\beta"'] & b \end{tikzcd}$$

with α *monic, the following square commutes up to a natural isomorphism:*

$$\begin{tikzcd} \mathcal{P}(b''') \arrow[d, "\overline{\alpha}_!"'] & \mathcal{P}(b') \arrow[l, "\overline{\beta}^*"'] \arrow[d, "\alpha_!"] \\ \mathcal{P}(b'') & \mathcal{P}(b) \arrow[l, "\beta^*"] \end{tikzcd} \ .$$

- **Fröbenius Reciprocity Law:** *Suppose now that for every* $b \in \mid \mathcal{B} \mid$*,* $\mathcal{P}(b)$ *has binary products. Say that* $\mathcal{P}$ *satisfies the* Fröbenius Reciprocity Law *if for every* $\beta : b' \to b$ *and* $c \in \mid \mathcal{P}(b) \mid$ *the following square commutes up to a natural isomorphism:*

$$\begin{tikzcd} b' \arrow[d, "\beta"'] \\ b \end{tikzcd} \qquad \begin{tikzcd} \mathcal{P}(b') \arrow[d, "\beta_!"'] & \mathcal{P}(b') \arrow[l, "\beta^*(c)\times -"'] \arrow[d, "\beta_!"] \\ \mathcal{P}(b) & \mathcal{P}(b) \arrow[l, "c\times -"] \end{tikzcd} \ .$$

We end up this section by listing a few facts about fibrations and indexed categories that we shall use later.

Proposition 1.4.11 *Let $\mathcal{P} : \mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$ be a pseudo-functor satisfying the Beck-Chevalley condition, then for any monic arrow $\beta : b' \rightarrowtail b$,*

$$\beta^* \beta_! \cong 1_{\mathcal{P}(b')} \ .$$

If $\mathcal{P}$ satisfies also the Fröbenius reciprocity law, then for any monic arrow $\beta : b' \rightarrowtail b$, the functor $\beta_!$ preserves products.

Proof: For the first statement, observe that if β is a mono, then the following diagram is a pullback

$$\begin{tikzcd} b' \arrow[r, "1"] \arrow[d, "1"'] & b' \arrow[d, "\beta"] \\ b' \arrow[r, "\beta"'] & b \ . \end{tikzcd}$$

Because of the Beck-Chevalley condition, $\beta^*\beta_! \cong 1_! 1^*$, but both $1_!$ and 1^* are naturally isomorphic to $1_{\mathcal{P}b'}$, hence $\beta^*\beta_! \cong 1_{\mathcal{P}b'}$. For the second one, let c', d' be two objects of $\mathcal{P}(b')$, then

$$\begin{array}{rcll} \beta_!(c' \times d') & \cong & \beta_!(\beta^*\beta_! c' \times d') & \text{(from the property above)} \\ & \cong & \beta_! c' \times \beta_! d' & \text{(by Fröbenius reciprocity law)} \end{array}$$

□

We can use the result above to prove Proposition 1.4.13 below that will be used in Chapter 3 to prove the congruence of bisimulation with respect to product. We first need a lemma.

Lemma 1.4.12 *Let $\mathcal{P} : \mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$ be a pseudo-functor, satisfying both the Beck-Chevalley condition and the Fröbenius Reciprocity law. Let the following square be a pullback of monomorphisms is $\mathcal{B}$:*

$$\begin{tikzcd} a \arrow[r, tail, "\beta"] \arrow[d, tail, "\alpha"'] & b \arrow[d, tail, "\gamma"] \\ c \arrow[r, tail, "\delta"'] & d \ . \end{tikzcd}$$

If X and Y are two objects of $\mathcal{P}(d)$ such that $\delta_!\delta^* X \cong X$ and $\gamma_!\gamma^* Y \cong Y$, then

$$X \times Y \cong \delta_!\alpha_!\alpha^*\delta^*(X \times Y) \ .$$

Proof:

$$\begin{array}{rcll} X \times Y & \cong & \delta_!\delta^* X \times Y & \text{(by hypothesis)} \\ & \cong & \delta_!(\delta^* X \times \delta^* Y) & \text{(by Fröbenius)} \\ & \cong & \delta_!\delta^* X \times \delta_!\delta^* Y & \text{(by Proposition 1.4.11)} \\ & \cong & X \times \delta_!\delta^* Y & \text{(by hypothesis)} \\ & \cong & X \times \delta_!\delta^*\gamma_!\gamma^* Y & \text{(by hypothesis)} \\ & \cong & X \times \delta_!\alpha_!\beta^*\gamma^* Y & \text{(by Beck-Chevalley)} \\ & \cong & X \times \delta_!\alpha_!\alpha^*\delta^* Y & \text{(since } \delta\alpha \cong \gamma\beta) \\ & \cong & \delta_!\alpha_!(\alpha^*\delta^* X \times \alpha^*\delta^* Y & \text{(by Fröbenius)} \\ & \cong & \delta_!\alpha_!\alpha^*\delta^*(X \times Y) & \text{(since } \alpha^* \text{ and } \delta^* \text{ are right adjoints).} \end{array}$$

□

Proposition 1.4.13 *Let $\mathcal{P} : \mathcal{B}^{\mathrm{op}} \to \mathbf{CAT}$ be a pseudo-functor, satisfying both the Beck-Chevalley condition and the Fröbenius Reciprocity law. If a diagram*

$$\begin{tikzcd} a & c \arrow[l, "\pi_a"'] \arrow[r, "\pi_b"] & b \\ d \arrow[u, hook, "i"] & f \arrow[l, "l"] \arrow[r, "r"'] \arrow[u, hook, "k"'] & e \arrow[u, hook, "j"'] \end{tikzcd}$$

is a limiting cone in $\mathcal{B}$, then for any object $X \in\mid \mathcal{P}(d) \mid$ and $Y \in\mid \mathcal{P}(e) \mid$, there is an isomorphism in $\mathcal{P}(c)$,

$$k_!(l^*X \times r^*Y) \cong \pi_a^* i_! X \times \pi_b^* j_! Y \ .$$

Proof: Observe first of all that the limit of the diagram

$$\begin{tikzcd} a & c \arrow[l, "\pi_a"'] \arrow[r, "\pi_b"] & b \\ d \arrow[u, hook, "i"] & & e \arrow[u, hook, "j"'] \end{tikzcd}$$

is obtained by taking three pullbacks, i.e., the limiting cone can be constructed as follows,

$$\begin{tikzcd} a & & c \arrow[ll, "\pi_a"'] \arrow[rr, "\pi_b"] & & b \\ & \cdot \arrow[ur, hook, "\delta"] \arrow[dl, "\pi_d"] & & \cdot \arrow[ul, hook, "\gamma"'] \arrow[dr, "\pi_e"'] & \\ d \arrow[uu, hook, "i"] & & & & e \arrow[uu, hook, "j"'] \\ & & f \arrow[uul, hook, "\alpha"] \arrow[uur, hook, "\beta"'] & & \end{tikzcd}$$

where all the quadrilaterals in the diagram above are pullbacks. Without loss of generality we can then assume $l = \pi_d\alpha$, $r = \pi_l\beta$ and $k = \delta\alpha = \beta\gamma$. We deduce the following natural isomorphism:

$$\begin{array}{rcll} \delta_!\delta^*\pi_a^* i_! & \cong & \delta_!\delta^*\delta_!\pi_d^* & \text{(by Beck-Chevalley)} \\ & \cong & \delta_!\pi_d^* & \text{(by Proposition 1.4.11)} \\ & \cong & \pi_a^* i_! & \text{(by Beck-Chevalley).} \end{array}$$

Similarly one deduces that $\gamma_! \gamma^* \pi_b^* j_! \cong \pi_b^* j_!$. Hence,

$$\begin{array}{rcll} \pi_a^* i_! X \times \pi_b^* j_! Y & \cong & \delta_! \alpha_! \alpha^* \delta^* (\pi_a^* i_! X \times \pi_b^* j_! Y) & \text{(by Lemma 1.4.12)} \\ & \cong & k_! (\alpha^* \delta^* \pi_a^* i_! X \times \alpha^* \delta^* \pi_b^* j_! Y) & \\ & \cong & k_! (\alpha^* \delta^* \delta_! \pi_d^* X \times \alpha^* \alpha_! \beta^* \pi_e^* Y) & \text{(by Beck-Chevalley)} \\ & \cong & k_! (\alpha^* \pi_d^* X \times \beta^* \pi_e^* Y) & \text{(by Proposition 1.4.11)} \\ & \cong & k_! (l^* X \times r^* Y)\ . & \end{array}$$

□

Proposition 1.4.14 *Let $\pi : \mathcal{E} \to \mathcal{B}$ be a fibration (cofibration). Let Δ be a class of diagram shapes (i.e., a class of categories). Suppose that for every object b of $\mathcal{B}$, the fibre $\mathcal{E}_b$ has limits (colimits) of diagrams of shape δ for every $\delta \in \Delta$ and suppose that $\mathcal{B}$ has limits (colimits) of diagrams of shape δ for every $\delta \in \Delta$ too. Then $\mathcal{E}$ has limits (colimits) of diagrams of shape δ for every $\delta \in \Delta$.*

Proof: We simply give the description of how to build a limiting cone in $\mathcal{E}$ for a diagram of shape $\delta \in \Delta$. Let then $F : \delta \to \mathcal{E}$ be a functor. Consider $\pi F : \delta \to \mathcal{B}$. By assumption there exists a limiting cone for πF. Let $b = \lim \pi F$ and for any $d \in |\delta|$, let $\beta_d : \pi F(d) \to b$ be the corresponding edge of the cone. Let $(\overline{\beta_d^*} : \beta_d^*(F(d)) \to F(d))_{d \in |\delta|}$ be a family of cartesian arrows. This family induces a functor, $\beta_{(-)}^* F : \delta \to \mathcal{E}_b$. By assumption there exists a limiting cone to such functor. Let $(e, f_d : e \to \beta_d^* F(d))$ be such a cone, then by post-composing with the corresponding cartesian arrows one obtains a limiting cone $(e, \overline{\beta_d^*} f_d : e \to F(d))$ in $\mathcal{E}$. □

1.5 Pseudo concepts

As we mentioned at the beginning of this chapter, 2-categories and bicategories will play sometimes a role in this thesis, especially in Chapter 4 and 6. The space of this chapter is too small for any reasonably full introduction to the concepts that we shall need so we avoid any attempt altogether. Relevant bibliographic references are supplied in Section 1.6 below. A rather small set of definitions is reported in Appendix A and is meant essentially to fix some notation and terminology that we shall use throughout the thesis. Roughly a 2-category is a category, $\mathcal{K}$, in which the collection of arrows, $\mathcal{K}(A, B)$, between any two objects, A and B organises as a category as well, whose arrows are called 2-cells. Typical examples are locally ordered categories like **Rel**, the category of sets and relations, where the relations between two sets are ordered by the inclusion order, or **Cpo**, the category of cpo's and continuous functions that are ordered with the pointwise ordering. A non locally ordered paradigmatic example is given by **Cat**, the 2-category of small categories, functors and natural transformations.

The presence of this extra structure allows one to reproduce in a general 2-category, essentially all categorical concepts, e.g., adjoint pairs, equivalences, ... In particular, now, between objects one can have two "different" notion of equality (as it is between categories), there is the usual isomorphism but, using the 2-cells, one can also consider the less strict notion of equivalence. Similarly limits can now be taken "up to isomorphism" or "up to equivalence" and for categorical properties the latter seems to be the right notion. We shall then be interested in what we call *pseudo limits* and dually *pseudo colimits*. As an illustrative example of this change of perspective, we give here explicitly the definition of pseudo-initial object.

Definition 1.5.1 (Pseudo-initial object) *An object* 0 *of a* 2*-category* $\mathcal{K}$ *is* pseudo-initial *if for every object* k*,* $\mathcal{K}(0,k)$ *is equivalent to the category* $\mathbf{1}$ *with only one object and one morphism. In other words* 0 *is initial if for every object* k*, there exists an arrow* $0_k : 0 \to k$ *and for every pair of arrows* $f, g : 0 \to k$ *there exists a unique* 2*-cell,* $\alpha : f \Longrightarrow g$*.*

Pseudo-limits, just like ordinary limits can be given a definition in terms of representability of objects [56] where now the "functors" do not range over **Set** but over **CAT**. We shall avoid such presentation and, just like above, always spell out in elementary terms the conditions for a (suitably enriched) cone to be a pseudo limiting one.

For the knowledgeable reader we notice here a small clash of terminology between our use of the *pseudo* prefix and what one finds often in the literature. Beside *pseudo* functors, we shall talk of *pseudo* natural transformations, *pseudo* limits and colimits where elsewhere, notably [127], these are called, *strong* transformations and *bi*limits or *bicategorical* limits and colimits, while the prefix pseudo is used for some stricter notion.

1.6 Some references

Everything we said about presheaf categories and Kan extensions can be found in [76, 17, 77]. Any other notion of category theory that we have not introduced here but that we shall make occasional use of in the remainder of this thesis can be found in [76]. Despite the large number of paper involving fibrations there is, to our knowledge, no text-book available. A valuable introduction is given by Paul Taylor's notes of a course by Peter Johnstone [61]. A few recent PhD theses [57, 52, 97] contains reasonable introductions that expand what we said here. For a discussion on the relevance of fibred category theory to category theory and mathematics one can look at [11]. The Fröbenius Reciprocity law and Beck-Chevalley condition are introduced, for instance, in [71, 77]. One has to note that sometimes Beck-Chevalley is given with respect to all pullback diagrams in the base categories and some other times (as for us) only with respect to pullbacks of monomorphisms. Kelly's book [65] provides an excellent introduction to enriched categories, while Kelly and Street paper [67] and Gray's monography [47] deal with many 2-, pseudo- and bi-categorical concepts. Bicategories were introduced by Bénabou [9]. Street's paper [127] contains definitions and results about pseudo-limits (bilimits in that paper's jargon) in bicategories. Borceux's monography [17] covers some material about enriched concepts as well.

Chapter 2

Open Map Bisimulation

In this chapter we will mainly summarise the definitions and results of [64] where the notion of open map bisimulation was defined. We refer to that paper and to the relevant part of the introduction to this thesis for a more detailed discussion about the rationale behind this abstract notion of bisimulation. We will also use this chapter to recall the definitions of some basic categories of models for concurrency that have been extensively studied over the past years [141, 119]. These will include, categories of labelled transition systems, synchronisation trees and event structures.

2.1 Traditional models

A *transition system* is a structure

$$(S, i, L, \mathit{tran})$$

where

- S is a set of *states* with *initial state* i,
- L is a set of *labels*,
- $\mathit{tran} \subseteq S \times L \times S$ is the *transition relation*. Usually, a transition (s, a, s') is written as $s \xrightarrow{a} s'$.

Let

$$T_0 = (S_0, i_0, L_0, \mathit{tran}_0) \text{ and } T_1 = (S_1, i_1, L_1, \mathit{tran}_1)$$

be transition systems. A *morphism* $f : T_0 \to T_1$ is a pair $f = (\sigma, \lambda)$ where

- $\sigma : S_0 \to S_1$, such that $\sigma(i_0) = i_1$, and
- $\lambda : L_0 \rightharpoonup L_1$, a partial function, which together satisfy

$$\begin{aligned}
&(s, a, s') \in \mathit{tran}_0 \ \&\ \lambda(a) \text{ defined} \\
&\qquad\qquad \Rightarrow (\sigma(s), \lambda(a), \sigma(s')) \in \mathit{tran}_1, \text{ and} \\
&(s, a, s') \in \mathit{tran}_0 \ \&\ \lambda(a) \text{ undefined} \Rightarrow \sigma(s) = \sigma(s').
\end{aligned}$$

A *synchronisation tree* is a transition system whose transition graph has the form of a tree with root the initial state.

Definition 2.1.1 (The Categories $\mathcal{TS}$ and $\mathcal{ST}$) *Define $\mathcal{TS}$ to be the category of objects transition systems and arrows transition systems morphisms. The composition of arrows is defined componentwise.*

Define $\mathcal{ST}$ to be the full subcategory of $\mathcal{TS}$ of Synchronisation Trees.

Transition systems and synchronisation trees are often called "interleaving models" because they represent parallel/concurrent composition by nondeterministically interleaving the actions of processes. In contrast, event structures represent a class of "independence models" (among them Petri nets) in which concurrency is represented directly as a form of causal independence. Define a *(labelled) event structure* to be a structure $(E, \leq, Con, l)$ consisting of a set E, of *events* which are partially ordered by $\leq$, the *causal dependency relation*, a *consistency relation Con* consisting of finite subsets of events, and a *labelling function* $l : E \to L$, which satisfy

$$\begin{aligned}
&\{e' \mid e' \leq e\} \text{ is finite,}\\
&\{e\} \in Con,\\
&Y \subseteq X \in Con \Rightarrow Y \in Con,\\
&X \in Con \;\&\; e \leq e' \in X \Rightarrow X \cup \{e\} \in Con,
\end{aligned}$$

for all events e, e' and their subsets X, Y.

Two events $e, e' \in E$ are said to be *concurrent* (causally independent) iff

$$(e \not\leq e' \;\&\; e' \not\leq e \;\&\; \{e, e'\} \in Con).$$

A set, x, of events in E is said to be a *configuration* if it is

downwards-closed: $\forall e, e'.\ e' \leq e \in x \Rightarrow e' \in x$, and
consistent: $\forall X.\ X$ finite & $X \subseteq x \Rightarrow X \in Con$.

A *morphism* of event structures consists of

$$(\eta, \lambda) : E \to E',$$

where $E = (E, \leq, Con, l), E' = (E', \leq', Con', l')$ are event structures, $\eta : E \rightharpoonup E'$ is a partial function on events, $\lambda : L \rightharpoonup L'$ is a partial function on labelling sets such that

(i) $l' \circ \eta = \lambda \circ l$,

(ii) If x is a configuration of E, then ηx is a configuration of E' and if for $e_1, e_2 \in x$ their images are both defined with $\eta(e_1) = \eta(e_2)$, then $e_1 = e_2$.

Definition 2.1.2 (The Category of Event Structures) *Define $\mathcal{ES}$ to be the category of objects event structures and arrows event structures morphisms. The composition of arrows is defined componentwise.*

The definition of morphism on event structures is given rather abruptly—see [141] for motivation. The categories $\mathcal{TS}, \mathcal{ST}$ and $\mathcal{ES}$ are related by coreflections: the inclusion functor $\mathcal{ST} \hookrightarrow \mathcal{TS}$ has a right adjoint unfolding transition systems to trees; the functor $\mathcal{ST} \to \mathcal{ES}$ identifying a synchronisation tree with an event structure has a right adjoint serialising an event structure to a synchronisation tree.

Inspired by the analysis of [141] we will axiomatise, in Chapter 3, the notion of a categorical model for a general **CCS**-like language. The role played by adjunctions in relating the semantics given in different categories of models will become apparent at that point.

As we saw in the definitions above, every morphism between any two objects of the defined categories consists of two components, the second being a partial (relabelling) function. It is immediately clear the existence of forgetful functors

$$p_{\mathcal{M}} : \mathcal{M} \to \mathbf{Set}_* \ ,$$

for $\mathcal{M} \in \{\mathcal{ST}, \mathcal{ES}, \mathcal{TS}\}$ and $\mathbf{Set}_*$ the category of sets and partial functions.

Proposition 2.1.3 (Implicit in [141]) *The functors $p_{\mathcal{TS}} : \mathcal{TS} \to \mathbf{Set}_*$ and $p_{\mathcal{ST}} : \mathcal{ST} \to \mathbf{Set}_*$ are bifibrations. The functor $p_{\mathcal{ES}} : \mathcal{ES} \to \mathbf{Set}_*$ is a cofibration but there exist cartesian liftings of all monomorphisms.*

As described in [141] and as we will see later in Chapter 3, considering the categories of models as fibred over $\mathbf{Set}_*$ is a crucial step for an abstract description of the operations associated to relabelling and restriction in the semantics of **CCS**-like languages.

2.2 Bisimulation from open maps

We now move to describing the characterisation of bisimulation relations via open maps. We need to have in our model an idea of computation paths. For instance a (computation) path of a transition system with labelling set L is reasonably taken to be a finite sequence of transitions that the transition system can perform. It takes the shape of a string of labels in L.

Definition 2.2.1 (Finite strings as a partial order category) *Let L be a set. Define L^* to be the partial order category of objects the finite strings over L with the order relation given by saying that a string s is below another string s' if s is an initial prefix of s'. It is convenient to identify strings L^* with the equivalent subcategory of $\mathcal{ST}$ consisting of those special synchronisation trees consisting of a finite single branch.*

To take account of the added independence structure of event structures, the shape of their computation paths is taken to be a finite *pomset* [111].

Definition 2.2.2 (Labelled pomsets) *Let L be a set. The category $\mathbf{Pom}_L$ is taken to be the subcategory of $\mathcal{ES}_L$, for a labelling set L, consisting of those finite event structures in which all subsets of events are in the consistency relation. In other words the objects P of $\mathbf{Pom}_L$ are triples $P = (P, \leq, l)$ where P is a finite set, $\leq$ is a partial order on P and $l : P \to L$ is a function. A morphism $f : P \to Q$ in $\mathbf{Pom}_L$ is given by a injective* *function that preserve the labelling and send downward closed sets of P to downward closed sets of Q.*

We can obtain a general definition of bisimulation from *open maps*, which roughly speaking are morphisms with the property that any extension of a computation path in the range can be matched by an extension in its domain.

Definition 2.2.3 ($\mathbb{P}$-open maps) *Assume a category of models $\mathcal{M}$ and a choice of path category, a subcategory $\mathbb{P} \hookrightarrow \mathcal{M}$ consisting of path objects together with morphisms expressing how they can be extended. Let $f : M \to M'$ be an arrow in $\mathcal{M}$. We say that f is a* $\mathbb{P}$-open map *if, whenever, for $m : P \to Q$ a morphism in $\mathbb{P}$, a "square"*

$$\begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & M \arrow[d, "f"] \\ Q \arrow[r, "q"] & M' \end{tikzcd}$$

in $\mathcal{M}$ commutes, i.e. $q \circ m = f \circ p$, meaning the path $f \circ p$ in M' can be extended via m to a path q in M', then there is a (not necessarily unique) morphism p' such that in the diagram

$$\begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & M \arrow[d, "f"] \\ Q \arrow[r, "q"] \arrow[ur, "p'"] & M' \end{tikzcd}$$

the two "triangles" commute, i.e. $p' \circ m = p$ and $f \circ p' = q$, meaning the path p can be extended via m to a path p' in M which matches q.

Two objects M_1, M_2 of $\mathcal{M}$ are said to be $\mathbb{P}$-bisimilar *iff there is a* span *of $\mathbb{P}$-open morphisms f_1, f_2:*

$$\begin{tikzcd} & M \arrow[dl, "f_1"'] \arrow[dr, "f_2"] & \\ M_1 & & M_2 \end{tikzcd} .$$

The following is an immediate consequence of the definition.

Proposition 2.2.4 *Given any category $\mathcal{M}$ and a subcategory $\mathbb{P}$; every isomorphism of $\mathcal{M}$ is $\mathbb{P}$-open. Hence any two isomorphic objects of $\mathcal{M}$ are $\mathbb{P}$-open bisimilar.*

In the case of traditional models we obtain known equivalences. In $\mathcal{ST}_L$, L^*-bisimulation coincides with Park and Milner's strong bisimulation; for event structures $\mathcal{ES}_L$, $\mathbf{Pom}_L$ bisimulation coincides with strong history-preserving bisimulation due to Bednarczyk refining ideas of van Glabbeek and Goltz, Rabinovitch and Traktenbrot [8, 41, 113].

Theorem 2.2.5 (Theorem 2 and Theorem 10(i) of [64])

- *Two transition systems (and so synchronisation trees) over the labelling set L are L^*-bisimilar iff they are strongly bisimilar in the sense of [82]*
- *Two event structures with labelling sets L are $\mathbf{Pom}_L$-open bisimilar if and only if they are strong history-preserving bisimilar (as defined in [64], Section 3).*

Observe that the given definition of open map bisimulation does not necessarily induce that 'being open map bisimilar' is an equivalence relation. This is because we did not make any assumption on the category of models $\mathcal{M}$. In particular any category, in order to support composition of relations, is normally required to have, at least, pullbacks. In fact given a category $\mathcal{M}$ with pullbacks it is possible to define the bicategory of spans (generalised relations) in $\mathcal{M}$, $Spn(\mathcal{M})$ to consists of:

- **Objects:** The same objects of $\mathcal{M}$,
- **Arrows:** Spans of arrows of $\mathcal{M}$, i.e., an arrow from M_1 to M_2 is given by a span $M_1 \xleftarrow{f} M \xrightarrow{g} M_2$.
- **2-Cells:** Given a span $M_1 \xleftarrow{f_1} M \xrightarrow{f_2} M_2$ and a span $M_1 \xleftarrow{g_1} N \xrightarrow{g_2} M_2$, a 2-cell from the first one to the second one is given by an arrow in $\mathcal{M}$, $h : M \to N$ such that $hg_i = f_i$, for $i = 1, 2$.

Both the composition of arrows and the horizontal composition of 2-cells rely on pullbacks to exist in order to be defined. Spans of open maps represent particular relations between the models, viz. bisimulation relations. In the case that they can be composed i.e., if $\mathcal{M}$ has pullbacks then being open map bisimilar is an equivalence relation on the objects of the category $\mathcal{M}$.

Proposition 2.2.6 (Proposition 3 of [64]) *Pullbacks of $\mathbb{P}$-open morphisms are $\mathbb{P}$-open. If $f : M \to N$ is a $\mathbb{P}$-open map and*

$$\begin{tikzcd} M \times_N O \arrow[r, "\overline{g}"] \arrow[d, "\overline{f}"'] & M \arrow[d, "f"] \\ O \arrow[r, "g"'] & N \end{tikzcd}$$

is a pullback square, then $\overline{f}$ is $\mathbb{P}$-open too.

If $\mathcal{M}$ has products, the product of $\mathbb{P}$-open morphisms is $\mathbb{P}$-open.

Proposition 2.2.7 (Joyal-Moerdijk) *Suppose $\mathcal{M}$ has products. Let $f_1 : M_1 \to N_1$ and $f_2 : M_2 \to N_2$ be $\mathbb{P}$-open maps. Then $f_1 \times f_2 : M_1 \times M_2 \to N_1 \times N_2$ is $\mathbb{P}$-open.*

Proof: Suppose that the following square, with $m : P \to Q$ in $\mathbb{P}$, commutes:

$$\begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & M_1 \times M_2 \arrow[d, "f_1 \times f_2"] \\ Q \arrow[r, "q"'] & N_1 \times N_2 \end{tikzcd} .$$

We need to find an $r : Q \to M_1 \times M_2$ such that $rm = p$ and $(f_1 \times f_2)r = q$. By universality of the product, any $r : Q \to M_1 \times M_2$ is uniquely determined by two arrows

$r_1 : Q \to M_1$ and $r_2 : Q \to M_2$. Consider the commutative diagrams:

$$\begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & M_1 \times M_2 \arrow[r, "\pi_{M_1}"] \arrow[d, "f_1 \times f_2"] & M_1 \arrow[d, "f_1"] \\ Q \arrow[r, "q"'] & N_1 \times N_2 \arrow[r, "\pi_{N_1}"'] & N_1 \end{tikzcd} \qquad \begin{tikzcd} P \arrow[r, "p"] \arrow[d, "m"'] & M_1 \times M_2 \arrow[r, "\pi_{M_2}"] \arrow[d, "f_1 \times f_2"] & M_2 \arrow[d, "f_2"] \\ Q \arrow[r, "q"'] & N_1 \times N_2 \arrow[r, "\pi_{N_2}"'] & N_2 \end{tikzcd} .$$

By $\mathbb{P}$-openness of f and g there exist $r_1 : Q \to M_1$ and $r_2 : Q \to M_2$ such that (for $i = 1, 2$),

$$r_i m = \pi_{M_i} p \qquad \text{and} \qquad f_i r_i = \pi_{N_i} q \ .$$

Let $r = \langle r_1, r_2 \rangle$, then

$$\begin{aligned} rm &= \langle r_1, r_2 \rangle m \\ &= \langle r_1 m, r_2 m \rangle \\ &= \langle \pi_{M_1} p, \pi_{M_2} p \rangle \\ &= \langle \pi_{M_1}, \pi_{M_2} \rangle p \\ &= p \end{aligned}$$

$$\begin{aligned} (f_1 \times f_2) r &= \langle f_1 \pi_{M_1}, f_2 \pi_{M_2} \rangle r \\ &= \langle f_1 \pi_{M_1} r, f_2 \pi_{M_2} r \rangle \\ &= \langle f_1 r_1, f_2 r_2 \rangle \\ &= \langle \pi_{N_1} q, \pi_{N_2} q \rangle \\ &= q \end{aligned}$$

where all the steps above are justified by the properties of the r_i's or by the universal property of products. □

2.2.1 Presheaves as models

The notion of open map was originally developed to be applied in a (pre)topos [62, 63]. A key example there was given by the following:

Definition 2.2.8 *Let $\mathbb{C}$ be a small category. Let X, Y be presheaves over $\mathbb{C}$ and let $f : X \to Y$ be an arrow in $\widehat{\mathbb{C}}$ (viz. a natural transformation between X and Y). Say that f is* open *if every naturality square is a quasi pullback in* **Set**. *This means that f is open if for every arrow in $\mathbb{C}$, $m : c \to c'$, the square*

$$\begin{tikzcd} X(c') \arrow[r, "X(m)"] \arrow[d, "f_{c'}"'] & X(c) \arrow[d, "f_c"] \\ Y(c') \arrow[r, "Y(m)"'] & Y(c) \end{tikzcd}$$

of functions is a quasi pullback, i.e., for every $x \in X(c)$, $y' \in Y(c')$ such that $f_c(x) = Y(m)(y')$, there exists an $x' \in X(c')$ such that

$$X(m)(x') = x \quad \textit{and} \quad f_{c'}(x') = y' \ .$$

It is not difficult to see that, via the Yoneda lemma, the condition of Definition 2.2.8 is equivalent to a path lifting property:

Proposition 2.2.9 *Let $\mathbb{C}$ be a small category. An arrow $f : X \to Y$ in $\widehat{\mathbb{C}}$ is open if and only if , whenever $m : c \to c'$ is an arrow in $\mathbb{C}$, $g : \mathrm{y}(c) \to X$ and $h : \mathrm{y}(c') \to Y$ are arrows in $\widehat{\mathbb{C}}$ and the square*

$$\begin{tikzcd} \mathrm{y}(c) \arrow[r, "g"] \arrow[d, "\mathrm{y}(m)"'] & X \arrow[d, "f"] \\ \mathrm{y}(c') \arrow[r, "h"'] & Y \end{tikzcd}$$

commutes, there exists an arrow $k : \mathrm{y}(c') \to X$, such that

$$k\mathrm{y}(m) = g \quad \textit{and} \quad fk = \mathrm{y}(h).$$

Given a category $\mathcal{M}$ and a small subcategory of path objects $\mathbb{P}$ it is possible to compare the two notions of open maps using the 'canonical' functor $c_{\mathcal{M}} : \mathcal{M} \to \widehat{\mathbb{P}}$ (cf. Definition 1.2.7).

Proposition 2.2.10 (Proposition 12 in [64]) *Let $\mathbb{P}$ be a small dense full subcategory of $\mathcal{M}$. A morphism $f : M \to M'$ in $\mathcal{M}$ is $\mathbb{P}$-open if and only if $c_M(f) : \mathcal{M}[-, M] \to \mathcal{M}[-, M']$ is open (in the sense of Definition 2.2.8).*

When bisimulation in $\mathcal{M}$ and $\widehat{\mathbb{P}}$ has to be related more care is needed. In fact in $\widehat{\mathbb{P}}$ the unique arrow $\emptyset \to X$ from the initial presheaf to any presheaf is always open. A natural way to remove this anomaly is to require in the definition of open map bisimulation an extra surjectivity condition.

Definition 2.2.11 (Open map bisimulation for presheaves) *Let $\mathbb{P}$ be a category. Say that two presheaves X, Y over $\mathbb{P}$ are* $\mathbb{P}$-open bisimilar *if they are connected by a span of epimorphic $\mathbb{P}$-open maps.*

Since in presheaf categories epimorphisms are natural transformations that are pointwise surjective, in the reminder we shall often use the word "surjective" as a synonym of epimorphic. If $\mathbb{P}$ has an initial object and $\mathbb{P}^+$ is the full subcategory of $\mathbb{P}$ to which all initial objects have been removed, then we have already said that the category of rooted presheaves over $\mathbb{P}$ is equivalent to $\widehat{\mathbb{P}^+}$. In particular $\mathbb{P}$-open maps in $\overline{\mathbb{P}}$ correspond to surjective $\mathbb{P}^+$-open maps in $\widehat{\mathbb{P}^+}$.

As one can immediately see, the full subcategory of rooted presheaves of $\widehat{L^*}$ is equivalent to the category $\mathcal{ST}_L$ that in turn is isomorphic to the presheaf category $\widehat{L^+}$, where the objects of L^+ are the non-empty finite strings of elements of L. While the canonical functor from $\mathcal{ES}_L$ to $\widehat{\mathbf{Pom}_L}$ always yields a rooted presheaf, not all rooted presheaves in $\widehat{\mathbf{Pom}_L}$ are obtained in this way. Full subcategories of rooted presheaves play an important role in our approach. Bisimulation in the subcategories of rooted presheaves coincides with bisimulation in the categories of concrete models:

Proposition 2.2.12 (Joyal-Nielsen-Winskel)

(i) *Two synchronisation trees, over labelling set L, are L^*-bisimilar (i.e. strong bisimilar) iff their corresponding presheaves, under the canonical embedding, are related by a span of open maps in the full subcategory of rooted presheaves of $\widehat{L^*}$.*

(ii) *Two event structures, over labelling set L, are $\mathbf{Pom}_L$-bisimilar (i.e. strong history-preserving bisimilar) iff their corresponding presheaves, under the canonical embedding, are related by a span of open maps in the full subcategory of rooted presheaves of $\widehat{\mathbf{Pom}_L}$.*

We have now established the link between categories of models and categories of presheaves over appropriate path categories. We can now look for general constructions that preserve openness and hence bisimulation and that can be used to model process constructions. Left Kan extensions and their right adjoints will be among those.

Before concluding this chapter we just add a simple property, with straightforward verification, of the canonical embeddings $c_{\mathcal{ST}_L}$ and $c_{\mathcal{ES}_L}$ that we shall need in the next chapter.

Proposition 2.2.13 *If the canonical embeddings, $c_{\mathcal{ST}_L}$ and $c_{\mathcal{ES}_L}$ are defined onto the categories of rooted presheaves, $\overline{L^*}$ and $\overline{\mathbf{Pom}_L}$, then they preserve coproducts and initial objects.*

Chapter 3

Presheaf Models for CCS-like languages

Prompted by results like Proposition 2.2.12, in this chapter we take up the suggestion of [64] of considering presheaf models for concurrent computation. There are, already at this preliminary stage, several reasons for doing this.

One reason is that, once one passes the barrier of unfamiliarity, presheaves are an intuitively appealing model of nondeterministic computation. Starting with a category of path objects (or observations) in which morphisms stand for an extension of one path by another, nondeterministic computations are represented essentially by gluing together computation paths in a manner reminiscent of the way a powerdomain is built from a domain as a completion of its finite elements. More accurately, as we saw in Chapter 1 forming presheaves is equivalent to adjoining all colimits to a category, which corresponds to more than just adding directed colimits—the reason why nondeterministic branching is also introduced.

As was argued in [64] presheaf models are promising generalisations of existing models; in fact on the one hand well-known models like synchronisation trees and labelled event structures embed fully and faithfully into appropriate presheaf categories, and, on the other hand, for general reasons, presheaves support operations such as those coming from Kan extensions (cf. Section 1.3). One particular Kan extension, resulting in a functor between presheaves over pomsets, was advanced in [64] as a good candidate for an operation of refinement of the kind proposed for event structures. In Section 3.5 it is shown that this Kan extension acts, when restricted to presheaves associated with event structures, in the same way as the refinement operation in [41]. More generally, working at this level of abstraction yields the possibility of achieving general congruence results like Proposition 3.2.5, that are used to show that a broad class of operations, obtained as left Kan extensions, automatically preserve open maps. It is interesting then to specialise to concrete cases and transfer this congruence properties to traditional models like synchronisation trees and event structures. In particular, for instance, we specialise to show that the refinement, obtained as a Kan extension, preserves open maps and so bisimulation.

3.1 A general process language and its categorical models

In this section we introduce the process language **Proc** of [141] as a template for so-called **CCS**-like languages. The distinctive feature of **Proc** is that the parallel composition operator is removed in favour of a more general product out of which different definitions of parallel compositions can be constructed with the help of restriction and relabelling operations. As noted in [141], if the terms of **Proc** are to be interpreted in categories of labelled structures (transition systems, event structures, ...) then it is convenient to regard such categories as fibred over the labelling system, namely over the category $\mathbf{Set}_*$ of sets and partial maps. By doing this, in fact, it is easy to recover the universal content of the operators associated with restriction, relabelling and (to a certain extent) prefixing in terms of (co)cartesian liftings. Assume then a universe of labels (the class of elements of some set), define the terms t of **Proc** by the following grammar:

$$t ::= \mathbf{Nil} \mid at \mid t_0 \oplus t_1 \mid t_0 \times t_1 \mid t\lceil\Lambda \mid t\{\Xi\} \mid x \mid rec\, x.t$$

where a is a label, Λ is a set of labels, Ξ is a total function from labels to labels and x a variable drawn from some distinguished infinite set that we indicate as $Vars$. In [141] an analysis of the categorical status of the operations involved in the semantics of **Proc** was conducted. Inspired by it, we propose an axiomatisation of the structure required for a category to give a model of the language. In particular we emphasise the role of partial relabelling functions as substitution operators and therefore we impose upon them the Fröbenius reciprocity law and the Beck-Chevalley condition that were introduced in Chapter 1

Definition 3.1.1 (Models for Proc) *A categorical model for* **Proc** *is given by a functor $\pi : \mathcal{M} \to \mathbf{Set}_*$ such that:*

- *$\mathcal{M}$ has binary products $(\times)$.*
- *For every set L, the fibre $\mathcal{M}_L$ has initial object (0_L), binary coproducts $(+_L)$ and colimits of ω-chains.*
- *For every inclusion $i : L \hookrightarrow M$ of sets, there exists a cartesian lifting functor $i^* : \mathcal{M}_M \to \mathcal{M}_L$.*
- *For every total function $f : L \to M$, there exists a cocartesian lifting functor $f_! : \mathcal{M}_L \to \mathcal{M}_M$.*
- *For every set L and label $a \in L$, there exists a prefixing endofunctor*

$$pre_{a,L} : \mathcal{M}_L \to \mathcal{M}_L$$

which preserves ω-colimits as well as existing cocartesian lifting functors for partial maps $f : L \rightharpoonup M$, that are defined on a, i.e., if f is a partial map from L to M such that $f(a)$ is defined and such that $f_!$ exists, then the following square commutes, up to natural isomorphism:

$$\begin{array}{ccc} \mathcal{M}_L & \xrightarrow{pre_{a,L}} & \mathcal{M}_L \\ {\scriptstyle f_!}\downarrow & & \downarrow{\scriptstyle f_!} \\ \mathcal{M}_M & \xrightarrow[pre_{f(a),M}]{} & \mathcal{M}_M. \end{array}$$

- *Whenever applicable, i.e., whenever the required (co)cartesian arrows exist, the Fröbenius reciprocity law and Beck-Chevalley condition of Definition 1.4.10 hold.*

In [141] a few models were considered ranging from 'interleaving' ones, like transition systems and synchronisation trees, to 'non-interleaving' ones, like event structures, Petri nets or transition systems with independence. Here we recall briefly, how the structure required in the definition above is used to give semantics to terms of **Proc** .

We first deduce some properties of models. In fact a model for **Proc** , as described in Definition 3.1.1, is not a cofibration, still it has enough cocartesian liftings to deduce the following corollary (of the proof) of Proposition 1.4.14:

Corollary 3.1.2 *If $\pi : \mathcal{M} \to \mathbf{Set}_*$ is a model for* **Proc** *, then $\mathcal{M}$ has initial object, binary coproducts and colimits of ω-chains.*

3.1.1 Denotational semantics of Proc

Before actually giving the semantics of **Proc** terms for a categorical model $\mathcal{M}$, we introduce the operation $\oplus$ that will be used to model the non-deterministic sum. As we shall see this contrasts with the categorical sum in the choice that it operates with respect to the labelling sets (taking their union and not their disjoint sum).

Definition 3.1.3 *Let $\pi : \mathcal{M} \to \mathbf{Set}_*$ be a model for* **Proc** *. If $M \in\mid \mathcal{M}_L \mid$ and $N \in\mid \mathcal{M}_{L'} \mid$, define $M \oplus N \in\mid \mathcal{M}_{L\cup L'} \mid$ to be*

$$M \oplus N = i_{L,!}(M) +_{L\cup L'} i_{L',!}(N) \ .$$

As one immediately sees, for any two sets L, L', this construction induces a functor $(-\oplus-) : \mathcal{M}_L \times \mathcal{M}_{L'} \to \mathcal{M}_{L\cup L'}$. Still, because of the choice of taking the union and not the disjoint set of labelling sets, these do not lift to a functor $\mathcal{M} \times \mathcal{M} \to \mathcal{M}$, because it does not have a well defined action on arrows. In fact for any two arrows of $\mathcal{M}$, $f : M \to M'$ and $g : N \to N'$ it is not possible to define what $f \oplus g$ must be. Still this is possible when $\pi(f)$ and $\pi(g)$ agree on their action on the elements of $\pi(M)\cap\pi(N)$. This is in particular true for the arrows in the fibres since in that case, $\pi(f)$ and $\pi(g)$ are the identity functions. More generally one can define a bifunctor $\oplus : \mathcal{M}_{in} \times \mathcal{M}_{in} \to \mathcal{M}_{in}$ where $\mathcal{M}_{in}$ is the subcategory of $\mathcal{M}$ of those arrows, f, such that $\pi(f)$ is an inclusion of sets.

Let $\pi : \mathcal{M} \to \mathbf{Set}_*$ be a model as in Definition 3.1.1. We describe the denotational semantics of **Proc** inductively on the structure of the terms, assuming an environment function $\rho : Vars \to\mid \mathcal{M} \mid$:

Nil: $[\![\mathbf{Nil}]\!]_\rho = 0_\emptyset$ an initial object of $\mathcal{M}$
Variables: $[\![x]\!]_\rho = \rho(x)$
Sum: $[\![t_1 \oplus t_2]\!]_\rho = [\![t_1]\!]_\rho \oplus [\![t_2]\!]_\rho$
Product: $[\![t_1 \times t_2]\!]_\rho = [\![t_1]\!]_\rho \times [\![t_2]\!]_\rho$
Restriction: Let Λ be a set. $[\![t_1 \restriction \Lambda]\!]_\rho = i^*([\![t_1]\!]_\rho)$, where $i : \Lambda \cap L \hookrightarrow L$ and $L = \pi([\![t_1]\!]_\rho)$

Relabelling: Let $\Xi : L \to M$ be total. $[\![t_1[\Xi]]\!]_\rho = \Xi'_{N,!}([\![t_1]\!]_\rho)$, where $N = \pi([\![t_1]\!]_\rho)$ and Ξ'_N is the truncation to its image of the function,

$$\begin{array}{rccl} \Xi_N : & L \cap N & \longrightarrow & M \cup N \\ & x & \mapsto & \begin{cases} \Xi(x) & \text{if } x \in L \\ x & \text{otherwise} \end{cases} \end{array}$$

Prefixing: Let a be a label, then $[\![at]\!]_\rho = pre_{a,L\cup\{a\}}(i_!([\![t]\!]_\rho))$, where $\pi([\![t]\!]_\rho) = L$ and $i : L \hookrightarrow L \cup \{a\}$.

Recursion: Let t be any term, and let x be a variable (possibly free in t). Given any environment ρ the term t and the variable x determine an endofunctor

$$\begin{array}{rccl} t^x_\rho : & \mathcal{M}_{in} & \to & \mathcal{M}_{in} \\ & M & \mapsto & [\![t]\!]_{\rho[M/x]} \end{array} .$$

From t^x_ρ, the following ω-chain is derivable:

$$\begin{array}{rccll} T : & \omega & \to & \mathcal{M} & \\ & 0 & \mapsto & [\![t]\!]_{\rho[[\![\mathbf{Nil}]\!]_\rho/x]} & \\ & n & \mapsto & [\![t]\!]_{\rho[T^{n-1}/x]} & \text{for } n > 0 \end{array}$$

Define $[\![recx.t]\!]\rho = \operatorname{colim} T$. Since all the constructions involved in the denotation of a term t are ω-colimits preserving functors, then colim T is a fixed point for t^x_ρ.

In giving the interpretation of the terms of **Proc** as objects of $\mathcal{M}$ we had not been constrained by the Fröbenius reciprocity law and the Beck-Chevalley condition. So, where does their use lie? As we shall see in the next section it is when equipping our models with a notion of bisimulation that it will matter that we have these conditions around: they constrain the action that the cartesian arrows have on the objects of $\mathcal{M}$.

3.2 Presheaf models for Proc

As we saw in the previous section, the denotation of a term of **Proc** is given mainly by means of universal constructions: (co)products, (co)cartesian liftings and fixed points.[1] Since we have an abstract definition of bisimulation in mind (see the previous chapter) it is natural to look for abstract proofs of the expected congruence properties of the term constructors with respect to the bisimulation relation. Given the generality of the data required for both the model and the bisimulation (the latter requiring an 'arbitrary' choice of categories of path objects), the task looks quite impossible to achieve. This might not be the case, on the contrary, if one manages to link the path category parametrisation over which bisimulation is defined with the description of the model $\mathcal{M}$. We have already seen that whenever we choose as a path category for bisimulation a (small) dense subcategory ($\mathbb{P}$) of $\mathcal{M}$, we can regard $\mathcal{M}$ itself as a full subcategory

[1] Prefixing with its *ad hoc* requirements forms an exception to this general treatment.

of $\widehat{\mathbb{P}}$ (cf. Proposition 1.2.8). We also saw that open maps and therefore bisimulation is always preserved in moving from $\mathcal{M}$ to $\widehat{\mathbb{P}}$; in examples of interest bisimulation is reflected, too (cf. Proposition 2.2.12). This motivated the idea that the study of open map bisimulation in presheaf categories could help deriving properties, like the congruence property of bisimulation, in an abstract setting that might be instantiated later to specific cases of interest. This, indeed, turned out to be the case and a special role in this has been played by Proposition 3.2.5 below, that states preservation of open maps and hence bisimulation along colimit preserving functors between presheaf categories. Let's proceed in good order though. We start by describing what we need to build a presheaf model for **Proc** .

Definition 3.2.1 (Pre-presheaf Models for Proc) *A pre-*presheaf model *for* **Proc** *consists of a functor $\mathbb{P}_{(-)}$ from* $\mathbf{Set}_*$ *to* **Cat***, the category of small categories, which sends $\lambda : L \rightharpoonup M$ to $\overline{\lambda} : \mathbb{P}_L \to \mathbb{P}_M$ such that:*

- *For each set L, the category $\mathbb{P}_L$ has an initial object; the functors $\overline{\lambda}$, for $\lambda : L \rightharpoonup M$, preserve initial objects.*
- *For each set L and element $a \in L$, there is an explicitly given* prefixing *functor $pre_{a,L} : \mathbb{P}_L \to \mathbb{P}_L$ satisfying commutativity of the following diagram*

$$\begin{tikzcd} \mathbb{P}_L \arrow[r, "pre_{a,L}"] \arrow[d, "\overline{\lambda}"'] & \mathbb{P}_L \arrow[d, "\overline{\lambda}"] \\ \mathbb{P}_{L'} \arrow[r, "pre_{\lambda(a),L'}"'] & \mathbb{P}_{L'} \end{tikzcd}$$

  for any $\lambda : L \rightharpoonup L'$ that is defined on a.

A process with labelling set L is to denote a rooted presheaf over $\mathbb{P}_L$.

With the Grothendieck construction of Section 1.4.1 in mind, one sees that a pre-presheaf model defines a split cofibration in $\mathbf{Set}_*$. We shall return to this in Section 3.4 when dealing with the example provided by pomsets and event structures.

Example 3.2.2 *Our two examples here will be given by the presheaf models that "cover" synchronisation trees and event structures. Namely:*

1. *Define $(-)^* : \mathbf{Set}_* \to \mathbf{Cat}$ be the functor that associates to each set L the partial ordered set (regarded as a category) L^* of finite (possibly empty) strings of elements of L and to each partial map $\lambda : L \rightharpoonup M$ the monotone map (i.e., the functor) that pointwise relabel every string over L to a string over M according to λ, sending every letter on which λ is undefined to the empty string (ε). The prefixing functors are defined by usual prefixing of strings, i.e., $pre_{a,L}(\sigma) = a\sigma$.*
2. *Define $\mathbf{Pom}_{(-)} : \mathbf{Set}_* \to \mathbf{Cat}$ to be functor that associates to each set L the category of pomsets labelled in L. If $\lambda : L \rightharpoonup M$, then $\overline{\lambda} : \mathbf{Pom}_L \to \mathbf{Pom}_M$ is the following functor:*
   - **On objects:** *Given a pomset $P = (P, \leq, l)$ in $|\mathbf{Pom}_L|$, $\overline{\lambda}(P) = (P', \leq', l')$ with $P' = \{e \in P \mid \lambda(l(e)) \text{ is defined}\}$, $\leq' = \leq \cap (P' \times P')$, $l'(e) = \lambda(l(e))$.*

- **On arrows:** *If $f : P \to Q$ is an arrow in* $\mathbf{Pom}_L$, $\overline{\lambda}(f)$ *is simply the restriction of f to P' and Q'*

The prefixing functors are again the obvious ones, i.e., the prefixing $pre_{a,L}(P)$ of a pomset P is obtained by adding a new event, labelled 'a', which is placed below all the events of P in the causal order relation.

Recall that from every functor $F : \mathbb{C} \to \mathbb{D}$ between small categories one can derive a triple of adjoint functors $F_! \dashv F^* \dashv F_*$:

$$\begin{tikzcd} \widehat{\mathbb{C}} \arrow[r, bend left, "F_!"] \arrow[r, bend right, "F_*"'] & \widehat{\mathbb{D}}. \arrow[l, "F^*"'] \end{tikzcd}$$

Hence from the data defining a presheaf model we can derive a bifibration using the Grothendieck construction.

3.2.1 The Grothendieck construction in presheaf models

Given a presheaf model $\mathbb{P}_{(-)}$, we can glue together all the fibres, consisting of categories of rooted presheaves over $\mathbb{P}_L$, to form a fibration over $\mathbf{Set}_*$ which we call $Groth(\mathbb{P}_{(-)})$:

Objects: pairs $\langle X, L\rangle$ with $L \in |\mathbf{Set}_*|$ and X a rooted presheaf over $\mathbb{P}_L$,
Arrows: pairs $\langle f, \lambda\rangle : \langle X, L\rangle \to \langle Y, M\rangle$ with $\lambda : L \rightharpoonup M$ and $f : X \to \overline{\lambda}^*(Y)$.

The composition of arrows is $\langle g, \mu\rangle \circ \langle f, \lambda\rangle = \langle\overline{\lambda}^*(g) \circ f, \mu \circ \lambda\rangle$. Clearly the projection $\langle X, L\rangle \mapsto L$ is the object part of a functor $\pi : Groth(\mathbb{P}_{(-)}) \to \mathbf{Set}_*$. Intuitively, the Grothendieck construction glues the various fibres together; it adds arrows between presheaves (possibly over different fibres), to allow for the possibility of a partial relabelling of actions.

For $\lambda : L \rightharpoonup M$ the adjunction $\overline{\lambda}_! \dashv \overline{\lambda}^*$ between presheaf categories $\widehat{\mathbb{P}}_L$ and $\widehat{\mathbb{P}}_M$ cuts down to an adjunction between the fibres of rooted presheaves. The adjunctions ensure that the Grothendieck fibration is in fact a bifibration [57]; the cocartesian lifting of λ with respect to X is $(\eta_X, \lambda) : X \to \overline{\lambda}_!(X)$ where $\eta_X : X \to \overline{\lambda}^*\overline{\lambda}_!(X)$ is the component of the unit of the adjunction at X. Since the fibres are presheaf categories they satisfy all the colimit completions required in Definition 3.1.1. Moreover by applying Proposition 1.4.14 we can deduce that $Groth(\mathbb{P}_{(-)})$ has binary products.

Even if the functor $\mathbb{P}_{(-)}$ induces a split cofibration, whose fibres are the categories $\mathbb{P}_L$, for L a set, when extended to $Groth(\mathbb{P}_{(-)})$ this property is lost. On the other hand, since the $\overline{\lambda}^*$'s are defined by composition, $Groth(\mathbb{P}_{(-)})$ is a split fibration.

Definition 3.2.3 (Presheaf Models for Proc) *A presheaf model for* **Proc** *, consists of a functor $\mathbb{P}_{(-)}$ as in Definition 3.2.1 satisfying the extra condition that the induced bifibration $Groth(\mathbb{P}_{(-)})$ satisfies both the Fröbenius reciprocity law and the Beck-Chevalley condition.*

Remark: Observe that following the statement of Proposition 2.2.12, we consider here rooted presheaves (assuming the base category has initial objects). This restriction might look slightly odd from the categorical point of view. We decided on it, since it will help us, at this early stage of the development of presheaf models, to give a smooth treatment of prefixing with an immediate proof that it preserves open map bisimulation. Later, when we will start describing the base categories using recursive domain equations, we will give a more detailed account of prefixing using a notion of lifting, and therefore we will remove the 'anomaly' of the restriction to rooted presheaves. Note once again that from a categorical point of view, the choice of rooted presheaves does not reduce the generality of our approach since, as we already said a few times, the category of rooted presheaves over a category with initial object is equivalent to the category of presheaves over the same base category to which all the initial objects has been removed.

It is immediately seen that given a presheaf models for **Proc** , $\mathbb{P}_{(-)}$, $Groth(\mathbb{P}_{(-)})$ with the obvious projection functor forms a categorical model for **Proc** in the sense of Definition 3.1.1. It is easy now to equip our model with a notion of bisimulation. We first bring the two objects over a common fibre and then see whether they are open map bisimilar there.

Definition 3.2.4 (Open map bisimulation in $Groth(\mathbb{P}_{(-)})$) *Let $\langle X, L\rangle$ and $\langle Y, M\rangle$ be two objects in $Groth(\mathbb{P}_{(-)})$. We say that they are* (open map) bisimilar *if $\overline{i}_{L,!}(\langle X, L\rangle)$ and $\overline{i}_{M,!}(\langle Y, M\rangle)$ are related by a span of (surjective) $\mathbb{P}_{L\cup M}$-open maps, where i_L and i_M are the set inclusions $L \stackrel{i_L}{\hookrightarrow} L\cup M \stackrel{i_M}{\hookleftarrow} M$.*

Notation: In the reminder of this chapter, we shall write $\langle X, L\rangle \sim \langle Y, M\rangle$ to mean that they are open map bisimilar, and $X \sim_L Y$ to say that both X and Y are in $|\widehat{\mathbb{P}_L}|$ and that they are $\mathbb{P}_L$-open bisimilar. Hence we have that

$$\langle X, L\rangle \sim \langle Y, M\rangle \text{ if and only if } \overline{i}_{L,!}(X) \sim_{L\cup M} \overline{i}_{M,!}(Y)\ .$$

From the above definition we immediately see that moving objects across different fibres along cocartesian liftings preserves bisimulation. But can we deduce more, i.e., can we deduce that bisimulation is a congruence with respect to the operations of the denotational semantics of processes? The answer is yes and it relies mainly on the following result (Corollary 4 in [26]).

Proposition 3.2.5 (Colimit preserving functors preserve open maps) *If $\mathbb{C}$ and $\mathbb{D}$ are two small categories and $F : \widehat{\mathbb{C}} \to \widehat{\mathbb{D}}$ is a colimit-preserving functor, then F preserves open maps:*

> *If $\alpha : X \stackrel{\cdot}{\longrightarrow} Y$ is a $\mathbb{C}$-open map, then $F(\alpha) : F(X) \stackrel{\cdot}{\longrightarrow} F(Y)$ is a $\mathbb{D}$-open map.*

Proof: The proof is postponed to Chapter 4 where the proposition reappears as Corollary 4.6.6 of the more general Theorem 4.6.5. $\square$

In any category, an epimorphism $e : c \twoheadrightarrow d$ gives rise to a pushout square

$$\begin{tikzcd} c \arrow[r, "e"] \arrow[d, "e"'] & d \arrow[d, "1"] \\ d \arrow[r, "1"'] & d \end{tikzcd} .$$

Hence any colimit preserving functor preserves epimorphisms. This implies the following Corollary.

Corollary 3.2.6 *Let $\mathbb{C}$ and $\mathbb{D}$ be two small categories. Let $F : \widehat{\mathbb{C}} \to \widehat{\mathbb{D}}$ be a colimit-preserving functor. If X and Y are two $\mathbb{C}$-open bisimilar presheaves, then $F(X)$ and $F(Y)$ are $\mathbb{D}$-open bisimilar.*

Using the Proposition and Corollary above we can deduce that for any $\langle X, L\rangle$ in $Groth(\mathbb{P}_{(-)})$, there exists a least subset L' of L with inclusion function $i : L' \hookrightarrow L$, such that, $\langle \overline{i}^* X, L'\rangle \sim \langle X, L\rangle$.

Proposition 3.2.7 *Let $\langle X, L\rangle \in\mid Groth(\mathbb{P}_-)\mid$. Let $M \overset{i}{\hookrightarrow} L \overset{j}{\hookleftarrow} N$ be two inclusions, such that $\langle \overline{i}^* X, M\rangle \sim \langle X, L\rangle \sim \langle \overline{j}^* X, N\rangle$, then if*

$$\begin{tikzcd} M \cap N \arrow[r, hook, "l"] \arrow[d, hook, "k"'] & M \arrow[d, hook, "i"] \\ N \arrow[r, hook, "j"'] & L \end{tikzcd}$$

is the obvious pullback square of inclusions,

$$\langle \overline{l}^* \overline{i}^* X, M \cap N\rangle \sim \langle X, L\rangle \ .$$

Proof: We need to show that $\overline{i}_! \overline{l}_! \overline{l}^* \overline{i}^* X \sim_L X$. By assumption $\overline{i}_! \overline{i}^* X \sim_L X$, hence by Corollary 3.2.6, $\overline{j}^* \overline{i}_! \overline{i}^* X \sim_N \overline{j}^* X$. By the Beck-Chevalley condition, $\overline{j}^* \overline{i}_! \overline{i}^* X \cong \overline{k}_! \overline{l}^* \overline{i}^* X$, hence by composing with $\overline{j}_!$, $\overline{j}_! \overline{k}_! \overline{l}^* \overline{i}^* X \sim_L \overline{j}_! \overline{j}^* X \sim_L X$. Since the square of inclusions is a commutative one, we know that $\overline{j}_! \overline{k}_! \cong \overline{i}_! \overline{l}_!$, hence $\overline{i}_! \overline{l}_! \overline{l}^* \overline{i}^* X \sim_L X$. $\square$

Corollary 3.2.8 *For any $\langle X, L\rangle \in\mid Groth(\mathbb{P}_-)\mid$, there exists a least subset L' of L such that $\langle \overline{i}^* X, L'\rangle \sim \langle X, L\rangle$, where $i : L' \hookrightarrow L$ is the inclusion function.*

Proof: Just take L' to be equal to the intersection of all $M \subseteq L$, such that $\langle \overline{i}_M^* X, M\rangle \sim \langle X, L\rangle$, where $i_M : M \hookrightarrow L$ is the inclusion function. $\square$

Definition 3.2.9 *For any $\langle X, L\rangle \in\mid Groth(\mathbb{P}_-)\mid$, say that X* reduces to *M, if M is a subset of L, and $\langle X, L\rangle \sim \langle \overline{i}^* X, M\rangle$. Is L' is the least subset of L for which X can be reduced to, say that X* is essentially on *L'. If the least subset L' is L itself, say that X* is reduced*.*

The notion of "reduction" captures the idea that the presheaf X is essentially described by path objects of $\mathbb{P}_{L'}$ even if one is regarding it as an object of $\widehat{\mathbb{P}_L}$.

The above results yields the following characterisations of bisimilarity in $Groth(\mathbb{P}_{(-)})$.

Proposition 3.2.10 *Two objects $\langle X, L\rangle, \langle Y, M\rangle \in\mid Groth(\mathbb{P}_-)\mid$ are bisimilar iff they both reduce to $L \cap M$ and $\overline{i}^* X \sim_{L\cap M} \overline{j}^* Y$, for $L \overset{i}{\hookleftarrow} L \cap M \overset{j}{\hookrightarrow} M$.*

Corollary 3.2.11 *Two objects $\langle X, L\rangle, \langle Y, M\rangle \in\mid Groth(\mathbb{P}_-)\mid$ are bisimilar iff they are essentially on the same set N with $L \overset{i}{\hookleftarrow} N \overset{j}{\hookrightarrow} M$ and $\overline{i}^* X \sim_N \overline{j}^* Y$.*

As we shall see all the operations involved in the semantics of **Proc** preserve bisimulation.

Before going into the semantics of **Proc** in presheaf models we recall the following instantiation of Proposition 1.4.11 to $Groth(\mathbb{P}_{(-)})$.

Proposition 3.2.12 *Let $\mathbb{P}_{(-)} : \mathbf{Set}_* \to \mathbf{Cat}$ be a presheaf model for* **Proc** *. Let $i : L \rightarrowtail M$ be a monomorphism in $\mathbf{Set}_*$, i.e., an injective (total) function. Then the following facts hold:*

- $\overline{i}^*\overline{i}_! \cong id_{\overline{\mathbb{P}_L}}$.
- *$\overline{i}_!$ preserves products in the fibres, i.e., for every two rooted presheaves $X, Y \in\mid\overline{\mathbb{P}_L}\mid$,*

$$\overline{i}_!(X \times_L Y) \cong \overline{i}_!(X) \times_M \overline{i}_!(Y) \ .$$

3.3 Semantic constructions in $Groth(\mathbb{P}_{(-)})$

We analyse now the constructions used in $Groth(\mathbb{P}_{(-)})$ to give the semantics of **Proc** according to Section 3.1.1 and show that they preserve open map bisimulation.

Products: As we have already said the category $Groth(\mathbb{P}_{(-)})$ has products. They can be constructed (cf. Proposition 1.4.14) using the products in the fibres as follows.[2] Given $\langle X, L\rangle, \langle Y, M\rangle \in |Groth(\mathbb{P}_{(-)})|$. Define

$$\langle X, L\rangle \times \langle Y, M\rangle = \langle \overline{\pi_L}^*(X) \times \overline{\pi_M}^*(Y), L \times_* M\rangle$$

where $L \overset{\pi_L}{\leftharpoondown} L \times_* M \overset{\pi_M}{\rightharpoondown} M$ are the projections of the product in $\mathbf{Set}_*$.

Proposition 3.3.1 *If $\langle X, L\rangle$ is open map map bisimilar to $\langle X', L'\rangle$ and $\langle Y, M\rangle$ is open map bisimilar to $\langle Y', M'\rangle$ then the product $\langle X, L\rangle \times \langle Y, M\rangle$ is open map bisimilar to the product $\langle X', L'\rangle \times \langle Y', M'\rangle$.*

Proof: Let $N = L \times_* M$ and $N' = L' \times_* M'$. Consider the diagram

$$\begin{tikzcd}
L \arrow[d, hook, "i_L"'] & N \arrow[l, "\pi_L"'] \arrow[r, "\pi_M"] \arrow[d, hook, "i_N"'] & M \arrow[d, hook, "i_M"] \\
L \cup L' & N \cup N' \arrow[l, "\alpha"'] \arrow[r, "\beta"] & M \cup M' \\
L' \arrow[u, hook, "i'_L"] & N' \arrow[l, "\pi_{L'}"'] \arrow[r, "\pi_{M'}"] \arrow[u, hook, "i_{N'}"'] & M' \arrow[u, hook, "i_{M'}"']
\end{tikzcd} ,$$

[2]We tend to use ($\mathbf{Set}_*$ is an exception) the same symbol "$\times$" to indicate the product of two objects in a category. This is irrespective of the fact that some time, as here, products taken in different categories appear in the same expression. We hope that the reader can disambiguate from the context.

with α and β the obvious projecting partial functions. Observe that since the set $N \cup N'$ is included in $(L \cup L') \times_* (M \cup M')$, then both the upper diagram and the lower one are limiting cones of vertex N and N', respectively. Hence by Proposition 1.4.13 we have the following two isomorphisms in $\overline{\mathbb{P}_{N \cup N'}}$:

$$\begin{array}{rcl} \overline{i}_{N,!}(\overline{\pi}_L^* X \times \overline{\pi}_M^* Y) & \cong & \overline{\alpha}^* \overline{i}_{L,!} X \times \overline{\beta}^* \overline{i}_{M,!} Y \\ \overline{i}_{N',!}(\overline{\pi}_{L'}^* X' \times \overline{\pi}_{M'}^* Y') & \cong & \overline{\alpha}^* \overline{i}_{L',!} X' \times \overline{\beta}^* \overline{i}_{M',!} Y' \ . \end{array}$$

This means that what we have to prove is that assuming $\overline{i}_{L,!} X \sim_{L \cup L'} \overline{i}_{L',!} X'$ and $\overline{i}_{M,!} Y \sim_{M \cup M'} \overline{i}_{M',!} Y'$ we have that

$$\overline{\alpha}^* \overline{i}_{L,!} X \times \overline{\beta}^* \overline{i}_{M,!} Y \sim_{N \cup N'} \overline{\alpha}^* \overline{i}_{L',!} X' \times \overline{\beta}^* \overline{i}_{M',!} Y' \ .$$

In fact, by Corollary 3.2.6, we deduce that $\overline{\alpha}^* \overline{i}_{L,!} X \sim_{N \cup N'} \overline{\alpha}^* \overline{i}_{L',!} X'$ and $\overline{\beta}^* \overline{i}_{M,!} Y \sim_{N \cup N'} \overline{\beta}^* \overline{i}_{M',!} Y'$. Hence, by Proposition 2.2.7 combined with the fact that the product of surjective natural transformations is surjective,

$$\overline{\alpha}^* \overline{i}_{L,!} X \times \overline{\beta}^* \overline{i}_{M,!} Y \sim_{N \cup N'} \overline{\alpha}^* \overline{i}_{L',!} X' \times \overline{\beta}^* \overline{i}_{M',!} Y' \ .$$

□

Sum: Let $\langle X, L\rangle, \langle Y, M\rangle \in |Groth(\mathbb{P}_{(-)})|$. Define

$$\langle X, L\rangle \oplus \langle Y, M\rangle = \langle \overline{i_{L!}}(X) + \overline{i_{M!}}(Y), L \cup M\rangle$$

where $L \xrightarrow{i_L} L \cup M \xleftarrow{i_M} M$ are the obvious set inclusions.

Proposition 3.3.2 *The functor $\oplus$ preserves open map bisimulation; if $\langle X, L\rangle$ is open map bisimilar to $\langle X', L'\rangle$ and $\langle Y, M\rangle$ is open map bisimilar to $\langle Y', M'\rangle$ then $\langle X, L\rangle \oplus \langle Y, M\rangle$ is open map bisimilar to $\langle X', L'\rangle \oplus \langle Y', M'\rangle$*

Proof: Let $N = L \cup M$, $N' = L' \cup M'$, $L'' = L \cup L'$ and $N'' = N \cup N'$. Consider the following diagram of inclusions:

$$\begin{tikzcd} L \arrow[r, hook, "j_L"] \arrow[d, hook, "i_L"'] & N \arrow[d, hook, "i_N"'] & M \arrow[l, hook', "j_M"'] \arrow[d, hook, "i_M"] \\ L'' \arrow[r, hook, "j_{L''}"] & N'' & M'' \arrow[l, hook', "j_{M''}"'] \\ L' \arrow[u, hook, "i'_L"] \arrow[r, hook, "j_{L'}"'] & N' \arrow[u, hook, "i_{N'}"'] & M' \arrow[l, hook', "j_{M'}"] \arrow[u, hook, "i_{M'}"'] \end{tikzcd} \ .$$

By assumption $\overline{i}_{L,!} X \sim_{L''} \overline{i}_{L',!} X'$ and $\overline{i}_{M,!} X \sim_{M''} \overline{i}_{M',!} X'$, hence

$$\overline{i}_{N,!}(\overline{j}_{L,!} X + \overline{j}_{M,!} Y) \quad \cong \quad \overline{i}_{N,!} \overline{j}_{L,!} X + \overline{i}_{N,!} \overline{j}_{M,!} Y$$

$$\begin{array}{cl} \cong & \overline{j}_{L'',!}\overline{i}_{L,!}X + \overline{j}_{M'',!}\overline{i}_{M,!}Y \\ \sim_{N''} & \overline{j}_{L'',!}\overline{i}_{L',!}X' + \overline{j}_{M'',!}\overline{i}_{M',!}Y' \\ \cong & \overline{i}_{N',!}\overline{j}_{L',!}X' + \overline{i}_{N',!}\overline{j}_{M',!}Y' \\ \cong & \overline{i}_{N',!}(\overline{j}_{L',!}X' + \overline{j}_{M',!}Y') \ . \end{array}$$

□

REMARK: This sum construction is not the coproduct because of the choice of labelling set for the sum. It can be shown that, if $[i_L, i_M] : L + M \to L \cup M$ is the mediating map from the coproduct of sets, then

$$\langle X, L\rangle \oplus \langle Y, M\rangle \cong \overline{[i_L, i_M]}_!(\langle X, L\rangle + \langle Y, M\rangle).$$

Restriction: Let Λ be a set and let $\langle X, L\rangle \in |Groth(\mathbb{P}_{(-)})|$. Then consider the inclusion map $i : \Lambda \cap L \hookrightarrow L$ and define the restriction of X to $\Lambda \cap L$ to be

$$\langle X, L\rangle\lceil\Lambda = \langle \overline{i}^*(X), \Lambda \cap L\rangle \ .$$

Proposition 3.3.3 *The functor $(-)\lceil\Lambda$ preserves open map bisimulation; if $\langle X, L\rangle$ is open map bisimilar to $\langle X', L'\rangle$ then $\langle X, L\rangle\lceil\Lambda$ is open map bisimilar to $\langle X', L'\rangle\lceil\Lambda$*

Proof: Consider the diagram of inclusions

$$\begin{tikzcd} \Lambda \cap L \arrow[r, hook, "k"] \arrow[d, hook, "j_L"'] & L \arrow[d, hook, "i_L"] \\ \Lambda \cap (L \cup L') \arrow[r, hook, "l"] & L \cup L' \\ \Lambda \cap L' \arrow[r, hook, "m"] \arrow[u, hook, "j_{L'}"] & L' \arrow[u, hook', "i_{L'}"'] \end{tikzcd}$$

Both squares are readily seen to be pullbacks, hence by Beck-Chevalley, $\overline{j}_{L,!}\overline{k}^* \cong \overline{l}^*\overline{i}_{L,!}$ and $\overline{j}_{L',!}\overline{m}^* \cong \overline{l}^*\overline{i}_{L',!}$. Assuming, $\overline{i}_{L,!}X \sim_{L\cup L'} \overline{i}_{L',!}X'$, we can deduce

$$\begin{array}{rcl} \overline{j}_{L,!}\overline{k}^*X & \cong & \overline{l}^*\overline{i}_{L,!}X \\ & \sim_{\Lambda\cap(L\cup L')} & \overline{l}^*\overline{i}_{L',!}X' \\ & \cong & \overline{j}_{L',!}\overline{m}^*X' \ . \end{array}$$

□

Relabelling: Let $\Xi : N \to M$ be total. Take $\langle X, L\rangle$ as usual, define $\Xi_L : L \to M \cup L$ with

$$\Xi_L(x) = \begin{cases} \Xi(x) & \text{if } x \in L \\ x & \text{otherwise} \end{cases}$$

Consider the truncation Ξ'_L of Ξ_L to its image set, i.e. $\Xi_L : L \to \Xi_L L$. Define the relabelling to be

$$\langle X, L\rangle[\Xi] = \langle \overline{\Xi_{L!}}(X), \Xi_L L\rangle \ .$$

Relabelling preserves bisimulation:

Proposition 3.3.4 *If $\langle X, L\rangle$ is open map bisimilar to $\langle X', L'\rangle$ then $\langle X, L\rangle[\Xi]$ is open map bisimilar to $\langle X', L'\rangle[\Xi]$*

Proof: Take the commuting diagram

$$\begin{tikzcd}
L \arrow[r, "\Xi_L"] \arrow[d, hook, "i_L"'] & \Xi_L L \arrow[d, hook, "j_L"] \\
L \cup L' \arrow[r, "\Xi_{L\cup L'}"] & \Xi_{L\cup L'}(L \cup L') \\
L' \arrow[r, "\Xi_{L'}"] \arrow[u, hook, "i_{L'}"] & L' \arrow[u, hook, "j_{L'}"'] .
\end{tikzcd}$$

Knowing that $\bar{i}_{L,!}X \sim_{L\cup L'} \bar{i}_{L',!}X'$, we obtain

$$\begin{array}{rcl}
\bar{j}_{L,!}\overline{\Xi}_{L,!}X & \cong & \overline{\Xi}_{L\cup L',!}\bar{i}_{L,!}X \\
& \sim_{\Xi_{L\cup L'}L\cup L'} & \overline{\Xi}_{L\cup L',!}\bar{i}_{L',!}X' \\
& \cong & \bar{j}_{L',!}\overline{\Xi}_{L',!}X' .
\end{array}$$

□

Prefixing: Suppose we have a label set L and an element $a \in L$. By taking a left Kan extension we extend the prefixing functors to $pre_{a,L,!} : \widehat{\mathbb{P}}_L \to \widehat{\mathbb{P}}_{L\cup\{a\}}$.

Proposition 3.3.5 *Let $a \in L \cap L'$. If $\langle X, L\rangle$ is open map bisimilar to $\langle X', L'\rangle$ then $pre_{a,L,!}(\langle X, L\rangle)$ is open map bisimilar to $pre_{a,L',!}(\langle X', L'\rangle)$*

Proof: By definition the following diagram commutes

$$\begin{tikzcd}
\mathbb{P}_L \arrow[r, "pre_{a,L}"] \arrow[d, "\bar{i}_L"'] & \mathbb{P}_L \arrow[d, "\bar{i}_L"] \\
\mathbb{P}_{L\cup L'} \arrow[r, "pre_{a,L\cup L'}"] & \mathbb{P}_{L\cup L'} \\
\mathbb{P}_{L'} \arrow[r, "pre_{a,L'}"'] \arrow[u, "\bar{i}_{L'}"] & \mathbb{P}_{L'} \arrow[u, "\bar{i}_{L'}"'] .
\end{tikzcd}$$

Hence

$$\begin{array}{rcl}
\bar{i}_{L,!}pre_{a,L,!}X & \cong & pre_{a,L\cup L',!}\bar{i}_{L,!}X \\
& \sim_{L\cup L'} & pre_{a,L\cup L',!}\bar{i}_{L',!}X' \\
& \cong & \bar{i}_{L',!}pre_{a,L',!}X' .
\end{array}$$

□

Recursion: Letting $F : Groth(\mathbb{P}_{(-)}) \to Groth(\mathbb{P}_{(-)})$ be a functor, define $rec(F)$ to be the colimit colim ω_F where

$$\begin{array}{rrcl} \omega_F : & \omega & \to & Groth(\mathbb{P}_{(-)}) \\ & n & \mapsto & F^n(\langle 0, \emptyset \rangle). \end{array}$$

Here 0 is the unique, up to isomorphism, rooted presheaf over $\mathbb{P}_\emptyset$. Any $F^n(\langle 0, \emptyset \rangle)$ consists of a pair $\langle X_n, L_n \rangle$ with $X_n \in |\overline{\mathbb{P}_{L_n}}|$, and we can express the colimit as a pair $\langle X, L \rangle$, where L is the colimit in $\mathbf{Set}_*$ of the L_n and X is the colimit in $\overline{\mathbb{P}_L}$ of all the cocartesian liftings of the X_n, along the edges of the cocone $i_n : L_n \to L$.

As we already noticed the operations in $Groth(\mathbb{P}_{(-)})$ associated with the term constructors are all functors but for $(- \oplus -)$, that nevertheless becomes functorial if one restricts to $Groth(\mathbb{P}_{(-)})_{in}$, the subcategory of $Groth(\mathbb{P}_{(-)})$ with the same object but with morphisms given by pairs $\langle f, i \rangle$ where i is an inclusion of sets. Having $F : Groth(\mathbb{P}_{(-)})_{in} \to Groth(\mathbb{P}_{(-)})_{in}$, we can define $rec(F)$ as above. Actually we shall have now that $L = \cup_n L_n$ and every $i_n : L_n \to L$ will actually be an inclusion of sets.

All our constructions are continuous with respect to ω-chains and restrict to work in $Groth(\mathbb{P}_{(-)})_{in}$, hence $rec(F)$ determines a fixed point if F is deduced from a denotation of a term t as in Section 3.1.1. So the construction above yields a denotation for a recursively defined process in terms of an ω-colimit of presheaves over a common path category. We would like to deduce the bisimulation of recursive processes $rec\ x.t$, $rec\ y.u$ from bisimulation between the open terms t and u. Such open terms give rise to endofunctors on $Groth(\mathbb{P}_{(-)})_{in}$ that includes in $Groth(\mathbb{P}_{(-)})$. Thus, we start by extending the notion of open map, and therefore bisimulation, to functors. Following Definition 3.2.4 and 3.2.9, we start by saying what we mean when asserting that an arrow $\langle f, i \rangle$ in $Groth(\mathbb{P}_{(-)})_{in}$ (and hence in $Groth(\mathbb{P}_{(-)})$) is open.

Definition 3.3.6 *An arrow $\langle f, i \rangle : \langle X, L \rangle \to \langle Y, M \rangle$ in $Groth(\mathbb{P}_{(-)})_{in}$ is open if the transpose $f' : \overline{i}_! X \to Y$ of f, with respect to the adjunction $\overline{i}_! \dashv \overline{i}^*$, is $\mathbb{P}_M$-open.*

Proposition 3.3.7 *If $\langle f, i \rangle : \langle X, L \rangle \to \langle Y, M \rangle$ is open in the sense of Definition 3.3.6 above, then f is $\mathbb{P}_L$-open and Y reduces to L.*

Proof: By the adjunction $f = \overline{i}^*(f')\eta_X$, where η_X is the unit of the adjunction $\overline{i}_! \dashv \overline{i}^*$. But we know that η_X is an isomorphism, since i is a monomorphism (cf. Proposition 3.2.12), hence f is the composite of two open maps and therefore is open. So we have that $X \sim_L \overline{i}^* Y$, hence $\overline{i}_! X \sim_M \overline{i}_! \overline{i}^* Y$. Therefore $Y \sim_M \overline{i}_! X \sim_M \overline{i}_! \overline{i}^* Y$. □

An obvious question is whether the Proposition 3.3.7 above can be made into an "if and only if" statement. This is indeed an intuitive expectation, but to our knowledge is not generally true. One has to put some extra assumption on the presheaf model. For instance using Lemma 6(ii) of [64] one can get the following:

Proposition 3.3.8 *Let $\mathbb{P}_{(-)} : \mathbf{Set}_* \to \mathbf{Cat}$ be a presheaf model such that for any injective total function (i.e., any monomorphism of $\mathbf{Set}_*$), $i : L \to M$ and for any two objects $P \in |\mathbb{P}_L|$ and $Q \in |\mathbb{P}_M|$,*

$$\mathbb{P}_M[Q, \overline{i}P] \neq \emptyset \textit{ only if } Q \cong \overline{i}P' \textit{ for some } P' \in |\mathbb{P}_L| \ .$$

Then, for any $Y \in|\widehat{\mathbb{P}_M}|$ that is essentially on L, the counit, ε_Y, of the adjunction $\overline{i}_! \dashv \overline{i}^*$ is $\mathbb{P}_M$-open.

It is immediately seen now that the proposition above induces the converse of Proposition 3.3.7, since $f' = \varepsilon_Y \overline{i}_! f$. It is worth noticing that both the presheaf models of Example 3.2.2 satisfy the condition required by Proposition 3.3.8.

Back to recursion:

Definition 3.3.9 *Let $F, G : \mathcal{C} \to Groth(\mathbb{P}_{(-)})_{in}$ be two functors. Let $\alpha : F \stackrel{\cdot}{\longrightarrow} G$ be a natural transformation. Say that α is open if for every $c \in|\mathcal{C}|$, α_c is open according to Definition 3.3.6.*

We consider two endofunctors F, G on $Groth(\mathbb{P}_{(-)})_{in}$ bisimilar *if there is another endofunctor R and a span of surjective open natural transformations $\alpha : R \stackrel{\cdot}{\to} F$ and $\beta : R \stackrel{\cdot}{\to} G$ relating them.*

Proposition 3.3.10 *Let $\mathcal{C}$ be a category with initial object 0. Every natural transformation $\alpha : R \stackrel{\cdot}{\longrightarrow} F$, with $R, F : \mathcal{C} \to \mathcal{C}$ endofunctors induces a natural transformation $\omega_\alpha : \omega_R \stackrel{\cdot}{\longrightarrow} \omega_F$ where ω_R (ω_F) are defined inductively by:*

- $\omega_R(0) = 0$ ($\omega_F(0) = 0$)
- $\omega_R(n+1) = R(\omega_R(n))$ ($\omega_F(n+1) = F(\omega_F(n))$)
- $\omega_R(0 \le 1) = 0_{R0}$ ($\omega_F(0 \le 1) = 0_{F0}$)
- $\omega_R(n+1 \le n+2) = R(\omega_R(n \le n+1)$ ($\omega_F(n+1 \le n+2) = F(\omega_F(n \le n+1))$)

Proof: Define inductively

- $(\omega_\alpha)_0 \stackrel{\text{def}}{=} id_0$
- $(\omega_\alpha)_{n+1} \stackrel{\text{def}}{=} \alpha_{F^n 0} R((\omega_\alpha)_n) = F((\omega_\alpha)_n)\alpha_{R^n 0}$ where the second equality holds by naturality of α.

To check that ω_α is a natural transformation, we need to show that the following square commutes for any $n \ge 0$:

$$\begin{tikzcd} R^n 0 \arrow[r, "\omega_R(n \le n+1)"] \arrow[d, "(\omega_\alpha)_n"'] & R^{n+1} 0 \arrow[d, "(\omega_\alpha)_{n+1}"] \\ F^n 0 \arrow[r, "\omega_F(n \le n+1)"'] & F^{n+1} 0 \end{tikzcd}\ .$$

The proof goes, obviously, by induction. The base case follows immediately by initiality of 0. Assume then $n > 0$.

$$\begin{array}{rcl}
(\omega_\alpha)_{n+1}\omega_R(n \leq n+1) & = & (\omega_\alpha)_{n+1}R(\omega_R(n-1 \leq n)) \\
& & \qquad\text{(by definition of } \omega_R) \\
& = & \alpha_{F^n 0}R((\omega_\alpha)_n)R(\omega_R(n-1 \leq n)) \\
& & \qquad\text{(by definition of } \omega_\alpha) \\
& = & \alpha_{F^n 0}R((\omega_\alpha)_n\omega_R(n-1 \leq n)) \\
& & \qquad\text{(by functoriality of } R) \\
& = & \alpha_{F^n 0}R(\omega_F(n-1 \leq n)(\omega_\alpha)_{n-1}) \\
& & \qquad\text{(by inductive hypothesis)} \\
& = & \alpha_{F^n 0}R(\omega_F(n-1 \leq n))R((\omega_\alpha)_{n-1}) \\
& & \qquad\text{(by functoriality of } R) \\
& = & F(\omega_F(n-1 \leq n))\alpha_{F^{n-1}0}R((\omega_\alpha)_{n-1}) \\
& & \qquad\text{(by naturality of } \alpha) \\
& = & \omega_F(n \leq n+1)(\omega_\alpha)_n \\
& & \qquad\text{(by definition)}
\end{array}$$

□

The above proposition, when instantiated to $Groth(\mathbb{P}_{(-)})_{in}$ proves the first part of the following statement. The second part is immediately verified by looking at the definition of ω_α.

Proposition 3.3.11 *Let F, R be endofunctors of $Groth(\mathbb{P}_{(-)})_{in}$ and let $\alpha : R \dot{\rightarrow} X$ be a natural transformation. Then there is a natural transformation $\omega_\alpha : \omega_R \dot{\rightarrow} \omega_F$. Moreover if α is open and X preserve open morphisms, then ω_α is open.*

Open maps are preserved in passing to the colimit, in particular:

Proposition 3.3.12 *Let $F, R : \omega \rightarrow \widehat{\mathbb{P}}$ be functors and $\alpha : R \dot{\rightarrow} F$ a natural transformation such that for every n, α_n is a (surjective) $\mathbb{P}$-open map. Then the map* colim α : colim $R \rightarrow$ colim F, *uniquely determined by the universal property of colimits, is a (surjective) $\mathbb{P}$-open map.*

Proof: Let the following be a commutative square with P and Q objects of $\mathbb{P}$:

$$\begin{tikzcd}
P \arrow[r, "p"] \arrow[d, "m"'] & \mathrm{colim}\ R \arrow[d, "\mathrm{colim}\ \alpha"] \\
Q \arrow[r, "q"'] & \mathrm{colim}\ F\ .
\end{tikzcd} \qquad (3.1)$$

Since colim R and colim F are colimits of ω-chains there exists a number n and arrows

$$p_n : P \rightarrow R(n) \qquad \text{and} \qquad q_n : Q \rightarrow F(n)$$

such that the following diagram commutes:

$$\begin{tikzcd}
P \arrow[r, "p_n"] \arrow[d, "m"'] \arrow[rr, bend left, "p"] & R(n) \arrow[r, "R_{n,\infty}"] \arrow[d, "\alpha_n"] & \text{colim } R \arrow[d, "\text{colim } \alpha"] \\
Q \arrow[r, "q_n"'] \arrow[rr, bend right, "q"'] & F(n) \arrow[r, "F_{n,\infty}"'] & \text{colim } F
\end{tikzcd} ,$$

where $R_{n,\infty}$ and $F_{n,\infty}$ are edges of the corresponding colimiting cones. By assumption, α_n is $\mathbb{P}$-open, hence there exists $r_n : Q \to R(n)$ splitting the leftmost square in two commutative triangles. Then

$$r \stackrel{\text{def}}{=} R_{n,\infty} r_n$$

is an arrow from Q to colim R that splits the Diagram (3.1) in two commutative triangles.

If every α_n is an epimorphic natural transformation, then obviously, since colimits in presheaf categories are calculated pointwise, colim α is epimorphic as well. □

Since the calculation of colimits of ω-chains in $Groth(\mathbb{P}_{(-)})_{in}$ is reduced to calculating them in the colimiting fibre, the above proposition yields the following:

Proposition 3.3.13 *Let $\omega_F, \omega_R : \omega \to Groth(\mathbb{P}_{(-)})_{in}$ and $\omega_\alpha : \omega_R \dot{\to} \omega_F$, be as in Proposition 3.3.11, with ω_α open, then the (adjoint transpose of the) arrow*

$$\text{colim } \omega_\alpha : \text{colim } \omega_R \to \text{colim } \omega_F \ ,$$

uniquely determined by the universal property of the colimit, is an open map in the fibre over the colimiting labelling set.

Proof: We have already remarked that colimits of ω-chains

$$\cdots \to \langle X_n, L_n \rangle \to \langle X_{n+1}, L_{n+1} \rangle \to \cdots \ ,$$

in $Groth(\mathbb{P}_{(-)})_{in}$ are obtained by taking first the union

$$L = \cup_{n \in \omega} L_n$$

of the all the labelling sets in the chain and then calculating the colimit of the chain induced in the fibre over L, by cocartesian liftings of all the X_n. We now make this explanation more precise, in order to show that the functor part of colim α arises from a situation satisfying the hypothesis of Proposition 3.3.12. We need some notation first. Let's write $\langle R_n, L_n \rangle$ for $\omega_R(n)$, $\langle F_n, M_n \rangle$ for $\omega_F(n)$. For every n, let $\langle r_n, i_n \rangle$ and $\langle f_n, j_n \rangle$ be $\omega_R(n \leq n+1)$ and $\omega_F(n \leq n+1)$, where, for simplicity we already assume that $r_n : \overline{(i_n)}_! R_n \to R_{n+1}$ and $f_n : \overline{(j_n)}_! F_n \to F_{n+1}$, rather than taking their transposes. For every n, let $\langle \alpha_n, k_n\rangle$ be $(\omega_\alpha)_n$, where again we take $\alpha_n : \overline{(k_n)_!}R_n \to F_n$. Naturality of ω_α means that the following square commutes (where the indicated isomorphisms are uniquely determined by the universal property of left Kan extensions):

$$\begin{tikzcd}
\overline{(j_nk_n)_!}R_n = \overline{(k_{n+1}i_n)_!}R_n \arrow[r, "\simeq"] \arrow[d, "\cong"'] & \overline{(k_{n+1})_!}\,\overline{(i_n)_!}R_n \arrow[r, "\overline{(k_{n+1})_!}r_n"] & \overline{(k_{n+1})_!}R_{n+1} \arrow[dd, "\alpha_{n+1}"] \\
\overline{(j_n)_!}\,\overline{(k_n)_!}R_n \arrow[d, "\overline{(j_n)_!}\alpha_n"'] & & \\
\overline{(j_n)_!}F_n \arrow[rr, "f_n"'] & & F_{n+1}\ .
\end{tikzcd}$$

Now, writing for every n

$$i_{n,\infty} : L_n \to L = \cup_{n\in\omega} L_n\ ,$$

and $\langle R_\infty, L\rangle$ for colim ω_R, we have that R_∞ can be calculated as the colimit of the following chain in $\widehat{\mathbb{P}_L}$:

$$\cdots \longrightarrow \overline{(i_{n,\infty})_!}R_n \xrightarrow{\ \simeq\ } \overline{(i_{n+1,\infty})_!}\,\overline{(i_n)_!}R_n \xrightarrow{\overline{(i_{n+1,\infty})_!}r_n} \overline{(i_{n+1,\infty})_!}R_{n+1} \longrightarrow \cdots\ .$$

Similarly one calculates colim F, moreover from the commutativity of the above diagram it follows that for every n, the following diagram commutes too (where $k : L \hookrightarrow M$ is the inclusion function):

$$\begin{tikzcd}
\overline{k}_!\overline{(i_{n,\infty})_!}R_n \arrow[r, "\simeq"] \arrow[d, "\cong"'] & \overline{k}_!\overline{(i_{n+1,\infty})_!}\,\overline{(i_n)_!}R_n \arrow[r, "\overline{k}_!\overline{(i_{n+1,\infty})_!}r_n"] & \overline{k}_!\overline{(i_{n+1,\infty})_!}R_{n+1} \arrow[d, "\cong"] \\
\overline{(j_{n,\infty}k_n)_!} = \overline{(ki_{n,\infty})_!}R_n \arrow[d, "\cong"'] & & \overline{(j_{n+1,\infty}k_{n+1})_!} = \overline{(ki_{n+1,\infty})_!}R_{n+1} \arrow[d, "\cong"] \\
\overline{(j_{n,\infty})_!}\,\overline{(k_n)_!}R_n \arrow[d, "\overline{(j_{n,\infty})_!}\alpha_n"'] & & \overline{(j_{n+1,\infty})_!}\,\overline{(k_{n+1})_!}R_{n+1} \arrow[d, "\overline{(j_{n+1,\infty})_!}\alpha_{n+1}"] \\
\overline{(j_{n,\infty})_!}F_n \arrow[r, "\simeq"] & \overline{(j_{n+1,\infty})_!}\,\overline{(j_n)_!}F_n \arrow[r, "\overline{(j_{n+1,\infty})_!}f_n"] & \overline{(j_{n+1,\infty})_!}F_{n+1}\ .
\end{tikzcd}$$

Hence $\alpha_\infty : \overline{k}_!R_\infty \to F_\infty$ is the unique mediating morphism between two colimiting cones connected by a natural transformation that is pointwise an open map, since every vertical arrow in the diagram above is either an isomorphism (hence an open map) or the transformation along a colimit preserving functor of an open map (hence an open map because of Proposition 3.2.5). Therefore we fall within the hypothesis of Proposition 3.3.12 and so α_∞ is $\mathbb{P}_M$-open. □

Consequently, if two endofunctors F, G ranging over $Groth(\mathbb{P}_{(-)})_{in}$ are bisimilar and preserve open maps, then the colimits $rec(F)$, $rec(G)$ are bisimilar. A term with a free variable, built-up from the constructions of this section, will determine an endofunctor on $Groth(\mathbb{P}_{(-)})_{in}$ which preserves open maps by this section propositions. It follows that if two open terms t and u are bisimilar, *i.e.* induce bisimilar functors, then the recursive definitions $rec\, x.t$ and $rec\, y.u$ are bisimilar.

3.4 Concrete models revisited

We wrote that a motivation for moving up to abstract models was the hope of being able to deduce general congruence properties, in the abstract setting, and then transfer them back to the concrete cases.

We have already mentioned the full embeddings

$$\mathcal{ST}_L \hookrightarrow \overline{L^*}$$
$$\mathcal{ES}_L \hookrightarrow \overline{\mathbf{Pom}_L}$$

The first one is actually an equivalence of categories, while the second one is a strict inclusion (for instance the terminal presheaf cannot be represented as an event structure) that not only preserves but reflects bisimulation (see Proposition 2.2.12). We consider the presheaf models $Groth((-)^*)$ and $Groth(\mathbf{Pom}_{(-)})$ of Example 3.2.2. We can now transfer the results from the presheaf models to the concrete models of synchronisation trees and event structures by noting that the canonical embeddings between fibres $\mathcal{ST}_L \to \widehat{L^*}$ and $\mathcal{ES}_L \to \widehat{\mathbf{Pom}_L}$ extend to full and faithful embeddings from $\mathcal{ST}$ and $\mathcal{ES}$ to the corresponding presheaf models. In particular we again have that the embedding $\mathcal{ST} \hookrightarrow Groth((-)^*)$ is an equivalence. We illustrate then the situation with event structures.

We have already recalled in Proposition 2.1.3 that the functor $p_{\mathcal{ES}} : \mathcal{ES} \to \mathbf{Set}_*$ is a cofibration. We provide now a choice of a cleavage that makes it into a split cofibration, that is one for which the associated pseudo-functor (cf. Section 1.4.1) is functorial on the nose and not just up to isomorphism.

Let $E = (E, \leq, Con, l : E \to L)$ be an event structure and let $\alpha : L \rightharpoonup L'$ be a partial function. define $\alpha_!(E) = (E', \leq', Con', l' : E' \to L')$ as follows:

- $E' = \{e \in E \mid \alpha(l(e)) \text{ is defined}\}$
- $\leq' = \leq \cap (E' \times E')$
- $Con' = \{x \in Con \mid x \subseteq E'\}$
- $l'(e') = \alpha(l(e'))$, for every $e' \in E'$.

It is straightforward to verify that $\alpha_!(E)$ is an event structure and that the pair $\langle 1', \alpha \rangle$, with $1' : E \rightharpoonup E'$ acting as the identity and defined only on those events of E that belongs to E' as well, is an event structure morphism.

The following two propositions have straightforward verifications.

Proposition 3.4.1 *Given an event structure $E = (E, \leq, Con, l : E \to L)$ and a partial function $\alpha : L \rightharpoonup L'$, the event structures morphism $\langle 1', \alpha \rangle$ is a cocartesian arrow.*

We shall write $\alpha_{!,E}$ for $\langle 1', \alpha\rangle$. There is an induced cocartesian lifting functor $\alpha_! : \mathcal{ES}_L \to \mathcal{ES}_{L'}$. Observe that if we restrict the construction above to pomsets we capture exactly the functor $\overline{\alpha} : \mathbf{Pom}_L \to \mathbf{Pom}_{L'}$ of Example 3.2.2.

The following is easily verifiable:

Proposition 3.4.2 *Given an event structure $E = (E, \leq, Con, l : E \to L)$ and a partial function $\alpha : L \rightharpoonup L'$ then the following holds:*

1. *If $\alpha = 1_L$, then $\alpha_{!,E}$ is the identity morphism $\langle 1_E, 1_L\rangle : E \to E$.*
2. *If $\beta : L' \to L''$ is another partial function, then $\beta_{!,\alpha_!(E)}\alpha_{!,E} = (\beta\alpha)_{!,E}$.*

Corollary 3.4.3 *There exists a functor $(-)_! : \mathbf{Set}_* \to \mathbf{Cat}$, that induces the cofibration $p_{\mathcal{ES}} : \mathcal{ES} \to \mathbf{Set}_*$ by means of the Grothendieck construction.*

NOTATION: If $\langle f, \alpha\rangle : E \to \overline{E}$ is an even structure morphism, write f_α for the unique function such that $\langle f, \alpha\rangle = \langle f_\alpha, 1'_L\rangle\alpha_{!,E}$, that is f_α is the restriction of f to the elements of $\alpha_!(E)$ that, by the way, is equal to the set $\{e \in E \mid f(e) \text{ is defined}\}$. Call f_α the transpose of f.

With this notation in mind, define $c : \mathcal{ES} \to Groth(\mathbf{Pom}_{(-)})$ to be

- **On objects:** $c(E, \leq, Con, l : E \to L) = \langle c_L(E), L\rangle$
- **On arrows:** If $\langle f, \alpha\rangle : E \to E'$ with $\alpha : L \rightharpoonup L'$, then $c(\langle f, \alpha\rangle) = \langle c(f), \alpha\rangle$ where

$$c(f) : c_L(E) = \mathcal{ES}_L[-, E] \to \mathcal{ES}_{L'}[\overline{\alpha}(-), E'] = \overline{\alpha}^*(c_{L'}(E'))$$

  is defined by composition and transposition (recall that on pomsets, $\alpha_!$ is another name for $\overline{\alpha}$), i.e., $c(f)_P(p) = (fp)_\alpha$.

This defines a functor because from Proposition 3.4.2 one has that $(1_E p)_{1_L} = p$ and $(g(fp)_\alpha)_\beta = (gfp)_{\beta\alpha}$ and from these equalities one deduces that $c\langle 1_E, 1_L\rangle = \langle 1_E, 1_L\rangle$ and $c(\langle g, \beta\rangle\langle f, \alpha\rangle) = c(\langle gf, \beta\alpha\rangle)$. Moreover, again form Proposition 3.4.2, one sees that for any arrow $\langle f, 1_L\rangle : E \to E'$, $(fp)_{1_L} = fp$, hence c acts as c_L when restricted to $\mathcal{ES}_L$.

Proposition 3.4.4 *The functor $c : \mathcal{ES} \to Groth(\mathbf{Pom}_-)$ is a dense full embedding.*

Proof: Straightforward from the fact that c extends the c_L's that were dense full embeddings and the fact that via cocartesian liftings, every arrow between objects of $Groth(\mathbf{Pom}_-)$ in different fibres is uniquely determined by an arrow in a fibre. $\square$

It is a known fact [144] that every dense full embedding preserves limits. Moreover a direct calculation would show that c respects relabelling (i.e., cocartesian liftings) and cartesian liftings of inclusions.

Proposition 3.4.5 *Let $\alpha : L \rightharpoonup M$ be a partial function, then there is a natural isomorphism*

$$c_M\alpha_! \cong \overline{\alpha}_! c_L \ .$$

Let $\lambda : L \hookrightarrow M$ be an inclusion map, then

$$c_L\lambda^* \cong \overline{\lambda}^* c_M \ ,$$

where λ^* is the right adjoint of $\lambda_!$ defined on objects as follows:

$$\lambda^*(E, \leq, Con, l) = (E', \leq', Con', l') \ ,$$

where $E' = \{e \in E | \forall e' \leq e \, \exists a \in L \, \lambda(a) = l(e')\}$, $\leq' = \leq \cap E' \times E'$, $Con' = \{x \in Con \mid x \subseteq E'\}$ and $l'(e) = a$, where a is the unique element of L, such that $\lambda(a) = l(e)$.

We have already noticed that c_L preserves coproducts (Proposition 2.2.13). After Proposition 1.4.14 we know that coproducts in a cofibred category are built using coproducts in the fibres and cocartesian liftings, hence c preserves coproducts. Summarising:

Proposition 3.4.6 *The embedding $c : \mathcal{ES} \to Groth(\mathbf{Pom}_{-})$ preserves all limits that exists in $\mathcal{ES}$, coproducts, cocartesian liftings and cartesian liftings of inclusion.*

A denotational semantics of **Proc** in $\mathcal{ES}$ was given in [141] and corresponds to the one described abstractly in Section 3.1.1. Proposition 3.4.6 above ensure that (after the embedding with c) the semantics in $\mathcal{ES}$ correspond to the one in $Groth(\mathbf{Pom}_{-})$, hence the following:

Theorem 3.4.7 *Let $\mathcal{ES}[\![\cdot]\!]$ and $Groth(\mathbf{Pom}_{-})[\![\cdot]\!]$ stands for corresponding semantics of* **Proc** *. Let $\rho : Vars \to |\mathcal{ES}|$ be an environment function, then*

$$c(\mathcal{ES}[\![t]\!]_{\rho}) \cong Groth(\mathbf{Pom}_{-})[\![t]\!]_{c\circ\rho} \ .$$

By Proposition 2.2.12(ii), open maps and bisimulation coincide, via the canonical embeddings, in $\mathcal{ES}_L$ and the fibre over L in $Groth(\mathbf{Pom}_{(-)})$. Hence we can transfer the congruence property deduced for the presheaf semantics to deduce, in particular, that strong history-preserving bisimulation is a congruence for the language **Proc**.

Theorem 3.4.8 *Strong history preserving bisimulation for event structure is a congruence for the semantics of the language* **Proc** *.*

3.5 Refinement for event structures

As a further example of an application of Corollary 3.2.6 we give here a proof of the fact that a refinement operator for event structures proposed in [41] preserves strong history preserving bisimulation (shpb for short).

Definition 3.5.1 (cf. [41], Section 2) *A* refinement function

$$r : L \to |\mathbf{Pom}_M| - \{0_M\}$$

is a map that takes any element of $a \in L$ to a non empty pomset $r(a)$ over M.

Definition 3.5.2 (A refinement functor) *A refinement function as in the definition above induces a* refinement functor

$$R : \mathbf{Pom}_L \to \mathbf{Pom}_M$$

acting as follows:

- **On objects:** *If* $P = (P, \leq_P, \lambda_P, L)$ *is a pomset over* L*, then define* $R(P) = (R(P), \leq_{R(P)}, \lambda_{R(P)}, M)$ *with:*
  - $R(P) = \{(x, x') | x \in P \wedge x' \in r(\lambda_P(x))\}$
  - $(x, x') \leq_{R(P)} (y, y')$ *if either* $x \leq_P y$ *and* $x \neq y$ *or* $x = y$ *and* $x' \leq_{r(\lambda_P(x))} y'$
  - $\lambda_{R(P)}(x, x') = \lambda_{r(\lambda_P(x))}(x')$.
- **On arrows:** *If* $f : P \to Q$ *in* $\mathbf{Pom}_L$*, define* $R(f)(x, x') = (f(x), x')$.

One can actually see r *inducing a refinement functor, say* $R^{\mathcal{ES}}$*, on event structures as well. If* $(E, \leq, Con, l)$ *is an event structure over* L*,* $R^{\mathcal{ES}}(E)$ *is defined on* E*,* $\leq$ *and* l *as for pomsets, while* $X \in Con_{R^{\mathcal{ES}}(E)}$ *iff* $\{x \in E \mid \exists x' .(x, x') \in X\} \in Con$.

As remarked in [64], the functor $R_!$, obtained as a left Kan extension, is a good candidate for the extension of this refinement to presheaves including those corresponding to event structures. But does the functor $R_!$ act like the operation of refinement $R^{\mathcal{ES}}$ on event structures? More precisely, if we let $c_L : \mathcal{ES}_L \rightharpoonup \widehat{\mathbf{Pom}_L}$ and $c_M : \mathcal{ES}_M \rightharpoonup \widehat{\mathbf{Pom}_M}$ denote the canonical embeddings, do we have that the following square commutes (up to a natural isomorphism)?

$$\begin{tikzcd}
\mathcal{ES}_L \arrow[r, "c_L"] \arrow[d, "R^{\mathcal{ES}}"'] & \widehat{\mathbf{Pom}_L} \arrow[d, "R_!"] \\
\mathcal{ES}_M \arrow[r, "c_M"'] & \widehat{\mathbf{Pom}_M}
\end{tikzcd}$$

The answer is yes and we embark now on proving it.

Lemma 3.5.3 *Let* E *be an event structure in* $\mathcal{ES}_L$*, let* Q *be a pomset over* M *and let* $R^{\mathcal{ES}} : \mathcal{ES}_L \to \mathcal{ES}_M$ *be a refinement functor. Then for any* $q : Q \to R^{\mathcal{ES}}(E)$ *there exists a pomset* $P_q \in |\mathbf{Pom}_L|$ *and a morphism* $p : P \to E$ *such that*

- *There exists a morphism* $p_q : Q \to R^{\mathcal{ES}}(P) = R(P)$ *such that* $q = R^{\mathcal{ES}}(p)p_q$.
- *For any other factorisation*

$$\begin{tikzcd}
Q \arrow[rr, "q"] \arrow[dr, "p'_q"'] & & R^{\mathcal{ES}}(E) \\
& R^{\mathcal{ES}}(P') \arrow[ur, "R^{\mathcal{ES}}(p')"'] &
\end{tikzcd}$$

there exists a unique mediating morphism of pomsets, $m : P \to P'$*, such that*

$$p'_q = R^{\mathcal{ES}}(m)p_q \qquad \text{and} \qquad p'm = p\ .$$

Proof: Define $P = \{e \in E | \exists (e, f) \in R(E)\, \exists y \in Q\, q(y) = (e, f)\}$, with the order relation induced by Q, i.e., $e \leq_P e'$ if either $e = e'$ or there exist $y \leq_Q y'$ with $q(y) = (e, f)$ and $q(y') = (e', f')$. The verification of the properties is straightforward. $\square$

Proposition 3.5.4 *Let $i_L : \mathbf{Pom}_L \to \mathcal{ES}_L$ and $i_M : \mathbf{Pom}_M \to \mathcal{ES}_M$ be the inclusion functors, then*

$$R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(i_M \circ R).$$

Proof: Recall that $\mathbf{Pom}_L$ is dense in $\mathcal{ES}_L$, i.e., for every $E \in |\mathcal{ES}_L|$,

$$E \cong \mathrm{colim}\ i_L/E \to \mathbf{Pom}_L \xrightarrow{i_L} \mathcal{ES}_L.$$

Using the Lemma 3.5.3 above it is not difficult to verify that

$$R^{\mathcal{ES}}(E) \cong \mathrm{colim}\ i_L/E \to \mathbf{Pom}_L \xrightarrow{R} \mathbf{Pom}_M \xrightarrow{i_M} \mathcal{ES}_M\ . \tag{3.2}$$

From this we can deduce that $R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(i_M \circ R)$. In fact $R^{\mathcal{ES}} \circ i_L = i_M \circ R$ and moreover if $F : \mathcal{ES}_L \to \mathcal{ES}_M$ is a functor and $\alpha : F \circ i_L \stackrel{\cdot}{\longrightarrow} R \circ i_m$ is a natural transformation, there exists a unique $\beta : R^{\mathcal{ES}} \stackrel{\cdot}{\longrightarrow} F$ such that

$$\beta_{i_L} = \alpha. \tag{3.3}$$

To show it observe first of all that if β is a natural transformation satisfying (3.3), then for any $E \in |\mathcal{ES}_L|$ and $f : P \to E$,

$$\beta_E \circ R^{\mathcal{ES}}(f) = F(f) \circ \alpha_P\ . \tag{3.4}$$

In fact

$$\begin{array}{rcll} \beta_E \circ R^{\mathcal{ES}}(f) & = & F(f) \circ \beta_{i_L(P)} & \text{(by naturality of } \beta) \\ & = & F(f) \circ \alpha_P & \text{(by equality (3.3)).} \end{array}$$

But since, for any $E \in |\mathcal{ES}_L|$, (3.2) holds, there exists a unique $\beta_E : R^{\mathcal{ES}}(E) \to F(E)$ satisfying $\beta_E \circ R^{\mathcal{ES}}(f) = F(f) \circ \alpha_P$. Commutativity of the naturality squares follows as well from the universal property of colimits. We need to prove that for any $g : E \to E'$ in $\mathcal{ES}_L$, the following diagram commutes:

$$\begin{tikzcd} R^{\mathcal{ES}}(E) \arrow[r, "\beta_E"] \arrow[d, "R^{\mathcal{ES}}(g)"'] & F(E) \arrow[d, "F(g)"] \\ R^{\mathcal{ES}}(E') \arrow[r, "\beta_{E'}"'] & F(E') \end{tikzcd}$$

It is enough to show that for any $f : P \to E$, $F(g) \circ \beta_E \circ R^{\mathcal{ES}}(f) = \beta_{E'} \circ R^{\mathcal{ES}}(g) \circ R^{\mathcal{ES}}(f)$. This follows from the following calculation:

$$\begin{array}{rcll} F(g) \circ \beta_E \circ R^{\mathcal{ES}}(f) & = & F(g) \circ F(f) \circ \alpha_P & \text{(by the equality (3.4)} \\ & = & F(gf) \circ \alpha_P & \text{(by functoriality of } F) \\ & = & \beta_{E'} \circ R^{\mathcal{ES}}(gf) & \text{(by equality (3.4))} \\ & = & \beta_{E'} \circ R^{\mathcal{ES}}(g) \circ R^{\mathcal{ES}}(f) & \text{(by functoriality of } R^{\mathcal{ES}}) \end{array}$$

□

We then have a functor $R : \mathbf{Pom}_L \to \mathbf{Pom}_M$ that can be extended as follows:

$$\begin{tikzcd}
\mathbf{Pom}_L \arrow[rr, "i_L"] \arrow[dr, "R"'] & & \mathcal{ES}_L \arrow[r, "c_L"] \arrow[dd, "R^{\mathcal{ES}}=\mathrm{Lan}_{i_L}(i_M R)"] & \widehat{\mathbf{Pom}_L} \arrow[dd, "\mathrm{Lan}_{\mathrm{y}_{\mathbf{Pom}_L}}(\mathrm{y}_{\mathbf{Pom}_M} R)=R_!"] \\
& \mathbf{Pom}_M \arrow[dr, "i_M"'] & & \\
& & \mathcal{ES}_M \arrow[r, "c_M"'] & \widehat{\mathbf{Pom}_M}
\end{tikzcd} \ .$$

We want to show that the square on the right commutes up to a natural isomorphism. We show first of all that

$$c_M R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(c_M i_M R) = \mathrm{Lan}_{i_L}(\mathrm{y}_{\mathbf{Pom}_M} R) \ .$$

From this in fact it will follow that (using that left Kan extensions compose, cf. Section 1.3),

$$\begin{aligned} \mathrm{Lan}_{\mathrm{y}_{\mathbf{Pom}_L}}(\mathrm{y}_{\mathbf{Pom}_M} R) &\cong \mathrm{Lan}_{c_L}(\mathrm{Lan}_{i_L}(\mathrm{y}_{\mathbf{Pom}_M} R) \\ &\cong \mathrm{Lan}_{c_L}(c_M R^{\mathcal{ES}}) \ . \end{aligned}$$

Hence, since c_L is full and faithful

$$\mathrm{Lan}_{\mathrm{y}_{\mathbf{Pom}_L}}(\mathrm{y}_{\mathbf{Pom}_M} R) c_L \cong \mathrm{Lan}_{c_L}(c_M R^{\mathcal{ES}}) c_L \cong c_M R^{\mathcal{ES}} \ .$$

To prove that $c_M R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(c_M i_M R)$, we apply Proposition 1.2.9.

Proposition 3.5.5 *There is a natural isomorphism*

$$c_M R^{\mathcal{ES}} \cong \mathrm{Lan}_{i_L}(c_M i_M R) \ .$$

Proof: The proof is an immediate consequence of Proposition 1.2.9 and of Lemma 3.5.3. In fact, as we saw in the proof of Proposition 3.5.4, $\mathrm{Lan}_{i_L}(i_M R)$ can be expressed as the colimit (3.2) and Lemma 3.5.3 ensures that the condition of Proposition 1.2.9 are met. □

Now we can use again Corollary 3.2.6 to deduce that $R_!$ preserves open map bisimulation.

Proposition 3.5.6 *For any refinement function $r : L \to \mid \mathbf{Pom}_M \mid$, the associated refinement functor $R_! : \widehat{\mathbf{Pom}_L} \to \widehat{\mathbf{Pom}_M}$ preserves open map bisimulation;*

> *If X and Y are two $\mathbf{Pom}_L$-open bisimilar presheaves, then $R_!(X)$ and $R_!(Y)$ are $\mathbf{Pom}_M$-open bisimilar.*

As a consequence, using Proposition 2.2.12, we have the following:

Corollary 3.5.7 *For any refinement function $r : L \to \mid \mathbf{Pom}_M \mid$, the associated refinement functor $R^{\mathcal{ES}} : \mathcal{ES}_L \to \mathcal{ES}_M$ preserves strong history preserving bisimulation;*

If E and E' are two strong history preserving bisimilar event structures in $\mathcal{ES}_L$, then $R^{\mathcal{ES}}(E)$ and $R^{\mathcal{ES}}(E')$ are strong history preserving bisimilar.

Proof:

$$\begin{array}{rcl} E \text{ shpb } E' & \Longrightarrow & c_L(E)\ \mathbf{Pom}_L\text{-open bisimilar to } c_L(E') \\ & \Longrightarrow & R_! c_L(E)\ \mathbf{Pom}_M\text{-open bisimilar to } R_! c_L(E') \\ & \Longleftrightarrow & c_M R^{\mathcal{ES}}(E)\ \mathbf{Pom}_M\text{-open bisimilar to } c_M R^{\mathcal{ES}}(E') \\ & \Longleftrightarrow & R^{\mathcal{ES}}(E) \text{ shpb } R^{\mathcal{ES}}(E')\ . \end{array}$$

□

Chapter 4

Profunctors

As we have seen presheaf categories provide an abstract setting for proving congruence results that can be transferred to traditional models. Here we study a way to organise presheaf categories into a (bi)category themselves and how its maps respect bisimulation. The result can be viewed as a form of domain theory in which traditional domains as partial orders are replaced by domains as presheaf categories. More details of this new domain theory will be presented in Chapter 6, here we introduce the basic bicategory that shall serve as our category of domains, analyse its structure and prove important congruence results with respect to the bisimulation relation.

4.1 Left Kan extensions via coend formulae

We start by recalling from [76] the notion of coend that leads to a useful way of calculating left Kan extensions [76, 65, 56]. Coends are special kinds of colimits defined by universal wedges in place of universal cocones. The notion of a wedge is in turn connected to that of dinatural transformation.

Definition 4.1.1 (Dinatural transformations [76], page 214) *Let*

$$F, G : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{D}$$

be two functors. A dinatural transformation $\alpha : F \overset{\cdot\cdot}{\longrightarrow} G$ *from F to G consists of a family of arrows $(\alpha_c : F(c,c) \to G(c,c))_{c\in\mathcal{C}}$, such that for every arrow of $\mathcal{C}$, $f : c \to c'$ the following hexagonal diagram commutes:*

$$\begin{tikzcd}
 & F(c,c) \arrow[r, "\alpha_c"] & G(c,c) \arrow[dr, "G(1_c,f)"] & \\
F(c',c) \arrow[ur, "F(f,1_c)"] \arrow[dr, "F(1_{c'},f)"'] & & & G(c,c') \\
 & F(c',c') \arrow[r, "\alpha_{c'}"'] & G(c',c') \arrow[ur, "G(f,1_{c'})"'] & .
\end{tikzcd}$$

NOTATION: Any object d of $\mathcal{D}$, gives rise to a constant functor, $d : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{D}$, always returning d on objects and 1_d on arrows.

Definition 4.1.2 (Wedges) *Let $F : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{D}$ be a functor and let d be an object of $\mathcal{D}$. A* wedge *from F to d is a dinatural transformation $\alpha : F \stackrel{..}{\longrightarrow} d$. In other words such a wedge consists of components $\alpha_c : F(c,c) \to d$, such that for any $f : c \to c'$ the following square commutes:*

$$\begin{tikzcd} F(c',c) \arrow[r, "F(1_{c'},f)"] \arrow[d, "F(f,1_c)"'] & F(c',c') \arrow[d, "\alpha_{c'}"] \\ F(c,c) \arrow[r, "\alpha_c"'] & d \end{tikzcd} .$$

Definition 4.1.3 (Coends) *A* coend *of a functor $F : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{D}$ is a universal wedge of F, i.e., it consists of a pair (d,ω) where $d \in |\mathcal{D}|$ and ω is a wedge from F to d such that, given any other wedge $\alpha : F \stackrel{..}{\longrightarrow} d'$, there exists a unique arrow $h : d \to d'$ such that for every $c \in |\mathcal{C}|$, $\alpha_c = h\omega_c$. As usual with colimits (and limits), by abuse of language the object d itself will be often call the coend of F and written with the integral notation:*

$$d = \int^c F(c,c) = \text{Coend of } F \ .$$

More generally one can reduce the existence (and calculation) of colimits to the existence (and calculation) of coends.

Proposition 4.1.4 ([76]) *Let $\mathcal{C}$ be any category, then $\mathcal{C}$ has all small colimits if and only if has all small coends.*

In particular the colimit of a functor $F : \mathbb{D} \to \mathcal{C}$, can be calculated as the coend $\int^d F\pi_2(d,d)$, where $\pi_2 : \mathbb{D}^{\mathrm{op}} \times \mathbb{D} \to \mathbb{D}$ is the obvious projection functor. This means that we are then allowed to write the colimit of a functor F as a coend, without explicitly mentioning the first (dummy) variable, i.e., we can (and we shall often do so in the sequel) write

$$\int^d F(d)$$

for colim F. The following two results from [76, 56] show some formal advantages of the integral notation.

Theorem 4.1.5 (Parametricity) *If $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \times \mathbb{D} \to \mathcal{C}$ is a functor such that for every $d \in |\mathbb{D}|$, $\int^c F(c,c,d)$ exists. Then, for a choice of coends, the mapping*

$$d \longmapsto \int^c F(c,c,d) \ ,$$

extends uniquely to a functor

$$\int^c F(c,c,-) : \mathbb{D} \to \mathcal{C} \ .$$

In line with the notation of the Theorem above, we shall write

$$\int^{c} F(c,c,f) : \int^{c} F(c,c,d) \to \int^{c} F(c,c,d')$$

for the action of the functor above on the arrows $f : d \to d'$ of $\mathbb{D}$.

Theorem 4.1.6 (Fubini) *If $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \times \mathbb{D}^{\mathrm{op}} \times \mathbb{D} \to \mathcal{C}$ is a functor, then*

$$\int^{c} \int^{d} F(c,c,d,d) \cong \int^{c,d} F(c,c,d,d) \cong \int^{d} \int^{c} F(c,c,d,d),$$

to be understood as meaning that if one of the three coends exists then so do the other two and they are isomorphic.

Since the calculation of colimits can be reduced to the calculation of coends, we can in particular give a description of (pointwise) left Kan extensions in terms of coends (Exercise 4, p. 239 of [76]). We begin with a simple observation.

Proposition 4.1.7 *The category of functors $F : \mathbb{P} \to \widehat{\mathbb{Q}}$ and natural transformations between them, $\mathbf{CAT}(\mathbb{P}, \widehat{\mathbb{Q}})$, is isomorphic to the category of functors $F : \mathbb{P} \times \mathbb{Q}^{op} \to \mathbf{Set}$, and natural transformations between them, $\mathbf{CAT}(\mathbb{P} \times \mathbb{Q}^{op}, \mathbf{Set})$.*

Given a functor $F : \mathbb{P} \to \widehat{\mathbb{Q}}$, we will talk of the *exponential transpose* $\overline{F}$ of F to indicate the functor $\overline{F} : \mathbb{P} \times \mathbb{Q}^{\mathrm{op}} \to \mathbf{Set}$ that corresponds to F in the isomorphism above. Similarly for a functor $G : \mathbb{P} \times \mathbb{Q}^{\mathrm{op}} \to \mathbf{Set}$, we write $\overline{G}$ for the corresponding functor from $\mathbb{P}$ to $\widehat{\mathbb{Q}}$.

Definition 4.1.8 *Let A be a set and X be a presheaf over a category $\mathbb{P}$. Define the* copower *$A \,.\, X$ to be the presheaf $P \mapsto A \times X(P)$, with obvious morphism action.*

Put in a different way the copower $A \,.\, X$ is isomorphic to $\sum_{a \in A} X$ and to the product $A' \times X$ where A' is the presheaf that is constantly A on objects and identities on arrows. Suppose then, that we have a functor $F : \mathbb{P} \to \widehat{\mathbb{Q}}$. Note that using the copower construction we obtain from X and F the following functor:

$$\mathbb{P}^{op} \times \mathbb{P} \xrightarrow{X \,.\, F} \widehat{\mathbb{Q}} \ ,$$

that is defined on objects as

$$(X \,.\, F)(P, P') \stackrel{\mathrm{def}}{=} X(P) \,.\, F(P')$$

and on arrows as

$$((X \,.\, F)(f^{\mathrm{op}}, g))_Q(x, y) \stackrel{\mathrm{def}}{=} (X(f)x, F(g)_Q(y)) \ .$$

Given this, one has the following natural isomorphism:

$$\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(F)(X) \cong \int^{P} X(P) \,.\, F(P) \ .$$

This means that for every Q object of $\mathbb{Q}$, one has

$$\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(F)(X)(Q) \cong \int^{P} X(P) \times F(P)(Q) \ , \tag{4.1}$$

with the action on morphisms, $q : Q \to Q'$, written as

$$\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(F)(q) = \int^{P} X(P) \times F(P)(q) \ .$$

Finally, given $\alpha : F \stackrel{\cdot}{\longrightarrow} F'$, one has

$$\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(\alpha)_Q = \int^{P} X(P) \times \alpha_{\langle P,Q \rangle}.$$

Since the left Kan extension of a functor along itself always exists and is equal to the identity, we can restate the density of the Yoneda embedding (see Proposition 1.2.17) in terms of the following *density formula*:

$$X \cong \int^{P} X(P) \,.\, \mathrm{y}_{\mathbb{P}}(P) \ .$$

Following what we said here about coends and in Section 1.3 the coend description of Kan extension is a sound one for any cocomplete category $\mathcal{V}$. The extra feature that **Set** has (for instance when computing the colimit at a "point" Q as in equation (4.1)) is that we can compute such coends, being colimits, as appropriate equivalence relations.[1] This is exactly what we are going to describe now:

$$\int^{P} X(P) \times \overline{F}(P,Q) \cong \coprod_{P \in |\mathbb{P}|} X(P) \times \overline{F}(P,Q)/\sim$$

where $\sim$ is the equivalence relation generated by

> $(P,x,f) \sim (P',x',f')$ if there exists an arrow $p : P \to P'$ in $\mathbb{P}$ such that $x = X(p)(x')$ and $f' = \overline{F}(p,1_Q)(f)$.

This means that two such triples, say (P,x,f) and (P',x',f') are related, if and only if there exists a chain of morphisms of $\mathbb{P}$,

$$\begin{tikzcd}
P = A_0 & & A_2 & & \cdots & & \cdots & & A_n = P' \\
& A_1 \arrow[ul, "h_0"] \arrow[ur, "h_1"'] & & A_3 \arrow[ul, "h_2"] \arrow[ur] & & \cdots \arrow[ul] \arrow[ur] & & A_{n-1} \arrow[ul] \arrow[ur, "h_{n-1}"] &
\end{tikzcd}$$

and elements $a_i \in X(A_i)$, $b_i \in F(A_i,Q)$ such that

- $(P,x,f) = (A_0,a_0,b_0)$ and $(P',x',f') = (A_n,a_n,b_n)$
- for k odd, $a_k = X(h_k)(a_{k+1})$ and $b_{k+1} = \overline{F}(h_k,1_Q)(b_k)$
- for k even, $a_{k+1} = X(h_k)(a_k)$ and $b_k = \overline{F}(h_k,1_Q)(b_{k+1})$.

[1] See [76] or [17], for a description of how to compute limits and colimits in **Set**.

4.2 The bicategory Prof and the 2-category Cocont

We introduce now the 2-category of presheaf categories, colimit preserving functors and natural transformations. In fact we shall give three equivalent presentations of it; the first two will actually define it as a bicategory rather than a 2-category. The reason for doing so is that the bicategorical presentation gives a greater emphasis to the base categories and, as we shall see later, we will be very concerned with operations that are best presented on the base categories.

Definition 4.2.1 (Prof) *Define the bicategory* **Prof** *of profunctors as follows*

- **objects:** *small categories,* $\mathbb{P}, \mathbb{Q}, \mathbb{R}$*...*
- **arrows:** $F : \mathbb{P} \dashrightarrow \mathbb{Q}$*, functors* $F : \mathbb{P} \times \mathbb{Q}^{op} \to \mathbf{Set}$ *(these are the* profunctors *[72])*
- **2-cells:** $\alpha : F \Rightarrow G$*, natural transformations between such functors.*

The vertical composition of 2*-cells is the usual (vertical) composition of natural transformations. Horizontal composition of both arrows and 2-cells is described in term of coends formulae. Given two arrows* $\mathbb{P} \overset{F}{\dashrightarrow} \mathbb{Q} \overset{G}{\dashrightarrow} \mathbb{R}$*, define the following functor,*

$$\mathbb{P} \times \mathbb{Q}^{op} \times \mathbb{Q} \times \mathbb{R}^{op} \overset{F \times G}{\longrightarrow} \mathbf{Set} \times \mathbf{Set} \overset{\times}{\longrightarrow} \mathbf{Set}$$

that at each 4-tuple of objects P, Q, Q', R *associates the set* $F(P,Q) \times G(Q',R)$*, with the obvious actions on morphisms derived from those of* F *and* G*. Then, one defines the composition of* F *and* G *as arrows of* **Prof** *as*

$$F;G(P,R) = \int^{Q} F(P,Q) \times G(Q,R)$$

and for any $f : P \to P'$ *and* $g : R' \to R$*, define*

$$F;G(f,g) = \int^{Q} F(f,Q) \times G(Q,g) \ .$$

So:

$$\textstyle\int^{Q} F(P,Q) \times G(Q,R) \xrightarrow{\int^{Q} F(f,Q) \times G(Q,g)} \int^{Q} F(P',Q) \times G(Q,R')$$

Concerning 2*-cells, suppose we have the following situation*

$$\begin{tikzcd} \mathbb{P} \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathbb{Q} \arrow[r, bend left, "G"] \arrow[r, bend right, "G'"'] \arrow[r, phantom, "\Downarrow\beta"] & \mathbb{R} \end{tikzcd} \ .$$

Then define

$$(\alpha;\beta)_{\langle P,R\rangle} = \int^{Q} \alpha_{\langle P,Q\rangle} \times \beta_{\langle Q,R\rangle} \ .$$

Concerning the identities, these are just the hom-functors. Given any small category $\mathbb{P}$ *define*

$$1_{\mathbb{P}} : \mathbb{P} \times \mathbb{P}^{\mathrm{op}} \to \mathbf{Set} \quad (p, p') \mapsto \mathbb{P}(p', p)$$

Following the Proposition 4.1.7, it is immediately seen that the exponential transpose of $1_{\mathbb{P}}$ *is the Yoneda embedding* $\mathrm{y}_{\mathbb{P}}$*. The associativity morphisms and left and right identities are derived from the universal property that defines coends.*

REMARK: In giving the definition of **Prof**, we exploited the fact that any small category can be regarded as a **Set**-category in the terminology of enriched category theory [65]. This implies that much of what we say could be rephrased in term of generic $\mathcal{V}$-categories for $\mathcal{V}$ a cocomplete category.

Profunctors subsume presheaf categories as the following proposition states.

Proposition 4.2.2 *Any presheaf category,* $\widehat{\mathbb{P}}$*, is equivalent to the category* $\mathbf{Prof}(\mathbf{1}, \mathbb{P})$ *of profunctors from the one object and one arrow category to the category* $\mathbb{P}$*.*

Proposition 4.1.7, together with Proposition 1.2.4 can be used to give a different, but equivalent, definition of **Prof** that uses left Kan extensions to define the composition of arrows. More precisely, we could have described **Prof** as the following bicategory:

- **objects:** small categories, $\mathbb{P}, \mathbb{Q}, \mathbb{R}$...
- **arrows:** $F : \mathbb{P} \rightarrow\!\!\!\!\!\!\mapsto \mathbb{Q}$, functors $F : \mathbb{P} \to \widehat{\mathbb{Q}}$
- **2-cells:** natural transformations between such functors

The composition of arrows $\mathbb{P} \xrightarrow{F} \mathbb{Q} \xrightarrow{G} \mathbb{R}$ is defined, using a choice for left Kan extensions, as $(G \circ F) = \mathrm{Lan}_{\mathrm{y}_Q}(G) \circ F$, where the second composition is the usual composition of functors. Using the description of Kan extensions via coends, this implies that for any $P \in |\mathbb{P}|$ and $R \in |\mathbb{R}|$,

$$(G \circ F)(P)(R) \cong \int^{Q} G(Q, R) \times F(P, Q)$$

and this ensures us that this definition is equivalent to the previous one.

The above description helps in understanding the tight relationship that arrows in **Prof** holds with respect to colimit preserving functors between presheaf categories. We can make this formal, by considering the following 2-category **Cocont**.

Definition 4.2.3 *Define* **Cocont** *as follows:*

- **objects:** *small categories,* $\mathbb{P}, \mathbb{Q}, \mathbb{R}, \ldots$
- **arrows:** *colimit preserving functors between the corresponding presheaf categories, i.e.,* F *is an arrow from* $\mathbb{P}$ *to* $\mathbb{Q}$*, if it is a colimit preserving functor* $F : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$*.*
- **2-cells:** *natural transformations between such functors*

Since any 2-category is a particular bicategory, it makes sense to look for bicategorical functors (cf. [9] or [17]) that relate **Prof** to **Cocont**. Following the terminology of [9]:

Proposition 4.2.4 *There exists a pair of strictly unitary morphisms*

$$\mathbf{Prof} \underset{\Xi}{\overset{\Sigma}{\rightleftarrows}} \mathbf{Cocont}\ ,$$

such that for any two small categories $\mathbb{P}, \mathbb{Q}$, $\Sigma_{(\mathbb{P},\mathbb{Q})}$ *and* $\Xi_{(\mathbb{P},\mathbb{Q})}$ *are equivalences of categories, pseudo inverses to each other.*

In other words **Prof** and **Cocont** can be regarded as 'equivalent' bicategories, that moreover have the same class of objects and such that the equivalences Σ and Ξ send identity arrows to identity arrows.

4.2.1 A set theoretic analogy

It is often said that a profunctor[2] is to a functor what a relation is to a mapping. That is, profunctors are to be regarded as generalised relations, when we moved up a level from sets to categories. This short section uses Proposition 1.2.15 to illustrate a result that backs up the above claim and that we shall use in Chapter 6. Recall that a relation between two sets X, Y, is given by a function $R : X \times Y \to \mathbf{2}$, where $\mathbf{2}$ is the two elements set. A relation R is actually a function from X to Y if it satisfies the following two conditions:

$$\begin{array}{ll} \text{Totality} & \Delta_X \subseteq R^\circ R \\ \text{Functionality} & RR^\circ \subseteq \Delta_Y \end{array} \quad ,$$

where the Δ's are the diagonal relations, i.e., the identity functions. In other (categorical) words, regarding **Rel** as an order-enriched category and so a bicategory, a relation is a function if it has a right adjoint. We state now a known result of category theory that shows how profunctors with right adjoints correspond (up to the notion of Cauchy completion) to functors between the underlying base categories.

Theorem 4.2.5 *Let* $F : \mathbb{P} \underset{\not\leftarrow}{\overset{\not\to}{\perp}} \mathbb{Q} : G$ *be adjoint profunctors*[3] *and let* $\mathbb{Q}$ *be a Cauchy complete category. Then there exists a functor* $H : \mathbb{P} \to \mathbb{Q}$ *such that* $F(-,+) = \mathbb{Q}(+, H(-))$ *and* $G(+,-) = \mathbb{Q}(H(-),+)$.

Proof:[Sketch] As we said, profunctors correspond to colimit preserving functors and therefore adjoint pairs of profunctors correspond to adjoint pairs of colimit preserving functors. Any colimit preserving functor between presheaf categories has a right adjoint (cf. Section 1.2), hence adjoint pairs of colimit preserving functors correspond to essential geometric morphisms that in turn correspond (Proposition 1.2.15) to functors between the base categories. $\square$

If $\mathbb{Q}$ was not Cauchy complete, then the statement would have to be modified to require the functor H to be defined from $\mathbb{P}$ to the Cauchy completion $\mathbb{Q}^c$ of $\mathbb{Q}$.

[2] Some people use the word distributor or bimodule [10, 72].

[3] Adjoint pairs in bicategories are defined in analogy with the 2-categorical case, but for some extra care needed to take account of the coherence isomorphisms [47].

In terms of Kan extensions, the theorem above is saying that $Lan_{y_{\mathbb{P}}}(F) \cong H_!$ and $Lan_{y_{\mathbb{Q}}}(D) \cong H^*$. As we shall see later, just like **Rel**, also **Prof** (or better **Cocont** as far as our definitions will be concerned) is compact closed [35, 36], though in a bicategorical sense.

4.2.2 A domain theoretic analogy

We discuss now the intuition that the presheaf construction is analogue to a powerdomain [101] one. As remarked in [138] and [25], in fact, **Prof** can be described as the bicategory of free algebras for a pseudo-monad over the categorical analogue of the category of ω-algebraic cpo's. If presheaf categories are analogues of powerdomains, then **Prof** can be regarded as a bicategory of non-deterministic domains [51]. But let's not proceed too hastily.

Definition 4.2.6 (Completion by filtered colimits) *Let $\mathbb{P}$ be a small category. We write $\widetilde{\mathbb{P}}$ for its* free filtered colimit completion*; that is, $\widetilde{\mathbb{P}}$ has colimits of filtered diagrams [78, 5] and any filtered colimit preserving functor $F : \widetilde{\mathbb{P}} \to \mathcal{C}$, where $\mathcal{C}$ is a category with filtered colimits, is uniquely (up to a natural isomorphism) determined by its restriction to $\mathbb{P}$.*

The categorical analogue of ω-algebraic cpo's is then defined to be the 2-category of such freely generated categories with filtered colimits.

Definition 4.2.7 ($\omega-$Acc) *Define $\omega-\mathbf{Acc}$ to be the following 2-category:*

- **objects:** *small categories, $\mathbb{P}, \mathbb{Q}, \mathbb{R}, \ldots$*
- **arrows:** *filtered colimit preserving functors between the respective filtered colimit completions i.e., F is an arrow from $\mathbb{P}$ to $\mathbb{Q}$, if it is a filtered colimit preserving functor $F : \widetilde{\mathbb{P}} \to \widetilde{\mathbb{Q}}$.*
- **2-cells:** *natural transformations between such functors.*

We shall write $\mathbf{Filt}(\widetilde{\mathbb{P}}, \widetilde{\mathbb{Q}})$ for $\omega-\mathbf{Acc}(\mathbb{P}, \mathbb{Q})$.

Since we are dealing with freely generated categories, we have the following equivalence:

Proposition 4.2.8 *The functor category $\mathbf{CAT}(\mathbb{P}, \widetilde{\mathbb{Q}})$ is equivalent to $\mathbf{Filt}(\widetilde{\mathbb{P}}, \widetilde{\mathbb{Q}})$.*

This means that we could have given, just like with profunctors an equivalent bicategorical presentation of $\omega-\mathbf{Acc}$ that have functors $\mathbb{P} \to \widetilde{\mathbb{Q}}$ as arrows and uses the freeness property to perform composition of arrows.

For more on the notion of κ-accessible category (for κ any regular cardinal) one can consult [78]. We want to describe now an endo pseudo-functor on $\omega-\mathbf{Acc}$ that is based on the free completion of a (small) category under finite colimits. Having linear logic [38] in mind, we shall denote it by the exclamation mark symbol (!). We need a preliminary result that can be deduced from results in Chapter VI of [60] (in particular from the Theorem on page 232):

Proposition 4.2.9 *If $\mathbb{P}$ is a small category, writing $!\mathbb{P}$ for the free finite colimit completion of $\mathbb{C}$, one has that there is an equivalence of categories*

$$\widehat{\mathbb{P}} \simeq \widetilde{!\mathbb{P}} \ .$$

Definition 4.2.10 *Define* $! : \omega{-}\mathbf{Acc} \to \omega{-}\mathbf{Acc}$ *to be the following pseudo-functor*

- **on objects:** $!\mathbb{P}$ *returns a (small) description of the free finite colimit completion of* $\mathbb{P}$.
- **on arrows:** *If* $F : \widetilde{\mathbb{P}} \to \widetilde{\mathbb{Q}}$ *is a filtered-colimit preserving functor, define*

$$!F : \widetilde{!\mathbb{P}} \to \widetilde{!\mathbb{Q}}$$

using the equivalences of Proposition 4.2.9 as follows

$$\widetilde{!\mathbb{P}} \xrightarrow{\sim} \widehat{\mathbb{P}} \xrightarrow{\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(c_{i_{\mathbb{Q}}} F i_{\mathbb{P}})} \widehat{\mathbb{Q}} \xrightarrow{\sim} \widetilde{!\mathbb{Q}} ,$$

where $i_{\mathbb{P}} : \mathbb{P} \to \widetilde{\mathbb{P}}$ *and* $i_{\mathbb{Q}} : \mathbb{Q} \to \widetilde{\mathbb{Q}}$ *are the obvious inclusion functors.*
- **On 2-cells:** *The action on 2-cells is uniquely determined by the universal property of left Kan extensions.*

The pseudo-functor ! can be equipped with multiplication, unit and corresponding coherence modifications in order to form a *pseudo-monad* [110] or a *doctrine* in the terminology of [127]. Using Proposition 4.2.9, we can prove the following result that allows us to represent filtered colimit preserving functors between presheaf categories as profunctors.

Proposition 4.2.11 *For any two small category* $\mathbb{P}$ *and* $\mathbb{Q}$ *there is an equivalence of categories*

$$\mathbf{Prof}(!\mathbb{P}, \mathbb{Q}) \simeq \mathbf{Filt}(\widehat{\mathbb{P}}, \widehat{\mathbb{Q}}) .$$

Proof:

$$\begin{aligned}
\mathbf{Prof}(!\mathbb{P}, \mathbb{Q}) &\cong \mathbf{CAT}(!\mathbb{P}, \widehat{\mathbb{Q}}) \\
&\simeq \mathbf{CAT}(!\mathbb{P}, \widetilde{!\mathbb{Q}}) \\
&\simeq \mathbf{Filt}(\widetilde{!\mathbb{P}}, \widetilde{!\mathbb{Q}}) \\
&\simeq \mathbf{Filt}(\widehat{\mathbb{P}}, \widehat{\mathbb{Q}})
\end{aligned}$$

□

The above proposition suggest that **Prof** can be regarded as the category of free algebras, that is the Kleisli category, for !.[4] In fact we have the following adjoint situation

$$\begin{tikzcd} \omega{-}\mathbf{Acc} \arrow[r, bend left, "L"] \arrow[r, phantom, "\perp"] & \mathbf{Prof} \arrow[l, bend left, "R"] \end{tikzcd}$$

where L is the identity on objects and send every arrow $F : \widetilde{\mathbb{P}} \to \widetilde{\mathbb{Q}}$, to

$$\mathbb{P} \hookrightarrow \widetilde{\mathbb{P}} \xrightarrow{F} \widetilde{\mathbb{Q}} \hookrightarrow \widehat{\mathbb{Q}}$$

[4]Of course to make this statement precise we should consider the notion of Kleisli category for a pseudo-monad.

(since the filtered colimit completion of $\mathbb{Q}$ includes into $\widehat{\mathbb{Q}}$, by freeness [60]). The action on 2-cells is given by composition with the two inclusions. On the other side, $R\mathbb{P} = !\mathbb{P}$ and one uses the equivalence of Proposition 4.2.11 to map $\mathbf{Prof}(\mathbb{P},\mathbb{Q})$ to $\mathbf{Filt}(!\mathbb{P},!\mathbb{Q})$, by first including (using the freeness of the ! construction) $\mathbf{Prof}(\mathbb{P},\mathbb{Q})$ into $\mathbf{Prof}(!\mathbb{P},\mathbb{Q})$.

If we now look at the monad from the "**Prof** point of view", then we obtain a comonad on **Prof**, that we indicate as well with the bang symbol, !. In Section 4.3, we will use this comonad to show that **Prof** can be regarded as a (bicategorical) Seely model [122] of classical linear logic [38].

4.3 The structure of Prof

In this section we show that **Prof** has enough structure to be considered what could be called a *compact closed bicategory*.[5] To do so, we first need to define explicitly some pseudo-limits.[6]

Definition 4.3.1 (Pseudo-products and -coproducts) *In a bicategory $\mathcal{B}$, a* pseudo-product *of two objects b, c, is given by an object d and an equivalence of categories*

$$\mathcal{B}(e,b) \times \mathcal{B}(e,c) \simeq \mathcal{B}(e,d)$$

pseudo-natural in e; more explicitly a pseudo-product is given by a span of arrows

$$b \xleftarrow{\pi_1} d \xrightarrow{\pi_2} c$$

such that

1. *For any other span, $b \xleftarrow{f} e \xrightarrow{g} c$, there exists an $h : e \to d$ and isomorphic 2-cells, $\Phi : \pi_1 h \overset{\sim}{\Rightarrow} f$ and $\Gamma : \pi_2 h \overset{\sim}{\Rightarrow} g$.*
2. *For any two arrows $h, k : e \to d$ and 2-cells, $\sigma_i : \pi_i h \Rightarrow \pi_i k$, for $i = 1, 2$, there exists a unique $\sigma : h \Rightarrow k$, such that $\sigma_i = \pi_i \sigma$.*

If the equivalences are isomorphisms, we shall say that that product is strict.

Dually one can define pseudo-coproducts

We have already seen, as an illustrative example the definition of pseudo-initial object (Definition 1.5.1). A dual definition would give a *pseudo-terminal* one.

Definition 4.3.2 (Pseudo-zero object) *In a bicategory $\mathcal{B}$ a* pseudo-zero *object is an object that is both pseudo-initial and -terminal.*

Proposition 4.3.3 **Prof** *has strict pseudo products* ($\&$) *and coproducts* ($\oplus$) *and they coincide on objects.*

[5] As for instance is remarked in [66].

[6] We recall, once more that what for us is a pseudo-limit is often called in other contexts, e.g., [127], a bicategorical limit or, shortly, a bilimit.

Proof: Let $\mathbb{P}$ and $\mathbb{Q}$ be two small categories, define

$$\mathbb{P}\&\mathbb{Q} \stackrel{\text{def}}{=} \mathbb{P} + \mathbb{Q} \stackrel{\text{def}}{=} \mathbb{P} \oplus \mathbb{Q} \ ,$$

where $\mathbb{P} + \mathbb{Q}$ is the usual disjoint union of small categories with inclusions $in_{\mathbb{P}}$ and $in_{\mathbb{Q}}$. Further define $\pi_{\mathbb{P}} : \mathbb{P}\&\mathbb{Q} \to\!\!\!\!\mapsto \mathbb{P}$ by $\pi_{\mathbb{P}}(in_{\mathbb{P}}(P), P') = \mathbb{P}(P', P)$ and $\pi_{\mathbb{P}}(in_{\mathbb{Q}}(d), P') = \emptyset$ and symmetrically $\pi_{\mathbb{Q}}$. While $i_{\mathbb{P}} : \mathbb{P} \to\!\!\!\!\mapsto \mathbb{P} \oplus \mathbb{Q}$ is defined as the transpose of $\mathrm{y}_{\mathbb{P}+\mathbb{Q}} in_{\mathbb{P}}$, i.e., $i_{\mathbb{P}}(P, in_{\mathbb{P}}(P')) = \mathbb{P}(P', P)$ and $i_{\mathbb{P}}(P, in_{\mathbb{Q}}(d)) = \emptyset$. □

In connection with the above proposition, it is worth stating the following that accounts for strictness of product and coproduct.

Proposition 4.3.4 *If $\mathbb{P}$ and $\mathbb{Q}$ are two small categories, then $\widehat{\mathbb{P} + \mathbb{Q}}$ is isomorphic to $\widehat{\mathbb{P}} \times \widehat{\mathbb{Q}}$.*

Proposition 4.3.5 **Prof** *has a (strict) pseudo-zero object*

Proof: Just take the initial category, **0**, with no objects and no arrows. □

It is also immediately seen that the zero object is the unit for the product/coproduct bifunctor. Other pseudo-functors are definable and they make **Prof** into what might be called a *-autonomous bicategory [7].

Definition 4.3.6 *If $\mathcal{B}$ is a bicategory, we write $\mathcal{B}^{\mathrm{op}}$ for the* opposite bicategory *which reverses the direction of the 1-cells but not that of the 2-cells.*

Definition 4.3.7 *We define a* tensor *and a* dualiser *in* **Prof***.*

- **Tensor:** *Define $\otimes$:* **Prof** $\times$ **Prof** $\to$ **Prof** *as follows:*
  - **On objects:** $\mathbb{P} \otimes \mathbb{Q} \stackrel{\text{def}}{=} \mathbb{P} \times \mathbb{Q}$*, the product of categories*
  - **On arrows:** *If $F : \mathbb{P} \to\!\!\!\!\mapsto \mathbb{P}'$ and $G : \mathbb{Q} \to\!\!\!\!\mapsto \mathbb{Q}'$,*

$$\begin{array}{rccl} F \otimes G : & \mathbb{P} \times \mathbb{Q} \times \mathbb{P}'^{\mathrm{op}} \times \mathbb{Q}'^{\mathrm{op}} & \to & \mathbf{Set} \\ & (P, Q, P', Q') & \mapsto & F(P, P') \times G(Q, Q') \end{array}$$

  - **On 2-cells:** *if $\alpha : F \Rightarrow F'$ and $\beta : G \Rightarrow G'$, then*

$$\alpha \otimes \beta_{(P,Q,P',Q')} = \alpha_{(P,P')} \times \beta_{(Q,Q')} \ .$$

  The terminal category **1** *is neutral element for $\otimes$.*
- **Dualizer:** *Define $(-)^*$:* **Prof** $\to$ **Prof**$^{\mathrm{op}}$ *as follows*
  - **On objects:** $\mathbb{P}^* = \mathbb{P}^{\mathrm{op}}$
  - **On arrows:** *Given $F : \mathbb{P} \to\!\!\!\!\mapsto \mathbb{Q}$, define $F^* : \mathbb{Q}^* \to\!\!\!\!\mapsto \mathbb{P}^*$ as $F^*(Q, P) = F(P, Q)$.*
  - **On** 2**-cells:** *If $\alpha : F \Rightarrow F'$, then $\alpha^* : F^* \Rightarrow F'^*$, with $\alpha^*_{\langle Q,P\rangle} = \alpha_{\langle P,Q\rangle}$.*

Combining tensor and dualiser, yields a "linear function space".

Definition 4.3.8 *Define the pseudo functor* $\multimap\colon \mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof} \to \mathbf{Prof}$ *as*

$$\multimap = \otimes \circ ((-)^* \times 1) \ ,$$

i.e., for every two categories $\mathbb{P}$ *and* $\mathbb{Q}$*,* $\mathbb{P} \multimap \mathbb{Q} = \mathbb{P}^{\mathrm{op}} \times \mathbb{Q}$*.*

Proposition 4.3.9 *For any three categories,* $\mathbb{P}, \mathbb{Q}, \mathbb{R}$*,*

$$\mathbf{Prof}(\mathbb{P} \otimes \mathbb{Q}, \mathbb{R}) \cong \mathbf{Prof}(\mathbb{P}, \mathbb{Q} \multimap \mathbb{R}) \ .$$

Proof: The following chain of natural isomorphisms holds trivially:

$$\begin{array}{rcl}
\mathbf{Prof}(\mathbb{P} \otimes \mathbb{Q}, \mathbb{R}) & \stackrel{\mathrm{def}}{=} & \mathbf{CAT}(\mathbb{P} \times \mathbb{Q} \times \mathbb{R}^{\mathrm{op}}, \mathbf{Set}) \\
& \cong & \mathbf{CAT}(\mathbb{P}, \widehat{\mathbb{Q}^{\mathrm{op}} \times \mathbb{R}}) \\
& \stackrel{\mathrm{def}}{=} & \mathbf{CAT}(\mathbb{P}, \widehat{\mathbb{Q}^* \otimes \mathbb{R}}) \\
& \cong & \mathbf{CAT}(\mathbb{P} \times (\mathbb{Q}^* \otimes \mathbb{R})^{\mathrm{op}}, \mathbf{Set}) \\
& \stackrel{\mathrm{def}}{=} & \mathbf{Prof}(\mathbb{P}, \mathbb{Q}^* \otimes \mathbb{R}) \\
& \stackrel{\mathrm{def}}{=} & \mathbf{Prof}(\mathbb{P}, \mathbb{Q} \multimap \mathbb{R})
\end{array}$$

□

We can summarise the above by saying that **Prof** is a compact closed bicategory. From a linear logic point of view, this also implies that $\parr$ and $\otimes$ coincide and hence the degeneracy of the model, adding to the degeneracy that & and $\oplus$ coincide too. It is worth remarking that the correspondence $\mathbf{Prof}(\mathbb{P} \otimes \mathbb{Q}, \mathbb{R}) \cong \mathbf{Prof}(\mathbb{P}, \mathbb{Q}^* \otimes \mathbb{R})$ does lead to an adjunction:

Proposition 4.3.10 *For any small category* $\mathbb{P}$*, the pseudo functor* $\mathbb{P} \otimes -$ *is left adjoint to* $\mathbb{P} \multimap -$*.*

Proof: What we lack are unit and counit for the adjunction, but these are immediately defined (for any $\mathbb{Q}$) as follows:

$$\begin{array}{rccl}
\eta_{\mathbb{Q}} : & \mathbb{Q} \times (\mathbb{P} \multimap (\mathbb{P} \otimes \mathbb{Q}))^{\mathrm{op}} & \to & \mathbf{Set} \\
& (Q, P, P', Q') & \mapsto & \mathbb{P}(P', P) \times \mathbb{Q}(Q', Q)
\end{array}$$

$$\begin{array}{rccl}
\varepsilon_{\mathbb{Q}} : & (\mathbb{P} \otimes (\mathbb{P} \multimap \mathbb{Q})) \times \mathbb{Q}^{\mathrm{op}} & \to & \mathbf{Set} \\
& (P, P', Q, Q') & \mapsto & \mathbb{P}(P', P) \times \mathbb{Q}(Q', Q)
\end{array}$$

□

4.3.1 Lifting

In this paragraph we draw the attention to another endofunctor of **Prof** that will play a crucial role in the following chapters and that, as we shall see in the next section allows us to represent connected colimit preserving functors between presheaf categories as profunctors.

Definition 4.3.11 (Lifting) *Define* $(-)_\perp : \mathbf{Prof} \to \mathbf{Prof}$ *to be the following pseudo functor:*

- **On object:** $\mathbb{P}_\perp$ *is the category* $\mathbb{P}$ *to which it has been added a new strict initial object, often referred to as* $\perp$. *The objects of* $\mathbb{P}_\perp$ *other than* $\perp$ *are often written* $\lfloor P \rfloor$ *for* P *an object of* $\mathbb{P}$.
- **On arrows:** *If* $F : \mathbb{P} \mathrel{-\!\!\!\mapsto} \mathbb{Q}$, $F_\perp$ *is defined by:*

$$F_\perp(x, y) = \begin{cases} F(P, Q) & \text{if } x = \lfloor P \rfloor \text{ and } y = \lfloor Q \rfloor \\ \{*\} & \text{if } y = \perp_{\mathbb{Q}} \\ \emptyset & \text{otherwise} \end{cases}$$

- **On 2-cells:** *A 2-cell* $\alpha : F \Rightarrow G$ *is extended to cover the new cases with identity functions.*

Observe that in **Cat**, the operation $\mathbb{P}_\perp$ corresponds to the free completion of $\mathbb{P}$ with the colimit of the empty diagram.

4.4 Connected colimits

In Section 4.2.2 it was shown how to represent filtered colimit preserving functors between presheaf categories in **Prof** using an exponential, !, that if regarded as a 2-functor in **Cat** amounts to the free finite colimit completion 2-monad. We now concentrate on another class of colimits that will be important for us. These are the colimits of connected diagrams [95]. Using the lifting functor, i.e., the free empty colimit completion, that freely adds to a category an initial object, we can similarly describe connected colimit preserving functors between presheaf categories as arrows in **Prof**.

NOTATION: Let $\mathbb{P}$ be a small category, write $l : \mathbb{P} \to \mathbb{P}_\perp$, for the inclusion functor, $P \mapsto \lfloor P \rfloor$.

We know from Chapter 1, that such a functor induces a triple of adjoints

$$l_! \dashv l^* \dashv l_* : \widehat{\mathbb{P}} \to \widehat{\mathbb{P}_\perp} \ .$$

The functor $l^* \mathrm{y}_{\mathbb{P}_\perp}$, is the universal functor from $\mathbb{P}_\perp$ to $\widehat{\mathbb{P}}$ that exists by freeness and that sends every non bottom object to the corresponding representable and the bottom one to the empty presheaf. In the sequel, we shall often write $j_{\mathbb{P}}$ for $l^* \mathrm{y}_{\mathbb{P}_\perp}$. The functor l_* takes any presheaf X over $\mathbb{P}$ and returns a rooted presheaf $\lfloor X \rfloor$ over $\mathbb{P}_\perp$, such that $\lfloor X \rfloor(\lfloor P \rfloor) = X(P)$. In fact, if we restrict, l^* to range over rooted presheaves, the pair $l^* : \overline{\mathbb{P}_\perp} \leftrightarrows \widehat{\mathbb{P}} : l_*$ defines the equivalence we have been talking about in previous chapters. In particular note that $l_* l^* \mathrm{y}_{\mathbb{P}_\perp} \cong \mathrm{y}_{\mathbb{P}_\perp}$ since any representable is a rooted presheaf.

Back to connected colimit preserving functors, we have:

Proposition 4.4.1 *The functor* $l_* : \widehat{\mathbb{P}} \to \widehat{\mathbb{P}_\perp}$ *preserves connected colimits.*

Proposition 4.4.2 *A connected colimit preserving functor* $G : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ *is uniquely (up to natural isomorphism) determined by its action on the representables of* $\widehat{\mathbb{P}}$ *and on the initial presheaf* $\emptyset$.

Proof: Let X be any presheaf in $\widehat{\mathbb{P}}$. We know that X is a colimit of representables

$$X \cong \mathrm{colim}\ \mathcal{E}l(X) \xrightarrow{\mathrm{y}_{\mathbb{P}}\pi} \widehat{\mathbb{P}}\ .$$

Consider the connected diagram $(\mathrm{y}_{\mathbb{P}}\pi)_\perp : (\mathcal{E}l(X))_\perp \to \widehat{\mathbb{P}}$ that extends $\mathrm{y}_{\mathbb{P}}\pi$ by sending $\perp$ to $\emptyset$. Clearly $X \cong \mathrm{colim}\ (\mathrm{y}_{\mathbb{P}}\pi)_\perp$, hence

$$GX \cong G\mathrm{colim}\ (\mathrm{y}_{\mathbb{P}}\pi)_\perp \cong \mathrm{colim}\ G(\mathrm{y}_{\mathbb{P}}\pi)_\perp\ .$$

Therefore to know the action of G on X we need to know

$$G(\mathrm{y}_{\mathbb{P}}\pi)_\perp(x,P) = G(\mathrm{y}_{\mathbb{P}}(P)), \text{ for any } (x,P) \in \mathcal{E}l(X) \text{ and}$$

$$G(\mathrm{y}_{\mathbb{P}}\pi)_\perp(\perp) = G\emptyset\ .$$

□

Proposition 4.4.3 *There exists an equivalence of categories*

$$\mathbf{Prof}(\mathbb{P}_\perp, \mathbb{Q}) \simeq \mathbf{Conn}(\widehat{\mathbb{P}}, \widehat{\mathbb{Q}})\ ,$$

for any two small categories $\mathbb{P}$ and $\mathbb{Q}$, where $\mathbf{Conn}(\widehat{\mathbb{P}}, \widehat{\mathbb{Q}})$ is the category of connected colimit preserving functors and natural transformations.

Proof: Actually we shall prove the existence of an equivalence

$$\varphi : \mathbf{Cocont}(\widehat{\mathbb{P}_\perp}, \widehat{\mathbb{Q}}) \stackrel{\sim}{\longrightarrow} \mathbf{Conn}(\widehat{\mathbb{P}}, \widehat{\mathbb{Q}})\ .$$

For any $F \in\mid \mathbf{Cocont}(\widehat{\mathbb{P}_\perp}, \widehat{\mathbb{Q}})\mid$, define $\varphi(F) = Fl_*$. If $\alpha : F \Rightarrow F'$, define $\varphi(\alpha) = \alpha_{l^*}$.

If $G \in\mid \mathbf{Conn}(\widehat{\mathbb{P}}, \widehat{\mathbb{Q}})\mid$, define $\psi(G) = \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}_\perp}}(Gl^*\mathrm{y}_{\mathbb{P}_\perp})$, while if $\beta : G \Rightarrow G'$, $\psi(\beta)$ is determined by the universal property of left Kan extensions.

From Proposition 4.4.1, it immediately follows that if F is colimit preserving, then Fl_* is connected colimit preserving. While $\psi(G)$ is colimit preserving by definition. Hence φ and ψ are well defined. Moreover is not difficult to see that φ is full and faithful using that presheaves, X, over $\mathbb{P}_\perp$ are expressible as sums of rooted presheaf and that the objects of $\mathbf{Cocont}(\widehat{\mathbb{P}_\perp}, \widehat{\mathbb{Q}})$ preserves colimits and hence sums. In fact $X \cong \sum_{x \in X(\perp)} \lfloor X_{|x} \rfloor$, where $X_{|x}$ is the presheaf over $\mathbb{P}$ defined by

$$X_{|x}(P) = \{y \in X(P) | x = X(\perp \le P)y\}\ .$$

And therefore one has that a natural transformation $\alpha : F \Rightarrow F'$ is uniquely determined by its action on rooted presheaves.

Now,

$$\begin{array}{rcl} \psi\varphi(F) & = & \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}_\perp}}(\varphi(F)l^*\mathrm{y}_{\mathbb{P}_\perp}) \\ & = & \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}_\perp}}(Fl_*l^*\mathrm{y}_{\mathbb{P}_\perp}) \\ & \cong & \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}_\perp}}(F\mathrm{y}_{\mathbb{P}_\perp}) \\ & \cong & F\ , \end{array}$$

while

$$\begin{array}{rcl} \varphi\psi(G) & = & \psi(G)l_* \\ & = & \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}_\perp}}(Gl^*\mathrm{y}_{\mathbb{P}_\perp})l_* \ . \end{array}$$

Observe that,

$$\begin{array}{rcccl} \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}_\perp}}(Gl^*\mathrm{y}_{\mathbb{P}_\perp})l_*(\mathrm{y}_{\mathbb{P}}(P)) & = & Gl^*\mathrm{y}_{\mathbb{P}_\perp}(\lfloor P\rfloor) & = & G\mathrm{y}_{\mathbb{P}}(P) \\ \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}_\perp}}(Gl^*\mathrm{y}_{\mathbb{P}_\perp})l_*(\emptyset) & = & Gl^*\mathrm{y}_{\mathbb{P}_\perp}(\perp) & = & G\emptyset \ . \end{array}$$

Hence, because of Proposition 4.4.2, $\varphi\psi(G) \cong G$. So φ is full and faithful and essentially surjective on objects and so an equivalence. □

Connected colimit preserving functors will play an important role in the semantics of process calculi. Proposition 4.4.3 and Proposition 4.4.4 below will be used in Section 4.6 to prove that connected colimit preserving functors preserve surjective open maps.

Proposition 4.4.4 *The functor $l_* : \widehat{\mathbb{P}} \to \widehat{\mathbb{P}_\perp}$ preserve surjective open maps.*

Proof: Let $f : X \to Y$ be surjective open, to ensure that $\lfloor f\rfloor : \lfloor X\rfloor \to \lfloor Y\rfloor$ is surjective open, it is enough to check that the "new" naturality squares

$$\begin{tikzcd} \lfloor X\rfloor(\lfloor P\rfloor) = X(P) \arrow[r] \arrow[d, "\lfloor f\rfloor_{\lfloor P\rfloor} = f_P"'] & \{*\} = \lfloor X\rfloor(\perp) \arrow[d] \\ \lfloor Y\rfloor(\lfloor P\rfloor) = X(P) \arrow[r] & \{*\} = \lfloor X\rfloor(\perp) \end{tikzcd}$$

are quasi-pullbacks, but this amounts to claiming surjectivity of f that is granted by hypothesis. □

4.5 A type theory of domains for concurrency

We can put together all the informations of the previous section to give interpretation in **Prof** to the types of a simple grammar. This will be the basis for the description of the presheaf models of Chapter 5, 7 and 8. In Chapter 6 in fact we will prove a theorem of limit-colimit coincidence that will enable us to extend the grammar with recursive types and therefore allow the recursive definition of path categories for presheaf models. The grammar is the following:

$$t \ ::= \ 0 \mid 1 \mid t \oplus t' \mid t \otimes t' \mid t^* \mid \ !t \mid \vartheta \mid \textstyle\sum_{i\in I} t_i \mid t_\perp \mid t \multimap t' \ .$$

These types are that of *compact closed categories* extended with type *variables* (ϑ), arbitrary *sums* ($\sum$), a *lifting* operator ($(-)_\perp$). For a list of distinct type variables Θ, we write $\Theta \vdash t$ to indicate that t is a well-formed type with free type variables amongst those in Θ. Type judgements $\Theta \vdash t$ are interpreted as pseudo functors

$$[\![\Theta \vdash t]\!] : (\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \longrightarrow \mathbf{Prof} \ .$$

The interpretation is given in terms of the constructors of Section 4.3 inductively as follows:

- $[\![\Theta \vdash 0]\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \to \mathbf{I} \xrightarrow{\mathbf{0}} \mathbf{Prof}\ ,$$
where $\mathbf{I}$ is the terminal bicategory and $\mathbf{0}$ is the functor that picks the initial category $\mathbf{0}$.
- $[\![\Theta \vdash 1]\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \to \mathbf{I} \xrightarrow{\mathbf{1}} \mathbf{Prof}\ ,$$
where $\mathbf{1}$ is the functor that picks the terminal category $\mathbf{1}$.
- $[\![\Theta \vdash t \oplus t']\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{\langle [\![\Theta \vdash t]\!], [\![\Theta \vdash t']\!] \rangle} \mathbf{Prof} \times \mathbf{Prof} \xrightarrow{\oplus} \mathbf{Prof}$$
- $[\![\Theta \vdash t \otimes t']\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{\langle [\![\Theta \vdash t]\!], [\![\Theta \vdash t']\!] \rangle} \mathbf{Prof} \times \mathbf{Prof} \xrightarrow{\otimes} \mathbf{Prof}$$
- $[\![\Theta \vdash t^*]\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{(((-)^* \times ((-)^*)^{\mathrm{op}})\sigma)^{|\Theta|}} (\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{[\![\Theta \vdash t]\!]} \mathbf{Prof}\ ,$$
where $\sigma : \mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof} \to \mathbf{Prof} \times \mathbf{Prof}^{\mathrm{op}}$ is the symmetry functor which on objects acts as follows: $\sigma(\mathbb{P}, \mathbb{Q}) = (\mathbb{Q}, \mathbb{P})$.
- $[\![\Theta \vdash !t]\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{[\![\Theta \vdash t]\!]} \mathbf{Prof} \xrightarrow{!} \mathbf{Prof}$$
- $[\![\Theta \vdash \vartheta]\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{\pi_\vartheta} \mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof} \xrightarrow{\pi_2} \mathbf{Prof}\ ,$$
where the π's are the obvious projection functors.
- $[\![\Theta \vdash \sum_{i \in I} t_i]\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{\langle [\![\Theta \vdash t_i]\!] \rangle_{i \in I}} \prod_{i \in I} \mathbf{Prof} \xrightarrow{\sum_{i \in I}} \mathbf{Prof}$$
where we write $\sum_{i \in I}$, for the extended sum functor.
- $[\![\Theta \vdash t_\perp]\!]$:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{[\![\Theta \vdash t]\!]} \mathbf{Prof} \xrightarrow{(-)_\perp} \mathbf{Prof}$$
- $[\![\Theta \vdash t \multimap t']\!] \stackrel{\mathrm{def}}{=} [\![\Theta \vdash t^* \otimes t']\!]$, that is:
$$(\mathbf{Prof}^{\mathrm{op}} \times \mathbf{Prof})^{|\Theta|} \xrightarrow{\langle [\![\Theta \vdash t^*]\!], [\![\Theta \vdash t']\!] \rangle} \mathbf{Prof} \times \mathbf{Prof} \xrightarrow{\otimes} \mathbf{Prof}$$

4.5.1 An alternative exponential

As we saw in previous sections an 'exponential' operator (!) naturally presented itself as a candidate comonad to obtain a cartesian closed structure in the bicategory of free coalgebras out of the symmetric monoidal closed of **Prof**, the key fact being that ! satisfies the Seely [122] condition:

$$!(\mathbb{P}\&\mathbb{Q}) \cong !\mathbb{P} \otimes !\mathbb{Q}.$$

An attractive feature of this operator is that it arises from domain theoretical considerations and it gives a way of describing in **Cocont** (the 2-category equivalent to **Prof**) the notion of filtered colimit preserving functor between presheaf categories, using the equivalence

$$\omega\text{−}\mathbf{Acc}(!\mathbb{P}, !\mathbb{Q}] \cong \mathbf{Filt}(\widehat{\mathbb{P}}, \widehat{\mathbb{Q}}) \cong \mathbf{Cocont}(!\mathbb{P}, \mathbb{Q}).$$

Its disadvantage is that, being, on objects, the free finite colimit completion, gives always rise to complicated categories that might be quite difficult to handle when, for instance, operational characterisation of the bisimulation induced by open maps is considered. Therefore it makes sense to keep our eyes open to the possibility of taking other choices for the exponential, choices sufficient for special purposes and perhaps easier to work with. An example is induced by the free finite coproduct completion construction.

Definition 4.5.1 (Finite families) *If $\mathbb{P}$ is a small category, let $Fam_f(\mathbb{P})$ be the following (small) category of finite families of objects of $\mathbb{P}$:*

- **Objects:** *Finite families $(P_i)_{i\in I}$ of objects of $\mathbb{P}$.*
- **Arrows:** *A pair $\langle(\alpha_i)_{i\in I}, f\rangle$ is an arrow from $(P_i)_{i\in I}$ to $(Q_j)_{j\in J}$, if $f : I \to J$ is a function and for every $i \in I$, $\alpha_i : P_i \to Q_{f(i)}$ is an arrow in $\mathbb{P}$.*

Proposition 4.5.2 *The construction Fam_f induces, just like ! and $(-)_\perp$ a 2-monad on the 2-category* **Cat**. *In particular on objects, $Fam_f(\mathbb{P})$ is the free completion of $\mathbb{P}$ with finite coproducts.*

The 2-monad Fam_f induces a pseudo-comonad [110] on **Prof** whose underlying pseudo endofunctor is given by the following definition:

Definition 4.5.3 *Define a pseudo functor fam_f :* **Prof** $\to$ **Prof** *as follows:*

- **On objects:** $fam_f(\mathbb{P}) = Fam_f(\mathbb{P})$
- **On arrows:** *If $F : \mathbb{P} \mathrel{-\!\!\!\mapsto} \mathbb{Q}$, then*

$$fam_f(F)((P_i)_{i\in I}, (Q_j)_{j\in J}) = \prod_{j\in J}\coprod_{i\in I} F(P_i, Q_j)\ .$$

- **On 2-cells:** *The pointwise extension according to the definition on arrows.*

It is not difficult to see that the above pseudo functor satisfies the Seely condition too, namely, for any two small categories, $\mathbb{P}$ and $\mathbb{Q}$,

$$Fam_f(\mathbb{P}\&\mathbb{Q}) \cong Fam_f(\mathbb{P}) \otimes Fam_F(\mathbb{Q})\ .$$

There are tie-ups between the Fam_f construction and the categorical powerdomain [73, 1] (the latter being the dual to the category yielded by Fam_f) but we have not yet explored how deep these connections are.

4.6 Open map bisimulation in Prof

Building on the fact that $\mathbf{Prof}(\mathbb{P},\mathbb{Q}) = \widehat{\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}}$, for any two small categories, $\mathbb{P},\mathbb{Q}$, we now give a definition of open 2-cells in **Prof**, based on that of open map and show that the horizontal composition of open 2-cells gives an open 2-cell. This will imply as a corollary that colimit preserving functors between presheaf categories, preserve open maps (recall Proposition 3.2.5).

Definition 4.6.1 *Let $\alpha : F \Rightarrow F'$, be a 2-cell between two profunctors $F, F' : \mathbb{P} \to\!\!\!\!\!\!\mapsto \mathbb{Q}$. Define α to be open, if it is open as an arrow of $\widehat{\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}}$*

Let's unpack this definition, to see what it really means. Since α is regarded as a natural transformation between two presheaves, being open amounts, for it, to satisfy the quasi-pullback condition of Definition 2.2.8. Suppose, that $\langle f^{\mathrm{op}}, g\rangle : \langle P, Q\rangle \to \langle P', Q'\rangle$ is an arrow in $\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}$, then the following square must be a quasi-pullback in **Set**:

$$\begin{tikzcd} F(P',Q') \arrow[r, "F(f^{\mathrm{op}},g)"] \arrow[d, "\alpha_{\langle P',Q'\rangle}"'] & F(P,Q) \arrow[d, "\alpha_{\langle P,Q\rangle}"] \\ F'(P',Q') \arrow[r, "F'(f^{\mathrm{op}},g)"'] & F'(P,Q)\ . \end{tikzcd} \tag{4.2}$$

If we instantiate one of the two arguments f or g to be the identity arrow, on P and Q, respectively, this immediately implies that the corresponding natural transformations,

$$\begin{aligned} \alpha_P &: F(P,-) \dot{\to} F'(P,-) \\ \alpha_Q &: F(-,Q) \dot{\to} F'(-,Q) \end{aligned}$$

are $\mathbb{Q}$-open and $\mathbb{P}^{\mathrm{op}}$-open, respectively. Actually, the converse holds, too.

Proposition 4.6.2 *Let $\alpha : F \dot{\to} F'$ be a natural transformation between two presheaves $F, F' \in \widehat{\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}}$, then α is $\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}$-open if and only if for any object P of $\mathbb{P}$ and Q of $\mathbb{Q}$, the corresponding natural transformations α_P and α_Q are $\mathbb{Q}$-open and $\mathbb{P}^{\mathrm{op}}$-open, respectively.*

Proof: The discussion above, proves the "only if" part. For the converse, note that the diagram (4.2) above, via the functoriality of F can be rewritten as the following:

$$\begin{tikzcd} F(P',Q') \arrow[r, "F(f^{\mathrm{op}},1_{Q'})"] \arrow[d, "\alpha_{\langle P',Q'\rangle}"'] & F(P,Q') \arrow[r, "F(1_P,g)"] \arrow[d, "\alpha_{\langle P,Q'\rangle}"] & F(P,Q) \arrow[d, "\alpha_{\langle P,Q\rangle}"] \\ F'(P',Q') \arrow[r, "F'(f^{\mathrm{op}},1_{Q'})"'] & F'(P,Q') \arrow[r, "F'(1_P,g)"'] & F'(P,Q)\ . \end{tikzcd}$$

Now, it is immediately seen that such gluing of quasi-pullback squares is a quasi-pullback square, too. □

The following observations, though having trivial proofs, are worth mentioning.

Proposition 4.6.3 *Let $\alpha : F \Rightarrow F'$ be an open 2-cell in* **Prof**, *then $\alpha^* : F^* \Rightarrow F'^*$ is open, as well.*

Proposition 4.6.4 *If we regard a presheaf category $\widehat{\mathbb{P}}$ as the hom-category* $\mathbf{Prof}(\mathbf{1}, \mathbb{P})$ *as in Proposition 4.2.2, then a natural transformation between two presheaves is open if and only if it is open as a two cell between the corresponding profunctors.*

Our aim is that of proving that the horizontal composition of 2-cells preserves open maps.[7]

Theorem 4.6.5 *If*

$$\begin{tikzcd} \mathbb{P} \arrow[r, bend left, "F"] \arrow[r, bend right, "F'"'] \arrow[r, phantom, "\Downarrow\alpha"] & \mathbb{Q} \arrow[r, bend left, "G"] \arrow[r, bend right, "G'"'] \arrow[r, phantom, "\Downarrow\beta"] & \mathbb{R} \end{tikzcd}$$

are two consecutive open 2-cells of **Prof**, *then their composition $\beta\alpha$ is an open 2-cell, too.*

Proof: We need to prove that for each two pairs of objects $\langle P, R \rangle$ and $\langle P', R' \rangle$ and arrows $f : P \to P'$, $g : R' \to R$ of $\mathbb{P}$ and $\mathbb{R}$, the following square is a quasi-pullback:

$$\begin{tikzcd} \int^Q F(P,Q) \times G(Q,R) \arrow[r, "\int^Q F(f,Q)\times G(Q,g)"] \arrow[d, "\int^Q \alpha_{\langle P,Q\rangle}\times\beta_{\langle Q,R\rangle}"'] & \int^Q F(P',Q) \times G(Q,R') \arrow[d, "\int^Q \alpha_{\langle P',Q\rangle}\times\beta_{\langle Q,R'\rangle}"] \\ \int^Q F'(P,Q) \times G'(Q,R) \arrow[r, "\int^Q F'(f,Q)\times G'(Q,g)"'] & \int^Q F'(P',Q) \times G'(Q,R') \end{tikzcd}$$

Recall that for each pair $\langle P, R \rangle$,

$$\int^Q F(P,Q) \times G(Q,R) \cong \Big(\coprod_{Q \in \mathbb{Q}} F(P,Q) \times G(Q,R)\Big)/\sim$$

where the equivalence relation $\sim$ is generated by

$$(Q, x, G(q, 1_R)y')) \sim (Q', F(1_P, q)x, y')$$

where $x \in F(P,Q)$, $y' \in G(Q', R)$ and $q : Q' \to Q$ is an arrow in $\mathbb{Q}$.

Suppose now that the equivalence classes

$$[Q_1, x_1, y_1] \in \int^Q F(P',Q) \times G(Q,R') \text{ and } [Q_2, x_2, y_2] \in \int^Q F'(P,Q) \times G'(Q,R)$$

[7] Of course since open maps compose, it should be already clear that the vertical composition of two open 2-cells is an open 2-cell.

are such that

$$
\begin{aligned}
(\int^{Q} \alpha_{\langle P',Q\rangle} \times \beta_{\langle Q,R'\rangle})[Q_1, x_1, y_1] &= [Q_1, \alpha_{\langle P',Q_1\rangle}(x_1), \beta_{\langle Q_1,R'\rangle}(y_1)] \\
&= [Q_2, F'(f, 1_{Q_2})(x_2), G'(1_{Q_2}, g)(y_2)] \\
&= (\int^{Q} F'(f, 1_Q) \times G'(1_Q, g))[Q_2, x_2, y_2]
\end{aligned}
$$

By the definition of the equivalence relation, this means that there exists a chain of morphisms in $\mathbb{Q}$:

$$
\begin{tikzcd}
Q_1 = A_0 & & A_2 & & \cdots & & \cdots & & A_n = Q_2 \\
& A_1 \arrow[ul, "h_0"] \arrow[ur, "h_1"'] & & A_3 \arrow[ul, "h_2"] \arrow[ur] & & \cdots \arrow[ul] \arrow[ur] & & A_{n-1} \arrow[ul] \arrow[ur, "h_{n-1}"] &
\end{tikzcd}
$$

and elements $a_i \in F'(P', A_i)$, $b_i \in G'(A_i, R')$ such that

- $(Q_1, \alpha_{\langle P',Q_1\rangle}(x_1), \beta_{\langle Q_1,R'\rangle}(y_1)) = (A_0, a_0, b_0)$
- $(Q_2, F'(f, 1_{Q_2})(x_2), G'(1_{Q_2}, g)(y_2)) = (A_n, a_n, b_n)$
- for k odd, $a_k = F'(1_{P'}, h_k)(a_{k+1})$ and $b_{k+1} = G'(h_k, 1_{R'})(b_k)$
- for k even, $a_{k+1} = F'(1_{P'}, h_k)(a_k)$ and $b_k = G'(h_k, 1_{R'})(b_{k+1})$

Our first step is to show the existence of elements $a'_i \in F(P', A_i)$ and $b'_i \in G(A_i, R')$ satisfying the following four conditions:

1. $a'_0 = x_1$ and $b'_0 = y_1$
2. $\alpha_{\langle P',A_k\rangle}(a'_k) = a_k$ and $\beta_{\langle A_k,R'\rangle}(b'_k) = b_k$
3. for k odd, $a'_k = F(1_{P'}, h_k)(a'_{k+1})$ and $b'_{k+1} = G(h_k, 1_{R'})(b'_k)$
4. for k even, $a'_{k+1} = F(1_{P'}, h_k)(a'_k)$ and $b'_k = G(h_k, 1_{R'})(b'_{k+1})$.

In other words we want to lift the chain of elements, that connects (A_0, a_0, b_0) to (A_n, a_n, b_n), to a chain that connects (Q_1, x_1, y_1) to something, whose image under α and β is (A_n, a_n, b_n).

We do this by induction, showing first how to deal with h_0 to obtain (a'_1, b'_1)[8] and afterwards how to obtain (a'_{k+1}, b'_{k+1}) from (a'_k, b'_k).

Let's consider then $h_0 : A_1 \to Q_1$. Following condition 4 we define

$$a'_1 = F(1_{P'}, h_0)(x_1)\ .$$

It immediately follows that

$$
\begin{aligned}
\alpha_{\langle P',A_k\rangle}(a'_1) &= \alpha_{\langle P',A_k\rangle}(F(1_{P'}, h_0)(x_1)) \\
&= F'(1_{P'}, h_0)(\alpha_{\langle P',Q_1\rangle}(x_1)) \\
&= F'(1_{P'}, h_0)(a_0)
\end{aligned}
$$

[8]Recall that a'_0 and b'_0 are already decided to be x_1 and y_1, respectively, because of condition 1.

$$= \quad a_1 \; .$$

Therefore condition 2 is satisfied by a'_1. To find b'_1 consider the square, with a choice of elements as on the right hand side:

$$\begin{array}{ccccc}
G(A_1,R') & \xrightarrow{G(h_0,1_{R'})} & G(Q_1,R') & & y_1 \\
\downarrow{\scriptstyle \beta_{\langle A_1,R'\rangle}} & & \downarrow{\scriptstyle \beta_{\langle Q_1,R'\rangle}} & & \big\downarrow \\
G'(A_1,R') & \xrightarrow[G'(h_0,1_{R'})]{} & G'(Q_1,R') & b_1 \longmapsto & b_0 = \beta_{\langle Q_1,R'\rangle}(y_1)
\end{array}$$

therefore, since β is open, there exists $b'_1 \in G(A_1,R')$ such that

$$b'_0 \stackrel{\text{def}}{=} y_1 = G(h_0,1_{R'})(b'_1) \text{ and } b_1 = \beta_{\langle A_1,R'\rangle}(b'_1).$$

This tells us that b'_1 satisfies conditions 2 and 4 too.

Suppose now $0 \leq k \langle n-1$ odd and such that there exist (b'_k, a'_k) satisfying condition 2. Define $b'_{k+1} = G(h_k,1_{R'})(b'_k)$. Then

$$\begin{aligned}
\beta_{\langle A_{k+1},R'\rangle}(b'_{k+1}) &= \beta_{\langle A_{k+1},R'\rangle}(G(h_k,1_{R'})(b'_k)) \\
&= G'(h_k,1_{R'})(\beta_{\langle A_k,R'\rangle}(b'_k)) \\
&= G'(h_k,1_{R'})(b_k) \\
&= b_{k+1}
\end{aligned}$$

To find a suitable a'_{k+1}, consider the square, with corresponding elements as below:

$$\begin{array}{ccccc}
F(P',A_{k+1}) & \xrightarrow{F(1_{P'},h_k)} & F(P',A_k) & & a'_k \\
\downarrow{\scriptstyle \alpha_{\langle P',A_{k+1}\rangle}} & & \downarrow{\scriptstyle \alpha_{\langle P',A_k\rangle}} & & \big\downarrow \\
F'(P',A_{k+1}) & \xrightarrow[F'(1_{P'},h_k,)]{} & F'(P',A_k) & a_{k+1} \longmapsto & a_k = \alpha_{\langle P',A_k\rangle}(a'_k)
\end{array}$$

since α is open, there exists $a'_{k+1} \in F(P',A_{k+1})$ such that

$$\alpha_{\langle P',A_{k+1}\rangle}(a'_{k+1}) = a_{k+1} \text{ and } a'_k = F(1_{P'},h_k)(a'_{k+1})$$

i.e., a'_{k+1} satisfies both conditions 2 and 3.

The case k even is treated in exactly the same way by substituting in the argument F for G, α for β and vice versa (it is actually the general instance of the argument used in the base case $k = 0$).

We are now almost finished with the proof of the theorem. We simply need to apply openness of α and β once more. Consider the following diagram (we concentrate on α first)

$$\begin{tikzcd}
F(P,Q_2) \arrow[r, "F(f{,}1_{Q_2})"] \arrow[d, "\alpha_{\langle P,Q_2\rangle}"'] & F(P',Q_2) \arrow[d, "\alpha_{\langle P',Q_2\rangle}"] & & a'_n \arrow[d, mapsto] \\
F'(P,Q_2) \arrow[r, "F'(f{,}1_{Q_2})"'] & F'(P',Q_2) & x_2 \arrow[r, mapsto] & F'(f,1_{Q_2})(x_2) = a_n
\end{tikzcd}$$

Since α is open, there exists $x_3 \in F(P,Q_2)$ such that

$$\alpha_{\langle P,Q_2\rangle}(x_3) = x_2 \text{ and } F(f,1_{Q_2})(x_3) = a'_n$$

In complete analogy, considering

$$\begin{tikzcd}
G(Q_2,R) \arrow[r, "G(1_{Q_2}{,}g)"] \arrow[d, "\beta_{\langle Q_2,R\rangle}"'] & G(Q_2,R') \arrow[d, "\beta_{\langle Q_2,R'\rangle}"] & & b'_n \arrow[d, mapsto] \\
G'(Q_2,R) \arrow[r, "G'(1_{Q_2}{,}g)"'] & G'(Q_2,R') & y_2 \arrow[r, mapsto] & G'(1_{Q_2},g)(y_2) = b_n
\end{tikzcd}$$

by openness of β, we can conclude the existence of $y_3 \in G(Q_2,R)$ such that

$$\beta_{\langle Q_2,R\rangle}(y_3) = y_2 \text{ and } G(1_{Q_2},g)(y_3) = b'_n \ .$$

Finally, we consider the equivalence class

$$[Q_2,x_3,y_3] \in \int^Q F(P,Q) \times G(Q,R)$$

and using the above results immediately verify that

$$\begin{aligned}
(\int^Q \alpha_{\langle P,Q\rangle} \times \beta_{\langle Q,R\rangle})[Q_2,x_3,y_3] &= [Q_2, \alpha_{\langle P,Q_2\rangle}(x_3), \beta_{\langle Q_2,R\rangle}(y_3)] \\
&= [Q_2,x_2,y_2]
\end{aligned}$$

and that

$$\begin{aligned}
(\int^Q F(f,1_Q) \times G(1_Q,g))[Q_2,x_3,y_3] &= [Q_2, F(f,1_{Q_2})(x_3), G(1_{Q_2},g)(y_3)] \\
&= [Q_2, a'_n, b'_n] \\
&= [Q_1, a'_0, b'_0] \\
&= [Q_1, x_1, y_1] \ .
\end{aligned}$$

□

By instantiating to particular situations, whiskering on the right or left, we can immediately deduce interesting corollaries to the theorem. Notably, Corollary 4.6.6 (already seen in this thesis as Proposition 3.2.5) below was a key result in [26] and Theorem 4.6.5 seems to be its natural generalisation.

Corollary 4.6.6 *Colimit preserving functors between presheaf categories preserve open maps*

Proof: Let $F_! : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ be a colimit preserving functor and let $\alpha : X \Rightarrow Y$ be a natural transformation between two presheaves over $\mathbb{P}$. Via Proposition 4.2.4 and Proposition 4.2.2 we can redraw this situation in **Prof** as follows:

$$\begin{tikzcd} \mathbf{1} \arrow[r, bend left, "X" , "\shortmid" marking] \arrow[r, bend right, "Y"', "\shortmid" marking] \arrow[r, phantom, "\Downarrow\alpha"] & \mathbb{P} \arrow[r, "F", "\shortmid" marking] & \mathbb{Q}. \end{tikzcd}$$

We know that $F_!(X)$ is the same as the composite FX in **Prof** and the same goes for Y. Moreover, by Theorem 4.6.5 the composition $F\alpha$ is open and, if we regard it as a natural transformation between presheaves, $F\alpha = F_!(\alpha)$. But now Proposition 4.6.4 concludes the proof. □

Corollary 4.6.7 *Let $\alpha : F \Rightarrow F'$ be an open 2-cell between $F, F' : \mathbb{P} \mathrel{-\!\!\shortmid\!\!\!\to} \mathbb{Q}$. Recall that α, via the Proposition 4.2.4, can be seen as a natural transformation between the corresponding two colimit preserving functors, $F_!$ and $F'_!$. Then α is open if and only if for each $X \in \widehat{\mathbb{P}}$, α_X is a $\mathbb{Q}$-open map and for each $Y \in \widehat{\mathbb{Q}^{\mathrm{op}}}$, α^*_Y is a $\mathbb{P}^{\mathrm{op}}$-open map.*

Proof: The 'if' part is trivial given Proposition 4.6.2. The 'only if' follows via an argument analogous to that of Corollary 4.6.6 above, applied to the following pictures.

$$\begin{tikzcd} \mathbf{1} \arrow[r, "X", "\shortmid" marking] & \mathbb{P} \arrow[r, bend left, "F", "\shortmid" marking] \arrow[r, bend right, "F'"', "\shortmid" marking] \arrow[r, phantom, "\Downarrow\alpha"] & \mathbb{Q} \end{tikzcd} \qquad \begin{tikzcd} \mathbf{1} \arrow[r, "Y", "\shortmid" marking] & \mathbb{Q}^{\mathrm{op}} \arrow[r, bend left, "F^*", "\shortmid" marking] \arrow[r, bend right, "F'^*"', "\shortmid" marking] \arrow[r, phantom, "\Downarrow\alpha^*"] & \mathbb{P}^{\mathrm{op}} \end{tikzcd} .$$

□

We can use Corollary 4.6.6 to deduce the preservation of surjective open maps also along connected colimit preserving functors.

Theorem 4.6.8 *Let $G : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ be a connected colimit preserving functor. Then G preserves surjective open maps.*

Proof: From Proposition 4.4.3, we know that $G \cong Fl_*$ for $l_* : \widehat{\mathbb{P}} \to \widehat{\mathbb{P}_\perp}$ and $F : \widehat{\mathbb{P}_\perp} \to \widehat{\mathbb{Q}}$ a colimit preserving functor. Now, from Proposition 4.4.4 we know that l_* preserves surjective open maps, hence Fl_* preserves surjective open maps too. □

Chapter 5

Two Examples

In the next chapter we will develop a theory of domains in 2-categories that generalise the order enriched categories case [125] and that deals with pseudo-functors (and pseudo-limits) rather than strict ones. **Cocont** (or *equivalently* **Prof**) will naturally fall in the list of examples. As motivating examples we present two process languages to which one gives presheaf models by deriving suitable base (path) categories as solutions to recursive domain equations. The fact that the solutions we present are indeed least fixed points will be justified intuitively. The results of the next chapter will make this in precise terms.

5.1 CCS

Here we show how to obtain synchronisation trees as presheaf models and we sketch the semantics of **CCS** terms using arrows in **Prof**. Following Corollary 4.6.6 and Proposition 4.6.8 this will entail that strong bisimulation [82] is a congruence with respect to the term constructors of the language. This analysis shall also serve as a reason to introduce some of our "standard" techniques for reasoning about presheaf models and open map bisimulation. These are represented by decomposition results (see Proposition 5.1.10) and transition relations for presheaves (see Section 5.1.4).

5.1.1 The term language

We briefly recall the main definitions of process terms with their transition semantics. The reader unfamiliar with process algebra and **CCS** in particular is advised to look in [82] for more detailed explanations. Assume then a set Ch of channels not including as an element the special symbol τ. Elements of Ch will be indicated with the letters $a, b, c, \ldots$. Let $\overline{Ch} = \{\overline{a} \mid a \in Ch\}$ be the set of "cochannels". Define the set of labels $L = Ch \cup \overline{Ch} \cup \{\tau\}$. Elements of L will be denoted by Greek letters $\alpha, \beta, \gamma, \ldots$. The process terms are defined by the following grammar:

$$t \;::=\; \mathbf{Nil} \mid \alpha.t \mid t_1 \,|\, t_2 \mid \textstyle\sum_{i\in I} t_i \mid t[f] \mid t/\Lambda \mid x \mid \mathrm{rec}\, x.t\ ,$$

where x is a variable drawn from some distinguished set *Vars*, I is a non empty indexing set, **Nil** stands for the deadlocked process, $f : Ch \to Ch$ is a relabelling function and $\Lambda \subseteq Ch$ is a set of "restricted" channels. Alternatively, we could have avoided to explicitly consider the **Nil** process and extend the sums with the possibility of having an empty indexing set. As usual, in recursive expressions like $\mathrm{rec}\, x.t$, the variable x becomes bound and this is the only binder we have in the language. Following this, the set of free variables in a process term and the class of closed process terms are defined by structural induction. The transition semantics is defined on closed processes:

$$\frac{}{\alpha.t \xrightarrow{\alpha} t} \qquad\qquad \frac{t_j \xrightarrow{\alpha} t'_j}{\sum_{i\in I} t_i \xrightarrow{\alpha} t'_j}\ (j \in I)$$

$$\frac{t \xrightarrow{\alpha} t'}{t \mid u \xrightarrow{\alpha} t' \mid u} \qquad \frac{t \xrightarrow{l} t' \quad u \xrightarrow{\overline{l}} u'}{t \mid u \xrightarrow{\tau} t' \mid u'} \qquad \frac{u \xrightarrow{\alpha} u'}{t \mid u \xrightarrow{\alpha} t \mid u'}$$

$$\frac{t \xrightarrow{\alpha} t'}{t/\Lambda \xrightarrow{\alpha} t'/\Lambda}(\alpha, \overline{\alpha} \notin Ch) \qquad \frac{t \xrightarrow{\alpha} t'}{t[f] \xrightarrow{f'\alpha} t'[f]} \qquad \frac{t[\mathrm{rec}\, x.t/x] \xrightarrow{\alpha} t'}{\mathrm{rec}\, x.t \xrightarrow{\alpha} t'}\ ,$$

where $l \in Ch \cup \overline{Ch}$ with the convention that doing $\overline{(-)}$ twice is the same as not doing it at all and where f' is the extension of f to $Ch \cup \overline{Ch} \cup \{\tau\}$ by putting $f'(\overline{a}) = \overline{f(a)}$ and $f'\tau = \tau$.

5.1.2 An equation for (synchronisation) trees

As it is clear from the operational semantics of the language, what can be observed of the behaviour of a process is a sequence of actions of the "type" L. These can be classified in three different classes: Input actions, Output actions and Silent or internal ones. By convention the input actions are represented by the elements a of Ch (make an input along channel a), the output by elements $\overline{a}$ of $\overline{Ch}$ while the internal ones are represented by the silent action τ. A path category is then defined to be:

$$\mathbb{P} \stackrel{\mathrm{def}}{=} \mathbb{P}_\perp + \sum_{a\in Ch} \mathbb{P}_\perp + \sum_{\overline{a}\in \overline{Ch}} \mathbb{P}_\perp = \sum_{\alpha\in L} \mathbb{P}_\perp\ .$$

As we shall see in more detail in the next chapter, to find a solution of this kind of equation in **Prof** is the same as finding one in **Poset** (the locally ordered category of partial ordered sets and monotone functions) and then regard it as a category in the usual way. This simplifies considerably the description of the solution.

Briefly, as expected, $\mathbb{P}$ can be described as the category L^+ of finite non empty strings of letters in L with morphisms given by the prefix order relation. As we have already written previously synchronisation trees and presheaves over L^+ correspond to each other:

Proposition 5.1.1 *The category $\mathcal{ST}_L$ of synchronisation trees over L is equivalent to $\widehat{L^+}$.*

Proof: We describe how to derive a tree from a presheaf and vice versa. To verify that this induces an equivalence of categories is matter of routine verification.

Given a tree (i.e., a special transition system) $T = (S, i, \mathit{tran}, L)$, define the presheaf over L^+, X_T, inductively as follows:

$$\begin{array}{rcl} X_T(\alpha) & = & \{s \in S \mid \exists i \xrightarrow{\alpha} s\} \\ X_T(P\alpha) & = & \{s \in S \mid \exists s' \in X(P)\, s' \xrightarrow{\alpha} s\} \end{array}$$

Since any state is reached by a unique transition, the map

$$X_T(P \leq P\alpha) : s \mapsto s'$$

is well defined.

Vice versa, given a presheaf X over L^+, define $T = (S, i, \mathit{tran}, L)$ to be given by

$$S \stackrel{\text{def}}{=} \{i\} \uplus \{(P, x) \mid x \in X(P)\}$$

with $(i \xrightarrow{\alpha} (\alpha, x))$, for any $x \in X(\alpha)$ and $((P, x) \xrightarrow{\alpha} (P\alpha, x'))$, if $X(P \leq P\alpha)x' = x$. $\square$

It is well known [82] that **CCS** terms can be given a semantics in terms of synchronisation trees. In Chapter 3, we also gave an abstract description of the categorical operations involved in the semantics (that was shown then to be compositional) by placing $\mathcal{ST}_L$ within the larger category $\mathcal{ST}$. Still we had, for the axiomatic approach, to assume as given the prefixing operator. Here we resolve that problem by employing the lifting operation for the denotations of prefixing. We also describe a parallel composition functor between trees by left Kan extensions. To do so we need some preliminary analysis that we will carry out in the next two sections and that (as we shall see in other chapters) is part of our standard pattern for deducing properties of presheaf models.

5.1.3 Decomposition of presheaves

Recall that

$$L^+ = \mathbb{P} = \mathbb{P}_\perp + \sum_{a \in Ch} \mathbb{P}_\perp + \sum_{\overline{a} \in \overline{Ch}} \mathbb{P}_\perp \ .$$

We write In_τ, In_a and $In_{\overline{a}}$ for the injection (pro)functors $\mathbb{P}_\perp \mathrel{-\!\!\!+\!\!\!\!\rightarrow} \widehat{\mathbb{P}}$ in the appropriate components, e.g.,

$$In_\tau(\lfloor P \rfloor) = \mathrm{y}_{\mathbb{P}}(\tau P) \qquad \text{and} \qquad In_\tau(\perp) = \mathrm{y}_{\mathbb{P}}(\tau) \ .$$

In the reminder of this section, we shall always write $\mathbb{P}$ for L^+ and P for a (generic) object of $\mathbb{P}$.

Proposition 5.1.2 *For any presheaf $X \in |\widehat{\mathbb{P}}|$,*

$$In_{\alpha !}X(P) \cong \begin{cases} \sum_{x \in X(\perp)} X(\lfloor P' \rfloor) & \text{if } P = \alpha P' \\ X(\perp) & \text{if } P = \alpha \\ \emptyset & \text{otherwise} \end{cases}$$

Proof: We know that $In_{\alpha!}X(P)$ is given by the following coend formula

$$In_{\alpha!}X(P) \cong \int^{Q} X(Q) \times In_{\alpha}(Q)(P) \ .$$

If $P = \alpha$, then, by definition $In_{\alpha}(Q)(P)$ is a singleton and the coend reduces to $\int^{Q} X(Q)$, that since $\mathbb{P}_{\perp}$ has an initial object is the same as $X(\perp)$. If $P = \alpha P'$, then again by looking at the definition, $In_{\alpha}(Q)(P) = \mathrm{y}_{\mathbb{P}_{\perp}}(Q)(\lfloor P' \rfloor)$, hence the coend becomes $\int^{Q} X(Q) \times \mathrm{y}_{\mathbb{P}_{\perp}}(Q)(\lfloor P' \rfloor)$ that, by Yoneda, is exactly $X(\lfloor P' \rfloor)$. Finally, if P is anything else, then $In_{\alpha}(Q)(P) = \emptyset$ and so the coends induces a diagram of empty sets, whose colimit is obviously the empty set as well. $\square$

Given a synchronisation tree, any given finite run of it, uniquely identifies a subtree of the original tree, namely the subtree rooted at the final state of the run. This fact is nicely expressible in presheaf terms by the following definition.

Definition 5.1.3 *If X is a presheaf over $\mathbb{P}$ and $x \in X(P)$, we write $X_{|x}$ for the presheaf over P "rooted at x" defined by*

$$X_{|x}(Q) = X(P \leq PQ)^{-1}(\{x\}) \ ,$$

where PQ stands for the concatenation of the string Q to the string P (recall that $\mathbb{P}$ is the category of finite non empty strings over L ordered by prefix ordering).

In terms of trees, of course, $X_{|x}$ is nothing else than the subtree of X rooted at x. Things are not quite so smooth when one is dealing with base categories other than $\mathbb{P}$. Still a similar notion of a "subcomponent rooted at an element" of a presheaf can be given in terms of slice categories [37].

Definition 5.1.4 *Let $\mathbb{C}$ be a small category and let C be an object of $\mathbb{C}$. Define the slice category $C/\mathbb{C}$ to be the category of objects the arrows $f : C \to C'$ of domain C and morphisms $h : f \to g$ the arrows of $\mathbb{C}$, $h : C' \to C''$ such that $g = hf$.*

It is immediately seen that $C/\mathbb{C}$ has initial object, given for instance by 1_C. In fact it has as many initial objects as there are isomorphisms in $\mathbb{C}$ of domain C. We write $(C/\mathbb{C})^{+}$ for the full subcategory of $C/\mathbb{C}$ consisting of non-initial objects.

Definition 5.1.5 *Let $\mathbb{C}$ be a small category and let $x \in X(C)$ for X a presheaf over $\mathbb{C}$. Define the* resumption of X at x *to be the presheaf $X_{|x}$ over $(C/\mathbb{C})^{+}$ defined by*

$$X_{|x}(f) = X(f)^{-1}(\{x\}) \ .$$

From the point of view of open map bisimulation we have an interesting preservation property.

Definition 5.1.6 *Let $\mathbb{C}$ be a small category, let X, Y be presheaves over $\mathbb{C}$ and let $\alpha : X \to Y$ be a map in $\widehat{\mathbb{C}}$, i.e., a natural transformation. If $x \in X(C)$, for $C \in |\mathbb{C}|$ and $y = \alpha_C(x)$, define*

$$\alpha_{|x} : X_{|x} \to Y_{|y}$$

to be the restriction of α to the two resumption presheaves, i.e., for every $f : C \to D$ in $\mathbb{C}$,

$$(\alpha_{|x})_f : X_{|x}(f) = X(f)^{-1}(\{x\}) \longrightarrow Y(f)^{-1}(\{y\}) = Y_{|y}(f)$$

maps every $x' \in X(f)^{-1}(\{x\}) \subseteq X(D)$ to $\alpha_D(x')$.

Theorem 5.1.7 *Let $\mathbb{C}$ be a small category. Let X, Y be presheaves over $\mathbb{C}$ and let $\alpha : X \to Y$ be a $\mathbb{C}$-open map. If $x \in X(C)$, for $C \in |\mathbb{C}|$ and $y = \alpha_C(x)$, then $\alpha_{|x}$ is a surjective $(C/\mathbb{C})^+$-open map.*

Proof: Let $f : C \to D$ and $g : C \to E$ be two objects of $(C/\mathbb{C})^+$ and let $h : D \to E$ an arrow from f to g, i.e., an arrow in $\mathbb{C}$ such that $g = fh$. To show that $\alpha_{|x}$ is open we need to show that the following square is a quasi pullback in **Set**:

$$\begin{tikzcd} X_{|x}(g) = X(g)^{-1}(\{x\}) \arrow[r, "X_x(h)"] \arrow[d, "(\alpha_{|x})_g"'] & X_{|x}(f) = X(f)^{-1}(\{x\}) \arrow[d, "(\alpha_{|x})_f"] \\ Y_{|y}(g) = Y(g)^{-1}(\{y\}) \arrow[r, "Y_y(h)"'] & Y_{|y}(f) = Y(f)^{-1}(\{y\}) \end{tikzcd} .$$

Suppose that $x' \in X(f)^{-1}(\{x\})$ and $y' \in Y(f)^{-1}(\{y\})$ are such that

$$(\alpha_{|x})_f(x') = Y_{|y}(h)(y') .$$

By definition this means that $\alpha_D(x') = Y(h)(y')$. Since α is $\mathbb{C}$ open, the following square is a quasi pullback,

$$\begin{tikzcd} X(E) \arrow[r, "X(h)"] \arrow[d, "\alpha_E"'] & X(D) \arrow[d, "\alpha_D"] \\ Y(E) \arrow[r, "Y(h)"'] & Y(D) \end{tikzcd} .$$

Hence there exists $x'' \in X(E)$ such that $\alpha_E(x'') = y'$ and $X(h)x'' = x'$. To conclude we shall only show that $x'' \in X_{|x}(g) \subseteq X(E)$, but

$$X(g)(x'') = X(hf)(x'') = X(f)X(h)(x'') = X(f)x' = x ,$$

i.e., $x'' \in X(g)^{-1}(\{x\}) \stackrel{\text{def}}{=} X_{|x}(g)$. The fact that $\alpha_{|x}$ is surjective too is again a consequence of openness of α. In fact let $f : C \to D$ be an object in $(C/\mathbb{C})^+$, we need to show that $(\alpha_{|x})_f$ is a surjective function. Let $y' \in Y_{|y}(f) \subseteq Y(D)$. Since α is open the following is a quasi pullback:

$$\begin{tikzcd} X(D) \arrow[r, "X(f)"] \arrow[d, "\alpha_D"'] & X(C) \arrow[d, "\alpha_C"] \\ Y(D) \arrow[r, "Y(f)"'] & Y(C) \end{tikzcd} .$$

Moreover, by assumption, $Y(f)(y') = y = \alpha_C(x)$ holds and therefore there exists $x' \in X(D)$ such that $\alpha_D(x') = y'$ and $X(f)(x') = x$, that is $x' \in X_{|x}(f)$ and $(\alpha_{|x})_f(x') = y'$. $\square$

REMARK: In fact, for X in $|\,\widehat{\mathbb{C}}\,|$ and $x \in X(C)$ we could have considered presheaves rooted at x over $C/\mathbb{C}$. Then the restriction of α to $\alpha_{|x}$ would have preserved openness, and since open with respect to rooted presheaves is equivalent to surjective open when the initial objects have been removed (cf. Chapter 2) we have a different way of seeing why in the theorem above, an open map becomes surjective when restricted.

What is so special about our particular $\mathbb{P}$ is that it satisfies the following "closure property" with respect to the slice category construction.

Proposition 5.1.8 *For any $P \in |\mathbb{P}|$, there is an isomorphism of categories between $\mathbb{P}$ and $(P/\mathbb{P})^+$.*

So in the case of $\mathbb{P}$, Definition 5.1.5 reduces to Definition 5.1.3. We shall make use of the more general case in Chapter 8. Another special case is given by categories with a strict initial object, with respect to resumptions after an initial step has been taken:

Proposition 5.1.9 *Let $\mathbb{C}$ be a small category. There is an isomorphism of categories between $\mathbb{C}$ and $(\perp/\mathbb{C}_\perp)^+$.*

So we shall write for any presheaf X over $\mathbb{P}_\perp$ and $x \in X(\perp)$, $X_{|x}$ for the presheaf over $\mathbb{P}$ defined by

$$X_{|x}(P) = X(\perp \leq \lfloor P \rfloor)^{-1}(\{x\}) \ .$$

Proposition 5.1.10 (Decomposition of Trees) *Any presheaf X over $\mathbb{P}$ is isomorphic to the presheaf*

$$X' = \sum_{i \in X(\tau)} In_{\tau !}\lfloor X_{|i} \rfloor + \sum_{a \in Ch} \sum_{j \in X(a)} In_{a!}\lfloor X_{|j} \rfloor + \sum_{\overline{a} \in \overline{Ch}} \sum_{k \in X(\overline{a})} In_{\overline{a}!}\lfloor X_{|k} \rfloor \tag{5.1}$$

Proof: By induction on the structure of path objects, i.e., on the length of the strings, one proves that for every $P \in |\mathbb{P}|$, there is a bijection $X(P) \cong X'(P)$.

- **Base case:** *length*$(P) = 1$, hence $P = \alpha$, for an $\alpha \in L$:

$$X'(\alpha) \stackrel{\text{def}}{=} \sum_{x \in X(\alpha)} In_{\alpha !}\lfloor X_{|x} \rfloor(\alpha) \cong \sum_{x \in X(\alpha)} \{x\} \cong X(\alpha) \ .$$

- **Inductive Step:** *length*$(P) = n+1$, hence $P = \alpha P'$:

$$\begin{aligned} X'(\alpha P') &\stackrel{\text{def}}{=} \sum_{x \in X(\alpha)} In_{\alpha !}\lfloor X_{|x} \rfloor(\alpha P') \\ &\cong \sum_{x \in X(\alpha)} X_{|x}(P') \\ &= \sum_{x \in X(\alpha)} X(\alpha \leq P)^{-1}(\{x\}) \\ &\cong X(P) \ . \end{aligned}$$

That these bijections are natural in P is trivial verification. $\square$

A way of looking at the decomposition or presheaves above is to think of it in terms of the "expansion law" [82] for process terms of **CCS**.

5.1.4 A transition relation for presheaves

Given the decomposition result it is natural to define a transition relation for presheaves. This can be done in two different ways, by decorating the transition arrow with the observed action only or with the observed action and the corresponding element of the presheaf, too. As it turns out, for **CCS** it does not really matter which one one chooses, in contrast to the situation we shall encounter in Chapter 8 - there one makes essential use of this more "intensional" information to characterise operationally the bisimulation from open maps.

Definition 5.1.11 (Transitions for Presheaves over L^+) *Let X and Y be presheaves over $\mathbb{P}$. We write*

$$X \xrightarrow{P}_x Y,$$

if $x \in X(P)$ and $Y = X_{|x}$. We write

$$X \xrightarrow{P} Y,$$

to mean that there exists $x \in X(P)$ such that $X \xrightarrow{P}_x Y$.

We shall be concerned exclusively with transitions of the form $X \xrightarrow{\alpha} Y$, for $\alpha \in L$ but since it does not require any extra effort, we prefer to give the definition more generally for all possible paths (these kinds of "long steps" transitions will play a big role in Chapter 8). Proposition 2.2.12 can now be read, for the part concerning synchronisation trees, as follows:

Proposition 5.1.12 *Two presheaves over L^+ are L^+-open map bisimilar if and only if the corresponding transitions systems are strongly bisimilar in the usual Park-Milner sense.*

Proof:[Hint] The proof relies on recognising the fact that the equivalence $\mathcal{ST}_L \simeq \widehat{L^+}$ takes a synchronisation tree to a presheaf that has the same transition system of the tree itself. $\square$

5.1.5 Denotational semantics

In Chapter 3 we reviewed a general way of giving a denotational semantics to **CCS** processes that mainly relied on the following three facts:

1. The category $\widehat{L^+}$ is included in a larger (fibred) category over all possible sets of labels and partial relabelling.
2. A distinguished "prefix" functor is available.
3. Parallel composition is expressed by combining product (in the larger category) with relabelling and restriction.

Here we take advantage of the domain equation that defines $\mathbb{P} = L^+$ in order to give a description directly in $\widehat{\mathbb{P}}$ of the operations involved in the semantics. In particular we reduce prefixing to lifting and parallel composition to a combination of liftings and left Kan extensions.

Prefixing: Let $\alpha \in L$, define $\alpha. : \widehat{\mathbb{P}} \to \widehat{\mathbb{P}}$ to be the functor

$$\widehat{\mathbb{P}} \xrightarrow{l_*} \widehat{\mathbb{P}_\perp} \xrightarrow{In_{\alpha !}} \widehat{\mathbb{P}} \ ,$$

i.e., for any presheaf $X \in |\widehat{\mathbb{P}}|$ and any path object P,

$$(\alpha.X)(P) = In_{\alpha !}(\lfloor X \rfloor)(P) = \begin{cases} X(P') & \text{if } P = \alpha.P' \\ \{*\} & \text{if } P = \alpha \\ \emptyset & \text{otherwise} \ . \end{cases}$$

Since l_* preserves connected colimits, $\alpha.$ preserves connected colimits too and therefore it preserves surjective open maps and open map bisimulation (see Proposition 4.6.8).

Observe moreover that the decomposition of presheaves of Proposition 5.1.10 can now be written as:

$$X \cong \sum_{i \in X(\tau)} \tau.(X_{|i}) + \sum_{a \in Ch} \sum_{j \in X(a)} a.(X_{|j}) + \sum_{\overline{a} \in \overline{Ch}} \sum_{k \in X(\overline{a})} \overline{a}.(X_{|k}) \ .$$

Parallel composition The parallel composition functor $(-|-) : \widehat{\mathbb{P}} \times \widehat{\mathbb{P}} \to \widehat{\mathbb{P}}$ is defined as follows:

$$(-|-) = (-||-)_! \circ w^*_{\mathbb{P}_\perp,\mathbb{P}_\perp} \circ (l_* \times l_*) \ ,$$

where

- The functor $w^*_{\mathbb{P}_\perp,\mathbb{P}_\perp} : \widehat{\mathbb{P}_\perp} \times \widehat{\mathbb{P}_\perp} \to \widehat{\mathbb{P}_\perp \times \mathbb{P}_\perp}$ is the right adjoint to the left Kan extension of

$$w_{\mathbb{P}_\perp,\mathbb{P}_\perp} : \mathbb{P}_\perp \times \mathbb{P}_\perp \,\mathrel{\circ\!\!\!\rightarrow}\, \mathbb{P}_\perp \& \mathbb{P}_\perp$$

  defined as the pairing of the "projections"

$$\mathbb{P}_\perp \times \mathbb{P}_\perp \xrightarrow{\pi_i} \mathbb{P}_\perp \xrightarrow{\mathrm{y}_{\mathbb{P}_\perp}} \widehat{\mathbb{P}_\perp} \ .$$

  Its action is given by $w^*_{\mathbb{P}_\perp,\mathbb{P}_\perp}(X,Y)(P,Q) = X(P) \times Y(Q)$.[1]

- The functor $(-|_\perp-) : \mathbb{P}_\perp \times \mathbb{P}_\perp \to \widehat{\mathbb{P}}$ is the symmetric functor defined inductively by

$$\begin{aligned} \perp || \perp &= \emptyset \\ \lfloor P \rfloor || \perp &= \mathrm{y}_{\mathbb{P}}(P) \\ \lfloor \alpha \rfloor || \lfloor \beta \rfloor &= \begin{cases} \mathrm{y}_{\mathbb{P}}(\alpha\beta) + \mathrm{y}_{\mathbb{P}}(\beta\alpha) & \text{if } \beta \neq \overline{\alpha} \text{ and } \alpha \neq \overline{\beta} \\ \mathrm{y}_{\mathbb{P}}(\alpha\beta) + \mathrm{y}_{\mathbb{P}}(\beta\alpha) + \mathrm{y}_{\mathbb{P}}(\tau) & \text{otherwise} \end{cases} \end{aligned}$$

[1] We shall use the functor w^* again in Chapters 7 and 8. We shall say a little more about it in Chapter 7.

$$\lfloor\alpha\rfloor||\lfloor\beta Q\rfloor \;=\; \begin{cases} \mathrm{y}_{\mathbb{P}}(\alpha\beta Q)+\beta.(\alpha||\lfloor Q\rfloor) & \text{if } \beta\neq\overline{\alpha} \text{ and } \alpha\neq\overline{\beta} \\ \mathrm{y}_{\mathbb{P}}(\alpha\beta Q)+\beta.(\alpha||\lfloor Q\rfloor)+\tau.\mathrm{y}_{\mathbb{P}}(Q) & \text{otherwise} \end{cases}$$

$$\lfloor\alpha P\rfloor||\lfloor\beta Q\rfloor \;=\; \begin{cases} \alpha.(\lfloor P\rfloor||\lfloor\beta Q\rfloor)+\beta.(\lfloor\alpha P\rfloor||\lfloor Q\rfloor) & \text{if } \beta\neq\overline{\alpha} \text{ and } \alpha\neq\overline{\beta} \\ \alpha.(\lfloor P\rfloor||\lfloor\beta Q\rfloor)+\beta.(\lfloor\alpha P\rfloor||\lfloor Q\rfloor) & \\ +\tau.(\lfloor P\rfloor||\lfloor Q\rfloor) & \text{otherwise} \end{cases}$$

Lemma 5.1.13 *Let X, Y be two presheaves over $\mathbb{P}$. Let $\alpha,\beta\in\{\tau\}\cup Ch\cup\overline{Ch}$ be two labels then the following holds:*

- *If α and β are not complementary, i.e., $\beta\neq\overline{\alpha}$ or vice versa, then*

$$\alpha.X|\beta.Y\cong\alpha.(X|\beta.Y)+\beta.(\alpha.X|Y)\ .$$

- *If α and β are complementary, then*

$$\alpha.X|\beta.Y\cong\alpha.(X|\beta.Y)+\beta.(\alpha.X|Y)+\tau.(X|Y)\ .$$

Proof: The following chain of isomorphisms prove the first case.

$$\begin{aligned}
\alpha.X|\beta.Y &= \textstyle\int^{r,s}\lfloor\alpha.X\rfloor(r)\times\lfloor\beta.Y\rfloor(s)\,.\,(r||s) \\
&\qquad\qquad \text{(by definition)} \\
&\cong \textstyle\int^{r',s'}\lfloor X\rfloor(r')\times\lfloor Y\rfloor(s')\,.\,(\lfloor In_\alpha r'\rfloor||\lfloor In_\beta s'\rfloor) \\
&\qquad\qquad (*) \\
&= \textstyle\int^{r',s'}\lfloor X\rfloor(r')\times\lfloor Y\rfloor(s')\,.\,(\alpha.(r'||\lfloor In_\beta s'\rfloor)+\beta.(\lfloor In_\alpha r'\rfloor||s')) \\
&\qquad\qquad \text{(by definition)} \\
&\cong \textstyle\int^{r',s'}\lfloor X\rfloor(r')\times\lfloor Y\rfloor(s')\,.\,(\alpha.(r'||\lfloor In_\beta s'\rfloor)) \\
&\quad + \textstyle\int^{r',s'}\lfloor X\rfloor(r')\times\lfloor Y\rfloor(s')\,.\,(\beta.(\lfloor In_\alpha r'\rfloor||s')) \\
&\qquad\qquad \text{(since sums distribute over coends)} \\
&\cong \alpha.\textstyle\int^{r',s'}\lfloor X\rfloor(r')\times\lfloor Y\rfloor(s')\,.\,(r'||\lfloor In_\beta s'\rfloor) \\
&\quad +\beta.\textstyle\int^{r',s'}\lfloor X\rfloor(r')\times\lfloor Y\rfloor(s')\,.\,(\lfloor In_\alpha r'\rfloor||s') \\
&\qquad\qquad \text{(since } \alpha. \text{ and } \beta. \text{ preserve connected colimits)} \\
&\cong \alpha.\textstyle\int^{r,s}\lfloor X\rfloor(r)\times\lfloor\beta.Y\rfloor(s)\,.\,(r||\lfloor s\rfloor) \\
&\quad +\beta.\textstyle\int^{r,s}\lfloor\alpha.X\rfloor(r)\times\lfloor Y\rfloor(s)\,.\,(r||s) \\
&\qquad\qquad \text{(no extra non-empty contribution is given)} \\
&= \alpha.(X|\beta.Y)+\beta.(\alpha.X|Y) \\
&\qquad\qquad \text{(by definition).}
\end{aligned}$$

The passage marked $(*)$ is justified for the following reasons. First of all, in the coend above it, when r (or s) is different from $\perp$ or $In_\alpha(r')$ ($\perp$ or $In_\alpha(s')$) then there is no contribution to the colimit since $\lfloor\alpha.X\rfloor(r)\times\lfloor\beta.Y\rfloor(s)=\emptyset$. Moreover, since $\lfloor\alpha.X\rfloor(\lfloor\alpha\rfloor)\times\lfloor\beta.Y\rfloor(\lfloor\beta\rfloor)$ is a singleton, we have that for any four lifted paths, r, r', s, s' the following is part of the diagram induced by the coend

$$\begin{tikzcd}
 & \lfloor\alpha\rfloor||\lfloor\beta\rfloor \arrow[dl] \arrow[dr] & \\
\lfloor In_\alpha r\rfloor||\lfloor In_\beta s\rfloor & & \lfloor In_\alpha r'\rfloor||\lfloor In_\beta s'\rfloor
\end{tikzcd} ,$$

and since we have the embeddings

$$\perp || \lfloor\beta\rfloor \hookrightarrow \lfloor\alpha\rfloor || \lfloor\beta\rfloor \hookleftarrow \lfloor\alpha\rfloor || \perp \ ,$$

the contribution of the (unique) pairs $\lfloor\alpha\rfloor||\perp$ and $\perp||\lfloor\beta\rfloor$ is subsumed by the contribution of $\lfloor\alpha\rfloor||\lfloor\beta\rfloor$.

The proof of the second case is just the same with the exception that, by definition, if α and β are complementary,

$$\lfloor In_\alpha r'\rfloor || \lfloor In_\beta s'\rfloor = \alpha.(r' || \lfloor In_\beta s'\rfloor) + \beta.(\lfloor In_\alpha r'\rfloor || s') + \tau.(r' || s') \ .$$

□

The above lemma together with the decomposition result on presheaves and the distribution property of sums with respect to coends immediately entail the following.

Proposition 5.1.14 *Let $X, Y \in |\widehat{\mathbb{P}}|$ with decompositions*

$$\begin{aligned} X &\cong \sum_{i \in X(\tau)} \tau.(X_{|i}) + \sum_{a \in Ch} \sum_{j \in X(a)} a.(X_{|j}) + \sum_{\overline{a} \in \overline{Ch}} \sum_{k \in X(\overline{a})} \overline{a}.(X_{|k}) \\ Y &\cong \sum_{l \in Y(\tau)} \tau.(Y_{|l}) + \sum_{a \in Ch} \sum_{m \in Y(a)} a.(Y_{|m}) + \sum_{\overline{a} \in \overline{Ch}} \sum_{n \in Y(\overline{a})} \overline{a}.(Y_{|n}) \end{aligned}$$

then $X|Y$ is isomorphic to

$$\begin{aligned} &\sum_{i \in X(\tau)} \tau.(X_{|i} \mid Y) + \sum_{a \in Ch} \sum_{j \in X(a)} a.(X_{|j} \mid Y) + \sum_{\overline{a} \in \overline{Ch}} \sum_{k \in X(\overline{a})} \overline{a}.(X_{|k} \mid Y) \\ &+ \sum_{l \in Y(\tau)} \tau.(X \mid Y_{|l}) + \sum_{a \in Ch} \sum_{m \in Y(a)} a.(X \mid Y_{|m}) + \sum_{\overline{a} \in \overline{Ch}} \sum_{n \in Y(\overline{a})} \overline{a}.(X \mid Y_{|n}) \\ &+ \sum_{a \in Ch} \sum_{j \in X(a)} \sum_{n \in Y(\overline{a})} \tau.(X_{|j} \mid Y_{|n}) + \sum_{a \in Ch} \sum_{m \in Y(a)} \sum_{k \in X(\overline{a})} \tau.(X_{|k} \mid Y_{|m}) \ . \end{aligned}$$

Restriction Let $\Lambda \subseteq Ch$ be a set of channels, define

$$(-/\Lambda) : \mathbb{P} \to \widehat{\mathbb{P}}$$

inductively as follows:

- $\alpha/\Lambda = \begin{cases} \mathrm{y}_{\mathbb{P}}(\alpha) & \text{if } \alpha \notin \Lambda \cup \overline{\Lambda} \\ \emptyset & \text{otherwise} \end{cases}$
- $(\alpha P)/\Lambda = \begin{cases} \alpha.(P/\Lambda) & \text{if } \alpha \notin \Lambda \cup \overline{\Lambda} \\ \emptyset & \text{otherwise} \end{cases}$

Referring back to the treatment of restriction in Chapter 3, it is easy to see that

$$(-/\Lambda)_! \cong \bar{i}_!((-\lceil\Lambda)) \ ,$$

where $i : (L/\Lambda) \hookrightarrow L$ is the inclusion map. We shall use the following characterisation of the left Kan extension of the restriction operator.

Proposition 5.1.15 *Let $X \in |\widehat{\mathbb{P}}|$ with decompositions*

$$X \cong \sum_{i \in X(\tau)} \tau.(X_{|i}) + \sum_{a \in Ch} \sum_{j \in X(a)} a.(X_{|j}) + \sum_{\overline{a} \in \overline{Ch}} \sum_{k \in X(\overline{a})} \overline{a}.(X_{|k})$$

and let Λ be a set of channels, then

$$(X/\Lambda_!) \cong \sum_{i \in X(\tau)} \tau.(X_{|i}/\Lambda_!) + \sum_{a \in Ch/\Lambda} \sum_{j \in X(a)} a.(X_{|j}/\Lambda_!) + \sum_{\overline{a} \in \overline{Ch}/\overline{\Lambda}} \sum_{k \in X(\overline{a})} \overline{a}.(X_{|k}/\Lambda_!) \ .$$

The proposition above is an immediate consequence of the following lemma.

Lemma 5.1.16 *Let X be a presheaf over $\mathbb{P}$ and $L \subseteq Ch$ a set of channel names, then the following holds:*

1. $(\tau.X)/\Lambda_! \cong \tau.(X/\Lambda_!)$
2. $(\alpha.X)/\Lambda_! \cong \alpha.(X/\Lambda_!)$ *if* $\alpha \notin \Lambda \cup \overline{\Lambda}$.
3. $(\alpha.X)/\Lambda_! \cong \emptyset$ *if* $\alpha \in \Lambda \cup \overline{\Lambda}$.

Proof: The proofs of all three items follow the same pattern. For the first case, recall that $\tau.X = In_{\tau!}\lfloor X \rfloor$. If by convention we write $j : \mathbb{P}_\perp \to \widehat{\mathbb{P}}$ for the functor such that $j(\perp) = \emptyset$ and $j(\lfloor P \rfloor) = \mathrm{y}_{\mathbb{P}}(P)$, we have the following:

$$\begin{array}{rcll}
(\tau.X)/\Lambda_! & = & \int^P \lfloor X \rfloor(P) \,.\, (In_\tau(P))/\Lambda & \text{(by definition)} \\
& = & \int^P \lfloor X \rfloor(P) \,.\, \tau.(j(P)/\Lambda_!) & \text{(by definition)} \\
& \cong & \tau.(\int^P \lfloor X \rfloor(P) \,.\, (j(P)/\Lambda_!)) & \text{(since } \tau. \text{ preserves connected colimits)} \\
& \cong & \tau.(\int^P X(P) \,.\, P/\Lambda) & \text{(since } j(\perp)/\Lambda_! = \emptyset) \\
& = & \tau.(X/\Lambda_!) & \text{(by definition).}
\end{array}$$

The proof of the second case is completely analogous, while in the third case all the contribution to the colimit are empty presheaves, hence the colimit itself is empty. □

Relabelling The interpretation is given exactly as in Chapter 3, since the function

$$f : Ch \to Ch$$

is naturally extended to a function

$$f' : L \to L$$

by putting $f'(\tau) = \tau$ and $f'(\overline{a}) = \overline{f(a)}$. In the details, $-[f] : \mathbb{P} \to \widehat{\mathbb{P}}$ is defined inductively as:

- $\alpha[f] = \mathrm{y}_{\mathbb{P}}(f'\alpha)$
- $(\alpha P)[f] = f'(\alpha).(\mathrm{y}_{\mathbb{P}}(P[f]))$.

We have the following characterisation, based on the decomposition of presheaves:

Proposition 5.1.17 *Let $X \in |\widehat{\mathbb{P}}|$ with decompositions*

$$X \cong \sum_{i \in X(\tau)} \tau.(X_{|i}) + \sum_{a \in Ch} \sum_{j \in X(a)} a.(X_{|j}) + \sum_{\overline{a} \in \overline{Ch}} \sum_{k \in X(\overline{a})} \overline{a}.(X_{|k})$$

and let $f : Ch \to Ch$ be a relabelling function, then

$$X[f]_! \cong \sum_{i \in X(\tau)} \tau.(X_{|i}[f]_!) \sum_{a \in Ch} \sum_{j \in X(a)} f(a).(X_{|j}[f]) + \sum_{\overline{a} \in \overline{Ch}} \sum_{k \in X(\overline{a})} \overline{f(a)}.(X_{|k}[f])$$

If t is a **CCS** term with free variables in $FV(t)$, its interpretation will be given by a functor

$$[\![t]\!] : \prod_{x \in FV(t)} \widehat{\mathbb{P}} \to \widehat{\mathbb{P}} \,,$$

defined compositionally as follows:

- $[\![\mathbf{Nil}]\!] = \emptyset$, the empty presheaf
- $[\![\alpha.t]\!]$:

$$\textstyle\prod_{x \in FV(t)} \widehat{\mathbb{P}} \xrightarrow{\quad [\![t]\!] \quad} \widehat{\mathbb{P}} \xrightarrow{\quad \alpha. \quad} \widehat{\mathbb{P}}$$

- $[\![t|t']\!]$:

$$\textstyle\prod_{x \in FV(t) \cup FV(t')} \widehat{\mathbb{P}} \xrightarrow{\langle [\![t]\!]\pi_t, [\![t']\!]\pi'_t \rangle} \widehat{\mathbb{P}} \times \widehat{\mathbb{P}} \xrightarrow{\quad | \quad} \widehat{\mathbb{P}} \,,$$

  where the π's are the obvious projection functors, e.g.,

$$\pi_t : \prod_{x \in FV(t) \cup FV(t')} \widehat{\mathbb{P}} \longrightarrow \prod_{x \in FV(t)} \widehat{\mathbb{P}}$$

  sends any tuple of presheaves indexed by the free variables of both t and t' to the tuple of those presheaves that corresponds to free variables of t only.
- $[\![t[f]]\!]$:

$$\textstyle\prod_{x \in FV(t)} \widehat{\mathbb{P}} \xrightarrow{\quad [\![t]\!] \quad} \widehat{\mathbb{P}} \xrightarrow{\quad -[f]_! \quad} \widehat{\mathbb{P}}$$

- $[\![t/\Lambda]\!]$:

$$\textstyle\prod_{x \in FV(t)} \widehat{\mathbb{P}} \xrightarrow{\quad [\![t]\!] \quad} \widehat{\mathbb{P}} \xrightarrow{\quad -/\Lambda_! \quad} \widehat{\mathbb{P}}$$

- $[\![x]\!]$:

$$\widehat{\mathbb{P}} \xrightarrow{1_{\widehat{\mathbb{P}}}} \widehat{\mathbb{P}}$$

- $[\![\mathrm{rec}\, y.t]\!]$: Let

$$t^y : \prod_{x \in FV(t)/\{y\}} \widehat{\mathbb{P}} \times \omega \longrightarrow \widehat{\mathbb{P}}$$

  be the functor

$t^y(\vec{X}, 0) = [\![t]\!](\vec{X}, \emptyset)$
$t^y(\vec{X}, n+1) = [\![t]\!](\vec{X}, t^y(\vec{X}, n))$.
Define $[\![\mathrm{rec}\, y.t]\!] = \int^n t^y(\vec{X}, n)$ (cf. Theorem 4.1.5). Since all the functors denoted by term constructors preserves colimits of ω-chain, the denotation of a recursively defined process is a fixed point, i.e.,

$$[\![\mathrm{rec}\, y.t]\!] \stackrel{\mathrm{def}}{=} \int^n t^y(\vec{X}, n) \cong [\![t]\!](\vec{X}, [\![\mathrm{rec}\, y.t]\!](\vec{X})) \ . \tag{5.2}$$

A categorical version of the usual Substitution Lemma holds:

Lemma 5.1.18 *Let t be a process term with free variables in $FV(t)$ and let $\vec{x}$ be a vector of free variables of t. Let $\vec{u}$ be a vector of closed process term of matching length with that of $\vec{x}$, then the following two functors are naturally isomorphic:*

$$[\![t]\!](\overrightarrow{[\![u]\!]}) \cong [\![t[\vec{u}/\vec{x}]]\!] \ .$$

Proof:[Sketch] Thanks to the compositional semantics the proof is an easy induction on the structure of the term t. The case when t is a recursively defined process is dealt with using the definition of $[\![\mathrm{rec}\, y.t]\!]$ as the colimit of an ω-chain. $\square$

Observe in particular that if t is a term with only one free variable y, then

$$[\![\mathrm{rec}\, y.t]\!] \cong [\![t[\mathrm{rec}\, y.t/y]]\!] \ .$$

In fact recursion is the reason for this non-standard formulation of the substitution lemma, since to prove, say, that

$$[\![\mathrm{rec}\, y.t]\!]\overrightarrow{[\![u]\!]} \cong [\![\mathrm{rec}\, y.t[\vec{u}/\vec{x}]]\!] \ ,$$

where $\vec{x}$ is now assumed to span over the whole of $FV(\mathrm{rec}\, y.t)$, one needs to show that the following isomorphism holds:

$$\int^n t^y(\overrightarrow{[\![u]\!]}, n) \cong \int^n (t[\vec{u}/\vec{x}])^y(n) \ .$$

And to have this, is not enough to know that at each n,

$$t^y(\overrightarrow{[\![u]\!]}, n) \cong (t[\vec{u}/\vec{x}])^y(n) \ ,$$

but one needs also to know that these isomorphisms are natural with respect to the relation $\leq$ on numbers.

We now show the agreement between the operational and denotational semantics using the transition relation on presheaves.

Theorem 5.1.19 *Let t be a closed* **CCS** *term. Then*

1. $t \stackrel{\alpha}{\longrightarrow} t'$ *implies* $\exists X\ [\![t]\!] \stackrel{\alpha}{\longrightarrow} X$ *and* $X \cong [\![t']\!]$.

2. $[\![t]\!] \xrightarrow{\alpha} X$ *implies* $\exists\, t'\ t \xrightarrow{\alpha} t'$ *and* $[\![t']\!] \cong X$.

Proof:

1. The proof goes by rule induction [137]. We exemplify it by looking at the most interesting cases.

$$\frac{}{\alpha t \xrightarrow{\alpha} t} \quad :$$

$[\![\alpha t]\!] = \tau.[\![t]\!] \xrightarrow{\tau} [\![t]\!]$, by definition of τ.

$$\frac{t \xrightarrow{\alpha} u}{t/\Lambda \xrightarrow{\alpha} u/\Lambda}(\alpha, \overline{\alpha} \notin \Lambda) \quad :$$

By inductive hypothesis, $[\![t]\!] \xrightarrow{\alpha} X \cong [\![u]\!]$. Using the characterisation of Proposition 5.1.15 we have that

$$([\![t]\!]/\Lambda_!) \cong \sum_{i \in [\![t]\!](\tau)} \tau.([\![t]\!]_{|i}/\Lambda_!) + \sum_{a \in Ch/\Lambda} \sum_{j \in [\![t]\!](a)} a.([\![t]\!]_{|j}/\Lambda_!) + \sum_{\overline{a} \in \overline{Ch}/\overline{\Lambda}} \sum_{k \in [\![t]\!](\overline{a})} \overline{a}.([\![t]\!]_{|k}/\Lambda_!)\,.$$

Consequently if $\alpha, \overline{\alpha} \notin \Lambda$,

$$[\![t/\Lambda]\!] = ([\![t]\!]/\Lambda_!) \xrightarrow{\alpha} X/\Lambda_! \cong ([\![u]\!]/\Lambda_!) = [\![u/\Lambda]\!]\ .$$

$$\frac{t_1 \xrightarrow{a} u_1 \qquad t_2 \xrightarrow{\overline{a}} u_2}{t_1|u_1 \xrightarrow{\tau} t_2|u_2} \quad :$$

By inductive hypothesis one has that $[\![t_1]\!] \xrightarrow{a} X_1 \cong [\![u_1]\!]$ and $[\![t_2]\!] \xrightarrow{\overline{a}} X_2 \cong [\![u_2]\!]$. By definition there exist $i \in [\![t_1]\!](a)$ and $j \in [\![t_2]\!](\overline{a})$, such that

$$X_1 = [\![t_1]\!]_{|i} \qquad \text{and} \qquad X_2 = [\![t_2]\!]_{|j}\ .$$

Hence, by Proposition 5.1.14 there exists $k \in [\![t_1]\!]|[\![t_2]\!](\tau)$ such that

$$X_1|X_2 \cong ([\![t_1]\!]|[\![t_2]\!])_{|k}\ .$$

Hence we have that

$$[\![t_1]\!]|[\![t_2]\!] \xrightarrow{\tau} ([\![t_1]\!]|[\![t_2]\!])_{|k} \cong X_1|X_2 \cong [\![u_1]\!]|[\![u_2]\!] = [\![u_1|u_2]\!]\ .$$

$$\frac{t[\mathrm{rec}\, x.t/x] \xrightarrow{\alpha} u}{\mathrm{rec}\, x.t \xrightarrow{\alpha} u} \quad :$$

We know that the denotation of a recursively defined process is given by a fixed point $[\![\mathrm{rec}\, x.t]\!] \cong [\![t[\mathrm{rec}\, x.t/x]]\!]$. Therefore, using the inductive hypothesis,

$$[\![\mathrm{rec}\, x.t]\!] \cong [\![t[\mathrm{rec}\, x.t/x]]\!] \xrightarrow{\alpha} X \cong [\![u]\!]\ .$$

2. Here the proof goes by structural induction on the structure of t and it is a straightforward verification.

□

We know (Proposition 5.1.12) that two presheaves over L^+ are open map bisimilar if the associated transition systems are strongly bisimilar. Hence, combining with the Theorem above we have the following corollary:

Corollary 5.1.20 *Two closed* **CCS** *terms are strong bisimilar if and only if they denote L^+-open map bisimilar presheaves.*

5.1.6 Remarks

We have treated this (easy) example in full detail even if it might seem a little artificial to call for rather heavy categorical machinery to discuss something that is very well understood anyway. Our reason for doing so is that the pattern followed here sets a template that we shall employ also later when dealing with more complicated situations (see Chapter 7 and Chapter 8). This suggests, in particular, the possibility of considering a metalanguage for recursively defined path categories and presheaves in connection with (a fragment of) the type theory of Chapter 4 (that we shall extend with recursive types in the next chapter). The hope is that, at least for the fragment excluding the exponential, it will be possible to induce from the definition of the path category automatically decomposition results and transition relations for the corresponding presheaves and, especially, an operational characterisation of open map bisimulation. Such a metalanguage has been recently considered in [140]. We will not expand on this line of research here.

5.2 CCS with value passing

We briefly consider now another example drawn from [138]. We essentially take an extension of **CCS** obtained by allowing values to be sent along channels. A typical output action, then, will be represented instead that with the symbol $\overline{a}$ with the expression $\overline{a}v$, meaning that one is observing the output of the value v along the channel a. More complicated is the situation with input actions. As we shall see in fact there are (at least) two natural ways of thinking about input. This fact is reflected in the corresponding notion of bisimulation and in our setting is captured by a modification in the input part of the domain equation.

For illustrative purposes we slightly deviate from the presentation of the language given in [138] by introducing the notion of abstraction [85, 118] beside that of process. Still all the results we shall claim are easily derivable from those of [138] and we shall skip all the proofs.

An advantage of having both models as objects of the same (bi)category is that it is possible to formally relate them using the arrows of the category. We shall sketch this in the last section.[2]

5.2.1 The term language

The terms of the language we shall consider are defined by the following grammar:

$$\begin{array}{lcl} t & ::= & \mathbf{Nil} \mid \tau.t \mid \overline{a}e.t \mid af \mid t_1 \mid t_2 \mid \sum_{i\in I} t_i \mid X \mid \mathrm{rec}\, X.t \\ f & ::= & (x)t \ , \end{array}$$

where as before $a \in Ch$, while the e's stands for expressions that we do not specify any further other than saying that they might contain value variables $x \in VVars$ and that when evaluated they always return an element of a set of values V. The X's are process variables drawn from a set $PVars$ disjoint from $VVars$. With respect to **CCS** we have omitted here the relabelling and restriction operator that can be put back without much difficulty. With respect to [138] we have adopted the "π-calculus notation" for input and output instead of the ?, ! one ($a(x)$ for $a?x$ and $\overline{a}e$ for $a!e$). More relevant changes come from the introduction of abstractions (f, like function), by the omission of a matching operator ($[e_1 = e_2]t$) than can be put back without any problem. In his paper [138] Winskel also restrict variables in a recursive expression to occur guarded by prefixes. We adopt this restriction as well in order to be able to refer to it for the results that we shall quote.

As announced we shall have two operational semantics that will differ in the input clauses. The idea is that when a process, say $a(x)t$ performs an input action this can be done in two different ways. The process can simply communicate its will to receive some input along the channel a,

$$a(x)t \xrightarrow{a} (x)t$$

and then becomes an abstraction, i.e., a function that waits for the input (say v) to come and then proceed as t where v has been substituted for x:

$$\frac{t[v/x] \xrightarrow{\alpha} t'}{(x)t \xrightarrow{v\mapsto\alpha} t'} \ .$$

Because of this delay between the input action a and the actual receiving of a value v, this is usually denoted as the *late* semantics of input actions. Alternatively the two things can occur at the same time and one has the *early* transition rule

$$\frac{}{a(x)t \xrightarrow{av} t[v/x]} \ .$$

[2]An observation of this kind was also made by Ian Stark in our joint work [25] as we shall report in Section 7.6.

The operational semantics of the language is then given by the following rules,

$$\frac{}{\tau.t \xrightarrow{\tau} t} \qquad \frac{}{\overline{a}e.t \xrightarrow{\overline{a}v} t} \qquad \frac{t_j \xrightarrow{\alpha} t'_j}{\sum_{i\in I} t_i \xrightarrow{\alpha} t'_j} \ (j \in I)$$

$$\frac{t \xrightarrow{\alpha} t'}{t \mid u \xrightarrow{\alpha} t' \mid u} \quad \frac{u \xrightarrow{\alpha} u'}{t \mid u \xrightarrow{\alpha} t \mid u'} \quad \frac{t[\mathrm{rec}\, x.t/x] \xrightarrow{\alpha} t'}{\mathrm{rec}\, x.t \xrightarrow{\alpha} t'} \ ,$$

where in the axiom for the output actions, e is taken to be closed and evaluating to v. One also takes the rules

$$\frac{}{a(x)t \xrightarrow{av} t[v/x]}$$

$$\frac{t \xrightarrow{\overline{a}v} t' \quad u \xrightarrow{av} u'}{t \mid u \xrightarrow{\tau} t' \mid u'} \qquad \frac{t \xrightarrow{av} t' \quad u \xrightarrow{\overline{a}v} u'}{t \mid u \xrightarrow{\tau} t' \mid u'}$$

for the *early* semantics or the rules

$$\frac{}{a.f \xrightarrow{a} f} \qquad \frac{t[v/x] \xrightarrow{\alpha} u}{(x)t \xrightarrow{v\mapsto\alpha} u}$$

$$\frac{t \xrightarrow{\overline{a}v} t' \quad u \xrightarrow{a} (x)u'}{t \mid u \xrightarrow{\tau} t' \mid u'[v/x]} \qquad \frac{t \xrightarrow{a} (x)t' \quad u \xrightarrow{\overline{a}v} u'}{t \mid u \xrightarrow{\tau} t'[v/x] \mid u'}$$

for the late semantics. In the case of the late semantics, expressions like $(x)t|t'$ that can be derived by application of the rules has to be understood as standing for $(x)(t|t')$, where every free occurrence of x in t' has been renamed to avoid capturing.

When thinking of transitions derived within the early set of rules, we shall write a subscript E to the transition arrows ($\longrightarrow_E$). Analogously for the late transitions we shall write ($\longrightarrow_L$). Consistently with the two different operational semantics, one has two different notions of bisimulation.

Definition 5.2.1 *A symmetric relation R between closed process terms is an* early bisimulation *if*
$t\,R\,t'$ implies

$$t \xrightarrow{\alpha}_E u \Longrightarrow \exists u' \,.\, (t' \xrightarrow{\alpha}_E u' \quad \wedge \quad uRu') \ ,$$

with $\alpha \in (Ch \times V) \cup (\overline{Ch} \times V) \cup \{\tau\}$.

A symmetric relation R between closed process terms is an late bisimulation *if*
$t\,R\,t'$ implies

$$t \xrightarrow{\alpha}_L u \Longrightarrow \exists u' \,.\, (t' \xrightarrow{\alpha}_L u' \quad \wedge \quad uRu') \ ,$$

with $\alpha \in (\overline{Ch} \times V) \cup \{\tau\}$ and

$$t \xrightarrow{a}_L f = (x)u \Longrightarrow \exists f' = (y)u' \,.\, (t' \xrightarrow{a}_L f' \quad \wedge \quad \forall v \in V \,.\, f[v]Rf'[v/y]),$$

with $a \in Ch$.

REMARK: We could have made a more extended use of abstractions and equivalently define a *late bisimulation* to consist of two (typed) relations R^t and R^f, with:

- R^t a symmetric binary relation between closed processes as above as far as the first clause is concerned but with the second replaced by
$$t \xrightarrow{a}_L f \Longrightarrow \exists f' . (t' \xrightarrow{a}_L f' \quad \wedge \quad fR^f f'),$$
with $a \in Ch$.
- R^f a symmetric binary relation between closed abstractions such that $fR^f g$ implies
$$f \xrightarrow{v \mapsto \alpha}_L t \Longrightarrow \exists u . (g \xrightarrow{v \mapsto \alpha}_L u \quad \wedge \quad tR^t u) \ ,$$
with $\alpha \in (\overline{Ch} \times V) \cup \{\tau\}$ and
$$f \xrightarrow{v \mapsto a}_L f' \Longrightarrow \exists g' . (g \xrightarrow{v \mapsto a}_L g' \quad \wedge \quad f'R^f g') \ ,$$
with $a \in Ch$.

We shall not give the denotational semantics of this language as we did for **CCS** in the previous section and refer to [138] for details about it. We simply write down here the necessary equations that reflects the early vs. late approach:

$$\begin{aligned}
\mathbb{P}^E &= \mathbb{P}^E_\perp + \sum_{(\overline{a},v) \in \overline{Ch} \times V} \mathbb{P}^E_\perp + \sum_{(a,v) \in Ch \times V} \mathbb{P}^E_\perp \\
\mathbb{P}^L &= \mathbb{P}^L_\perp + \sum_{(\overline{a},v) \in \overline{Ch} \times V} \mathbb{P}^L_\perp + \sum_{a \in Ch} \mathbb{F}^L_\perp \\
\mathbb{F}^L &= (\mathbb{V} \multimap \mathbb{P}) \ ,
\end{aligned}$$

where $\mathbb{V}$ is the set V regarded as a discrete category.

After a denotational semantics is given the main result will obviously be that early bisimulation correspond to $\mathbb{P}^E$-open map bisimulation and late bisimulation correspond to $\mathbb{P}^L$-open map bisimulation.

5.2.2 A map between models

We end up this section by sketching how one can derive an arrow $\mathbb{P}^L \longrightarrow\!\!\!\!\!\!\!+\;\; \mathbb{P}^E$ in **Prof** that maps the late denotation of terms onto the early one.

A functor $\mathbb{P}^E \to \mathbb{P}^L$: The solutions to the equation defining $\mathbb{P}^E$ and $\mathbb{P}^L$ can be given as partial orders inductively defined as follows:

$\mathbb{P}^E$: There are "roots", $\tau.$, $a!v.$ and $a?v.$, corresponding to the silent action, output and input components, for any $a \in Ch$ and $v \in V$. Above these in the order relation we inductively find the following, for any $P \in \mathbb{P}^E$ and for any $P \leq P'$ in $\mathbb{P}^E$:

$$\begin{array}{ccc}
\tau. \leq \tau.P & a!v. \leq a!v.P & a?v. \leq a?v.P \\
\tau.P \leq \tau.P' & a!v.P \leq a!v.P' & a?v.P \leq a?v.P' \ .
\end{array}$$

$\mathbb{P}^L$: There are "roots", $\tau.$, $a!v.$ and $a?$, corresponding to the silent action, output and input components, for any $a \in Ch$ and $v \in V$. Above these in the order relation we inductively find the following, for any $v \in V$, $P \in \mathbb{P}^L$ and for any $P \leq P'$ in $\mathbb{P}^L$:

$$\begin{array}{ccc} \tau. \leq \tau.P & a!v. \leq a!v.P & a? \leq a?(v \mapsto P) \\ \tau.P \leq \tau.P' & a!v.P \leq a!v.P' & a?(v \mapsto P) \leq a?(v \mapsto P') \ . \end{array}$$

It should be clear that the expressions of the form $(v \mapsto P)$ denotes the elements of $\mathbb{F}^L$, that are ordered pointwise, i.e., $P \leq P'$ implies $(v \mapsto P) \leq (v \mapsto P')$.

It is then easy to define a functor (i.e., a monotone map) $EL : \mathbb{P}^E \to \mathbb{P}^L$ from *early* paths to *late* ones, by defining

$$\begin{array}{ccc} EL(\tau.) = \tau. & EL(a!v.) = a!v & EL(a?v.) = a? \\ EL(\tau.P) = \tau.EL(P) & EL(a!v.P) = a!v.EL(P) & EL(a?v.P) = a?(v \mapsto P) \ . \end{array}$$

The extension $EL_!$ of this functor will not map the early denotational semantics ($[\![\cdot]\!]^E$) onto the late one ($[\![\cdot]\!]^L$), since for instance a process $a?x$ will denote in $\widehat{\mathbb{P}^E}$ the presheaf $\sum_{v\in V} \mathrm{y}_{\mathbb{P}^E}(a?v.)$; while in $\widehat{\mathbb{P}^L}$ it will denote the presheaf $\mathrm{y}_{\mathbb{P}^L}(a?)$, but

$$EL_!([\![a?x]\!]^E) \cong \sum_{v\in V} \mathrm{y}_{\mathbb{P}^L}(a?) \cong \sum_{v\in V} [\![a?x]\!]^L \ncong [\![a?x]\!]^L \ .$$

Still the right adjoint EL^* to $EL_!$ will map the late semantics onto the early one. Since we have not given a precise definition of the two semantics, it is impossible to give a formal proof of this result here. In Chapter 7 we shall give a presheaf semantics for the π-calculus for both the late and early variant and prove an analogous result in that case. The argument we shall employ, in a simplified form, would work in this case as well. As an illustrative example, we can anyway sketch the claim to be correct for the case of the process $a?x$. In fact, for any $P \in \mid \mathbb{P}^E \mid$,

$$EL^*[\![a?x]\!]^L(P) = [\![a?x]\!]^L(EL(P)) \ .$$

Hence, if $P = a?v.$, $EL^*[\![a?x]\!]^L(P) \cong \{*\}$; in any other case $EL^*[\![a?x]\!]^L(P) = \emptyset$, therefore

$$EL^*[\![a?x]\!]^L \cong \sum_{v\in V} \mathrm{y}_{\mathbb{P}^E}(a?v.) \cong [\![a?x]\!]^E \ .$$

Since colimit preserving functors preserve bisimulation, one can use the denotational semantics to deduce that terms that are late bisimilar are also early bisimilar. In fact, if two closed process terms are late bisimilar then their denoted (late) presheaves are $\mathbb{P}^L$-open map bisimilar. By composing with EL^*, using the result that $EL^*[\![\cdot]\!]^L \cong [\![\cdot]\!]^E$, we deduce that their denoted (early) presheaves are $\mathbb{P}^E$-open map bisimilar and hence the terms are early bisimilar, too.

Chapter 6

A Theory of Recursive Domains

In Chapter 4 we introduced a bicategory, **Prof**, that we think of as being a category of domains with built-in notions of bisimulation (as given by open maps). In the previous chapter we saw examples of presheaf models for concurrent process calculi whose base categories had been provided by solving appropriate recursive domain equations.

In this chapter we give a generalisation of classical results [125, 120] about the solution of recursive domain equations that, following the axiomatic approach of [30, 33, 105, 34], will justify our intuitive understanding. In fact, after Freyd [35, 36], we shall consider the notion of algebraic compactness and use a "pseudo" version of the Basic Lemma (see [125]) to deduce pseudo algebraic compactness for a class of 2-categories that include **Cocont** which is the 2-categorical equivalent of **Prof**. We develop a domain theoretical approach to open map bisimulation using relational structures [94, 100] and induction/coinduction principles for recursively defined domains and coinduction properties based on bisimulation [99, 31].

For technical reasons (so as to have less coherence conditions to worry about) we shall state our results as holding for 2-categories. Thus, we prefer in this chapter the 2-category **Cocont** over **Prof** (cf. Chapter 4) and consider the interpretation of the type theory of Section 4.5 in **Cocont**.

We shall see in Section 6.2 some coherence results that allow further generalisations to bicategories but we have chosen not to pursue this aim any further as far as this thesis is concerned.

6.1 Local-characterisation theorem

In denotational semantics, domains are often specified using recursive equations. We are interested in tools for solving such equations. The first was given by Scott [120] with his inverse-limit construction in the category of countably based continuous lattices and continuous functions. Smyth and Plotkin [125] building on previous work of Wand [134] provided a categorical framework based on *order-enriched categories* and a general version of Scott's result applicable to a wider class of categories was given. This is our starting point. We want to generalise Smyth and Plotkin's results so as to cover the cases of interest to us. We generalise in three directions:

1. First of all, following some categorical folklore, e.g., [129, 73], we move from embedding projection pairs (viz. coreflections in general categorical terms) to adjoint pairs. Hence we shall consider chains of adjoint pairs of arrows in 2-categories. For the order-enriched case, this issue had already been addressed also in [124].
2. We move up a level from the order-enriched case and consider 2-categories whose hom-categories have colimits of ω-chains.
3. As a consequence of the two points above, we shall consider pseudo-limits rather than 2-limits.

We begin by giving some preliminary definitions, which also serve to fix some notation and terminology.

Definition 6.1.1 *Let $\mathcal{K}$ be a 2-category, write $\mathcal{K}_{\cong}$ for the sub-2-category with the same objects, same arrows but only isomorphic 2-cells.*

TERMINOLOGY: To improve readability we shall often write "pseudo cell" instead of "isomorphic 2-cell".

Definition 6.1.2 *Let $\mathcal{K}$ be a 2-category. We define $\mathcal{K}^{\mathrm{adj}}$ to be the* 2-category of adjunctions *as follows. The objects of $\mathcal{K}^{\mathrm{adj}}$ are those of $\mathcal{K}$; whilst $\mathcal{K}^{\mathrm{adj}}(A,B)$ is the category whose objects are tuples $(\eta, \varepsilon : f \dashv g : B \to A)$, where $f \dashv g$ is an adjunction in $\mathcal{K}$ with unit η and counit ε, and 2-cells $(\eta, \varepsilon : f \dashv g) \Rightarrow (\eta', \varepsilon' : f' \dashv g')$ are given by pairs of 2-cells $\sigma : f \Rightarrow f' : A \to B$ and $\tau : g \Rightarrow g' : B \to A$ in $\mathcal{K}$, such that the diagrams*

$$
\begin{tikzcd}
1_A \arrow[r, Rightarrow, "\eta"] \arrow[dr, Rightarrow, "\eta'"'] & gf \arrow[d, Rightarrow, "\tau\sigma"] \\
& g'f'
\end{tikzcd}
\qquad
\begin{tikzcd}
fg \arrow[d, Rightarrow, "\sigma\tau"'] \arrow[dr, Rightarrow, "\varepsilon"] & \\
f'g' \arrow[r, Rightarrow, "\varepsilon'"'] & 1_B
\end{tikzcd}
$$

commute.
The horizontal and vertical compositions of arrows and 2-cells are defined as follows:

Horizontal:

- *If $(\eta, \varepsilon : f \dashv g : B \to A)$ and $(\eta', \varepsilon' : f' \dashv g' : C \to B)$ are two arrows of $\mathcal{K}^{\mathrm{adj}}$, their horizontal composition is given by the following tuple:*

$$((g\eta' f) \cdot \eta, (f'\varepsilon g') \cdot \varepsilon' : f'f \dashv gg' : C \to A) \ .$$

- *If (σ, τ) and (σ', τ') are 2-cells between horizontally composable arrows, their horizontal composition is given by*

$$(\sigma'\sigma, \tau'\tau) \ .$$

Vertical: *If*

$$(\sigma, \tau) : (\eta, \varepsilon : f \dashv g : B \to A) \Longrightarrow (\eta', \varepsilon' : f' \dashv g' : B \to A)$$

and

$$(\sigma', \tau') : (\eta', \varepsilon' : f' \dashv g' : B \to A) \Longrightarrow (\eta'', \varepsilon'' : f'' \dashv g'' : B \to A)$$

are two 2-cells, then their vertical composition is given by

$$(\sigma' \cdot \sigma, \tau' \cdot \tau) \ .$$

We write $\mathcal{K}^{\mathrm{cor}}$ for the full sub-2-category of $\mathcal{K}^{\mathrm{adj}}$ consisting of coreflections*; i.e. tuples $(\eta, \varepsilon : f \dashv g)$ where η is a pseudo cell.*

We concentrate on pseudo-colimits of ω-chains.

Definition 6.1.3 (ω-Chains) *Let $\mathcal{K}$ be a 2-category. An ω-*chain *in $\mathcal{K}$ is given by an ω-indexed family of arrows $\langle f_n : A_n \to A_{n+1} \rangle$. For $l \geq n$, we write $f_{n,l} : A_n \to A_l$ for the inductively defined arrow $f_{n,l+1} \stackrel{\mathrm{def}}{=} f_l f_{n,l}$, where $f_{n,n} \stackrel{\mathrm{def}}{=} 1_{A_n}$. Dually an ω^{op}-chain is given by an indexed family of arrows $\langle g_n : A_{n+1} \to A_n \rangle$ with $g_{l+1,n} \stackrel{\mathrm{def}}{=} g_{l,n} g_{l+1}$ and $g_{n,n} \stackrel{\mathrm{def}}{=} 1_{A_n}$.*

Definition 6.1.4 (Pseudo-cones of ω-chains) *A* pseudo cone *for an ω-chain*

$$\langle f_n : A_n \to A_{n+1} \rangle$$

is given by the following data:

- *An object A.*
- *An ω-indexed family of arrows $\langle \varphi_n : A_n \to A \rangle$.*
- *An ω-indexed family of pseudo cells $\langle \Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n \rangle$.*

Dually a pseudo-cone for an ω^{op}-chain $\langle g_n : A_{n+1} \to A_n \rangle$ is given by:

- *An object A.*
- *An ω-indexed family of arrows $\langle \gamma_n : A \to A_n \rangle$.*
- *An ω-indexed family of pseudo cells $\langle \Gamma_n : g_n \gamma_{n+1} \stackrel{\sim}{\Longrightarrow} \gamma_n \rangle$.*

Given a 2-category $\mathcal{K}$ we will be interested in pseudo-cones of ω-chains in $\mathcal{K}^{\mathrm{adj}}$. We spell out in terms of data from $\mathcal{K}$ what a pseudo-cone of ω-chains in $\mathcal{K}^{\mathrm{adj}}$ amounts to.

Definition 6.1.5 (Pseudo-cones of ω-chains in $\mathcal{K}^{\mathrm{adj}}$) *A pseudo cone for an ω-chain*

$$\langle \eta_n, \varepsilon_n : f_n \dashv g_n : A_{n+1} \longrightarrow A_n \rangle$$

in $\mathcal{K}^{\mathrm{adj}}$ consists of:

- *An object A.*
- *An ω-indexed family $\langle \iota_n, \jmath_n : \varphi_n \dashv \gamma_n : A \to A_n \rangle$ of adjoint pairs.*
- *An ω-indexed family $\langle \Phi_n, \Gamma_n \rangle$ of pseudo cells $\Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n$ and $\Gamma_n : g_n \gamma_{n+1} \stackrel{\sim}{\Longrightarrow} \gamma_n$ such that the squares*

$$\mathcal{I}_n : \quad \begin{tikzcd} 1_{A_n} \arrow[rr, Rightarrow, "\eta_n"] \arrow[d, Rightarrow, "\iota_n"'] & & g_n f_n \arrow[d, Rightarrow, "g_n \iota_{n+1} f_n"] \\ \gamma_n \varphi_n & & g_n \gamma_{n+1} \varphi_{n+1} f_n \arrow[ll, Rightarrow, "\Gamma_n \Phi_n"] \end{tikzcd} \tag{6.1}$$

$$\mathcal{J}_n : \qquad \begin{tikzcd} \varphi_n\gamma_n \arrow[r, Rightarrow, "\Phi_n^{-1}\Gamma_n^{-1}"] \arrow[d, Rightarrow, "\jmath_n"'] & \varphi_{n+1}f_n g_n\gamma_{n+1} \arrow[d, Rightarrow, "\varphi_{n+1}\varepsilon_n\gamma_{n+1}"] \\ 1_A & \varphi_{n+1}\gamma_{n+1} \arrow[l, Rightarrow, "\jmath_{n+1}"] \end{tikzcd} \tag{6.2}$$

commute for all n.

It is important to observe that a pseudo cone for an ω-chain of adjunctions induces ω-chains in $\mathcal{K}(A_n, A_n)$ and $\mathcal{K}(A, A)$ and cones obtained from the diagrams (6.1) and (6.2) as follows:

- For every n, the chain

$$1_{A_n} \xRightarrow{\eta_n} g_n f_n \xRightarrow{g_n\eta_{n+1}f_n} \cdots \xRightarrow{g_{l,n}\eta_l f_{n,l}} g_{l+1,n}f_{n,l+1} \xRightarrow{g_{l+1,n}\eta_{l+1}f_{n,l+1}} \cdots$$

  in $\mathcal{K}(A_n, A_n)$ with cone

$$\begin{tikzcd} 1_{A_n} \arrow[r, Rightarrow] \arrow[d, Rightarrow] \arrow[dr, phantom, "\mathcal{I}_n"] & g_n f_n \arrow[r, Rightarrow] \arrow[d, Rightarrow] \arrow[dr, phantom, "g_n\mathcal{I}_{n+1}f_n"] & g_{n+2,n}f_{n,n+2} \arrow[r, Rightarrow] \arrow[d, Rightarrow] \arrow[dr, phantom, "\cdots"] & \cdots \\ \gamma_n\varphi_n & g_n\gamma_{n+1}\varphi_{n+1}f_n \arrow[l, Rightarrow] & g_{n+2,n}\gamma_{n+2}\varphi_{n+2}f_{n,n+2} \arrow[l, Rightarrow] & \cdots \arrow[l, Rightarrow] \end{tikzcd}$$

- In $\mathcal{K}(A, A)$,

$$\varphi_0\gamma_0 \xRightarrow{\Phi_0^{-1}\Gamma_0^{-1}} \xRightarrow{\varphi_1\varepsilon_0\gamma_1} \varphi_1\gamma_1 \xRightarrow{\Phi_1^{-1}\Gamma_1^{-1}} \xRightarrow{\varphi_2\varepsilon_1\gamma_2} \varphi_2\gamma_2 \Longrightarrow \cdots$$

  with cone

$$\begin{tikzcd} \varphi_0\gamma_0 \arrow[r, Rightarrow] \arrow[drr, Rightarrow] & {} \arrow[r, Rightarrow] \arrow[d, phantom, "\mathcal{J}_0"] & \varphi_1\gamma_1 \arrow[r, Rightarrow] \arrow[d, Rightarrow] & {} \arrow[r, Rightarrow] \arrow[d, phantom, "\mathcal{J}_1"] & \varphi_2\gamma_2 \arrow[r, Rightarrow] \arrow[dll, Rightarrow] \arrow[d, phantom, "\cdots"] & \cdots \\ & {} & 1_A & {} & {} & \end{tikzcd}$$

For the purpose of this chapter we shall call the above cones the *canonical cones*

$$\langle g_{l,n}f_{n,l}\rangle_l \stackrel{\cdot}{\Longrightarrow} \gamma_n\varphi_n$$

and

$$\langle \varphi_n\gamma_n\rangle \stackrel{\cdot}{\Longrightarrow} 1_A \ ,$$

respectively.

We give now explicitly an elementary description of pseudo-colimits of ω-chains in a 2-category.

Definition 6.1.6 (Pseudo-colimits of ω-chains) *A pseudo cone*

$$\langle \Phi_n : \varphi_{n+1}f_n \stackrel{\sim}{\Longrightarrow} \varphi_n : A_n \to A\rangle$$

for an ω-chain $\langle f_n : A_n \to A_{n+1}\rangle$ in a 2-category $\mathcal{K}$ is said to be a pseudo colimit *if it satisfies the following universal property:*

1. *For every pseudo cone* $\langle \Psi_n : \psi_{n+1} f_n \overset{\sim}{\Longrightarrow} \psi_n : A_n \to X\rangle$ *there exists an arrow* $u : A \to X$ *and an* ω*-indexed family of pseudo cells* $\langle \mu_n : u\varphi_n \overset{\sim}{\Longrightarrow} \psi_n\rangle$ *such that the square*

$$\begin{tikzcd} u\varphi_{n+1}f_n \arrow[r, Rightarrow, "\mu_{n+1}f_n"] \arrow[d, Rightarrow, "u\Phi_n"'] & \Psi_{n+1}f_n \arrow[d, Rightarrow, "\Psi_n"] \\ u\varphi_n \arrow[r, Rightarrow, "\mu_n"'] & \psi_n \end{tikzcd}$$

commutes for every n.

2. *For every pair of arrows* $u, v : A \to X$ *and every* ω*-indexed family of 2-cells*

$$\langle \xi_n : u\varphi_n \Rightarrow v\varphi_n\rangle$$

satisfying

$$\begin{tikzcd} u\varphi_{n+1}f_n \arrow[r, Rightarrow, "\xi_{n+1}f_n"] \arrow[d, Rightarrow, "u\Phi_n"'] & v\varphi_{n+1}f_n \arrow[d, Rightarrow, "v\Phi_n"] \\ u\varphi_n \arrow[r, Rightarrow, "\xi_n"'] & v\varphi_n \end{tikzcd},$$

there exists a unique 2-cell $\xi : u \Rightarrow v$ *such that* $\xi_n = \xi\varphi_n$.

We shall concentrate on 2-categories $\mathcal{K}$, whose hom-categories, $\mathcal{K}(A, B)$ are categories with colimits of ω-chains:

Definition 6.1.7 (ωCat) *Define* ω**Cat** *to be the (large) category of (locally small) categories with colimits of* ω*-chains and* ω*-cocontinuous functors, that is functors which preserve colimits of* ω*-chains.*

Define $\omega\mathbf{Cat}_0$ *to be the subcategory of* ω**Cat** *consisting of those categories with an initial object and initial object preserving functors.*

Clearly ω**Cat** (and $\omega\mathbf{Cat}_0$) are straightforward generalisations of the categories **Cpo** (and $\mathbf{Cppo}_\perp$) of (pointed) cpos and (strict) continuous functions.

Example 6.1.8 Cocont *is an* $\omega\mathbf{Cat}_0$*-category. In fact, for any two small categories,* $\mathbb{C}$ *and* $\mathbb{D}$*,* $\mathbf{Cocont}(\mathbb{C}, \mathbb{D}) \simeq \widehat{\mathbb{C} \times \mathbb{D}^{\mathrm{op}}}$ *is a cocomplete category. Moreover since the arrows of* **Cocont** *are colimit preserving functors, the composition functor preserves colimits of* ω*-chains as well as the initial presheaf.*

The announced generalisation of [125, Theorem 2], obtained with Marcelo Fiore, can now be stated as follows:

Theorem 6.1.9 (Local-characterisation) *Let* $\mathcal{K}$ *be an* ω**Cat***-category. For an* ω*-chain of adjunctions* $\langle \eta_n, \varepsilon_n : f_n \dashv g_n : A_{n+1} \to A_n\rangle$ *and a pseudo cone*

$$\langle \Phi_n : \varphi_{n+1} f_n \overset{\sim}{\Longrightarrow} \varphi_n : A_n \to A\rangle$$

for the ω*-chain* $\langle f_n : A_n \to A_{n+1}\rangle$*, the following are equivalent:*

1. *$\langle \Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n : A_n \to A \rangle$ is a pseudo colimit for $\langle f_n : A_n \to A_{n+1} \rangle$ in $\mathcal{K}$.*
2. *$\langle \Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n : A_n \to A \rangle$ is a pseudo colimit for $\langle f_n : A_n \to A_{n+1} \rangle$ in $\mathcal{K}_{\cong}$.*
3. *There is a pseudo cone of adjunctions*

$$(\Phi_n, \Gamma_n) : (\iota_{n+1}, \jmath_{n+1} : \varphi_{n+1} \dashv \gamma_{n+1})(\eta_n, \varepsilon_n : f_n \dashv g_n) \stackrel{\sim}{\Longrightarrow} (\iota_n, \jmath_n : \varphi_n \dashv \gamma_n)$$

such that the canonical cones $\langle \varphi_n \gamma_n \rangle \stackrel{\cdot}{\Longrightarrow} id_A$ and $\langle g_{l,n} f_{n,l} \rangle_l \stackrel{\cdot}{\Longrightarrow} \gamma_n \varphi_n$ are colimiting.

Proof: We prove the chain of implications (1) ⇒ (2) ⇒ (3) ⇒ (1). Clearly the first implication holds trivially since any pseudo colimit in $\mathcal{K}$ is a pseudo colimit in $\mathcal{K}_{\cong}$, too. Therefore we concentrate on the remaining two.

[2 implies 3:] We start by looking for suitable definitions of the right adjoints, γ_n's, to the φ_n's together with units ι_n's and coherence pseudo-cells Γ_n's for all n, showing that they satisfy the commutativity of diagram (6.1). Afterwards we define the $\jmath_n$'s showing the commutativity of diagram (6.2). Finally we verify the triangular identities that show $(\iota_n, \jmath_n)$ to be unit and counit of an adjunction $\varphi_n \dashv \gamma_n$. From now on, when not quantified, let n be an arbitrary natural number. If $m \geq n$, let the following be a colimiting cone in the hom-category $\mathcal{K}(A_m, A_n)$:

$$\mathcal{G}_{m,n} : \quad \begin{tikzcd} g_{m,n} \arrow[r, Rightarrow, "g_{m,n}\eta_m"] \arrow[dr, Rightarrow, "\iota_m^{m,n}"'] & g_{m+1,n} f_m \arrow[r, Rightarrow, "g_{m+1,n}\eta_{m+1} f_m"] \arrow[d, Rightarrow, "\iota_{m+1}^{m,n}"] & g_{m+2,n} f_{m,m+2} \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\iota_{m+2}^{m,n}"] & \cdots \\ & \overline{g}_{m,n} & \cdots & \end{tikzcd} \qquad (m \geq n)4$$

By precomposing $\mathcal{G}_{m+1,n}$ with f_m one obtains a colimiting cone for the above chain, where the first element is removed, that is for

$$g_{m+1,n} f_m \xrightarrow{g_{m+1,n}\eta_{m+1} f_m} g_{m+2,n} f_{m,m+2} \xrightarrow{g_{m+2,n}\eta_{m+2} f_{m,m+2}} \cdots \qquad .$$

Hence there exists a universal pseudo-cell, $\varpi^f_{m,n} : \overline{g}_{m,n} \stackrel{\sim}{\Longrightarrow} \overline{g}_{m+1,n} f_m$, such that, for all $k \geq m+1$,

$$\varpi^f_{m,n} \cdot \iota_k^{m,n} \quad = \quad \iota_k^{m+1,n} f_m \ . \tag{6.3}$$

Similarly, if $m \geq n+1$, by post composing $\mathcal{G}_{m,n+1}$ with g_n one derives the existence of a universal pseudo-cell $\varpi^g_{m,n} : \overline{g}_{m,n} \stackrel{\sim}{\Longrightarrow} g_n \overline{g}_{m,n+1}$, such that

$$\varpi^g_{m,n} \cdot \iota_k^{m,n} \quad = \quad g_n \iota_k^{m,n+1} \tag{6.4}$$

for all $k \geq m$. Define $\varpi_{m,n} : \overline{g}_{m,n} \stackrel{\sim}{\Longrightarrow} g_n \overline{g}_{m+1,n+1} f_m$ to be the universal pseudo-cell such that, for all $k \geq m+1$,

$$\varpi_{m,n} \cdot \iota_k^{m,n} \quad = \quad g_n \iota_k^{m+1,n+1} f_m \ . \tag{6.5}$$

It follows that

$$\begin{aligned} \varpi_{m,n} &= (\varpi^g_{m+1,n} f_m) \cdot \varpi^f_{m,n} && (6.6) \\ &= (g_n \varpi^f_{m,n+1}) \cdot \varpi^g_{m,n} \;. && (6.7) \end{aligned}$$

In particular, since $\iota^{n,n}_n = \iota^{n,n}_{n+1} \cdot \eta_n$, we have from (6.6), (6.3) and (6.4) that

$$\varpi_{n,n} \cdot \iota^{n,n}_n \;=\; (g_n \iota^{n+1,n+1}_{n+1} f_n) \cdot \eta_n \;. \tag{6.8}$$

Consider now the cone for the ω-chain $\langle f_m : A_m \to A_{m+1} \rangle_{m \geq n}$ of vertex A_n, arrows $\langle \overline{g}_{m,n} \rangle_{m \geq n}$ and pseudo-cells $\langle (\varpi^f_{m,n})^{-1} \rangle_{m \geq n}$. Using that the cone of Φ's is a pseudo colimit, there exists an arrow $\gamma_n : A \to A_n$ and pseudo-cells, $\varpi^n_m : \gamma_n \varphi_m \overset{\sim}{\Longrightarrow} \overline{g}_{m,n}$ for all $m \geq n$, such that

$$\varpi^n_m \cdot (\gamma_n \Phi_m) \;=\; (\varpi^f_{m,n})^{-1} \cdot (\varpi^n_{m+1} f_m) \tag{6.9}$$

Define

$$\iota_n \;\overset{\text{def}}{=}\; (\varpi^n_n)^{-1} \cdot \iota^{n,n}_n : 1_{A_n} \Rightarrow \gamma_n \varphi_n \;. \tag{6.10}$$

To deduce the existence of $\Gamma_n : g_n \gamma_{n+1} \overset{\sim}{\Longrightarrow} \gamma_n$, we use again the fact that the Φ's form a pseudo colimit for the chain $\langle f_m : A_m \to A_{m+1} \rangle_{m \geq n+1}$. We describe, in fact a family of pseudo-cells

$$\Upsilon^n_m : g_n \gamma_{n+1} \varphi_m \overset{\sim}{\Longrightarrow} \gamma_n \varphi_m \qquad\qquad (m \geq n+1)$$

such that for all $m \geq n+1$,

$$\Upsilon^n_m \cdot (g_n \gamma_{n+1} \Phi_m) \;=\; (\gamma_n \Phi_m) \cdot (\Upsilon^n_{m+1} f_m) \;. \tag{6.11}$$

Hence it will follow the existence of a unique $\Gamma_n : g_n \gamma_{n+1} \overset{\sim}{\Longrightarrow} \gamma_n$ such that $\Gamma_n \varphi_m = \Upsilon^n_m$, for all $m \geq n+1$. To define the Υ^n_m's, observe that the following diagram of pseudo cells commutes, for every $m \geq n+1$:

$$\begin{tikzcd}
g_n \gamma_{n+1} \varphi_{m+1} f_m \arrow[r, Rightarrow, "g_n \varpi^{n+1}_{m+1} f_m"] \arrow[d, Rightarrow, "g_n \gamma_{n+1} \Phi_m"'] & g_n \overline{g}_{m+1,n+1} f_m \arrow[r, Rightarrow, "(\varpi^g_{m+1,n})^{-1} f_m"] \arrow[d, Rightarrow, "g_n (\varpi^f_{m,n+1})^{-1}"'] \arrow[dr, Rightarrow, "(\varpi_{m,n})^{-1}"] & \overline{g}_{m+1,n} f_m \arrow[r, Rightarrow, "(\varpi^n_{m+1})^{-1} f_m"] \arrow[d, Rightarrow, "(\varpi^f_{m,n})^{-1}"] & \gamma_n \varphi_{m+1} f_m \arrow[d, Rightarrow, "\gamma_n \Phi_m"] \\
g_n \gamma_{n+1} \varphi_m \arrow[r, Rightarrow, "g_n \varpi^{n+1}_m"'] & g_n \overline{g}_{m,n+1} \arrow[r, Rightarrow, "(\varpi^g_{m,n})^{-1}"'] & \overline{g}_{m,n} \arrow[r, Rightarrow, "(\varpi^n_m)^{-1}"'] & \gamma_n \varphi_m
\end{tikzcd}$$

Commutativity of the leftmost and rightmost squares is provided by equation (6.9), while the central one follows from equation (6.7). Define, Υ^n_m as the bottom 2-cell in the above diagram, namely

$$\Upsilon^n_m \;\overset{\text{def}}{=}\; (\varpi^n_m)^{-1} \cdot (\varpi^g_{m,n})^{-1} \cdot (g_n \varpi^{n+1}_m) \;.$$

Satisfaction of condition (6.11) is then immediately read off from the diagram above. We have to show now commutativity of diagram (6.1), i.e., that for every n,

$$\iota_n = (\Gamma_n \Phi_n) \cdot (g_n \iota_{n+1} f_n) \cdot \eta_n \;.$$

Aiming at this, observe first of all the following:

$$
\begin{aligned}
\Gamma_n\Phi_n &= (\gamma_n\Phi_n)\cdot(\Gamma_n\varphi_{n+1}f_n)\\
&\qquad\text{(by the interchange law)}\\
&= (\gamma_n\Phi_n)\cdot(\Upsilon^n_{n+1}f_n)\\
&\qquad\text{(by the property defining } \Gamma_n\text{)}\\
&= (\gamma_n\Phi_n)\cdot((\varpi^n_{n+1})^{-1}f_n)\cdot((\varpi^g_{n+1,n})^{-1}f_n)\cdot(g_n\varpi^{n+1}_{n+1}f_n)\\
&\qquad\text{(by definition of } \Upsilon^n_{n+1}\text{)}\\
&= (\gamma_n\Phi_n)\cdot(\gamma_n(\Phi_n)^{-1})\cdot(\varpi^n_n)^{-1}\cdot(\varpi^f_{n,n})^{-1}\cdot((\varpi^g_{n+1,n})^{-1}f_n)\cdot(g_n\varpi^{n+1}_{n+1}f_n)\\
&\qquad\text{(by equation (6.9))}\\
&= (\varpi^n_n)^{-1}\cdot(\varpi_{n,n})^{-1}\cdot(g_n\varpi^{n+1}_{n+1}f_n) \qquad\qquad (6.12)\\
&\qquad\text{(by equation (6.6))}
\end{aligned}
$$

Hence

$$
\begin{aligned}
\iota_n &= (\varpi^n_n)^{-1}\cdot\iota^{n,n}_n && \text{(by definition)}\\
&= (\varpi^n_n)^{-1}\cdot(\varpi_{n,n})^{-1}\cdot(g_n\iota^{n+1,n+1}_{n+1}f_n)\cdot\eta_n && \text{(by equation (6.8))}\\
&= (\varpi^n_n)^{-1}\cdot(\varpi_{n,n})^{-1}\cdot(g_n\varpi^{n+1}_{n+1}f_n)\cdot(g_n\iota_{n+1}f_n)\cdot\eta_n && \text{(by definition of } \iota_{n+1}\text{)}\\
&= (\Gamma_n\Phi_n)\cdot(g_n\iota_{n+1}f_n)\cdot\eta_n && \text{(from the identity (6.12))}
\end{aligned}
$$

The fact that the cone

$$\langle g_{m,n}f_{n,m}\rangle_{m\geq n} \overset{\cdot}{\Longrightarrow} \gamma_n\varphi_n$$

is colimiting is now an immediate consequence of the fact that it is obtained from the following pasting of diagrams:

$$
\begin{tikzcd}
1_{A_n} \arrow[r, Rightarrow, "\eta_n"] \arrow[d, Rightarrow, "\iota^{n,n}_n"'] & g_nf_n \arrow[r, Rightarrow, "g_n\eta_{n+1}f_n"] \arrow[d, Rightarrow, "g_n\iota^{n+1,n+1}_{n+1}f_n"] & g_{n+2,n}f_{n,n+2} \arrow[r, Rightarrow] \arrow[d, Rightarrow, "g_{n+2,n}\iota^{n+2.n+2}_{n+2}f_{n,n+2}"] & \cdots \\
\overline{g}_{n,n} \arrow[d, Rightarrow, "(\varpi^n_n)^{-1}"'] & g_n\overline{g}_{n+1,n+1}f_n \arrow[l, Rightarrow, "(\varpi_{n,n})^{-1}"'] \arrow[d, Rightarrow, "g_n(\varpi^{n+1}_{n+1})^{-1}f_n"] & g_{n+2,n}\overline{g}_{n+2,n+2}f_{n,n+2} \arrow[l, Rightarrow, "g_n(\varpi_{n+1,n+1})^{-1}f_n"'] \arrow[d, Rightarrow, "g_{n+2,n}(\varpi_{n+2,n+2})^{-1}f_{n,n+2}"] & \cdots \arrow[l, Rightarrow] \\
\gamma_n\varphi_n & g_n\gamma_{n+1}\varphi_{n+1}f_n \arrow[l, Rightarrow, "\Gamma_n\Phi_n"] & g_{n+2,n}\gamma_{n+2}\varphi_{n+2}f_{n,n+2} \arrow[l, Rightarrow, "g_n\Gamma_{n+1}\Phi_{n+1}f_n"] & \cdots \arrow[l, Rightarrow]
\end{tikzcd}
$$

where the upper one, because of equation (6.8), is the one that defines $\overline{g}_{n,n}$ as a colimit, while the lower one, which commutes by the identity (6.12), consists of pseudo cells.

We now look for the definition of suitable $\jmath_n$'s. Using the family of Γ_n's, we can describe the *canonical* chain in the hom-category $\mathcal{K}(A, A)$ that gives rise to the 'second' canonical cone; namely,

$$\langle\ \varphi_n\gamma_n \xRightarrow{(\Phi_n)^{-1}(\Gamma_n)^{-1}} \varphi_{n+1}f_ng_n\gamma_{n+1} \xRightarrow{\varphi_{n+1}\varepsilon_n\gamma_{n+1}} \varphi_{n+1}\gamma_{n+1}\ \rangle_n\ .$$

Let $\langle \alpha_n : \varphi_n\gamma_n \Rightarrow a\rangle$ be a colimiting cone for this chain. We aim to prove first of all that $a \cong 1_A$. This will induce a colimiting cone $\langle \jmath_n : \varphi_n\gamma_n \Rightarrow 1_A\rangle$ which, by construction, will make diagram (6.2) commute. Moreover, we will finally prove that we obtain adjunctions

$$\iota_n, \jmath_n : \varphi_n \dashv \gamma_n \ .$$

The hint for proving $a \cong 1_A$ comes from the following calculation, for k any natural number:

$$\begin{aligned}
(\mathrm{colim}\ _{n\in\omega}\varphi_n\gamma_n)\varphi_k &\cong \mathrm{colim}\ _{n\in\omega}\varphi_n\gamma_n\varphi_k \\
&\cong \mathrm{colim}\ _{n\geq k}\varphi_n\gamma_n\varphi_k \\
&\cong \mathrm{colim}\ _{n\geq k}\varphi_n\gamma_n\varphi_n f_{k,n} \\
&\cong \mathrm{colim}\ _{n\geq k}\varphi_n(\mathrm{colim}\ _{l\geq n} g_{l,n}f_{n,l})f_{k,n} \\
&\cong \mathrm{colim}\ _{n\geq k}\mathrm{colim}\ _{l\geq n}\varphi_n g_{l,n}f_{n,l}f_{k,n} && (6.13)\\
&\cong \mathrm{colim}\ _{n\geq k}\varphi_n g_{n,n}f_{n,n}f_{k,n} && (6.14)\\
&\cong \mathrm{colim}\ _{n\geq k}\varphi_n f_{k,n} \\
&\cong \mathrm{colim}\ _{n\geq k}\varphi_k \\
&\cong \varphi_k \ .
\end{aligned}$$

In fact we need to have a closer look at the matrix (6.13) and in particular at the diagonal (6.14). In Figure 6.1 we have sketched the (infinite) matrix completed with the colimit points. Using the interchange law and the fact that for every n, $(\varepsilon_n f_n)\cdot(f_n\eta_n) = 1_{f_n}$ it is not difficult to verify that

$$\begin{array}{rcll}
x_{l+1,n}y_{l,n} & = & y_{l+1,n}x_{l,n} & \text{for } l \geq n+1 \\
x_{n+1,n}y_{n,n} & = & \Phi_n^{-1}f_{k,n} & \\
(\varphi_{n+1}\gamma_{n+1}\Phi_{k,n+1})\cdot x_{\infty,n} & = & x'_{\infty,n}\cdot(\varphi_n\gamma_n\Phi_{k,n}) & \text{for } n \geq k+1 \\
(\varphi_{k+1}\gamma_{k+1}\Phi_k)\cdot x_{\infty,k} & = & x'_{\infty,k} & .
\end{array}$$

Therefore we have that for any $n \in \omega$ the chain of isomorphisms (the diagonal of the matrix):

$$\begin{tikzcd}
\varphi_n \arrow[r, Rightarrow, "(\Phi_n)^{-1}"] & \varphi_{n+1}f_n \arrow[r, Rightarrow, "(\Phi_{n+1})^{-1}f_n"] & \varphi_{n+2}f_{n,n+2} \arrow[r, Rightarrow] & \cdots
\end{tikzcd}$$

has colimiting cones given by the (trivial cone of) inverses

$$\begin{tikzcd}
\varphi_n \arrow[r, Rightarrow, "(\Phi_n)^{-1}"] \arrow[dr, Rightarrow, "1_{\varphi_n}"'] & \varphi_{n+1}f_n \arrow[r, Rightarrow, "(\Phi_{n+1})^{-1}f_n"] \arrow[d, Rightarrow, "\Phi_n"] & \varphi_{n+2}f_{n,n+2} \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\Phi_n\cdot(\Phi_{n+1}f_n)"] & \cdots \\
& \varphi_n & \cdots &
\end{tikzcd}$$

$$
\begin{tikzcd}
 & \varphi_{k+1}\gamma_{k+1}\varphi_k \arrow[r, Rightarrow, "x'_{\infty,k+1}"] & \varphi_{k+2}\gamma_{k+2}\varphi_k \arrow[r, dotted, no head] & \cdots \\
\varphi_k\gamma_k\varphi_k \arrow[r, Rightarrow, "x_{\infty,k}"] \arrow[d, dotted, no head] & \varphi_{k+1}\gamma_{k+1}\varphi_{k+1}f_k \arrow[u, Rightarrow, "\varphi_{k+1}\gamma_{k+1}\Phi_k"] \arrow[r, Rightarrow, "x_{\infty,k+1}"] \arrow[d, dotted, no head] & \varphi_{k+2}\gamma_{k+2}\varphi_{k+2}f_{k,k+2} \arrow[u, Rightarrow, "\varphi_{k+2}\gamma_{k+2}\Phi_{k,k+2}"] \arrow[r, dotted, no head] \arrow[d, dotted, no head] & \cdots \\
{} \arrow[r, dotted, no head] \arrow[d, dotted, no head] & {} \arrow[r, dotted, no head] \arrow[d, dotted, no head] & {} \arrow[r, dotted, no head] \arrow[d, dotted, no head] & \cdots \\
{} \arrow[r, dotted, no head] & {} \arrow[r, dotted, no head] & {} \arrow[r, dotted, no head] & {} & \cdots \\
\varphi_k g_{k+2,k} f_{k,k+2} \arrow[u, Rightarrow, "y_{k+2,k}"] \arrow[r, Rightarrow, "x_{k+2,k}"] & \varphi_{k+1} g_{k+1} f_{k,k+2} \arrow[u, Rightarrow, "y_{k+2,k+1}"] \arrow[r, Rightarrow, "x_{k+2,k+1}"] & \varphi_{k+2} f_{k,k+2} \arrow[u, Rightarrow, "y_{k+2,k+2}"] \arrow[ur, dotted, no head] \\
\varphi_k g_k f_k \arrow[u, Rightarrow, "y_{k+1,k}"] \arrow[r, Rightarrow, "x_{k+1,k}"] & \varphi_{k+1} f_k \arrow[u, Rightarrow, "y_{k+1,k+1}"] \arrow[ur, Rightarrow, "\Phi_{k+1}^{-1} f_k"'] \\
\varphi_k \arrow[u, Rightarrow, "y_{k,k}"] \arrow[ur, Rightarrow, "\Phi_k^{-1}"']
\end{tikzcd}
$$

$$
\begin{array}{rcll}
x_{l,n} & = & ((\varphi_{n+1}\varepsilon_n)\cdot(\Phi_n g_n))g_{l,n+1}f_{k,l} & \text{for } l \geq n+1 \\
y_{l,n} & = & \varphi_n g_{l+1,n}\eta_l f_{k,l} & \text{for } l \geq n \\
x_{\infty,n} & = & ((\varphi_{n+1}\varepsilon_n\gamma_{n+1})\cdot(\Phi_n^{-1}\Gamma_n^{-1}))(\Phi_n^{-1}f_{k,n}) & \\
x'_{\infty,n} & = & ((\varphi_{n+1}\varepsilon_n\gamma_{n+1})\cdot(\Phi_n^{-1}\Gamma_n^{-1}))\varphi_k &
\end{array}
$$

Figure 6.1: A matrix of 2-cells

and by

$$
\begin{tikzcd}
\varphi_n \arrow[r, Rightarrow, "(\Phi_n)^{-1}"] \arrow[d, Rightarrow, "\varphi_n\iota_n"'] & \varphi_{n+1}f_n \arrow[r, Rightarrow, "(\Phi_{n+1})^{-1}f_n"] \arrow[d, Rightarrow, "\varphi_{n+1}((\gamma_{n+1}\Phi_n)\cdot(\iota_{n+1}f_n))"] & \varphi_{n+2}f_{n,n+2} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \cdots \\
\varphi_n\gamma_n\varphi_n \arrow[r, Rightarrow, "x'_{\infty,n}"] \arrow[dr, Rightarrow, "\alpha_n\varphi_n"'] & \varphi_{n+1}\gamma_{n+1}\varphi_n \arrow[r, Rightarrow, "x'_{\infty,n+1}"] \arrow[d, Rightarrow, "\alpha_{n+1}\varphi_n"] & \varphi_{n+2}\gamma_{n+2}\varphi_n \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\alpha_{n+2}\varphi_n"] & \cdots \\
 & a\varphi_n & \ldots &
\end{tikzcd}
\tag{6.15}
$$

obtained by going along the edges of the matrix of Figure 6.1. Hence there exists a universal pseudo-cell $\mu_n : a\varphi_n \overset{\sim}{\Longrightarrow} \varphi_n$ such that

$$
\mu_n \cdot (\alpha_n\varphi_n) \cdot (\varphi_n\iota_n) \quad = \quad 1_{\varphi_n} \ . \tag{6.16}
$$

Namely μ_n is the inverse of $(\alpha_n\varphi_n)\cdot(\varphi_n\iota_n)$. To conclude the existence of an isomorphism $a \cong 1_A$, we use again that the $\Phi's$ form a pseudo-colimit for the chain $\langle f_n : A_n \to A_{n+1}\rangle$

and show that the family of μ_n's satisfy

$$\mu_n \cdot (a\Phi_n) \quad = \quad \Phi_n \cdot (\mu_{n+1} f_n) \ . \tag{6.17}$$

To deduce the above equality notice that it follows from diagram (6.15) that $a\varphi_{n+1} f_n$ is the colimit of the chain of pseudo-cells obtained by precomposing the chain

$$\langle (\Phi_m)^{-1} f_{n+1,m} : \varphi_m f_{n+1,m} \Rightarrow \varphi_{m+1} f_{n+1,m+1} \rangle_{m \geq n+1}$$

with f_n. By universality of this colimit and using that it consists of isomorphisms, to prove property (6.17) it is enough to establish that

$$\begin{aligned} &\mu_n \cdot (a\Phi_n) \cdot (\alpha_{n+1}\varphi_{n+1} f_n) \cdot (\varphi_{n+1}\iota_{n+1} f_n) \\ &\quad = \quad \Phi_n \cdot (\mu_{n+1} f_n) \cdot (\alpha_{n+1}\varphi_{n+1} f_n) \cdot (\varphi_{n+1}\iota_{n+1} f_n) \ . \end{aligned}$$

By the universal property defining μ_{n+1} (see equation (6.16)) the right hand side equals Φ_n. Let us then calculate the left hand side:

$$\begin{aligned} &\mu_n \cdot (a\Phi_n) \cdot (\alpha_{n+1}\varphi_{n+1} f_n) \cdot (\varphi_{n+1}\iota_{n+1} f_n) \\ &\quad = \quad \mu_n \cdot (\alpha_{n+1}\varphi_n) \cdot (\varphi_{n+1}\gamma_{n+1}\Phi_n) \cdot (\varphi_{n+1}\iota_{n+1} f_n) \\ &\qquad\qquad \text{(by the interchange law)} \\ &\quad = \quad \mu_n \cdot (\alpha_n \varphi_n) \cdot (\varphi_n \iota_n) \cdot \Phi_n \\ &\qquad\qquad \text{(see diagram (6.15))} \\ &\quad = \quad \Phi_n \\ &\qquad\qquad \text{(by equation (6.16))} \end{aligned} \tag{6.18}$$

Thus equation (6.17) holds and so there exists a unique $\jmath : a \overset{\sim}{\Longrightarrow} 1_A$ such that

$$\jmath\varphi_n \quad = \quad \mu_n \ . \tag{6.19}$$

Finally, define

$$\jmath_n \quad \overset{\text{def}}{=} \quad \jmath \cdot \alpha_n : \varphi_n\gamma_n \Rightarrow 1_A \ . \tag{6.20}$$

The following cone is colimiting, since $\jmath$ is an isomorphism:

$$\begin{tikzcd}
\varphi_0\gamma_0 \arrow[r, Rightarrow] \arrow[dr, Rightarrow, "\alpha_0"] \arrow[ddr, Rightarrow, "\jmath_0"'] & \varphi_1\gamma_1 \arrow[r, Rightarrow] \arrow[d, Rightarrow, "\alpha_1"] \arrow[dd, Rightarrow, bend right, "\jmath_1"'] & \varphi_2\gamma_2 \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\alpha_2"'] \arrow[ddl, Rightarrow, "\jmath_2"] & \cdots \\
& a \arrow[d, Rightarrow, "\jmath"] & \cdots & \\
& 1_A & &
\end{tikzcd}$$

Moreover, expanding the definition of $\jmath_n$, and using the identities (6.19) and (6.16), we obtain the first triangular identity; namely, $(\jmath_n\varphi_n) \cdot (\varphi_n\iota_n) = 1_{\varphi_n}$.

It follows from the first triangular identity that the composite $(\gamma_n\jmath_n)\cdot(\iota_n\gamma_n)$ is an idempotent. Thus, to deduce the second triangular identity (namely, $(\gamma_n\jmath_n)\cdot(\iota_n\gamma_n) = 1_{\gamma_n}$) we need only show that the composite $(\gamma_n\jmath_n)\cdot(\iota_n\gamma_n)$ is an isomorphism. Again we consider a chain of pseudo-cells. Take, the following colimiting cone:

$$\begin{tikzcd}
\gamma_n \arrow[r, Rightarrow, "(\Gamma_n)^{-1}"] \arrow[dr, Rightarrow, "1_{\gamma_n}"'] & \gamma_{n+1}f_n \arrow[r, Rightarrow, "(\Gamma_{n+1})^{-1}f_n"] \arrow[d, Rightarrow, "\Gamma_n"] & \gamma_{n+2}f_{n,n+2} \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\Gamma_n\cdot(\Gamma_{n+1}f_n)"] & \cdots \\
 & \gamma_n & \cdots &
\end{tikzcd}$$

Using again a matrix like in Figure 6.1 one can verify that the following cone

$$\begin{tikzcd}
\gamma_n \arrow[r, Rightarrow, "(\Gamma_n)^{-1}"] \arrow[d, Rightarrow, "\iota_n\gamma_n"'] & g_n\gamma_{n+1} \arrow[r, Rightarrow, "g_n(\Gamma_{n+1})^{-1}"] \arrow[d, Rightarrow, "((\Gamma_n\varphi_{n+1})\cdot(g_n\iota_{n+1}))\gamma_{n+1}"] & g_{n+2,n}\gamma_{n+2} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \cdots \\
\gamma_n\varphi_n\gamma_n \arrow[r, Rightarrow] \arrow[dr, Rightarrow, "\gamma_n\jmath_n"'] & \gamma_n\varphi_{n+1}\gamma_{n+1} \arrow[r, Rightarrow] \arrow[d, Rightarrow, "\gamma_n\jmath_{n+1}"] & \gamma_n\varphi_{n+2}\gamma_{n+2} \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\gamma_n\jmath_{n+2}"] & \cdots \\
 & \gamma_n & \cdots &
\end{tikzcd} \tag{6.21}$$

is colimiting for the same diagram. Thus, there exists an automorphism on γ_n of which $(\gamma_n\jmath_n)\cdot(\iota_n\gamma_n)$ is the inverse.

[3 implies 1:] Recall that in order to prove that $\langle \Phi_n : \varphi_{n+1}f_n \overset{\sim}{\Longrightarrow} \varphi_n : A_n \to A\rangle$ is a pseudo colimit for $\langle f_n : A_n \to A_{n+1}\rangle$ we need to show that the properties 1 and 2 of Definition 6.1.6 hold.

<u>*Proof of property 1:*</u> In order to find a suitable $\upsilon : A \to X$ consider the chain

$$\langle\ \psi_n\gamma_n \xRightarrow{(\Psi_n)^{-1}(\Gamma_n)^{-1}} \psi_{n+1}f_ng_n\gamma_{n+1} \xRightarrow{\psi_{n+1}\varepsilon_n\gamma_{n+1}} \psi_{n+1}\gamma_{n+1}\ \rangle_n$$

and let $\langle \Upsilon_n : \psi_n\gamma_n \Rightarrow \upsilon\rangle_n$ be a colimiting cone for it. We need to describe now a family of $\mu_k : \upsilon\varphi_k \Rightarrow \psi_k$ such that for every k the following diagram commutes:

$$\begin{tikzcd}
\upsilon\varphi_{k+1}f_k \arrow[r, Rightarrow, "\mu_{k+1}f_k"] \arrow[d, Rightarrow, "\upsilon\Phi_k"'] & \psi_{k+1}f_k \arrow[d, Rightarrow, "\Psi_k"] \\
\upsilon\varphi_k \arrow[r, Rightarrow, "\mu_k"'] & \psi_k
\end{tikzcd} .$$

Fix a natural number k, an observe, once again using a matrix like Figure 6.1 that the following diagram

$$\begin{tikzcd}
\psi_k \arrow[r, Rightarrow, "(\Psi_k)^{-1}"] \arrow[d, Rightarrow, "\psi_k\iota_k"'] & \psi_{k+1}f_k \arrow[r, Rightarrow, "(\Psi_{k+1})^{-1}f_k"] \arrow[d, Rightarrow, "\psi_{k+1}((\gamma_{k+1}\Phi_k)\cdot(\iota_{k+1}f_k))"] & \psi_{k+2}f_{k,k+2} \arrow[r, Rightarrow] \arrow[d, Rightarrow] & \cdots \\
\psi_k\gamma_k\varphi_k \arrow[r, Rightarrow] \arrow[dr, Rightarrow, "\Upsilon_k\varphi_k"'] & \psi_{k+1}\gamma_{k+1}\varphi_k \arrow[r, Rightarrow] \arrow[d, Rightarrow, "\Upsilon_{k+1}\varphi_k"] & \psi_{k+2}\gamma_{k+2}\varphi_k \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\Upsilon_{k+2}\varphi_k"] & \cdots \\
& u\varphi_k & \cdots &
\end{tikzcd} \tag{6.22}$$

yields a colimiting cone for the chain $\langle(\Psi_n)^{-1}f_{k,n} : \psi_n f_{k,n} \Rightarrow \psi_{n+1}f_{k,n+1}\rangle_{n\geq k}$. But since the above chain is made of pseudo cells, it admits also the (trivial) colimiting cone of inverses:

$$\begin{tikzcd}
\psi_k \arrow[r, Rightarrow, "(\Psi_k)^{-1}"] \arrow[dr, Rightarrow, "1_{\psi_k}"'] & \psi_{k+1}f_k \arrow[r, Rightarrow, "(\Psi_{k+1})^{-1}f_k"] \arrow[d, Rightarrow, "\Psi_k"] & \psi_{k+2}f_{k,k+2} \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\Psi_k\cdot(\Psi_{k+1}f_k)"] & \cdots \\
& \psi_k & \cdots &
\end{tikzcd}$$

Define $\mu_k : \upsilon\varphi_k \Rightarrow \psi_k$ to be the universal pseudo cell such that

$$\mu_k \cdot (\Upsilon_k\varphi_k) \cdot (\psi_k\iota_k) \quad = \quad 1_{\psi_k} \ .$$

We are then left with showing that for all $k \in \omega$,

$$\mu_k \cdot (\upsilon\Phi_k) = \Psi_k \cdot (\mu_{k+1}f_k) \ .$$

By the universal property of colimits, we know that two parallel arrows in the hom-category $\mathcal{K}(A, X)$ with domain $\upsilon\varphi_k$, are equal if they are equalised by all the edges of the cone (6.22). Moreover since the diagram is a diagram of isomorphisms it is enough to check the property for only one edge. Observe then that the following equality holds, for all k:

$$(\Psi_k)^{-1} \cdot \mu_k \cdot (\Upsilon_k\varphi_k) \cdot (\psi_k\iota_k) \quad = \quad (\mu_{k+1}f_k) \cdot (\upsilon(\Phi_k)^{-1}) \cdot (\Upsilon_k\varphi_k) \cdot (\psi_k\iota_k) \ .$$

In fact from the universal property defining μ_k, the left hand side of the equality above reduces immediately to $(\Psi_k)^{-1}$. From the diagram (6.22), instead, we see that

$$(\Upsilon_k\varphi_k) \cdot (\psi_k\iota_k) \quad = \quad (\Upsilon_{k+1}\varphi_k) \cdot (\psi_{k+1}\gamma_{k+1}\Phi_k) \cdot (\psi_{k+1}\iota_{k+1}f_k(\Psi_k)^{-1}) \ ,$$

hence,

$$\begin{aligned}
(\Psi_k)^{-1} \quad &= \quad (\mu_{k+1}f_k) \cdot (\Upsilon_{k+1}\varphi_{k+1}f_k) \cdot (\psi_{k+1}\iota_{k+1}f_k) \cdot (\Psi_k)^{-1} \\
&= \quad (\mu_{k+1}f_k) \cdot (\upsilon(\Phi_k)^{-1}) \cdot (\upsilon\Phi_k) \cdot (\Upsilon_{k+1}\varphi_{k+1}f_k) \cdot (\psi_{k+1}\iota_{k+1}f_k) \cdot (\Psi_k)^{-1} \\
&= \quad (\mu_{k+1}f_k) \cdot (\upsilon(\Phi_k)^{-1}) \cdot (\Upsilon_{k+1}\varphi_k) \cdot (\psi_{k+1}\gamma_{k+1}\Phi_k) \cdot (\psi_{k+1}\iota_{k+1}f_k(\Psi_k)^{-1}) \\
&= \quad (\mu_{k+1}f_k) \cdot (\upsilon(\Phi_k)^{-1}) \cdot (\Upsilon_k\varphi_k) \cdot (\psi_k\iota_k) \ .
\end{aligned}$$

Therefore $\Psi_k^{-1} \cdot \mu_k = (\mu_{k+1} f_k) \cdot (\upsilon \Phi_k^{-1})$. Hence by composing with Ψ_k on the left and with $(\upsilon\Phi_k)$ on the right of both the sides of the equations we get that $\mu_k \cdot (\upsilon\Phi_k) = \Psi_k \cdot (\mu_{k+1} f_k)$.

<u>*Proof of property 2:*</u> Suppose now that we are given two arrows $u, v : A \to X$ and a family of 2-cells $\Upsilon_n n : u\varphi_n \Rightarrow v\varphi_n$, as in condition 2 of Definition 6.1.6. We look for a 2-cell $\Upsilon : u \Rightarrow v$, such that $\Upsilon_n = \Upsilon\varphi_n$, for every n.

Recall that $1_{A_n} = \text{colim } (\varphi_n\gamma_n)$, hence $u = \text{colim } (u\varphi_n\gamma_n)$ and $v = \text{colim } (v\varphi_n\gamma_n)$. Because of the properties of the Υ_n's, we can describe two colimiting cones and the following map between them:

$$\begin{tikzcd}
 & \vdots & \vdots & \\
 & u\varphi_n\gamma_n \arrow[r, Rightarrow, "\Upsilon_n\gamma_n"] \arrow[d, Rightarrow] \arrow[dddl, Rightarrow, "u\jmath_n"'] & v\varphi_n\gamma_n \arrow[d, Rightarrow] \arrow[dddr, Rightarrow, "v\jmath_n"] & \\
 & u\varphi_{n+1}\gamma_{n+1} \arrow[r, Rightarrow, "\Upsilon_{n+1}\gamma_{n+1}"] \arrow[ddl, Rightarrow, "u\jmath_{n+1}"] & v\varphi_{n+1}\gamma_{n+1} \arrow[ddr, Rightarrow, "v\jmath_{n+1}"'] & \\
 & \vdots & \vdots & \\
u \arrow[rrr, Rightarrow, "\Upsilon"'] & & & v
\end{tikzcd}$$

where $\Upsilon : u \Rightarrow v$ is the unique 2-cell such that

$$\Upsilon \cdot (u\jmath_n) \quad = \quad (v\jmath_n) \cdot (\Upsilon_n\gamma_n) \ .$$

We claim that Υ satisfies the required property. In fact by whiskering, for any n, the above equalities with φ_n, we obtain that

$$(\Upsilon\varphi_n) \cdot (u\jmath_n\varphi_n) \quad = \quad (v\jmath_n\varphi_n) \cdot (\Upsilon_n\gamma_n\varphi_n) \ . \tag{6.23}$$

Now, since $\jmath_n$ is counit for the adjunction $\varphi_n \dashv \gamma_n$ with unit ι_n, by precomposing both sides of the equation with $u\varphi_n\iota_n$ we obtain:

$$\begin{array}{rcll}
\Upsilon\varphi_n & = & (\Upsilon\varphi_n) \cdot (u\jmath_n\varphi_n) \cdot (u\varphi_n\iota_n) & \text{(by a triangular identity)} \\
 & = & (v\jmath_n\varphi_n) \cdot (\Upsilon_n\gamma_n\varphi_n) \cdot (u\varphi_n\iota_n) & \text{(by the equation (6.23) above)} \\
 & = & (v\jmath_n\varphi_n) \cdot (v\varphi_n\iota_n) \cdot \Upsilon_n & \text{(by the interchange law)} \\
 & = & \Upsilon_n & \text{(by a triangular identity)}
\end{array}$$

Moreover Υ is uniquely determined by the property $\Upsilon\varphi_n = \Upsilon_n$. Indeed, for every n,

$$\Upsilon\varphi_n = \Upsilon_n \quad \text{implies} \quad \Upsilon\varphi_n\gamma_n = \Upsilon_n\gamma_n \tag{6.24}$$

and so

$$\begin{array}{rcll}
\Upsilon \cdot (u\jmath_n) & = & (v\jmath_n) \cdot (\Upsilon\varphi_n\gamma_n) & \text{(by the interchange law)} \\
 & = & (v\jmath_n) \cdot (\Upsilon_n\gamma_n) & \text{(by the property (6.24) above).}
\end{array}$$

But, by universality of colimits there exists a unique such Υ. $\square$

In the case of a chain of coreflections, all the 2-cells in the canonical cones $\langle g_{l,n} f_{n,l} \rangle_l \stackrel{\cdot}{\Longrightarrow} \gamma_n \varphi_n$ are pseudo-cells, hence the condition about these cones being colimiting becomes vacuous and so we have the following simplified version of the theorem:

Corollary 6.1.10 (Local characterisation for coreflections) *In an ω**Cat**-category $\mathcal{K}$, for an ω-chain of coreflections $\langle f_n \dashv g_n : A_{n+1} \to A_n \rangle$ and a pseudo cone*

$$\langle \Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n : A_n \to A \rangle$$

for the ω-chain $\langle f_n : A_n \to A_{n+1} \rangle$, the following are equivalent:

1. *$\langle \Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n : A_n \to A \rangle$ is a pseudo colimit for $\langle f_n : A_n \to A_{n+1} \rangle$ in $\mathcal{K}$.*
2. *$\langle \Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n : A_n \to A \rangle$ is a pseudo colimit for $\langle f_n : A_n \to A_{n+1} \rangle$ in $\mathcal{K}_{\cong}$.*
3. *There is a pseudo cone of coreflections*

$$(\Phi_n, \Gamma_n) : (\varphi_{n+1} \dashv \gamma_{n+1})(f_n \dashv g_n) \stackrel{\sim}{\Longrightarrow} (\varphi_n \dashv \gamma_n)$$

such that the canonical cone $\langle \varphi_n \gamma_n \rangle \stackrel{\cdot}{\Longrightarrow} 1_A$ is colimiting.

Proof: The only thing to check is that in this case the ι_n's of the previous proof are pseudo-cells. Recall that by definition (see equation (6.10) in the previous proof),

$$\iota_n \stackrel{\text{def}}{=} (\varpi_n^n)^{-1} \cdot \iota_n^{n,n} : 1_{A_n} \Rightarrow \gamma_n \varphi_n \ ,$$

where ϖ_n^n is a pseudo cell, while $\iota_n^{n,n}$ is an edge for the colimiting cone

$$\mathcal{G}_{n,n} : \quad \begin{tikzcd} g_{n,n} \arrow[r, Rightarrow, "g_{n,n}\eta_n"] \arrow[dr, Rightarrow, "\iota_n^{n,n}"'] & g_{n+1,n} f_n \arrow[r, Rightarrow, "g_{n+1,n}\eta_{n+1} f_n"] \arrow[d, Rightarrow, "\iota_{n+1}^{n,n}"] & g_{n+2,n} f_{n,n+2} \arrow[r, Rightarrow] \arrow[dl, Rightarrow, "\iota_{n+2}^{n,n}"] & \cdots \\ & \overline{g}_{m,n} & \cdots & \end{tikzcd} \quad .$$

But in the case of a chain of coreflections the diagram consists of pseudo cells only, hence the edges must be pseudo cells, too. $\Box$

For those familiar with Smyth and Plotkin's result, it should be clear that the condition about the canonical cone $\langle \varphi_n \gamma_n \rangle \stackrel{\cdot}{\Longrightarrow} id_A$ being colimiting generalises the analogous condition of [125, Theorem 2] asserting $\bigvee \varphi_n \gamma_n = 1_A$. Dual results to Theorem 6.1.9 and Corollary 6.1.10 with respect to the limit of the ω^{op}-chain of g_n's obviously hold too and provide the following corollary about limit/colimit coincidence.

Corollary 6.1.11 (Limit/Colimit coincidence) *In an ω**Cat**-category, the following are equivalent for an ω-chain of coreflections (adjunctions)*

$$\langle \eta_n, \varepsilon_n : f_n \dashv g_n : A_{n+1} \to A_n \rangle$$

and a pseudo cone of coreflections (adjunctions)

$$(\Phi_n, \Gamma_n) : (\varphi_{n+1} \dashv \gamma_{n+1})(f_n \dashv g_n) \stackrel{\sim}{\Longrightarrow} (\varphi_n \dashv \gamma_n) \ :$$

1. $\langle \Phi_n : \varphi_{n+1} f_n \overset{\sim}{\Longrightarrow} \varphi_n : A_n \to A\rangle$ *is a bicategorical colimit for* $\langle f_n : A_n \to A_{n+1}\rangle$.
2. $\langle \Gamma_n : g_n \gamma_{n+1} \overset{\sim}{\Longrightarrow} \gamma_n : A \to A_n\rangle$ *is a bicategorical limit for* $\langle g_n : A_{n+1} \to A_n\rangle$.

Corollary 6.1.11 in itself seems to be part of the categorical folklore [128, 143]. It also seems that the main reason why it holds (as an attempt to a direct proof would suggest) is due to the fact that the diagrams we are considering (ω-chains) have colimits in the enriching categories (objects of $\omega\mathbf{Cat}$) and the functors induced by whiskering preserve them. This fact was pointed out to us by Pino Rosolini.

Remark: An analysis of the statements and proofs of Theorem 6.1.9 and Corollary 6.1.10 suggests possible generalisations from ω-chains to classes of filtered diagrams satisfying certain closure properties. We do not expand on this possibility here.

We have already seen (cf. Proposition 1.2.15) that adjoint pairs in **Cocont** correspond to functors in **Cat**, whence we can deduce a way of calculating pseudo-colimits of ω-chains in **Cocont**.

Proposition 6.1.12 *Let* $\langle \eta_n, \varepsilon_n : f_n \dashv g_n : \widehat{\mathbb{A}_{n+1}} \to \widehat{\mathbb{A}_n}\rangle$ *be an* ω*-chain in* $\mathbf{Cocont}^{\mathrm{adj}}$. *Then the chain* $\langle f_n : \widehat{\mathbb{A}_n} \to \widehat{\mathbb{A}_{n+1}}\rangle$ *has a pseudo-colimit in* **Cocont**.

Proof:[Sketch] It is a known fact that the full embeddings,

$$\mathbf{Cat}[\mathbb{C},\mathbb{D}] \hookrightarrow \mathbf{Cocont}[\mathbb{C},\mathbb{D}]\ ,$$

given by $F \mapsto \mathrm{Lan}_{\mathrm{y}_{\mathbb{C}}}(\mathrm{y}_{\mathbb{D}}F)$, for any two small categories $\mathbb{C}$ and $\mathbb{D}$, extend to a pseudo functor (locally a full embedding)

$$\mathbf{Cat} \hookrightarrow \mathbf{Cocont}$$

that preserves pseudo colimits [143].

In Chapter 1 we proved (Proposition 1.2.15) an equivalence of categories

$$\mathbf{Cat}[\mathbb{C}^c,\mathbb{D}^c] \simeq \mathbf{EGeom}[\mathbb{C},\mathbb{D}]\ ,$$

to which we can now add an equivalence

$$\mathbf{EGeom}[\mathbb{C},\mathbb{D}] \simeq \mathbf{Cocont}^{\mathrm{adj}}[\mathbb{C},\mathbb{D}]\ ,$$

since any colimit preserving functor between presheaf categories has a right adjoint (cf. Section 1.2).

So given any chain of adjoint pairs $\langle f_n \dashv g_n : \widehat{\mathbb{A}_{n+1}} \to \widehat{\mathbb{A}_n}\rangle$, consider the "associated" chain of functors

$$\langle h_n : \mathbb{A}^c_n \to \mathbb{A}^c_{n+1}\rangle\ .$$

Let $\langle \mathbb{A}, k_n : \mathbb{A}^c_n \to \mathbb{A}\rangle$ be a colimit in **Cat** of the chain of h_n's. Let $\varphi_n : \widehat{\mathbb{A}_n} \to \widehat{\mathbb{A}}$ and $\Phi_n : \varphi_{n+1} f_n \overset{\sim}{\Longrightarrow} \varphi_n$ be the colimit preserving functors and natural transformations induced by the equivalences $\widehat{\mathbb{A}_n} \overset{\sim}{\longrightarrow} \widehat{\mathbb{A}^c_n}$ (cf. Proposition 1.2.14) and by the pseudo-functor $\mathbf{Cat} \hookrightarrow \mathbf{Cocont}$. The families of φ_n's and Φ_n's forms a pseudo-colimit for the chain $\langle f_n : \widehat{\mathbb{A}_n} \to \widehat{\mathbb{A}_{n+1}}\rangle$. $\square$

In the reminder of this chapter a sub-2-category of **Cocont** will play a role with respect to the analysis of open map bisimulation that we have in mind.

Definition 6.1.13 ($\mathbf{Cocont}_M$) *Define* $\mathbf{Cocont}_M$ *to be the sub-2-category of* **Cocont** *with the same objects of* **Cocont**, *2-cells given by monomorphic natural transformations and arrows colimit preserving functors that also preserve monomorphic natural transformations.*

The 2-category $\mathbf{Cocont}_M$ is an $\omega\mathbf{Cat}_0$ category.

Proposition 6.1.14 *The same construction outlined in the proof of the proposition above would produce a pseudo colimit in* $\mathbf{Cocont}_M$ *from a chain of arrows in* $\mathbf{Cocont}_M^{\mathrm{adj}}$.

6.2 Coherence

So far we have proved results concerning ω**Cat**-categories. In order to make these results directly applicable to **Prof** we should extend them to hold for bicategories with the ω**Cat**-property [9]. A way of doing this is by means of *coherence results* [106, 107, 109, 127, 42]. Roughly speaking these are results that state when an *up-to-isomorphism situation* can be replaced with a *strict one* without losing any property of interest. The category theory literature abounds with examples of such results. The first notable one is Mac Lane's coherence result for monoidal categories (see [76]).

To us the following will be of primary importance:

Theorem 6.2.1 (see [42]) *Any bicategory is pseudo equivalent to a 2-category.*

Definition 6.2.2 (ωCat-Bicategories) *An* ω**Cat**-*bicategory is a bicategory with the* ω**Cat**-*property, i.e., a bicategory,* $\mathcal{B}$, *such that for any two objects* A, B, *the hom category* $\mathcal{B}(A, B)$ *has colimits of* ω-*chains and the composition functors preserve them.*

Similarly one defines $\omega\mathbf{Cat}_0$-*bicategories.*

Theorem 6.2.1 is the key result to prove the following:

Theorem 6.2.3 *Any* ω**Cat**-*bicategory* ($\omega\mathbf{Cat}_0$-*bicategory) is pseudo equivalent to an* ω**Cat**-*category* ($\omega\mathbf{Cat}_0$-*category).*

Proof: We do the case for ω**Cat**, the other one is analogous.

Let $\mathcal{B}$ be an ω**Cat**-bicategory and let $\mathcal{K}$ be a pseudo equivalent 2-category (that exists because of Theorem 6.2.1). We show that $\mathcal{K}$ is in fact an ω**Cat**-category. Let $\varphi : \mathcal{B} \to \mathcal{K}$ be a pseudo equivalence with $\psi : \mathcal{K} \to \mathcal{B}$ as a pseudo inverse. Since any pseudo equivalence is locally an equivalence of categories, then for any two objects $A, B \in |\mathcal{K}|$,

$$\mathcal{K}(A, B) \simeq \mathcal{B}(\psi A, \psi B)$$

that is $\mathcal{K}(A, B)$ is an object of ω**Cat** since $\mathcal{B}$ has the ω**Cat** property. We are only left with showing that the composition functors in $\mathcal{K}$ are ω**Cat**-functors. But this is again trivial since, for any three objects $A, B, C \in \mid \mathcal{K} \mid$, we have the following natural isomorphism

$$\begin{tikzcd}
\mathcal{K}(B,C) \times \mathcal{K}(A,B) \arrow[rr, "c^{\mathcal{K}}_{A,B,C}"] \arrow[ddd, "\psi_{B,C} \times \psi_{A,C}"'] & & \mathcal{K}(A,C) \\
 & & \\
 & \cong & \mathcal{K}(\varphi\psi A, \varphi\psi C) \arrow[uu, "\simeq"'] \\
\mathcal{B}(\psi C, \psi B) \times \mathcal{B}(\psi A, \psi B) \arrow[rr, "c^{\mathcal{B}}_{\psi A,\psi B,\psi C}"'] & & \mathcal{B}(\psi A, \psi C) \arrow[u, "\varphi_{\psi A,\psi C}"']
\end{tikzcd} ,$$

where $c^{\mathcal{K}}$ and $c^{\mathcal{B}}$ are the composition functors in $\mathcal{K}$ and $\mathcal{B}$ respectively. Hence, for every A, B, C, the composition functor $c^{\mathcal{K}}_{A,B,C}$ is naturally isomorphic to a functor that preserves colimits of ω-chains and so it preserves such colimits as well. $\square$

Observe that, in fact we have proved that a pseudo equivalence preserves the ω**Cat**-property (and the $\omega\mathbf{Cat}_0$-property as well), hence we have the following:

Example 6.2.4 **Prof**, *being pseudo equivalent to* **Cocont**, *is an* $\omega\mathbf{Cat}_0$*-bicategory.*

Moreover since pseudo (co)limits are preserved by pseudo equivalences, the above result is telling us that the statements of Theorem 6.1.9, Corollary 6.1.10 and Corollary 6.1.11 can be generalised to ω**Cat**-bicategories. Of course this would imply taking some extra care especially in treating the notion of adjoint arrows in a bicategory [47].

In the next section we shall develop, after Freyd, the notion of *pseudo* algebraically complete and compact 2-categories. The following results here will help simplifying some of the proofs as well as pave the way to a possible further generalisation of the concepts and results to include bicategories as well.

We thank John Power for pointing out to us the following theorem which is a consequence of results in [15].

Theorem 6.2.5 *For any* 2*-category* $\mathcal{K}$ *there is another* 2*-category* $\mathcal{K}_q$ *and a pseudo equivalence* $q : \mathcal{K} \to \mathcal{K}_q$ *such that for every other* 2*-category* $\mathcal{L}$*, composing with* q *induces a pseudo equivalence of* 2*-categories*

$$Ps[\mathcal{K}_q, \mathcal{L}] \simeq Hom[\mathcal{K}, \mathcal{L}] \ ,$$

where $Ps[\mathcal{K}_q, \mathcal{L}]$ *is the* 2*-category of* 2*-functors from* $\mathcal{K}_q$ *to* $\mathcal{L}$*, pseudo natural transformations and modifications and* $Hom[\mathcal{K}, \mathcal{L}]$ *the* 2*-category of pseudo functors from* $\mathcal{K}$ *to* $\mathcal{L}$*, pseudo natural transformations and modifications.*

This means that for every pseudo functor between two 2-categories, $\mathcal{K}$ and $\mathcal{L}$, there exists an "equivalent" 2-functor between $\mathcal{K}_q$ and $\mathcal{L}$. This fact will be useful in simplifying the proof of Theorem 6.3.12.

Definition 6.2.6 (Pseudo ωCat-functors) *A pseudo functor, F, between two ω**Cat**-categories, $\mathcal{K}, \mathcal{K}'$, regarded as 2-categories is a* pseudo-ω**Cat**-functor *if for any two objects of $\mathcal{K}$, A, B, the functor*

$$F_{A,B} : \mathcal{K}(A, B) \to \mathcal{K}'(FA, FB)$$

*preserves colimits of ω-chains, i.e., if it is an arrow of ω***Cat***.*

Similarly it is a pseudo-$\omega\mathbf{Cat}_0$-functor *if it also preserves initial objects.*

The following two propositions are immediate consequences of the definitions (when given at the abstraction level of [127]).

Proposition 6.2.7 *If $T : \mathcal{K} \to \mathcal{L}$ is a pseudo equivalence of ω**Cat**-categories ($\omega\mathbf{Cat}_0$-categories) then T is a pseudo ω**Cat**-functor ($\omega\mathbf{Cat}_0$-functor).*

Proposition 6.2.8 *If $T, T' : \mathcal{K} \to \mathcal{L}$ are two equivalent pseudo functors in $Hom[\mathcal{K}, \mathcal{L}]$ and T is a pseudo ω**Cat**-functor ($\omega\mathbf{Cat}_0$-functor) then so is T', moreover if T preserves pseudo colimits of ω-chains so does T'.*

6.3 Pseudo algebraic compactness

Algebraic compactness [35, 36] is a notion due to Freyd that axiomatises canonical fixed points for (endo)functors of mixed variance. As common in the categorical analysis of fixed point theorems the notion of *least* fixed point is replaced by the more robust notion of *initial* prefixed point. In fact by a lemma due to Lambek [69], given any category $\mathcal{C}$ and an endofunctor $T : \mathcal{C} \to \mathcal{C}$, an initial algebra $i : TC \to C$ for T is always an isomorphism. Dually the same hold for any final coalgebra $f : C \to TC$. This motivates the definition of algebraic *completeness* first and, as a refining step for the treatment of mixed-variance functors, *compactness* after.

Definition 6.3.1 *Let $T : \mathcal{C} \to \mathcal{C}$ be a functor. Define the category of T-algebras, T-alg to consist of*

Objects: *Arrows of $\mathcal{C}$, $a : TC \to C$.*

Arrows: *An arrow $g : C \to D$ in $\mathcal{C}$ is an arrow between $a : TC \to C$ and $b : TD \to D$ if the following square commutes:*

$$\begin{tikzcd} TC \arrow[r, "a"] \arrow[d, "Tg"'] & C \arrow[d, "g"] \\ TD \arrow[r, "b"'] & D \end{tikzcd} .$$

An initial algebra for T *is an initial object in $T\!-\!alg$.*

Dually one defines the category $T\!-\!coalg$ of T-coalgebras and a final coalgebra is a terminal object of $T\!-\!coalg$.

Lemma 6.3.2 (Lambek) *Given a functor $T : \mathcal{C} \to \mathcal{C}$, every initial algebra for T is an isomorphism in $\mathcal{C}$.*

Definition 6.3.3 (Algebraic Completeness) *A category $\mathcal{C}$ is* algebraically complete *if every endofunctor $T : \mathcal{C} \to \mathcal{C}$ has an initial algebra. It is* algebraically bicomplete *if every endofunctor has both an initial algebra and a final coalgebra.*

As Freyd remarks [36], when giving this definition, the phrase "every endofunctor" has to be understood in a 2-categorical sense, i.e., it refers to a chosen class of functors. Fiore in his PhD thesis [30] tackles this remark by considering the enriched case and defines $\mathcal{V}$-algebraically complete and bicomplete categories. In the presentation of the results and definitions we take the more relaxed view of Freyd and when concrete cases will be presented we shall explicitly mention what class of functors we are referring to.

Since any final coalgebra $f : B \to TB$ for an endofunctor T is an isomorphism, if T has an initial algebra $i : TA \to A$ there exists a unique arrow of $\mathcal{C}$, $h : A \to B$, such that the following square commutes:

$$\begin{tikzcd} TA \arrow[r, "i"] \arrow[d, "Th"'] & A \arrow[d, "h"] \\ TB \arrow[r, "f^{-1}"'] & B \end{tikzcd} \quad .$$

Following Freyd, we call this the *canonical* morphism from the algebra i to the coalgebra f.

Definition 6.3.4 (Algebraic Compactness) *A category $\mathcal{C}$ is* algebraically compact *if it is algebraically bicomplete and every canonical morphism from an initial algebra to a final coalgebra is an isomorphism.*

As an immediate consequence of the definition we have the following:

Lemma 6.3.5 (Freyd) *In an algebraically compact category, the inverse of an initial algebra is a final coalgebra and vice versa.*

If a category is algebraically compact, then mixed-variance functors have particularly well-behaving fixed points satisfying a *minimal invariance* property (see [35, 36]). We also have the following:

Theorem 6.3.6 (Freyd) *Let $\mathcal{C}$ be an algebraically compact category. Let*

$$T : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$$

be a functor. Then there exists an isomorphism $i : T(A, A) \to A$ in $\mathcal{C}$ satisfying the following universal property: For every two objects, $B, C \in |\mathcal{C}|$ and morphisms

$$f : T(B, C) \to C \qquad \text{and} \qquad g : B \to T(C, B)\ ,$$

there exists a unique pair of morphisms

$$\mathrm{it}(f, g) : A \longrightarrow C \qquad \text{and} \qquad \mathrm{coit}(f, g) : B \longrightarrow A\ ,$$

such that the following squares commute,

$$\begin{tikzcd} T(A,A) \arrow[r, "i"] \arrow[d, "T(\mathrm{coit}(f,g),\mathrm{it}(f,g))"'] & A \arrow[d, "\mathrm{it}(f,g)"] \\ T(B,C) \arrow[r, "f"'] & C \end{tikzcd} \qquad \begin{tikzcd} B \arrow[r, "g"] \arrow[d, "\mathrm{coit}(f,g)"'] & T(C,B) \arrow[d, "T(\mathrm{it}(f,g),\mathrm{coit}(f,g))"] \\ A \arrow[r, "i^{-1}"'] & T(A,A) \end{tikzcd} .$$

We call such an $i : T(A,A) \to A$ a free *dialgebra* for T (more about properties of free dialgebras can be found in [35, 36, 30, 100]).

We are interested in fixed points up to equivalence and not isomorphism, hence we need the notion of pseudo initial algebras (cf. [15]), pseudo algebraic completeness and compactness.

Definition 6.3.7 (Pseudo initial algebras) *A* pseudo initial algebra *for a pseudo-functor T on a 2-category $\mathcal{K}$ is an algebra $a : TA \to A$ satisfying the following universal property:*

1. *For every algebra $x : TX \to X$ there exists* $(\mathrm{it}(x), \iota)$ *as in*

$$\begin{tikzcd} TA \arrow[r, "a"] \arrow[d, "T(\mathrm{it}(x))"'] \arrow[dr, phantom, "\overset{\iota}{\cong}"] & A \arrow[d, "\mathrm{it}(x)"] \\ TX \arrow[r, "x"'] & X \end{tikzcd} .$$

2. *For every*

$$\begin{tikzcd} TA \arrow[r, "a"] \arrow[d, "Tu"'] \arrow[dr, phantom, "\overset{\mu}{\cong}"] & A \arrow[d, "u"] \\ TX \arrow[r, "x"'] & X \end{tikzcd} \qquad \begin{tikzcd} TA \arrow[r, "a"] \arrow[d, "Tv"'] \arrow[dr, phantom, "\overset{\nu}{\cong}"] & A \arrow[d, "v"] \\ TX \arrow[r, "x"'] & X \end{tikzcd}$$

there exists a unique 2-cell $\xi : u \Rightarrow v$ such that

$$\begin{tikzcd} TA \arrow[r, "a"] \arrow[d, bend right, "Tu"'] \arrow[dr, phantom, "\overset{\mu}{\cong}"] & A \arrow[d, bend right, "u"'] \arrow[d, bend left, "v"] \arrow[d, phantom, "\overset{\xi}{\Rightarrow}"] \\ TX \arrow[r, "x"'] & X \end{tikzcd} = \begin{tikzcd} TA \arrow[r, "a"] \arrow[d, bend right, "Tu"'] \arrow[d, bend left, "Tv"] \arrow[d, phantom, "\overset{T\xi}{\Rightarrow}"] \arrow[dr, phantom, "\overset{\nu}{\cong}"] & A \arrow[d, bend left, "v"] \\ TX \arrow[r, "x"'] & X \end{tikzcd} ,$$

that is

$$(\xi a) \cdot \mu = \nu \cdot (xT\xi) \ .$$

Observe that from the universal property it immediately follows that ξ is a pseudo cell.

Lemma 6.3.8 (Pseudo Lambek) *If $T : \mathcal{K} \to \mathcal{K}$ is a pseudo functor, then any pseudo initial algebra*

$$a : TA \to A$$

for T is an equivalence, in the sense that there exists $b : A \to TA$ such that

$$ba \cong 1_{TA} \qquad \textit{and} \qquad ab \cong 1_A \ .$$

Proof: Consider the algebra $Ta : TTA \to TA$. By the universal property of pseudo initial algebras there exists (b, β) such that

$$
\begin{tikzcd}
TA \arrow[r, "a"] \arrow[d, "Tb"'] \arrow[dr, phantom, "\overset{\beta}{\cong}"] & A \arrow[d, "b"] \\
TTA \arrow[r, "Ta"'] & TA
\end{tikzcd}
.
$$

Thus there is a unique 2-cell $\xi : ab \Rightarrow 1_A$ such that:

$$
\begin{tikzcd}
TA \arrow[r, "a"] \arrow[d, "Tb"'] \arrow[dd, bend right=60, "T(ab)"'] \arrow[dr, phantom, "\overset{\beta}{\cong}"] & A \arrow[d, "b"] \arrow[dd, bend left=60, "1_A"] \\
TTA \arrow[r, "Ta"'] \arrow[d, "Ta"'] \arrow[l, phantom, "\overset{T_{b,a}}{\cong}"] & TA \arrow[d, "a"] \arrow[r, phantom, "\overset{\xi}{\Rightarrow}"] & {} \\
TA \arrow[r, "a"'] & A
\end{tikzcd}
=
\begin{tikzcd}
TA \arrow[r, "a"] \arrow[d, "T1_A"] \arrow[d, bend right=60, "T(ab)"', "\overset{T\xi}{\Rightarrow}"] \arrow[dr, phantom, "\overset{aT_A}{\cong}"] & A \arrow[d, "1_A"] \\
TA \arrow[r, "a"'] & A
\end{tikzcd}
.
$$

Clearly the unique ξ as above is a pseudo-cell, hence $ab \cong 1_A$. To show that $ba \cong 1_{TA}$ observe that

$$
ba \cong TaTb \cong T(ab) \cong T(1_A) \cong 1_{TA} \ ,
$$

where the first isomorphism is given by β above. □

TERMINOLOGY: If $a : A \to B$ is an equivalence in a 2-category, we call any $b : B \to A$ such that $ba \cong 1_A$ and $ab \cong 1_B$ a *pseudo inverse to a*.

It is immediately seen that any two pseudo inverses are isomorphic.

By a dual statement to the Pseudo Lambek Lemma, pseudo final coalgebras are equivalences. If $f : B \to TB$ is a pseudo final coalgebra for T, for any $g : TB \to B$ be pseudo inverse to f, if $(\mathrm{it}(g), \iota)$ is as in

$$
\begin{tikzcd}
TA \arrow[r, "a"] \arrow[d, "T(\mathrm{it}(g))"'] \arrow[dr, phantom, "\overset{\iota}{\cong}"] & A \arrow[d, "\mathrm{it}(g)"] \\
TB \arrow[r, "g"'] & B
\end{tikzcd}
$$

we shall call $\mathrm{it}(g) : A \to B$ a *canonical* arrow from the pseudo initial algebra a to the pseudo final coalgebra f.

Definition 6.3.9 (Pseudo Algebraic Completeness and Compactness) *Define a* 2*-category* $\mathcal{K}$ *to be* pseudo algebraically complete *if every pseudo endofunctor,* $T : \mathcal{K} \to \mathcal{K}$ *has a pseudo initial algebra.*

The 2*-category* $\mathcal{K}$ *is* pseudo algebraically bicomplete *if every pseudo endofunctor has both pseudo initial algebra and pseudo final coalgebra.*

The 2*-category* $\mathcal{K}$ *is* pseudo algebraically compact *if it is pseudo algebraically bicomplete and the* canonical *map from a pseudo initial algebra to a pseudo final coalgebra is an equivalence.*

We shall be interested in pseudo algebraic compactness with respect to pseudo ω**Cat**-functors.

The following generalisation of [125, Lemma 2] (Basic Lemma) to the "pseudo" case is due to Marcelo Fiore. It provides a tool for finding pseudo initial algebras of pseudo functors by iterating the application of the functor starting at some pseudo initial object.

Lemma 6.3.10 (Pseudo Basic Lemma) *Let $\mathcal{K}$ be a 2-category with pseudo initial object* 0 *and let $T : \mathcal{K} \to \mathcal{K}$ be a pseudo functor. For $\perp : 0 \to T0$ consider the ω-chain*

$$\langle T^n \perp : T^n 0 \to T^{n+1} 0 \rangle$$

and let $\Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n : T^n 0 \to A$ be a pseudo colimit for it.

If

$$\Phi'_n = T(\Phi_n) T_{f_n, \varphi_{n+1}} : T(\varphi_{n+1}) T(f_n) \stackrel{\sim}{\Longrightarrow} T(\varphi_{n+1} f_n) \stackrel{\sim}{\Longrightarrow} T\varphi_n : T^{n+1} 0 \to TA$$

is a pseudo colimit of the ω-chain $\langle T^{n+1} \perp : T^{n+1} 0 \to T^{n+2} 0 \rangle$ and $a : TA \to A$ mediates between the pseudo cones $\langle \Phi'_n \rangle$ and $\langle \Phi_{n+1} \rangle$, then a is a pseudo initial T-algebra.

We embark now on generalising part of the definitions and results of [30, Chapter 7]. In fact we identify a class of ω**Cat**-categories for which pseudo algebraic compactness will be guaranteed by the results above (cf. [30, Definition 7.3.11]).

Definition 6.3.11 *A* Kcat *is an $\omega\mathbf{Cat}_0$-category with pseudo initial object and pseudo colimits of ω-chains of coreflections.*

To ensure pseudo algebraic compactness of Kcats we need some preliminary results.

Theorem 6.3.12 *For $\mathcal{K}$ an ω**Cat**-category, let*

$$\langle \eta_n, \varepsilon_n : f_n \dashv g_n : A_{n+1} \longrightarrow A_n \rangle$$

be an ω-chain in $\mathcal{K}^{\mathrm{cor}}$ and let

$$\langle \Phi_n : \varphi_{n+1} f_n \stackrel{\sim}{\Longrightarrow} \varphi_n : A_n \to A \rangle$$

*be a pseudo colimit for the chain of f_n's. If $T : \mathcal{K} \to \mathcal{K}$ is a pseudo ω**Cat**-functor, then*

$$\langle T(\Phi_n) T_{f_n, \varphi_{n+1}} : T(\varphi_{n+1}) T(f_n) \stackrel{\sim}{\Longrightarrow} T(\varphi_n) : TA_n \to TA \rangle$$

is a pseudo colimit too.

Proof: Observe first of all that by means of the coherence results exposed in Section 6.2 we can reduce to assume T a 2-functor rather than a pseudo functor. In fact any pseudo functor from $\mathcal{K} \to \mathcal{K}$ (by Theorem 6.2.5) will be equivalent in $Hom[\mathcal{K}, \mathcal{K}]$ to a pseudo functor obtained by precomposing a 2-functor from $\mathcal{K}_q$ to $\mathcal{K}$ with $q : \mathcal{K} \to \mathcal{K}_q$. So if we know that the theorem holds for the 2-functors, say, T', from $\mathcal{K}_q$ to $\mathcal{K}$ then it will also hold for the pseudo functors of the form $T'q$, since q is a pseudo equivalence and hence, by Proposition 6.2.8 it will hold for all pseudo $\omega\mathbf{Cat}$-endofunctors of $\mathcal{K}$.

By the Local-Characterisation Theorem (Theorem 6.1.9), there exists a pseudo cone of adjoints

$$(\Phi_n, \Gamma_n) : (\iota_{n+1}, \jmath_{n+1} : \varphi_{n+1} \dashv \gamma_{n+1})(\eta_n, \varepsilon_n : f_n \dashv g_n) \stackrel{\sim}{\Longrightarrow} (\iota_n, \jmath_n : \varphi_n \dashv \gamma_n)$$

such that the canonical cones $\langle \varphi_n \gamma_n \rangle \stackrel{\cdot}{\Longrightarrow} 1_A$ and $\langle g_{l,n} f_{n,l} \rangle_l \stackrel{\cdot}{\Longrightarrow} \gamma_n \varphi_n$ are colimiting. Since 2-functors preserve adjoints, the following is also a pseudo cone of adjoints

$$(T\Phi_n, T\Gamma_n) : (T\iota_{n+1}, T\jmath_{n+1} : T\varphi_{n+1} \dashv T\gamma_{n+1})(T\eta_n, T\varepsilon_n : Tf_n \dashv Tg_n) \stackrel{\sim}{\Longrightarrow} (T\iota_n, T\jmath_n : T\varphi_n \dashv T\gamma_n)\,.$$

Moreover since T is assumed to be an $\omega\mathbf{Cat}$-functor, it preserves locally colimit of ω-chains, thus the canonical cones $\langle T\varphi_n T\gamma_n \rangle \stackrel{\cdot}{\Longrightarrow} 1_{TA}$ and $\langle Tg_{l,n} Tf_{n,l} \rangle_l \stackrel{\cdot}{\Longrightarrow} T\gamma_n T\varphi_n$ are colimiting. Again by the Local Characterisation Theorem this implies that the pseudo cone

$$\langle T(\Phi_n) : T(\varphi_{n+1}) T(f_n) \stackrel{\sim}{\Longrightarrow} T(\varphi_n) : TA_n \to TA \rangle$$

is a pseudo colimit. $\square$

Proposition 6.3.13 *For any Kcat $\mathcal{K}$ the pseudo initial object, 0, is also pseudo terminal and for every object A of $\mathcal{K}$, any pair of arrows*

$$\begin{tikzcd} 0 \arrow[r, bend left, "f"] & A \arrow[l, bend left, "g"] \end{tikzcd}$$

forms a coreflection, $f \dashv g$.

Proof: To see that 0 is pseudo terminal, observe first of all that for any object A of $\mathcal{K}$, $\mathcal{K}(A, 0)$ is non-empty since it has initial object. Moreover, any arrow $g : A \to 0$ is initial in $\mathcal{K}(A, 0)$, since $g = 1_0 g$ and composition preserves initiality (1_0 is initial in $\mathcal{K}(0, 0)$, just like any other arrow $0 \to 0$ since 0 is pseudo initial). In particular given any two arrows $g_1, g_2 : A \to 0$, by initiality of g_1 there exists a unique 2-cell, $g_1 \Rightarrow g_2$, i.e., $\mathcal{K}(A, 0) \simeq \mathbf{1}$.

If $f : 0 \to A$ and $g : A \to A$ are two arrows then there exists (again by initiality) unique $\eta : 1_0 \Rightarrow gf$ and $\varepsilon : fg \Rightarrow 1_A$ that satisfy the triangular identities because of the universal property of initial objects. Moreover η is an isomorphism just like any other arrow in $\mathcal{K}(0, 0)$. $\square$

Kcats are closed under duals and products.

Proposition 6.3.14 *If $\mathcal{K}$ and $\mathcal{K}'$ are Kcats, then $\mathcal{K}^{\mathrm{op}}$ and $\mathcal{K} \times \mathcal{K}'$ are Kcats.*

Proof: The proof is straightforward for the product case. For the dualisation process, observe that from Proposition 6.3.13 it immediately follows that $\mathcal{K}^{\mathrm{op}}$ has a pseudo initial object. Since 2-cells are not reversed, then $\mathcal{K}^{\mathrm{op}}$ is $\omega\mathbf{Cat}_0$-enriched and any ω-chain of coreflections

$$\langle \eta_n, \varepsilon_n : f_n^{\mathrm{op}} \dashv g_n^{\mathrm{op}} : A_{n+1} \longrightarrow A_n \rangle$$

in $\mathcal{K}^{\mathrm{op}}$ is derived from the chain of coreflections in $\mathcal{K}$ given by

$$\langle \eta_n, \varepsilon_n : g_n \dashv f_n : A_{n+1} \longrightarrow A_n \rangle$$

and a pseudo limiting cone for this chain (provided by the limit/colimit coincidence) will be a pseudo colimit for the chain of op-arrows. □

Definition 6.3.15 (Pseudo ωCat-algebraic completeness and compactness) *An $\omega\mathbf{Cat}$-category is said to be* pseudo $\omega\mathbf{Cat}$-algebraically complete *if it is pseudo algebraically complete with respect to pseudo $\omega\mathbf{Cat}$-functors.*

An $\omega\mathbf{Cat}$-category is said to be pseudo $\omega\mathbf{Cat}$-algebraically compact *if it is pseudo algebraically compact with respect to pseudo $\omega\mathbf{Cat}$-functors.*

Theorem 6.3.16 *Kcats are pseudo $\omega\mathbf{Cat}$-algebraically compact.*

Proof: Given a pseudo $\omega\mathbf{Cat}$-functor, $T : \mathcal{K} \to \mathcal{K}$, and a coreflection

$$\begin{tikzcd} 0 \arrow[r, bend left, "\perp"] & T0 \arrow[l, bend left, "!"] \end{tikzcd}$$

that certainly exists because of the Proposition 6.3.13, with unit η and counit ε, consider the chain of coreflections defined by

Objects: $A_0 = 0$, $A_{n+1} = TA_n$

Arrows: $f_0 = \perp$, $g_0 = !$, $f_{n+1} = Tf_n$, $g_{n+1} = Tg_n$ with unit η_{n+1} and counit ε_{n+1} given by

$$\begin{aligned} \eta_{n+1} &\stackrel{\text{def}}{=} T^{-1}_{f_n,g_n} \cdot T\eta_n \cdot T_{A_n} : 1_{TA_n} \stackrel{\sim}{\Longrightarrow} T1_{A_n} \Rightarrow T(g_n f_n) \stackrel{\sim}{\Longrightarrow} Tg_n Tf_n \\ \varepsilon_{n+1} &\stackrel{\text{def}}{=} T^{-1}_{A_{n+1}} \cdot T\varepsilon_n \cdot T_{g_n,f_n} : Tf_n Tg_n \stackrel{\sim}{\Longrightarrow} T(f_n g_n) \Rightarrow T(1_{A_{n+1}}) \stackrel{\sim}{\Longrightarrow} 1_{TA_{n+1}} \ . \end{aligned}$$

Since $\mathcal{K}$ is a Kcat the chain of coreflections has a pseudo colimit of vertex A in $\mathcal{K}^{\mathrm{cor}}$ and by Corollary 6.1.10 the cone of left adjoints is pseudo colimiting in $\mathcal{K}$. By Theorem 6.3.12 this is preserved by application of T and by the Pseudo Basic Lemma, any mediating equivalence $TA \to A$ is a pseudo initial algebra for T. By the limit colimit coincidence, a dual statement to the Pseudo Basic Lemma yields that any pseudo inverse to the initial algebra is a pseudo final coalgebra and this is easily seen to be equivalent to pseudo algebraic compactness. □

Thus, every pseudo $\omega\mathbf{Cat}$-functor $T : \mathcal{K}^{\mathrm{op}} \times \mathcal{K} \to \mathcal{K}$ on a Kcat $\mathcal{K}$ has a *free pseudo dialgebra*

$$T(A, A) \simeq A$$

characterised by the following universal property: for every $x' : X' \to T(X, X')$ and $x : T(X', X) \to X$, we have

$$\begin{tikzcd} X' \arrow[r, "x'"] \arrow[d, "\mathrm{coit}(x',x)"'] \arrow[dr, phantom, "\cong"] & T(X, X') \arrow[d, "T(\mathrm{it}(x',x),\mathrm{coit}(x',x))"] \\ A \arrow[r, "\simeq"] & T(A, A) \end{tikzcd}$$

$$\begin{tikzcd} T(A, A) \arrow[r, "\simeq"] \arrow[d, "T(\mathrm{coit}(x',x),\mathrm{it}(x',x))"'] \arrow[dr, phantom, "\cong"] & A \arrow[d, "\mathrm{it}(x',x)"] \\ T(X', X) \arrow[r, "x"'] & X \end{tikzcd}$$

given uniquely up to canonical coherent isomorphism (as defined for pseudo initial algebras).

Corollary 6.3.17 **Cocont** *and* $\mathbf{Cocont}_M$ *are pseudo* $\omega\mathbf{Cat}$*-algebraically compact.*

Definition 6.3.9, 6.3.11 and 6.3.15 can be generalised to ($\omega\mathbf{Cat}$-)bicategories. Similarly using the pseudo equivalences (and the related preservation properties) of Section 6.2 the Pseudo Lambek Lemma as well as the other theorems and propositions of this section could be restated for bicategories. We shall not pursue this generalisation effort in all details. For this reason we concentrate, in the remainder of this chapter, on the interpretation of the type theory of Section 4.5 in **Cocont** (and $\mathbf{Cocont}_M$ as we shall see) rather than **Prof**.

6.4 Recursive types

Using the results of the previous sections we can extend the type theory of Section 4.5 with recursive types

$$\mu\vartheta.t\ .$$

We begin with a parametrisation result (cf. [30, Theorem 7.1.12 and Definition 6.1.7]).

Theorem 6.4.1 *Let* $\mathcal{K}$ *and* $\mathcal{L}$ *be two pseudo* $\omega\mathbf{Cat}$*-algebraically complete (compact) categories and* $T : \mathcal{K} \times \mathcal{L} \to \mathcal{L}$ *be a pseudo* $\omega\mathbf{Cat}$*-functor. For any* $A \in |\mathcal{K}|$*, write*

$$\mu\, T_A$$

for the object part of a chosen pseudo-initial algebra for the endo pseudo functor, $T_A : \mathcal{L} \to \mathcal{L}$*, defined by freezing the first component to be, the object* A*, or* 1_A *or* 1_{1_A}*. Then the mapping* $A \to \mu\, T_A$ *extends canonically to a pseudo-*$\omega\mathbf{Cat}$*-functor,* $\mu\, T_{(-)} : \mathcal{K} \to \mathcal{L}$*.*

Proof: The proof of this fact relies on some analysis of the properties of canonical arrows from initial algebras to any other algebra (cf. [30, Chapter 7]). The details can be found in Appendix B. □

Finally we check that all the pseudo functors involved in giving the semantics of the type theory of Section 4.5 are pseudo-ω**Cat**-functors, but this is straightforward calculation.

Proposition 6.4.2 *The pseudo functors, $\otimes$, $+$, $(-)^*$, $!(-)$ and $(-)_\perp$ are pseudo-ω**Cat**-functors.*

NOTATION: If $T : (\mathcal{K}^{\mathrm{op}} \times \mathcal{K})^n \to \mathcal{L}$ is a pseudo-functor. We write $\breve{T}$ for the pseudo-functor,

$$\breve{T} : (\mathcal{K}^{\mathrm{op}} \times \mathcal{K})^n \to \mathcal{L}^{\mathrm{op}} \times \mathcal{L}$$

defined (on objects) by $\breve{T}(A, B) = (T(B, A), T(A, B))$.

Clearly $T = \pi_2 \breve{T}$, where $\pi_2 : \mathcal{L}^{\mathrm{op}} \times \mathcal{L} \to \mathcal{L}$ is the projection on the second component.

Consider now a term $\mu\,\vartheta.\, t$ in the type theory of Section 4.5 extended with recursive types. We need to provide a definition for

$$[\![\Theta \vdash \mu\,\vartheta.\, t]\!] : (\mathbf{Cocont}^{\mathrm{op}} \times \mathbf{Cocont})^{|\Theta|} \to \mathbf{Cocont}\ ,$$

where Θ is a set of variables including the free ones in $\mu\,\vartheta.\, t$.

Assume $T : (\mathbf{Cocont}^{\mathrm{op}} \times \mathbf{Cocont}) \times (\mathbf{Cocont}^{\mathrm{op}} \times \mathbf{Cocont})^{|\Theta|} \to \mathbf{Cocont}$ to be $[\![\Theta + \{\vartheta\} \vdash t]\!]$. Define

$$[\![\Theta \vdash \mu\,\vartheta.\, t]\!] = \pi_2(\mu\,\breve{T}_{(-)})\ ,$$

where $\mu\,\breve{T}_{(-)}$ is defined according to Theorem 6.4.1.

6.4.1 The two examples revisited

Back to the examples of Chapter 5 we understand now in terms of uniform fixed points for endofunctors in **Cocont**, how to give solutions to our domain equations; that is

$$\mathbb{P} = \mathbb{P}_\perp + \sum_{a \in Ch} \mathbb{P}_\perp + \sum_{\overline{a} \in \overline{Ch}} \mathbb{P}_\perp \text{ stands for } [\![\vdash \mu\,\vartheta.\, (\vartheta_\perp + \sum_{a \in Ch} \vartheta_\perp + \sum_{\overline{a} \in \overline{Ch}} \vartheta_\perp)]\!]$$

while

$$\mathbb{P} = \mathbb{P}_\perp + \sum_{a \in Ch} (\mathbb{V} \multimap \mathbb{P})_\perp + \sum_{(\overline{a}, v) \in \overline{Ch} \times V} \mathbb{P}_\perp \text{ stands for}$$

$$[\![\vdash \mu\,\vartheta.\, (\vartheta_\perp + \sum_{a \in Ch} (\sum_{v \in V} \vartheta)_\perp + \sum_{(\overline{a}, v) \in \overline{Ch} \times V} \vartheta_\perp)]\!]\ .$$

There is still one point that we shall clarify, namely the reason why for the equations above one can reduce to look for the solution in **Poset** the locally ordered category of partial ordered sets and monotone functions (ordered pointwise). This is due to a property of the embedding

$$e : \mathbf{Poset} \hookrightarrow \mathbf{Cat}\ ,$$

that takes any poset and regards it as a category with at most one arrow between any two objects (elements of the poset) and exactly one if the two elements are in the order relation. In our jargon such a category is called a *partial order category*. The following property is then particularly useful for our purposes.

Proposition 6.4.3 *The category* **Poset** *has small sums and small filtered colimits and the embedding* $e : \mathbf{Poset} \hookrightarrow \mathbf{Cat}$ *preserves them.*

Hence,

Corollary 6.4.4 *Let* $F : \mathbf{Cat} \to \mathbf{Cat}$ *be an endofunctor that preserves colimits of* ω*-chains and such that for every partial order category,* P*,* $F(P)$ *is a partial order category. That is, there exists a functor* $F' : \mathbf{Poset} \to \mathbf{Poset}$ *such that the following square commutes:*

$$\begin{tikzcd} \mathbf{Poset} \arrow[r, "F'"] \arrow[d, "e"'] & \mathbf{Poset} \arrow[d, "e"] \\ \mathbf{Cat} \arrow[r, "F"'] & \mathbf{Cat}\ . \end{tikzcd}$$

If $i : F'(P) \xrightarrow{\simeq} P$ *is an initial algebra for* F'*, then* $e(i)$ *is an initial algebra for* F*.*

Proof: The only thing to note is that if F (F') preserves colimits of ω-chains, then an initial algebra for F (F') is calculated by taking the colimit of the standard ω-chains obtained by iterating F (F'), starting with the empty category (poset) and this is a filtered colimit. $\Box$

Poset has other colimits as well (in fact it is a cocomplete category being locally finitely presentable [5]) but they are not all preserved by the embedding into **Cat** as the following counterexample that we learned from Peter Selinger shows.

Example 6.4.5 *Consider the pair of parallel arrows* $\mathbf{1} \overset{\perp}{\underset{\top}{\rightrightarrows}} \mathbf{2}$ *from the one element poset to the two elements one* $\perp \leq \top$ *that respectively pick the bottom and the top element (as their name suggests). The coequaliser of such a pair is given by* $\mathbf{1}$ *itself, while if we regard the partial orders as categories the coequaliser is given by the monoid of natural numbers regarded as a one object category.*

Clearly all the **Cat**-functors used to model the type theory (but for the exponential) take partial order categories to partial order categories. Therefore when looking for a solution to a domain equation not involving "!", we can find it in **Poset** and then transfer it to **Cocont** using the embeddings:

$$\mathbf{Poset} \hookrightarrow \mathbf{Cat} \hookrightarrow \mathbf{Prof} \simeq \mathbf{Cocont} \text{ (cf. Proposition 6.1.12).}$$

6.5 Relational structures

In the next chapters we shall consider more examples of concurrent process calculi to which we give a presheaf semantics. Before doing that, though, we want to use the results of the previous sections to begin with the study of (open map) bisimulation from a domain theoretical point of view in the style of [99, 53]. The idea is to equip $\omega\mathbf{Cat}_0$-categories with relational structures [94]. Any given (admissible) relational structure

$\mathcal{R}$ on a $\omega\mathbf{Cat}_0$-category $\mathcal{K}$ will induce, by the Grothendieck construction, an $\omega\mathbf{Cat}_0$-category, $\{\mathcal{K} \mid \mathcal{R}\}$, whose objects will be pairs $\{A \mid R\}$ consisting of an object of $\mathcal{K}$ and a relation drawn from a partial order of admissible relations. An important result here will be that is $\mathcal{K}$ is a Kcat then so is $\{\mathcal{K} \mid \mathcal{R}\}$. By focusing on two particular relational structures over **Cocont** and $\mathbf{Cocont}_M$ we carry out a study of bisimulation. In particular, using some extra "intensional" information provided by presheaf categories, we give a domain theoretic characterisation of bisimulation for arbitrary trees. To do so, after having defined relational structures we carry out a study of induced induction/coinduction principles [53, 99, 31, 100].

Definition 6.5.1 *Define* $(\mathbf{CPPO}_\perp)^*$ *to be the category of possibly large posets* P *such that* P^{op} *is pointed (i.e., has a least element) and* ω*-complete (i.e., has least upper bounds of* ω*-chains) and monotone functions* $f : P \to Q$ *such that* $f^{\mathrm{op}} : P^{\mathrm{op}} \to Q^{\mathrm{op}}$ *is strict (i.e., least element preserving) and continuous (i.e., monotone and least upper bound of* ω*-chains preserving).*

Definition 6.5.2 (Relational Structures) *A* relational structure *on a category* $\mathcal{C}$ *is a functor* $\mathcal{R} : \mathcal{C}^{\mathrm{op}} \to (\mathbf{CPPO}_\perp)^*$.

The order relation on a $\mathcal{R}(C)$*, for* $C \in |\mathcal{C}|$*, will usually be denoted with the subset symbol,* $\subset$.

If C *is an object of* $\mathcal{C}$*, the top element of* $\mathcal{R}(C)$ *will be written as* $\top_{\mathcal{R}(C)}$.

An admissible relational structure $\mathcal{R}$ *on an* $\omega\mathbf{Cat}_0$*-category* $\mathcal{K}$ *is a relational structure on the ordinary category underlying* $\mathcal{K}$*, such that*

1. *for a pair of morphisms* $f, g : A \to B$*, if* $f \cong g$ *then* $\mathcal{R}(f) = \mathcal{R}(g)$*;*
2. *for a morphism* $f : A \to B$ *and an element* $S \in \mathcal{R}(B)$*, if* f *is initial in* $\mathcal{K}(A, B)$ *then* $\mathcal{R}(f)(S) = \top_{\mathcal{R}(A)}$*;*
3. *for an* ω*-chain* $\langle f_n \rangle$ *in* $\mathcal{K}(A, B)$ *with colimit* $f : A \to B$*,*

   $$R \subset \mathcal{R}(f_n)(S), \text{ for all } n, \text{ implies } R \subset \mathcal{R}(f)(S)$$

   for all $R \in \mathcal{R}(A)$ *and* $S \in \mathcal{R}(B)$.

On a $\mathbf{Cppo}_\perp$-category any Pitts' relational structure admitting inverse images and intersections in which every relation is admissible (as defined in [100]) is an admissible relational structure in our sense, but not vice versa as we do not require, in general, $\mathcal{R}(f)$ to preserve greatest lower bounds.

NOTATION: If $\mathcal{R}$ is a relational structure on $\mathcal{C}$ and $f : C \to D$ is an arrow in $\mathcal{C}$, with $R \in \mathcal{R}(C)$ and $S \in \mathcal{S}(D)$ we write $f : R \subset S$ for $R \subset \mathcal{R}(f)S$.

An admissible relational structure on an $\omega\mathbf{Cat}_0$-category $\mathcal{K}$ induces a category of relations in $\mathcal{K}$.

Definition 6.5.3 *Let* $\mathcal{R}$ *be an admissible relational structure on an* $\omega\mathbf{Cat}_0$*-category* $\mathcal{K}$*. The* $\omega\mathbf{Cat}_0$-category of relations $\{\mathcal{K} \mid \mathcal{R}\}$ *has: objects given by pairs* $\{C \mid R\}$ *with*

$C \in |\mathcal{K}|$ and $R \in \mathcal{R}(C)$; hom-categories $\{\mathcal{K} \mid \mathcal{R}\}(\{A \mid R\}, \{B \mid S\})$ defined as the full subcategory of $\mathcal{K}(A,B)$ consisting of all those f such that $f : R \subset S$; and identities and compositions given as in $\mathcal{K}$.

In fact, since partial orders can be regarded as categories, the above definition is another example of the Grothendieck construction that we recalled in Section 1.4.1.

In [21] the following theorem is proved

Theorem 6.5.4 (Fiore) *Let $\mathcal{R}$ be an admissible relational structure on an $\omega\mathbf{Cat}_0$-category $\mathcal{K}$. The forgetful functor*

$$U : \{\mathcal{K}|\mathcal{R}\} \to \mathcal{K}$$

that projects each pair onto the first component is faithful, $\omega\mathbf{Cat}_0$-enriched, creates ([76]) pseudo initial objects and pseudo colimits of ω-chains of coreflections.

The above theorem allows us to deduce that if one has a relational structure on a $Kcat$ then the induced category of relations is a $Kcat$, too.

Corollary 6.5.5 *For an admissible relational structure $\mathcal{R}$ on a Kcat $\mathcal{K}$, the category of relations $\{\mathcal{K}|\mathcal{R}\}$ is a Kcat.*

As we have already said we shall be concerned with certain specific relational structures on **Cocont** and **Cocont**$_M$. These are defined below.

Definition 6.5.6 *1.* Admissible extensional relations on **Cocont**$_M$: Ext *is defined as follows.*
For every small category $\mathbb{C}$, Ext($\mathbb{C}$) *is the complete meet semilattice of relations $R \subseteq |\widehat{\mathbb{C}}\ ^2|$ such that*

(a) $X' \cong X \; R \; Y \cong Y'$ implies $X' \; R \; Y'$;

(b) $\emptyset \; R \; \emptyset$;

(c) for every pair of ω-chains of monomorphisms *$\vec{X}$ and $\vec{Y}$ with colimits X and Y respectively,*

$$\textit{if } \vec{X}_n \; R \; \vec{Y}_n, \textit{ for all } n, \textit{ then } X \; R \; Y.$$

These relations are ordered by inclusion and the action of Ext *on morphisms is by inverse image, i.e.,*

$$\mathrm{Ext}(F)(S) = \{(X,Y) \in |\widehat{\mathbb{C}}| \times |\widehat{\mathbb{C}}| \mid (FX, FY) \in S\} \; .$$

2. Admissible intensional relations on **Cocont**: Int *is defined as follows.*
For every small category $\mathbb{C}$, Int($\mathbb{C}$) *is the complete meet semilattice of* intensional *relations $R \subseteq |\widehat{\mathbb{C}}^{\swarrow\searrow}|$ such that*

(a) for every triple of isomorphisms $W \cong W'$, $X \cong X'$, and $Y \cong Y'$,

$$(X \leftarrow W \rightarrow Y) \in R \textit{ implies } (X' \cong X \leftarrow W \xleftarrow{\cong} W' \xrightarrow{\cong} W \rightarrow Y \cong Y') \in R\,;$$

(b) $(\emptyset \leftarrow \emptyset \rightarrow \emptyset) \in R$;

(c) *for every span of natural transformations* $\vec{X} \overset{p}{\Longleftarrow} \vec{W} \overset{q}{\Longrightarrow} \vec{Y}$ *where* $\vec{X}$, $\vec{W}$, *and* $\vec{Y}$ *are* ω*-chains with colimits* X, W, *and* Y *respectively,*

$$\textit{if } (\vec{X}_n \xleftarrow{p_n} \vec{W}_n \xrightarrow{q_n} \vec{Y}_n) \in R, \textit{ for all } n, \textit{ then } (X \xleftarrow{\operatorname{colim} p} W \xrightarrow{\operatorname{colim} q} Y) \in R.$$

These intensional relations are ordered by inclusion and the action of Int *on morphisms is by inverse image, i.e.,*

$$\mathrm{Int}(F)(S) = \{X \xleftarrow{f} Z \xrightarrow{g} Y \in |\widehat{\mathbb{C}}^{\swarrow\searrow}| \mid FX \xleftarrow{Ff} FZ \xrightarrow{Fg} FY \in S\} .$$

3. *Every relational structure* $\mathcal{R}$ *on* **Cocont** *induces a relational structure* $\mathcal{R}^*$ *on* $\mathbf{Cocont}^{\mathrm{op}}$ *with*

$$\mathcal{R}^*(\mathbb{C}) \stackrel{\text{def}}{=} \mathcal{R}(\mathbb{C}^{\mathrm{op}}) \textit{ and } \mathcal{R}^*(F) \stackrel{\text{def}}{=} \mathcal{R}(F^0) , \tag{6.25}$$

where, if $F : \widehat{\mathbb{C}} \to \widehat{\mathbb{D}}$, $F^0 : \widehat{\mathbb{D}^{\mathrm{op}}} \to \widehat{\mathbb{C}^{\mathrm{op}}}$ *is defined by extending the corresponding dualised profunctor (cf. Section 4.3 and Example 6.6.7).*

6.6 Coinduction and bisimulation

We use Theorem 6.5.4 to derive induction and coinduction principles for recursively defined domains [100, 53]. The coinduction principle will be used to prove a coinduction property based on bisimulation, i.e., to show that the relational part of a free pseudo algebra in a category of relations $\{\mathcal{K}\,|\,\mathcal{R}\}$ is the maximal bisimulation.

With an eye on applications to **Cocont** (see Section 6.7) we consider separately the case of covariant functors and of mixed-variance ones, the latter requiring the notion of *involutory category* (cf.[30, Chapter 6]).

6.6.1 Covariant functors

NOTATION: Let $\mathcal{R}$ be an admissible relational structure on a Kcat $\mathcal{K}$, and let T and $T^\#$ be pseudo ω**Cat**-functors such that the diagram

$$\begin{tikzcd} \{\mathcal{K}\,|\,\mathcal{R}\} \arrow[r, "T^\#"] \arrow[d, "U"'] & \{\mathcal{K}\,|\,\mathcal{R}\} \arrow[d, "U"] \\ \mathcal{K} \arrow[r, "T"'] & \mathcal{K} \end{tikzcd}$$

commutes, where U is the forgetful functor, we often write $(T(A), T_{\mathcal{R}}(R))$ in place of $T^\#(\{A\,|\,R\})$.

Similarly we shall often write (A, R) for $\{A\,|\,R\}$.

As a consequence of Theorem 6.5.4 we have the following:

Proposition 6.6.1 *Let $\mathcal{R}$ be an admissible relational structure on a Kcat $\mathcal{K}$. Consider pseudo ω**Cat**-functors T and $T^{\#}$ such that the diagram*

$$\begin{tikzcd} \{\mathcal{K}\,|\,\mathcal{R}\} \arrow[r, "T^{\#}"] \arrow[d, "U"'] & \{\mathcal{K}\,|\,\mathcal{R}\} \arrow[d, "U"] \\ \mathcal{K} \arrow[r, "T"'] & \mathcal{K} \end{tikzcd}$$

commutes, where U denotes the forgetful $\omega\mathbf{Cat}_0$-functor. Then, for every free pseudo T-algebra

$$\mathrm{fold} : T(D) \simeq D : \mathrm{unfold}$$

there exists (a necessarily unique) $\Delta \in \mathcal{R}(D)$ such that

$$\mathrm{fold} : T^{\#}(\{D\,|\,\Delta\}) \simeq \{D\,|\,\Delta\} : \mathrm{unfold}$$

is a free pseudo $T^{\#}$-algebra.

Proof: From the results of the previous sections we know that D is equivalent to the vertex of any pseudo colimit for the ω-chain

$$0 \to T0 \to T^2 0 \to T^3 0 \to \cdots ,$$

where 0 is a pseudo initial object. Hence, because of Theorem 6.5.4 there exists Δ, such that $\{D\,|\,\Delta\}$ is the pseudo colimit of

$$(0, \top_{\mathcal{R}(0)}) \to (T0, T_{\mathcal{R}}(\top_{\mathcal{R}(0)}) \to (T^2 0, T^2_{\mathcal{R}}(\top_{\mathcal{R}(0)})) \to \cdots .$$

Moreover, Δ is unique as $\mathcal{R}(D)$ is a partial order set. □

Proposition 6.6.2 (Induction/Coinduction principles) *Under the hypothesis of Proposition 6.6.1, the free pseudo algebra $\{D \mid \Delta\}$ enjoys induction and coinduction principles as expressed by the following rules [100, 53]:*

- *For $a : TA \to A$ and $R \in \mathcal{R}(A)$,*

$$\frac{a : T_{\mathcal{R}}(R) \subset R}{\mathrm{it}(a) : \Delta \subset R}$$

- *For $z : Z \to TZ$ and $R \in \mathcal{R}(Z)$,*

$$\frac{z : R \subset T_{\mathcal{R}}(R)}{\mathrm{coit}(z) : R \subset \Delta}$$

Proof: The rules above are an immediate consequence of the universal properties of the free algebra

$$\mathrm{fold} : T^{\#}(\{D\,|\,\Delta\}) \simeq \{D\,|\,\Delta\} : \mathrm{unfold} .$$

We look at the first rule only, the other case is analogous.

Since fold : $T^{\#}(\{D \mid \Delta\}) \xrightarrow{\sim} \{D \mid \Delta\}$ is a pseudo initial algebra, for any other algebra $a : (T(A), T_{\mathcal{R}}(R)) \to (A, R)$, there exists $\mathrm{it}(a) : (D, \Delta) \to (A, R)$ and ι such that

$$\begin{tikzcd}
(TD, \Delta) \arrow[r, "\mathrm{fold}"] \arrow[d, "T(\mathrm{it}(a))"'] \arrow[dr, phantom, "\overset{\iota}{\cong}"] & (D, \Delta) \arrow[d, "\mathrm{it}(a)"] \\
(T(A), T_{\mathcal{R}}(R)) \arrow[r, "a"'] & (A, R) \; .
\end{tikzcd}$$

But $a : T(A) \to A$ is an algebra for (A, R) if and only if, by definition, $a : T_{\mathcal{R}}(R) \subset R$, while $\mathrm{it}(a) : D \to A$ is an arrow $(D, \Delta) \to (A, R)$ if and only if $\mathrm{it}(a) : \Delta \subset R$, hence the rule is justified. □

Definition 6.6.3 *Let* fold : $T^{\#}(\{D \mid \Delta\}) \xrightarrow{\sim} \{D \mid \Delta\}$ *be a free algebra. Define a relation* $R \in \mathcal{R}(D)$ *to be a* $T^{\#}$-bisimulation *if it satisfies the condition*

$$\mathrm{unfold} : R \subset T_{\mathcal{R}}(R) \; .$$

Proposition 6.6.4 *Under the hypothesis of Proposition 6.6.1,* Δ *is a* $T^{\#}$*-bisimulation.*

But we can say more, in fact we can establish a coinduction property (cf. [99, 100, 31, 53]) that establishes Δ to be the maximal bisimulation.

Proposition 6.6.5 *Under the hypothesis of Proposition 6.6.1,*

$$\Delta = \bigvee \{R \in \mathcal{R}(D) \mid R \textit{ is a } T^{\#}\textit{-bisimulation}\} \; .$$

Proof: By Proposition 6.6.4 Δ is a $T^{\#}$-bisimulation. Moreover it satisfies the coinduction principle of Proposition 6.6.2

$$\frac{\mathrm{unfold} : R \subset T_{\mathcal{R}}(R)}{\mathrm{coit}(\mathrm{unfold}) : R \subset \Delta \; .}$$

But coit(unfold) is isomorphic to 1_D, hence $R \subset D$ for any $T^{\#}$-bisimulation. □

6.6.2 Mixed-variance functors

To treat mixed-variance functors we consider pseudo involutory 2-categories (viz. 2-categories which are *self dual via a pseudo involution* —see [30]).

Definition 6.6.6 *A* pseudo involutory 2-category *is given by a pair* $(\mathcal{K}, O)$ *with* $\mathcal{K}$ *a* 2*-category and* $O : \mathcal{K}^{\mathrm{op}} \to \mathcal{K}$ *a pseudo functor such that* $O \circ O^{\mathrm{op}} \cong 1_{\mathcal{K}}$.

Our main example is provided by **Cocont**.

Example 6.6.7 *The* 2*-category* **Cocont** *is pseudo involutory with involution given by*

$$\mathbf{Cocont}^{\mathrm{op}} \xrightarrow{\sim} \mathbf{Prof}^{\mathrm{op}} \xrightarrow{(-)^*} \mathbf{Prof} \xrightarrow{\sim} \mathbf{Cocont} \; .$$

Definition 6.6.8 *Let $(\mathcal{K}, O)$ be a pseudo involutory Kcat. Every admissible relational structure $\mathcal{R}$ on $\mathcal{K}$ induces by composition an admissible relational structure $\mathcal{R}^O$ on $\mathcal{K}^{\mathrm{op}}$, with $\mathcal{R}^O(C) \stackrel{\mathrm{def}}{=} \mathcal{R}(OC)$ and $\mathcal{R}^O(f) \stackrel{\mathrm{def}}{=} \mathcal{R}(Of)$ (cf. (6.25)).*

Observe that the relational structure in point 3 of Definition 6.5.6 is obtained from the construction above using the involution of Example 6.6.7.

Proposition 6.6.9 *Let $(\mathcal{K}, O)$ be an involutory Kcat, and let $\mathcal{R}$ be an admissible relational structure on $\mathcal{K}$. Consider pseudo ω**Cat**-functors $T : \mathcal{K}^{\mathrm{op}} \times \mathcal{K} \to \mathcal{K}$ and $T^\#$ such that the diagram*

$$\begin{array}{ccc}
\{\mathcal{K}^{\mathrm{op}} | \mathcal{R}^O\} \times \{\mathcal{K} | \mathcal{R}\} & \xrightarrow{T^\#} & \{\mathcal{K}^{\mathrm{op}} | \mathcal{R}^O\} \times \{\mathcal{K} | \mathcal{R}\} \\
{\scriptstyle U' \times U} \downarrow & & \downarrow {\scriptstyle U' \times U} \\
\mathcal{K}^{\mathrm{op}} \times \mathcal{K} & \xrightarrow[\check{T}]{} & \mathcal{K}^{\mathrm{op}} \times \mathcal{K}
\end{array}$$

*commutes, where U' and U denote forgetful ω**Cat**$_0$-functors. Then, for every free pseudo T-dialgebra*

$$\mathrm{fold} : T(D, D) \simeq D : \mathrm{unfold}$$

there exist (necessarily unique) $\Delta' \in \mathcal{R}(OD)$ and $\Delta \in \mathcal{R}(D)$ such that

$$(\mathrm{unfold}, \mathrm{fold}) : T^\#(\{D | \Delta'\}, \{D | \Delta\}) \simeq (\{D | \Delta'\}, \{D | \Delta\}) : (\mathrm{fold}, \mathrm{unfold})$$

is a free pseudo $T^\#$-algebra.

Proof:[Hint] The proof is as in Proposition 6.6.1 since $\mathcal{R}^O \times \mathcal{R}$ is an admissible relational structure on $\mathcal{K}^{\mathrm{op}} \times \mathcal{K}$. □

The induction and coinduction principles of Proposition 6.6.2 assume now a more general form but are deducible as well as before directly form the universal property of free pseudo dialgebras (cf. Theorem 6.3.6).

Proposition 6.6.10 *In the situation of the above proposition, let*

$$T^\#(\{A' | R'\}, \{A | R\}) = (\{T(A, A') | T'_{\mathcal{R}}(R', R)\}, \{T(A', A) | T_{\mathcal{R}}(R', R)\}) \ .$$

Then, $\Delta' \in \mathcal{R}(OD)$ and $\Delta \in \mathcal{R}(D)$ satisfy the following rules:

- *For $a' : A' \to T(A, A')$, $a : T(A', A) \to A$, $R' \in \mathcal{R}(OA')$, $R \in \mathcal{R}(A)$,*

$$\frac{O(a') : T'_{\mathcal{R}}(R', R) \subset R' \quad a : T_{\mathcal{R}}(R', R) \subset R}{O(\mathrm{coit}(a', a)) : \Delta' \subset R' \quad \mathrm{it}(a', a) : \Delta \subset R}$$

- *For $z' : T(Z, Z') \to Z'$, $z : Z \to T(Z', Z)$, $R' \in \mathcal{R}(OZ')$, $R \in \mathcal{R}(Z)$,*

$$\frac{O(z') : R' \subset T'_{\mathcal{R}}(R', R) \quad z : R \subset T_{\mathcal{R}}(R', R)}{O(\mathrm{it}(z, z')) : R' \subset \Delta' \quad \mathrm{coit}(z, z') : R \subset \Delta}$$

Bisimulations with respect to $T^{\#}$ are now given by pairs of relations.

Definition 6.6.11 *Define a* $T^{\#}$-bisimulation *to be a pair* $(R', R) \in \mathcal{R}(OD) \times \mathcal{R}(D)$ *such that*

$$O(\text{fold}) : R' \subset T'_{\mathcal{R}}(R', R) \quad \textit{and} \quad \text{unfold} : R \subset T_{\mathcal{R}}(R', R)\ .$$

By its defining property the pair (Δ', Δ) is immediately seen to be a $T^{\#}$-bisimulation and because of the coinduction principle is the maximal one.

Proposition 6.6.12 *Under the hypothesis of Proposition 6.6.9, the pair* (Δ', Δ) *is a* $T^{\#}$*-bisimulation and moreover it satisfies the following coinduction property:*

$$(\Delta', \Delta) = \bigvee \{(R', R) \mid (R', R) \textit{ is a } T^{\#}\textit{-bisimulation}\}\ .$$

6.7 Open map bisimulation from coinduction properties

We use the results of the previous two sections to study open map bisimulation in presheaf categories. We consider the relational structures of Definition 6.5.6 that were defined on $\mathbf{Cocont}_M$ and $\mathbf{Cocont}$ and lift the interpretation of (suitable restrictions of) the type grammar of Section 4.5 first to $\{\mathbf{Cocont}_M \,|\, \text{Ext}\}$ and then to $\{\mathbf{Cocont}^{\text{op}} \,|\, \text{Int}\}^* \times \{\mathbf{Cocont} \,|\, \text{Int}\}$.

6.7.1 Extensional relations

Proposition 6.7.1 *The pseudo functors,* $(\mathbb{V} \otimes -)$, $\sum_{i \in I}$, $(-)_\perp$ *restricts to* $\mathbf{Cocont}_M$ *when* $\mathbb{V}$ *is a discrete category.*

Proof: The only thing that is needed to check is that for any ω-colimit and monomorphism preserving functor, $F : \widehat{\mathbb{C}} \to \widehat{\mathbb{D}}$ and any indexed family $(F_i : \widehat{\mathbb{C}_i} \to \widehat{\mathbb{D}_i})_{i \in I}$ of such functors, $1_{\widehat{\mathbb{V}}} \otimes F$, $\sum_{i \in I} F_i$ and $F_\perp$ are monomorphism preserving.

$\underline{1_{\widehat{\mathbb{V}}} \otimes F}$:

$$1_{\widehat{\mathbb{V}}} \otimes F : \widehat{\mathbb{V} \otimes \mathbb{C}} \cong \textstyle\prod_{v \in \mathbb{M}} \widehat{\mathbb{C}} \xrightarrow{\prod_{v \in \mathbb{M}} F} \textstyle\prod_{v \in \mathbb{M}} \widehat{\mathbb{D}} \cong \widehat{\mathbb{V} \otimes \mathbb{D}}$$

is defined by (cf. Chapter 4)

$$\begin{array}{rcl} (1_{\widehat{\mathbb{V}}} \otimes F(\langle X_v \rangle_{v \in \mathbb{M}}) & = & \langle F(X_v) \rangle_{v \in \mathbb{M}} \\ (1_{\widehat{\mathbb{V}}} \otimes F(\langle \alpha_v \rangle_{v \in \mathbb{M}}) & = & \langle F(\alpha_v) \rangle_{v \in \mathbb{M}}\ , \end{array}$$

for $\langle X_v \rangle_{v \in \mathbb{M}} \in \mid \prod_{v \in \mathbb{M}} \widehat{\mathbb{C}} \mid$ and $\langle \alpha_v \rangle_{v \in \mathbb{M}}$ a natural transformation. Hence $(1_{\widehat{\mathbb{V}}} \otimes F)$ preserves monomorphic natural transformations if F does so.

$\underline{\sum_{i \in I} F_i}$: The sum is defined componentwise and hence it trivially preserves monomorphisms.

$\underline{F_\perp}$:

$$F_\perp : \widehat{\mathbb{C}_\perp} \to \widehat{\mathbb{C}_\perp}$$

is defined by $F_\perp(X) \cong \sum_{x\in X(\perp)} \lfloor F(X_{|x}) \rfloor$ on objects. If $\alpha : X \stackrel{\cdot}{\longrightarrow} Y$ in $\widehat{\mathbb{C}_\perp}$ then α is uniquely determined by $\alpha_\perp : X(\perp) \to Y(\perp)$ and a family $(\alpha_{|x} : X_{|x} \stackrel{\cdot}{\longrightarrow} Y_{|\alpha_\perp(x)})_{x\in X(\perp)}$ of natural transformations in $\widehat{\mathbb{C}}$. In particular α is a monomorphism if and only if $\alpha_\perp$ is an injective map and for every $x \in X(\perp)$, $\alpha_{|x}$ is a monomorphic natural transformation. $F_\perp(\alpha)$ is the natural transformation determined by $\alpha_\perp$ and $(F(\alpha_{|x}))_{x\in X(\perp)\cong F_\perp(X)(\perp)}$. Since F preserves monomorphisms, $F_\perp(\alpha)$ is a monomorphism if α is a monomorphism.

□

We consider the following fragment of our main type structure

$$t \quad ::= \quad 0 \mid 1 \mid \textstyle\sum_{i\in I} t_i \mid \mathbb{V} \multimap t \mid \vartheta \mid \mu\vartheta.t \ . \tag{6.26}$$

Previously we provided the semantics to the types t in **Prof**. We know that this was equivalent to giving it in **Cocont** and get mixed-variance pseudo functors

$$(\mathbf{Cocont}^{\mathrm{op}} \times \mathbf{Cocont})^{|\Theta|} \to \mathbf{Cocont} \ . \tag{6.27}$$

All the type constructors in the fragment above do not present any non-trivial situation where the contravariant part play any significant role, In fact the only case where we use contravariancy is in the denotation of $\mathbb{V} \multimap t$ but since $\mathbb{V}$ is assumed to be a discrete category

$$\mathbb{V} \multimap \mathbb{P} \stackrel{\mathrm{def}}{=} \mathbb{V}^{\mathrm{op}} \times \mathbb{P} = \mathbb{V} \times \mathbb{P} \ ,$$

for any category $\mathbb{P}$. Hence we can, for simplicity, discard the contravariant components in (6.27) and assume the interpretation to range over pseudo ω**Cat**-functors

$$[\![\Theta \vdash t]\!] : \mathbf{Cocont}^{|\Theta|} \to \mathbf{Cocont} \ .$$

Following Proposition 6.7.1 above we can actually restrict to

$$[\![\Theta \vdash t]\!] : \mathbf{Cocont}_M^{|\Theta|} \to \mathbf{Cocont}_M \ .$$

We wish to lift this interpretation to act on $\{\mathbf{Cocont}_M \,|\, \mathrm{Ext}\}$, i.e., we want to interpret a type t with free variables in Θ as a pseudo functor

$$\mathcal{E}[\![\Theta \vdash t]\!] : \{\mathbf{Cocont}_M \,|\, \mathrm{Ext}\}^{|\Theta|} \to \{\mathbf{Cocont}_M \,|\, \mathrm{Ext}\} \ .$$

To do so it suffices to describe how the action of taking constants, sum lifting and *discrete* function space extends from $\mathbf{Cocont}_M$ to $\{\mathbf{Cocont}_M \mid \mathrm{Ext}\}$. The inductive types case being taken care by the results of Section 6.6.1, since $\{\mathbf{Cocont}_M \mid \mathrm{Ext}\}$ is pseudo ω**Cat**-algebraically compact being a Kcat (Corollary 6.5.5 and Theorem 6.3.16).

Sums: Consider a presheaf X over $\sum_{i\in I} \mathbb{A}_i$. Its projection $(X)_i$, for $i \in I$, is the presheaf obtained as the restriction of X to $\mathbb{A}_i$. Define

$$\textstyle\sum_{i\in I}\{\mathbb{A}_i\,|\,R_i\} \stackrel{\text{def}}{=} \{\sum_{i\in I}\mathbb{A}_i\,|\,R\}$$

where

$$X\ R\ Y \quad \stackrel{\text{def}}{\iff} \quad \forall i \in I.\ (X)_i\ R_i\ (Y)_i\ .$$

It is easy to check that this extension is well-defined and that

$$(\forall i \in I.\ F_i : R_i \subset S_i) \quad\Rightarrow\quad \sum_{i\in I} F_i : \sum_{i\in I} R_i \subset \sum_{i\in I} S_i\ .$$

Lifting: Consider a presheaf X over $\mathbb{A}_\perp$. We saw in Chapter 5 that it decomposes into a sum

$$X \quad\cong\quad \textstyle\sum_{x\in X(\perp)} \lfloor X_{|x} \rfloor \tag{6.28}$$

where each presheaf $X_{|x}$ in $\widehat{\mathbb{A}}$ is the component subtended from the element x, and $\lfloor - \rfloor$ is the functor that puts a root to a presheaf. In Section 5.1.4 we have also used a transition relation for presheaves. Here, for $X' \in \widehat{\mathbb{A}}$, write

$$X \xrightarrow{\perp} X'$$

when there is $x \in X(\perp)$ such that $X' = X_{|x}$.

The "obvious" way to extend lifting to relations is, given a relation R between presheaves over $\mathbb{A}$ to define $(R)^0_\perp$ a relation between presheaves over $\mathbb{A}_\perp$ by taking: $X\ (R)^0_\perp\ Y$ iff

$$\forall X'.\ X \xrightarrow{\perp} X' \Rightarrow \exists Y'.\ Y \xrightarrow{\perp} Y'\ \&\ X'\ R\ Y',$$
$$\forall Y'.\ Y \xrightarrow{\perp} Y' \Rightarrow \exists X'.\ X \xrightarrow{\perp} X'\ \&\ X'\ R\ Y'\ .$$

But, unfortunately, the relation $(R)^0_\perp$ may fail to satisfy the ω-admissibility requirement (c) in the definition of admissible extensional relations even though R lies in $\mathrm{Ext}(\mathbb{A})$. We thus define $X\ (R)_\perp\ Y$ iff there are ω-chains of monomorphisms $\vec{X}$, $\vec{Y}$ with colimits X and Y respectively for which $\vec{X}_n\ (R)^0_\perp\ \vec{Y}_n$ for all $n \in \omega$. Finally we define

$$(\{\mathbb{A}\,|\,R\})_\perp \stackrel{\text{def}}{=} \{\mathbb{A}_\perp\,|\,(R)_\perp\}\ .$$

Suppose $F : R \subset S$ in $\{\mathbf{Cocont}_M \mid \mathrm{Ext}\}$. Then from F being colimit, and so sum, preserving, it follows that $(F)_\perp : (R)_\perp \subset (S)_\perp$.

Proposition 6.7.2 *If R is an admissible extensional relation on $\mathbb{A}$, then $R_\perp$ is an admissible extensional relation on $\mathbb{A}_\perp$.*

Proof: Clearly the only thing to check is the satisfaction of the ω-admissibility requirement.

Say that two presheaves X, Y over a category $\mathbb{A}$ are compatible, if for every arrow $f : A \to B$ in $\mathbb{A}$, if $x \in X(B) \cap Y(B)$, then $X(f)x = Y(f)x$. On compatible presheaves it is possible to define a union operation, $X \cup Y$, that is the union of sets pointwise.

It is easy to see that $R^0_\perp$ satisfies the following closure property with respect to the union of compatible presheaves.

> If $X\ R^0_\perp\ Y$ and $V\ R^0_\perp\ W$, with X compatible to V and Y compatible with W, then $(X \cup V)\ R^0_\perp (Y \cup W)$.

Let's suppose now that $\vec{X}$, $\vec{Y}$ is a chain of monomorphisms in $\widehat{\mathbb{A}_\perp}$, with colimits X and Y, respectively, for which $\vec{X}_n\ (R)_\perp\ \vec{Y}_n$ for all $n \in \omega$. Without loss of generalities we can assume (since all the natural transformations involved are taken to be monomorphisms) that the chains are given pointwise by inclusions. By definition of $(R)_\perp$, we have that each pair $X_n\ (R)_\perp\ Y_n$ in the chain is generated as the colimit of a chain of monomorphisms (w.l.o.g. pointwise inclusions) of presheaves X_{nm} and Y_{nm} with $X_{nm}\ (R)^0_\perp\ Y_{nm}$, i.e., we have the following "matrix" of natural inclusions:

$$\begin{tikzcd}
X_{00}\ (R)^0_\perp\ Y_{00} \arrow[r] & X_{01}\ (R)^0_\perp\ Y_{01} \arrow[r] & X_{02}\ (R)^0_\perp\ Y_{02} \arrow[r] & \cdots\cdots & X_0\ (R)_\perp\ Y_0 \arrow[d] \\
X_{10}\ (R)^0_\perp\ Y_{10} \arrow[r] & X_{11}\ (R)^0_\perp\ Y_{11} \arrow[r] & X_{12}\ (R)^0_\perp\ Y_{12} \arrow[r] & \cdots\cdots & X_1\ (R)_\perp\ Y_1 \arrow[d] \\
X_{20}\ (R)^0_\perp\ Y_{20} \arrow[r] & X_{21}\ (R)^0_\perp\ Y_{21} \arrow[r] & X_{22}\ (R)^0_\perp\ Y_{22} \arrow[r] & \cdots\cdots & X_2\ (R)_\perp\ Y_2 \arrow[d] \\
\cdots & \cdots & \cdots & \cdots & \cdots \\
 & & & & X?Y
\end{tikzcd}$$

We wish to deduce that

$$X\ R_\perp\ Y\ .$$

Since all the arrows are inclusions, all the presheaves are compatible. It can be easily verified that X is the colimit of the chain of inclusions

$$(\bigcup_{k \le n} X_{kk} \stackrel{\cdot}{\longrightarrow} \bigcup_{k \le n+1} X_{kk})_{n \in \omega}\ ,$$

and similarly for Y. By the closure property, quoted above, for every n,

$$\bigcup_{k \le n} X_{kk}\ (R)^0_\perp\ \bigcup_{k \le n} Y_{kk}$$

and therefore by definition,

$$X\ (R)_\perp\ Y\ .$$

□

It is a known fact that bisimulation does not close at ω for arbitrary trees (as a well known example reported below as Example 6.7.7 shows) and this fact is reflected here by $(R)^0_\perp$ not satisfying the ω-admissibility requirement. Still by restricting to a suitable class of presheaves (analogous to finitely branching trees) we can make $(R)^0_\perp$ coincide with $(R)_\perp$ as well as having open map bisimulation closing at ω.

Definition 6.7.3 (Locally-finite presheaves) *A presheaf X over a small category $\mathbb{C}$ is said to be* locally finite *if, for every object C of $\mathbb{C}$, the set $X(C)$ is finite.*

Locally finite presheaves satisfy the following ω-admissibility property with respect to open map bisimulation.

Lemma 6.7.4 *Let $\mathbb{C}$ be a small category. Let $\vec{X}$ and $\vec{Y}$ be two ω-chains of monomorphisms in $\widehat{\mathbb{C}}$ with colimits X and Y respectively. For locally finite X and Y, if $\vec{X}_n$ and $\vec{Y}_n$ are open-map bisimilar, for all n, then so are X and Y.*

Proof: Without loss of generality one can assume all the monomorphic natural transformations involved in the two chains to be pointwise inclusions of sets. Moreover any span of surjective open maps generates another one that is definable as a sub presheaf of the product of the related presheaves [63, page 68]. Thus from the hypothesis we can assume that for every $n \in \omega$ there exists a sub presheaf of $X_n \times Y_n$, Z_n, such that the projections $X_n \leftarrow Z_n \rightarrow Y_n$ are surjective $\mathbb{C}$-open maps. For every object C of $\mathbb{C}$, define

$$Z(C) = \{(x, y) \in X(C) \times Y(C) \mid \exists^\infty n \in \omega \,.\, (x, y) \in Z_n(C)\} \ .$$

It is immediately seen that this indeed defines a sub presheaf of $X \times Y$. We check now that the two projections onto X and Y are surjective open maps. We look at the projection $\pi : Z \rightarrow X$ onto X. The other case is analogous. We need to check that for every $f : C' \rightarrow C$, the following square is a quasi-pullback:

$$\begin{tikzcd} Z(C) \arrow[r, "\pi_C"] \arrow[d, "Z(f)"'] & X(C) \arrow[d, "X(f)"] \\ Z(C') \arrow[r, "\pi_{C'}"'] & X(C') \end{tikzcd} \quad .$$

Suppose $x \in X(C)$ and $(x', y') \in Z(C')$ are such that $x' = X(f)x$. By assumption for every n, the projection $Z_n \rightarrow X_n$ is open, and by definition of Z, there exist infinitely many n's such that $(x', y') \in Z_n(C')$. For any such n there exists $y_n \in Y_n(C)$ such that $(x, y_n) \in Z_n(C)$, but since $Y(C)$ is finite (and includes all the $Y_n(C)$'s), there must exists a y that appears in infinitely many pairs, i.e., a y such that $(x, y) \in Z(C)$.

Concerning surjectivity, let $x \in X(C)$, then there exists a number n such that for every $m \geq n$, $x \in X_m(C)$. Since. for every $k \in \omega$, $Z_k(C) \rightarrow X_k(C)$ is a surjective map, for every $m \geq n$, there exists y_m such that $(x, y_m) \in Z_m(C)$. But now since $Y(C)$ is finite (and includes all the $Y_k(C)$'s), there must exists a $y \in Y(C)$ that appears in infinitely many pairs, i.e., $(x, y) \in Z(C)$. □

This result generalises to larger cardinals, in the sense that the statement is still valid if, for any $n \in \omega$, one replaces ω-chains with ω_{n+1}-chains and assumes X and Y to be locally of size ω_n. We remark that the assumption that the ω-chains consist of monomorphisms is crucial as the following example due to Glynn Winskel shows; hence our restriction to $\mathbf{Cocont}_M$ when considering extensional relations.

Example 6.7.5 *Let T be the presheaf over the partial order category ω (i.e., the tree) defined by:*

$$T(0) = \{*\},\ T(1) = \{a, b\},\ T(2) = \{c\},\ T(n) = \emptyset \text{ for every } n \geq 3.$$

with $T(1 \leq 2)c = a$ and all the other actions on arrows $(n \leq m)$ uniquely determined by the cardinality of the associated sets. Consider the following two chains

$$T \xrightarrow{1_T} T \xrightarrow{1_T} T \xrightarrow{1_T} \cdots$$

$$T \xrightarrow{f} T \xrightarrow{f} T \xrightarrow{f} \cdots$$

where $f_1(a) = f_1(b) = a$ and $f_n = 1_{T(n)}$for any other n. Being identical, corresponding presheaves in the two chains are bisimilar, but the colimit of the first one is T itself, while the colimit of the second one is the string $T(0) \cong T(1) \cong T(2) \cong \{*\}$ and $T(n) = \emptyset$ for $n \geq 2$ that is clearly not bisimilar to T.

By a proof similar to that of Lemma 6.7.4, we can show that the two relations $(R)^0_\perp$ and $(R)_\perp$ coincide on locally finite presheaves.

Lemma 6.7.6 *Let X, Y be locally finite presheaves over $\mathbb{A}_\perp$. Suppose the ω-chains of monomorphisms $\vec{X}$, $\vec{Y}$ have colimits X and Y respectively. Then,*

$$(\forall n \in \omega.\ \vec{X}_n\ (R)^0_\perp\ \vec{Y}_n) \Rightarrow X\ (R)^0_\perp\ Y\ .$$

Consequently,

$$X\ (R)^0_\perp\ Y\ \Leftrightarrow X\ (R)_\perp\ Y\ .$$

Example 6.7.7 *Consider the synchronisation tree recursively defined by $T = a.T$, i.e., the tree consisting of one single branch of infinite length and whose transitions are all labelled a. For every $n \in \omega$, define a^n to be the tree consisting of one single branch of length n whose transitions are all labelled a. Define $U = \sum_{n \in \omega} a^n$ and $V = T + \sum_{n \in \omega} a^n$. For every k define $U_k = \sum_{n \in \omega} a^{\min(n,k)}$ and $V_k = a^k + \sum_{n \in \omega} a^{\min(n,k)}$. Clearly U is the colimit of the U_k's and V is the colimit of the V_k's. For every k, U_K is bisimilar to V_K, but U is not bisimilar to V since the latter has an infinitely long branch while all the branches of U are finite.*

Discrete function space: A presheaf X over $\mathbb{V} \multimap \mathbb{A}$ corresponds to a functor $\mathbb{V} \to \widehat{\mathbb{A}}$, and we write Xv for the presheaf in $\mathbb{A}$ resulting from the functor's application to $v \in \mathbb{V}$. Define

$$(\mathbb{V} \multimap \{\mathbb{A} \,|\, R\}) \stackrel{\text{def}}{=} \{(\mathbb{V} \multimap \mathbb{A}) \,|\, (\mathbb{V} \multimap R)\}$$

where

$$X\ (\mathbb{V} \multimap R)\ Y \quad \stackrel{\text{def}}{\iff} \quad (\forall v \in \mathbb{V}.\ (Xv)\ R\ (Yv))\ .$$

This extension is well-defined and such that

$$F : R \subset S \Rightarrow (\mathbb{V} \multimap F) : (\mathbb{V} \multimap R) \subset (\mathbb{V} \multimap S)\ .$$

Thus by structural induction any closed type t in the grammar (6.26) is associated with an extensional relation $\approx_t^{\mathrm{Ext}} \in \mathrm{Ext}([\![t]\!])$. Recursive types $\mu\vartheta.t$ are interpreted as parameterised free pseudo algebras in the Kcat $\{\mathbf{Cocont}_M \mid \mathrm{Ext}\}$; specialising the pseudo-colimit construction of the Pseudo Basic Lemma (using Theorem 6.5.4).

The relation $\approx_t^{\mathrm{Ext}}$ coincides with open-map bisimulation on locally finite presheaves.

Theorem 6.7.8 *Let t be a closed type in the grammar (6.26). Let X, Y be locally finite presheaves over $[\![t]\!]$. Then, $X \approx_t^{\mathrm{Ext}} Y$ iff X and Y are open-map bisimilar.*

Proof: The proof proceeds by structural induction on t. Write $OK\{\mathbb{A} \mid S\}$ when a relation $\{\mathbb{A} \mid S\}$ in $\{\mathbf{Cocont}_M \mid \mathrm{Ext}\}$ satisfies the condition that on locally finite presheaves X, Y over $\mathbb{A}$

$$X \; S \; Y \Leftrightarrow X, Y \text{ are open-map bisimilar .}$$

As the induction hypothesis, on type judgement $\vartheta_1, \cdots, \vartheta_k \vdash t$, we take

$$OK\{\mathbb{A}_1 \,|\, S_1\} \,\&\cdots\&\, OK\{\mathbb{A}_k \,|\, S_k\} \Longrightarrow OK([\![\vartheta_1, \cdots, \vartheta_k \vdash t]\!]\{\mathbb{A}_1 \,|\, S_1\} \cdots \{\mathbb{A}_k \,|\, S_k\}) \; .$$

It can be checked that each of the constructions lifting, sum, and discrete function space preserve the OK property on relations. This covers all cases of the induction but for recursive types.

Consider the relation interpreting a recursively-defined type

$$[\![\Theta \vdash \mu\vartheta.t]\!]\{\mathbb{A}_1 \,|\, S_1\} \cdots \{\mathbb{A}_{|\Theta|} \,|\, S_{|\Theta|}\}$$

in the environment where we assume

$$OK\{\mathbb{A}_1 \,|\, S_1\} \,\&\cdots\&\, OK\{\mathbb{A}_{|\Theta|} \,|\, S_{|\Theta|}\} \; .$$

The relation is a pseudo colimit $\{\mathbb{D} \,|\, R\}$ of the ω-chain $\{\mathbb{D}_n \,|\, R_n\}$ where

$$\{\mathbb{D}_0 \,|\, R_0\} \stackrel{\mathrm{def}}{=} \{\mathbf{0} \,|\, \{(\emptyset, \emptyset)\}\}$$

and

$$\{\mathbb{D}_{n+1} \,|\, R_{n+1}\} \stackrel{\mathrm{def}}{=} [\![\Theta, \vartheta \vdash t]\!]\{\mathbb{A}_1 \,|\, S_1\} \cdots \{\mathbb{A}_{|\Theta|} \,|\, S_{|\Theta|}\}\{\mathbb{D}_n \,|\, R_n\} \; .$$

Using the structural induction hypothesis, an induction on n shows that $OK\{\mathbb{D}_n \,|\, R_n\}$ at each stage n. Suppose $X \; R \; Y$. Projecting, we have $\gamma_n X \; R_n \; \gamma_n Y$ at each n. Each $\gamma_n X, \gamma_n Y$ is also locally finite (γ_n being part of a coreflection in $\mathbf{Cocont}_M$). Thus $\gamma_n X, \gamma_n Y$ are open-map bisimilar over $\mathbb{D}_n$. Injecting, we obtain ω-chains of monomorphisms $\langle X_n \rangle$, $\langle Y_n \rangle$ in $\widehat{\mathbb{D}}$ with pseudo colimits X and Y. But maps in $\mathbf{Cocont}_M$ preserve open-map bisimilarity, so X_n and Y_n are open-map bisimilar for each n. We now meet the conditions of Lemma 6.7.4, from which we conclude that X and Y are open-map bisimilar. $\square$

From the above and the results of Section 6.6.1 we obtain the following characterisation.

Corollary 6.7.9 *Let $\vartheta \vdash t$ be a type in the grammar (6.26), and let*

$$\mathcal{E}[\![\vartheta \vdash t]\!] : \{\mathbf{Cocont}_M \,|\, \mathrm{Ext}\} \to \{\mathbf{Cocont}_M \,|\, \mathrm{Ext}\}$$

be its interpretation as above. Then, for presheaves X, Y over $[\![\mu\vartheta.t]\!]$, the following are equivalent:

- $X \approx^{\mathrm{Ext}}_{\mu\vartheta.t} Y$.
- *X and Y are $\mathcal{E}[\![\vartheta \vdash t]\!]$-bisimilar as defined in Definition 6.6.3.*

Thus, for locally finite X and Y, a further equivalent statement is:

- *X and Y are open-map bisimilar.*

The two examples once more

1. Let $\mathbb{P} = \mu\vartheta.T$ where $T\vartheta = \sum_{a \in L}(\vartheta)_\perp$. As we saw (cf. Chapter 5 and Section 6.4.1) synchronisation tress over L corresponds to presheaves over $\mathbb{P}$ and $\approx^{\mathrm{Ext}}_{\mathbb{P}}$ to an ω-admissible version of Park and Milner's strong bisimulation. It specialises to the usual strong bisimulation on locally finite presheaves (i.e. finitely branching trees). Further, by Corollary 6.7.9, a $T^{\#}$-bisimulation between locally finite presheaves is a strong bisimulation. That is, for locally finite presheaves X, Y, whenever $X \; R \; Y$,
$$\forall X'.\ X \xrightarrow{a} X' \Rightarrow \exists Y'.\ Y \xrightarrow{a} Y' \;\&\; X' \; R \; Y',$$
$$\forall Y'.\ Y \xrightarrow{a} Y' \Rightarrow \exists X'.\ X \xrightarrow{a} X' \;\&\; X' \; R \; Y' \,.$$

2. Recall that a domain for value-passing with "late" semantics was obtained as $\mathbb{P} = \mu\vartheta.T$ where now
$$T\vartheta = \vartheta_\perp + \textstyle\sum_{a \in Ch,\ v \in \mathbb{V}} \vartheta_\perp + \sum_{a \in Ch}(\mathbb{V} \multimap \vartheta)_\perp$$
with sums over channels *Ch* and values $\mathbb{V}$. In [138], Winskel restricted the class of processes in the language by requiring recursive definitions to be guarded. Hence any process was denoting a locally finite presheaf. By Corollary 6.7.9, $T^{\#}$-bisimilarity between locally finite presheaves corresponds to open map bisimilarity that it was shown in [138] to be a *late bisimulation on presheaves*, Since all process terms denote locally finite presheaves over $\mathbb{P}$, the relation $\approx^{\mathrm{Ext}}_{\mathbb{P}}$ holds between denotations of closed terms iff they are late-bisimilar in the traditional sense.

REMARK: Our treatment thus coincides with that usually adopted in operational semantics of process languages *provided* we restrict to "finitely branching" processes whose denotations are locally finite presheaves. We expect that we could extend the treatment to "countably branching" processes whose denotations are locally countable presheaves if we generalise the results here from ω-colimits to ω_1-colimits. This would follow the pioneering work on countable nondeterminism described in [103]. Of course an even greater degree of branching would require even larger cardinals.

In [22] we also began to explore the possibility of enlarging our class of models to include colimit completions of a more restricted form, while retaining the same preservation properties of colimit preserving functors with respect to (open map) bisimulation. These restricted forms will in many cases directly provide classes of structures with a bounded degree of branching.

6.7.2 Intensional relations

We take now a larger fragment of the grammar of Section 4.5. We consider in fact the following extension of the grammar in (6.26):

$$t ::= \vartheta \mid \textstyle\sum_{i\in I} t_i \mid t_\perp \mid \mu\vartheta.t \mid t \otimes t' \mid t^* \tag{6.29}$$

obtained by adding tensors and duals, and give an interpretation of these types as pseudo $\omega\mathbf{Cat}$-functors

$$\mathcal{I}[\![\Theta \vdash t]\!] : (\{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\})^{|\Theta|} \to \{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\}\ .$$

To do so we use again the $(\breve{-})$ operator (cf. Sections 6.6.2 and 6.4) to obtain

$$[\![\Theta \breve{\vdash} t]\!] : (\mathbf{Cocont}^{\mathrm{op}} \times \mathbf{Cocont})^{|\Theta|} \longrightarrow \mathbf{Cocont}^{\mathrm{op}} \times \mathbf{Cocont}$$

from

$$[\![\Theta \vdash t]\!] : (\mathbf{Cocont}^{\mathrm{op}} \times \mathbf{Cocont})^{|\Theta|} \longrightarrow \mathbf{Cocont}\ .$$

In order to lift it to the category of relations, again, it suffices to show how sums, lifting, tensor, and the dualizer extend to pseudo $\omega\mathbf{Cat}$-functors.

Sums: For an I-indexed family of relations $\langle R_i \in \mathrm{Int}(\mathbb{A}_i)\rangle$ we define

$$\sum_{i\in I} R_i \in \mathrm{Int}(\sum_{i\in I} \mathbb{A}_i)$$

as follows: A span $X \Leftarrow W \Rightarrow Y$ is in $\sum_{i\in I} R_i$ iff, for every $i \in I$, the restriction

$$(X)_i \Leftarrow (W)_i \Rightarrow (Y)_i$$

is in R_i.

The interpretation of $\sum_{i\in I} t_i$ is done using the pseudo $\omega\mathbf{Cat}$-functor

$$(\{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\})^{|I|} \to \{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\}$$

sending $(\{\mathbb{A}'_i \,|\, R'_i\}, \{\mathbb{A}_i \,|\, R_i\})$ to $(\{\sum_{i\in I} \mathbb{A}'_i \,|\, \sum_{i\in I} R'_i\}, \{\sum_{i\in I} \mathbb{A}_i \,|\, \sum_{i\in I} R_i\})$.

Lifting: Consider presheaves W and X over $\mathbb{A}_\perp$ with decompositions

$$W \cong \sum_{w\in W(\perp)} \lfloor W_{|w} \rfloor \qquad \text{and} \qquad X \cong \sum_{x\in X(\perp)} \lfloor X_{|x} \rfloor\ ,$$

where the $W_{|w}$'s and $X_{|x}$'s are presheaves over $\mathbb{A}$. If $p : W \Rightarrow X$ is a natural transformation, we know that it is uniquely identified by $p_\perp : W(\perp) \to X(\perp)$ and $\langle p_{|w} : W_{|w} \Rightarrow X_{|p_\perp(w)}\rangle_{w \in W(\perp)}$ in $\widehat{\mathbb{A}}$. Then, for $R \in \mathrm{Int}(\mathbb{A})$, we define $(R)_\perp \in \mathrm{Int}(\mathbb{A}_\perp)$ as follows. A span

$$X \stackrel{p}{\Longleftarrow} W \stackrel{q}{\Longrightarrow} Y$$

is in $(R)_\perp$ iff the span $X(\perp) \stackrel{p_\perp}{\longleftarrow} W(\perp) \stackrel{q_\perp}{\longrightarrow} Y(\perp)$ consists of surjections and, for every $w \in W(\perp)$, the span $X_{|p_\perp(w)} \stackrel{p_{|w}}{\Longleftarrow} W_{|w} \stackrel{q_{|w}}{\Longrightarrow} Y_{|q_\perp(w)}$ is in R.

Moreover, for $R \in \mathrm{Int}(\mathbb{A}^{\mathrm{op}})$, we define $(R)^\top \in \mathrm{Int}((\mathbb{A}_\perp)^{\mathrm{op}})$ as follows. A span $X \Leftarrow W \Rightarrow Y$ is in $(R)^\top$ iff, writing l for the canonical inclusion $\mathbb{A} \hookrightarrow \mathbb{A}_\perp$, the span $l^*(X) \Leftarrow l^*(W) \Rightarrow l^*(Y)$ is in R and if, for all $A \in |\mathbb{A}|$, the naturality squares

$$\begin{array}{ccc} X(\perp) & \longleftarrow & W(\perp) \\ \downarrow & & \downarrow \\ X(A) & \longleftarrow & W(A) \end{array} \qquad \begin{array}{ccc} W(\perp) & \longrightarrow & Y(\perp) \\ \downarrow & & \downarrow \\ W(A) & \longrightarrow & W(A) \end{array}$$

are quasi-pullbacks.

Finally, the interpretation of $t_\perp$ uses the pseudo $\omega\mathbf{Cat}$-endofunctor

$$\{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\} \to \{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\}$$

sending $(\{\mathbb{A}' \,|\, R'\}, \{\mathbb{A} \,|\, R\})$ to $(\{\mathbb{A}'_\perp \,|\, (R')^\top\}, \{\mathbb{A}_\perp \,|\, (R)_\perp\})$.

Tensor: Let $R \in \mathrm{Int}(\mathbb{A})$ and $S \in \mathrm{Int}(\mathbb{B})$. A span $X \Leftarrow W \Rightarrow Y$ in $\widehat{\mathbb{A} \otimes \mathbb{B}}$ is defined to be in $R \otimes S$ iff, for every $A \in |\mathbb{A}|$,

$$X(A, -) \Leftarrow W(A, -) \Rightarrow Y(A, -) \text{ is in } S$$

and, for every $B \in |\mathbb{B}|$,

$$X(-, B) \Leftarrow W(-, B) \Rightarrow Y(-, B) \text{ is in } R \quad .$$

The interpretation of $\otimes$ is the pseudo $\omega\mathbf{Cat}$-functor

$$(\{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\})^2 \to \{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\}$$

sending $(\{\mathbb{A}' \,|\, R'\}, \{\mathbb{A} \,|\, R\}), (\{\mathbb{B}' \,|\, S'\}, \{\mathbb{B} \,|\, S\})$ to $(\{\mathbb{A}' \otimes \mathbb{B}' \,|\, R' \otimes S'\}, \{\mathbb{A} \otimes \mathbb{B} \,|\, R \otimes S\})$.
Dualizer: The interpretation of $(-)^*$ is the pseudo $\omega\mathbf{Cat}$-endofunctor

$$\{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\} \to \{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\}$$

sending $(\{\mathbb{A}' \,|\, R'\}, \{\mathbb{A} \,|\, R\})$ to $(\{\mathbb{A}^{\mathrm{op}} \,|\, R\}, \{\mathbb{A}'^{\mathrm{op}} \,|\, R'\})$.

Thus every closed type t in the grammar (6.29) is associated with intensional relations $\approx'^{\mathrm{Int}}_t \in \mathrm{Int}([\![t]\!]^{\mathrm{op}})$ and $\approx^{\mathrm{Int}}_t \in \mathrm{Int}([\![t]\!])$, which using a proof similar to that of Theorem 6.7.8 can be shown to coincide with open-map bisimulation.

Theorem 6.7.10 *Let t be a closed type in the grammar (6.29). Then,*

$$\approx_t'^{\mathrm{Int}} = \mathrm{sOs}_{[\![t]\!]^{\mathrm{op}}} \quad \textit{and} \quad \approx_t^{\mathrm{Int}} = \mathrm{sOs}_{[\![t]\!]}$$

where $\mathrm{sOs}_{\mathbb{C}}$ *denotes the class of surjective open spans in* $\widehat{\mathbb{C}}$.

Corollary 6.7.11 *Let $\vartheta \vdash t$ be a type in the grammar (6.29), and let*

$$\mathcal{I}[\![\vartheta \vdash t]\!] : \{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\} \to \{\mathbf{Cocont}^{\mathrm{op}} \,|\, \mathrm{Int}^*\} \times \{\mathbf{Cocont} \,|\, \mathrm{Int}\}$$

be its interpretation as above. Then,

$$\begin{aligned} \approx_{\mu\vartheta.t}^{\mathrm{Int}} &= \mathrm{sOs}_{[\![\mu\vartheta.t]\!]} \\ &= \bigcup \{R \mid (R', R) \textit{ is a } \mathcal{I}[\![\vartheta \dashv t]\!]\textit{-bisimulation}\} \ . \end{aligned}$$

Strong bisimulation revisited. Using intensional relations we can capture strong bisimulation for *arbitrary* trees in our domain theoretic setting. In fact by taking the equation for synchronisation trees, $\mathbb{P} = \mu\vartheta. \sum_{a \in L} (\vartheta)_\perp$, we have that by Corollary 6.7.11, two trees are connected by a span in $\approx_{\mathbb{P}}^{\mathrm{Int}}$ if and only if they are connected by a span of surjective $\mathbb{P}$-open maps, i.e., if and only if they are strong bisimilar. As far as we know, this is the first domain-theoretic characterisation of strong bisimulation for *arbitrary* trees.

Chapter 7

Presheaf Models for the π-Calculus

In this chapter we go back to concrete examples and address the issue of giving presheaf models for the π-calculus [87, 88]. In contrast with examples of process languages we have seen so far (cf. Chapter 5), the π-calculus via the ability of communicating channel names can express processes whose communication topology changes over time, while computation evolves. In [126, 32, 50] where domain theoretic models, i.e., based on partial orders, were given this characteristic has been tackled semantically by indexing the category of domains with a category of "finite sets of channel names", $\mathcal{I}$. Here we take a similar approach and index **Prof** with the same $\mathcal{I}$. As a result process terms will be interpreted by indexed families of presheaves. Open map bisimulation at a fibre will correspond to bisimulation in the language, while open map bisimulation at each fibre will correspond to the largest congruence included in bisimilarity.

We shall mainly deal with a model for the π-calculus with late bisimulation, but shall also show how one gets one for the early-bisimulation, something the domain-theoretic model [126, 32] does not seem to be able to capture, and how an arrow between models, mapping the late interpretation into the early one arises in this context.

There exist different variants of the π-calculus and people often concentrate on fragments of it [117, 18]. We shall consider the full calculus as presented in [87, 84]. This means that all the possible fragments one might wish to consider can be modelled directly by our model, still in these cases it might be possible to describe simpler domain equations that provide models which are tailored for the special fragment one is considering and that come together with an "embedding" which maps the restricted model into the general one.

7.1 The π-calculus

The version of the π-calculus we use is entirely standard. We summarise it only very briefly here: for discussion and further detail see the original papers [87, 84]. Processes have the following syntax

$$P \ ::= \ \overline{x}y.P \ \mid \ x(y).P \ \mid \ \nu x\, P \ \mid \ [x{=}y]P \ \mid \ 0 \ \mid \ P+P \ \mid \ P\,|\,P \ \mid \ !P$$

with x and y ranging over some infinite supply of *names*. Note that we include the match operator $[x{=}y]P$, unguarded sum and unguarded replication $!P$. This selection is fairly arbitrary: our model copes equally well with mismatch $[x{\neq}y]P$ and processes defined by recursion, guarded or unguarded. Similarly, it makes no difference if we restrict to one of the popular subsets, such as the asynchronous π-calculus [18].

To simplify presentation we identify processes up to a *structural congruence*, the smallest congruence relation satisfying

$$\begin{array}{cccc}
[x=x]P \equiv P & !P \equiv P\,|\,!P & \begin{array}{c} x(y).P \equiv x(z).P[z/y] \\ \nu y\, P \equiv \nu z\, P[z/y] \end{array} & \begin{array}{c} z \notin \mathit{fn}(P) \\ z \notin \mathit{fn}(P) \end{array} \\
P+0 \equiv P & P+Q \equiv Q+P & (P+Q)+R \equiv P+(Q+R) & \\
P\,|\,0 \equiv P & P\,|\,Q \equiv Q\,|\,P & (P\,|\,Q)\,|\,R \equiv P\,|(Q\,|\,R). &
\end{array}$$

Here $P[z/y]$ denotes capture-avoiding substitution — which may of course require in turn the α-conversion of subexpressions. This equivalence is not as aggressive as the structural congruence of, say, Definition 3.1 in [83], which allows name restriction $\nu x(-)$ to change its scope. Nevertheless it cuts down the operational rules we shall need, with none at all for matching and replication. All this is to some degree a matter of taste: if we treat process terms as concrete syntax, with no structural identification, the model is still valid. Indeed "completeness" of the model then allows us to read off the fact that α-conversion, commuting '+' and so forth all respect bisimilarity (replacing, for example, the proofs of Theorems 1 to 9 in [88, §3]).

The operational semantics of processes are given by transitions of four kinds: internal or 'silent' action τ, input $x(y)$, free output $\overline{x}y$ and bound output $\overline{x}(y)$. We denote a general transition by α, and define its free and bound names thus:

$$\begin{array}{lll}
\mathit{fn}(\tau) = \emptyset & \mathit{fn}(\overline{x}y) = \{x,y\} & \mathit{fn}(x(y)) = \mathit{fn}(\overline{x}(y)) = \{x\} \\
\mathit{bn}(\tau) = \emptyset & \mathit{bn}(\overline{x}y) = \emptyset & \mathit{bn}(x(y)) = \mathit{bn}(\overline{x}(y)) = \{y\}.
\end{array}$$

The transitions that a process may perform are given inductively by the rules in Figure 7.1. This is a *late* semantics, in that input substitution happens in the (COM) rule when communication actually occurs, rather than at (IN) (cf. Section 5.2). The chief difference between these rules and Table 2 of [88] is that we let structural congruence do some of the work. Thus there are no symmetric forms for the four right-hand rules, and *sometimes processes must be α-converted before they can interact.* Of course the possible transitions derived are exactly the same as with the original definitions.

Definition 7.1.1 *A symmetric relation $\mathcal{S}$ between processes is a* bisimulation *if for every $(P,Q) \in \mathcal{S}$ the following conditions hold.*

- *For $\alpha = \tau, \overline{x}y, \overline{x}(y)$, if $P \xrightarrow{\alpha} P'$ then there exists Q' such that $Q \xrightarrow{\alpha} Q'$ and $(P',Q') \in \mathcal{S}$.*

OUT	$\overline{x}y.P \xrightarrow{\overline{x}y} P$	SUM	$\dfrac{P \xrightarrow{\alpha} P'}{P+Q \xrightarrow{\alpha} P'}$	
IN	$x(y).P \xrightarrow{x(y)} P$	PAR	$\dfrac{P \xrightarrow{\alpha} P'}{P \mid Q \xrightarrow{\alpha} P' \mid Q}$	$bn(\alpha) \cap fn(Q) = \emptyset$
RES	$\dfrac{P \xrightarrow{\alpha} P'}{\nu x\, P \xrightarrow{\alpha} \nu x\, P'}$ $x \notin fn(\alpha)$	COM	$\dfrac{P \xrightarrow{x(y)} P' \quad Q \xrightarrow{\overline{x}z} Q'}{P \mid Q \xrightarrow{\tau} P'[z/y] \mid Q'}$	
OPEN	$\dfrac{P \xrightarrow{\overline{x}y} P'}{\nu y\, P \xrightarrow{\overline{x}(y)} P'}$ $x \neq y$	CLOSE	$\dfrac{P \xrightarrow{x(y)} P' \quad Q \xrightarrow{\overline{x}(y)} Q'}{P \mid Q \xrightarrow{\tau} \nu y(P' \mid Q')}$	

Figure 7.1: Transition rules for π-calculus processes

- *If* $P \xrightarrow{x(y)} P'$ *then there exists* Q' *such that* $Q \xrightarrow{x(y)} Q'$ *and for any name* z, $(P'[z/y], Q'[z/y]) \in \mathcal{S}$.

Two processes P *and* Q *are (strong, late)* bisimilar, $P \dot{\sim} Q$ *if there is some bisimulation relating them.*

To check the second condition in the definition above it is only necessary that z ranges over the free names of P and Q, and one fresh name. The bisimilarity relation is *strong*, in that τ-actions must match, and *late*, in that input actions must match before the transmitted value is known.

Bisimilarity is preserved by all process constructors except for input prefix $x(y).P$. This is because bisimilarity assumes all names are distinct, while the substitution that happens on input can cause names to become identified. The following definition then identifies the largest congruence included in bisimilarity.

Definition 7.1.2 *Define two processes* P, Q *to be* equivalent $P \sim Q$ *if they are bisimilar under all possible name substitutions.*

7.2 Indexing Prof

As a π-calculus process evolves, the ambient set of channel names, that is the set of names publicly available might change. In fact names that were distinct at a certain stage might get identified and new names can be created and then made public. To take account of this fact we index **Prof** with a category of name sets.

Definition 7.2.1 *Define* $\mathcal{I}$ *to be the (essentially small) category of finite sets and injective functions.*

NOTATION: If s is a finite set, we shall write $s+1$ for the "generic" set obtained by adding a new element to the set s. This new element will normally be indicated by $*_s$, omitting the subscript when it is not necessary.

Similarly if $i : s \to s'$ is an arrow in $\mathcal{I}$, we write $f+1 : s+1 \to s'+1$ for the function that acts like f on the element of s and that maps $*_s$ onto $*_{s'}$.

Finally, if $i : s \to s'$ is an arrow in $\mathcal{I}$ and $y \notin Im\, i$, we write the copair $[i, y]$ for the injective function $s+1 \to s'$ acting like i on the elements of s and mapping $*_s$ onto y.

Definition 7.2.2 *Define* $\mathbf{Prof}^{\mathcal{I}}$ *to be the bicategory of pseudo functors (from* $\mathcal{I}$ *to* **Prof***), pseudo natural transformations and modifications.*

Recall that to give a pseudo functor $F : \mathcal{I} \to \mathbf{Prof}$ is to give an indexed family $(F(s))_{s\in|\mathcal{I}|}$ of small categories together with coherent families of profunctors

$$(F(i) : F(s) \multimap F(s'))_{i:s\to s'} \ .$$

Remark: To shorten the presentation effort and building on the experience of the previous chapters, we shall intentionally gloss over many of the bicategorical, i.e., coherence, details in this chapter. We shall often talk of functors, natural transformation and commutative diagrams when instead, to be precise, we should talk of pseudo functors, pseudo natural transformation and diagrams commuting up to isomorphism. Similarly when giving most of the definitions we will not bother with an explicit checking that the necessary coherence conditions are met, since these will always be enforced by the universal property of left Kan extensions.

As usual with functor categories, much (but not all) of the structure of **Prof** extends to $\mathbf{Prof}^{\mathcal{I}}$, In particular, $+$, $\times$, $\otimes$, $(-)^*$ extend pointwise. Still this does not imply that the symmetric monoidal closed structure of **Prof** lifts to $\mathbf{Prof}^{\mathcal{I}}$. In fact this is an open question for us since, for example, we do not know whether **Prof** is closed under all small pseudo colimits (or equivalently pseudo limits), that is a sufficient condition to imply that the closed structure lifts to $\mathbf{Prof}^{\mathcal{I}}$ too [65]. Anyway there exist linear "function spaces" for special objects and a *Yoneda-like* lemma[1] and this is all we need to model the π-calculus.

Definition 7.2.3 *Let* $Y : \mathcal{I} \to \mathbf{Prof}^{\mathcal{I}}$ *be the pseudo functor that takes* s *to* $\mathcal{I}(s, -)$*, where* $\mathcal{I}(s, -)$ *is the pseudo functor that return to any* s' *the set* $\mathcal{I}(s, s')$ *regarded as a discrete category. Both the action of* Y *and of* $\mathcal{I}(s, -)$ *on arrows is by composition.*

Lemma 7.2.4 (Yoneda-like) *Let* $\mathbb{A} : \mathcal{I} \to \mathbf{Prof}$ *be a pseudo functor, then for every finite set* s*,*

$$\widehat{\mathbb{A}(s)} \simeq \mathbf{Prof}^{\mathcal{I}}(Ys, \mathbb{A}) \ .$$

Proof: Recall that $\mathbf{Prof}^{\mathcal{I}}(Ys, \mathbb{A})$ is the category of pseudo natural transformation from Ys to $\mathbb{A}$ and modifications. This means that an object of $\mathbf{Prof}^{\mathcal{I}}(Ys, \mathbb{A})$ is given by a

[1] Cf. Yoneda lemma in the enriched setting [65].

family of squares:

$$\begin{tikzcd} \mathcal{I}(s,s') \arrow[r, "\alpha_{s'}", "\shortmid"{marking}] \arrow[d, "i\circ-"', "\shortmid"{marking}] \arrow[dr, phantom, "\overset{\alpha_i}{\cong}"] & \mathbb{A}(s') \arrow[d, "\mathbb{A}(i)", "\shortmid"{marking}] \\ \mathcal{I}(s,s'') \arrow[r, "\alpha_{s''}"', "\shortmid"{marking}] & \mathbb{A}(s') \end{tikzcd} .$$

satisfying the usual coherence conditions [127]. Since $\mathcal{I}(s,s')$ is a discrete category, $\alpha_{s'}$ is uniquely determined by a family $(X_i)_{i\in\mathcal{I}(s,s')}$ of presheaves of $\widehat{A(s')}$. Moreover since there is an isomorphism

$$\begin{tikzcd} \mathcal{I}(s,s) \arrow[r, "\alpha_{s}", "\shortmid"{marking}] \arrow[d, "i\circ-"', "\shortmid"{marking}] \arrow[dr, phantom, "\overset{\alpha_i}{\cong}"] & \mathbb{A}(s) \arrow[d, "\mathbb{A}(i)", "\shortmid"{marking}] \\ \mathcal{I}(s,s') \arrow[r, "\alpha_{s'}"', "\shortmid"{marking}] & \mathbb{A}(s') \end{tikzcd} ,$$

for any $i : s \to s'$, one has that

$$X_i \cong \mathbb{A}(i)(Y_{1_s}) ,$$

for (Y_j) the family identified by α_s. Hence, up to isomorphism α is uniquely determined by $\alpha_s(1_s)$, i.e., by the choice of a presheaf over $\mathbb{A}(s)$. Similarly a modification, i.e., a coherent family of natural transformations,

$$\varphi_{s'} : \alpha_{s'} \stackrel{\cdot}{\longrightarrow} \beta_{s'}$$

is uniquely determined by $(\varphi_s)_{1_s}$. It is not difficult to verify that the functor mapping any pseudo natural transformation α to $\alpha_s(1_s)$ and any modification φ to $(\varphi_s)_{1_s}$ is an equivalence of categories

$$\mathbf{Prof}^{\mathcal{I}}(\mathcal{I}(s,-),\mathbb{A}) \simeq \widehat{\mathbb{A}(s)} .$$

$\square$

Because of the Lemma 7.2.4, we always have (cf. [77]) a "candidate" for a function space $\mathbb{A} \multimap \mathbb{B}$ given by

$$\text{“}(\widehat{\mathbb{A} \multimap \mathbb{B}})(s)\text{”} \;\cong\; \mathbf{Prof}^{\mathcal{I}}(\mathcal{I}(s,-)\otimes\mathbb{A},\mathbb{B}) ,$$

but this is only meaningful if we can exhibit it as an actual presheaf. Conveniently this is the case for the kind of function space we shall need.

Definition 7.2.5 (The object of names) *Define* $\mathbb{N} : \mathcal{I} \to \mathbf{Prof}$ *to be the functor,*

$$\begin{array}{rcl} \mathbb{N}(s) & = & s \textit{ regarded as a discrete category} \\ \mathbb{N}(i) & = & \mathrm{y}_{s'}\, i : s \to\!\!\!\!\shortmid\;\, s' , \end{array}$$

i.e., $\mathbb{N}$ *is given by the chain of embeddings*

$$\mathcal{I} \hookrightarrow \mathbf{Set} \hookrightarrow \mathbf{Cat} \hookrightarrow \mathbf{Prof} .$$

It turns out that when $\mathbb{N}$ is involved, the expression $\mathbb{N} \multimap \mathbb{A}$ does indeed makes sense, i.e.,

$$\mathbf{Prof}^{\mathcal{I}}(\mathcal{I}(s,-) \otimes \mathbb{N}, \mathbb{A})$$

is expressible as a presheaf category.

Definition 7.2.6 *For any pseudo functor $\mathbb{A} : \mathcal{I} \to$ **Prof**, define $\mathbb{N} \multimap \mathbb{A}$, for $x \in s$, $i : s \to s'$ in $\mathcal{I}$, $A \in |\mathbb{A}(s)|$ and $\overline{A} \in |\mathbb{A}(s+1)|$ by:*

$$\begin{array}{rcl} (\mathbb{N} \multimap \mathbb{A})(s) & = & s \times \mathbb{A}(s) + \mathbb{A}(s+1) \\ ((\mathbb{N} \multimap \mathbb{A})(i)(x, A))(x', A') & = & \begin{cases} (\mathbb{A}(i)A)(A') & \text{if } x' = i(x) \\ \emptyset & \text{otherwise} \end{cases} \\ ((\mathbb{N} \multimap \mathbb{A})(i)(\overline{A}))(x', A') & = & \begin{cases} (\mathbb{A}([i, x'])A)(A') & \text{if } x' \in s' - Im\, i \\ (\mathbb{A}(i+1)A)(A') & \text{if } x' = *_{s'} \\ \emptyset & \text{otherwise} \end{cases} \end{array}$$

where $(x', A') \in s \times \mathbb{A}(s) + \mathbb{A}(s+1)$ with the objects of $\mathbb{A}(s+1)$ in the sum represented by pairs $(*_{s'}, A')$.

Proposition 7.2.7 *For any pseudo functor $\mathbb{A} : \mathcal{I} \to$ **Prof**, for any finite set s,*

$$\widehat{\mathbb{N} \multimap \mathbb{A}(s)} \simeq \mathbf{Prof}^{\mathcal{I}}(\mathcal{I}(s,-) \otimes \mathbb{N}, \mathbb{A}) \ .$$

Proof: Observe first of all that

$$\begin{array}{rcl} \widehat{\mathbb{N} \multimap \mathbb{A}(s)} & = & \widehat{s \times \mathbb{A}(s) + \mathbb{A}(s+1)} \\ & \cong & \widehat{s \times \mathbb{A}(s)} \times \widehat{\mathbb{A}(s+1)} \\ & \cong & (\prod_{x \in s} \widehat{\mathbb{A}(s)}) \times \widehat{\mathbb{A}(s+1)} \ . \end{array} \tag{7.1}$$

This means that a presheaf over $(\mathbb{N} \multimap \mathbb{A})(s)$ is given by a family of presheaves over $\mathbb{A}(s)$ indexed by s plus a presheaf over $\mathbb{A}(s+1)$. Similarly an arrow between two presheaves in $(\mathbb{N} \multimap \mathbb{A})(s)$ is given by a family of natural transformations $(\alpha_x)_{x\in s}$ in $\widehat{\mathbb{A}(s)}$ plus a natural transformation β in $\widehat{\mathbb{A}(s+1)}$. Let's now look at how much information is needed in order to identify an object, F, of $\mathbf{Prof}^{\mathcal{I}}(\mathcal{I}(s,-) \otimes \mathbb{N}, \mathbb{A})$. By definition F is given by a family of squares

$$\begin{tikzcd} \mathcal{I}(s,s') \times s' \arrow[r, "F_{s'}", "|" marking] \arrow[d, "(i\circ -)\times i"', "-" marking] \arrow[dr, phantom, "\overset{F_i}{\cong}"] & \mathbb{A}(s') \arrow[d, "\mathbb{A}(i)", "-" marking] \\ \mathcal{I}(s,s'') \times s'' \arrow[r, "F_{s''}"', "|" marking] & \mathbb{A}(s'') \ , \end{tikzcd}$$

satisfying the usual coherence conditions [127]. Hence for any $i : s \to s'$ and $y = i(x)$ for a necessarily unique x,

$$F_{s'}(i, y) \cong \mathbb{A}(i)(F_s(1_s, x)) \ ;$$

while if $y \notin Im\, i$,

$$F(s')(i,y) \cong \mathbb{A}([i,y])(F_{s+1}(e,*_s)) \ ,$$

for $e : s \hookrightarrow s+1$ the inclusion of s into $s+1$.

In other words, $F(s')$ is uniquely determined (up to isomorphism) by

$$F(s)(1_s,-) : s \multimap\!\!\!\!\to \mathbb{A}(s) \qquad \text{and} \qquad F(s+1)(e,*_s) \in \mid \widehat{\mathbb{A}(s+1)} \mid \ .$$

It is not difficult now to use these observations to provide an equivalence

$$(\prod_{x \in s} \widehat{\mathbb{A}(s)}) \times \widehat{\mathbb{A}(s+1)} \simeq \mathbf{Prof}^{\mathcal{I}}(\mathcal{I}(s,-) \otimes \mathbb{N}, \mathbb{A}) \ .$$

□

NOTATION: Objects of $(\mathbb{N} \multimap \mathbb{A})(s)$ can be seen, rather loosely, as elements of the graph of a function. Thus we write an object in the $s \times \mathbb{A}(s)$-component of (7.1) as $(x \mapsto A)$ for name $x \in s$ and $A \in \mid \mathbb{A}(s) \mid$. An object in the $\mathbb{A}(s+1)$-component we write as $(* \mapsto A')$ for $A' \in \mid \mathbb{A}(s+1) \mid$. In a similar spirit we can inject a presheaf $X \in \widehat{\mathbb{A}(s)}$ into the left x-component as $(x \mapsto X)$, and a presheaf $Y \in \widehat{\mathbb{A}(s+1)}$ into the right component as $(* \mapsto Y)$.

Using the above notation, the action of $(N \multimap \mathbb{A})(i)$ can be written as

$$(x \mapsto A) \ \mapsto \ (i(x) \mapsto \mathbb{A}(i)(A)) \,, \tag{7.2}$$

on the component $s \times \mathbb{A}(s)$ and

$$(*_s \mapsto A') \ \mapsto \ \sum_{y \notin Im(i)} (y \mapsto \mathbb{A}[i,y](A')) + (*_{s'} \mapsto \mathbb{A}(i+1)(A')) \ . \tag{7.3}$$

on the component $\mathbb{A}(s+1)$.

With this notation the action of $(\mathbb{N} \multimap \mathbb{A})(i)_!$ can be characterised as follows.

Lemma 7.2.8 *Let $i : s \to s'$ be an injective function between finite sets. Let*

$$X = \sum_{x \in s} (x \mapsto X^x) + (* \mapsto X^*)$$

be a presheaf over $(\mathbb{N} \multimap A)(s)$, then $(\mathbb{N} \multimap \mathbb{A})(i)_!(X)$ is isomorphic to

$$\sum_{i(x) \in s'} (i(x) \mapsto \mathbb{A}(i)_!(X^x) + \sum_{w \notin Im\, i} (w \mapsto \mathbb{A}([i,w])_!(X^*)) + (* \mapsto \mathbb{A}(i+1)_!(X^*)) \ .$$

Proof: The proof of the lemma is given by the following coend calculation:

$$\begin{aligned}
(\mathbb{N} \multimap \mathbb{A})(i)_!(X) \ &= \ \int^R X(R) \,.\, (\mathbb{N} \multimap \mathbb{A})(i)(R) \\
&\cong \ \sum_{x \in s} (\int^P X^x(P) \,.\, (i(x) \mapsto \mathbb{A}(i)P))
\end{aligned}$$

$$
\begin{aligned}
& + \int^{Q} X^{*}(Q) \,.\, (\mathbb{N} \multimap \mathbb{A})(i)(* \mapsto Q) \\
& \text{(where we have considered all possible “shapes” that } R \text{ can have)} \\
\cong\ & \sum_{x \in s} i(x) \mapsto (\int^{P} X^{x}(P) \,.\, \mathbb{A}(i)(P)) \\
& + \int^{Q} X^{*}(Q) \,.\, (\sum_{w \notin Im\, i} (w \mapsto \mathbb{A}([i,w])(Q)) + (* \mapsto \mathbb{A}(i+1)(Q))) \\
\cong\ & \sum_{x \in s} (i(x) \mapsto \mathbb{A}(i)_{!}(X^{x})) \\
& + \sum_{w \notin Im\, i} (w \mapsto \int^{Q} X^{*}(Q) \,.\, \mathbb{A}([i,w])(Q)) \\
& + (* \mapsto \int^{Q} X^{*}(Q) \,.\, \mathbb{A}(i+1)(Q)) \\
\cong\ & \sum_{i(x) \in s'} (i(x) \mapsto \mathbb{A}(i)_{!}(X^{x}) \\
& + \sum_{w \notin Im\, i} (w \mapsto \mathbb{A}([i,w])_{!}(X^{*})) + (* \mapsto \mathbb{A}(i+1)_{!}(X^{*})) \ .
\end{aligned}
$$

□

7.2.1 Creation of new names

To handle the creation of new names we shall use the following construction in $\mathbf{Prof}^{\mathcal{I}}$ (cf. [32, 126]);

$$\delta : \mathbf{Prof}^{\mathcal{I}} \longrightarrow \mathbf{Prof}^{\mathcal{I}}$$

is the functor that takes $\mathbb{A}$ to $\mathbb{A}(-+1)$, i.e.,

$$
\begin{aligned}
\delta\mathbb{A}(s) &= \mathbb{A}(s+1) \\
\delta\mathbb{A}(i) &= \mathbb{A}(i+1) \ .
\end{aligned}
$$

The action on pseudo natural transformations and modifications is trivial since it simply rearranges the indexing structure, e.g., for $F : \mathbb{A} \to \mathbb{B}$,

$$
\begin{aligned}
(\delta F)_{s} &= F_{s+1} \\
(\delta F)_{i} &= F_{i+1} \ .
\end{aligned}
$$

In [32] the analogous construction that they use is presented in terms of computational monads [90], while Stark [126] insists on its “universality” and presents it as a form of function space $\mathbb{N} \to \mathbb{A}$ arising from a Day-like [28] construction on the monoidal structure of $\mathcal{I}$ given by the disjoint union of sets. Regarded in this way, the δ represents functions that will only accept a new name as input.

7.2.2 A tensor of presheaves

We analyse now in some detail the family of bifunctors

$$w^*_{\mathbb{P},\mathbb{Q}} : \widehat{\mathbb{P}} \times \widehat{\mathbb{Q}} \longrightarrow \widehat{\mathbb{P} \times \mathbb{Q}} \ ,$$

an instance of which we already met in Chapter 5 (Section 5.1.5). Its action is by definition given by (omitting the sub indices $\mathbb{P}$ and $\mathbb{Q}$)

$$\begin{aligned} w^*(X,Y)(P,Q) &= X(P) \times Y(Q) \\ w^*(f,g)_{P,Q} &= f_P \times g_Q \ , \end{aligned}$$

and categorically, as an easy calculation shows, it can be seen as a right adjoint to the colimit preserving functor obtained by Kan extending the profunctor

$$w = \langle \pi_{\mathbb{P}}, \pi_{\mathbb{Q}} \rangle : \mathbb{P} \times \mathbb{Q} \longrightarrow\!\!\!\!\!\!\!+\ \mathbb{P} + \mathbb{Q} \ ,$$

that is defined on objects by

$$w(P,Q) = \mathrm{y}_{\mathbb{P}+\mathbb{Q}}(i_{\mathbb{P}}(P)) + \mathrm{y}_{\mathbb{P}+\mathbb{Q}}(i_{\mathbb{Q}}(Q)) \ ,$$

where $i_{\mathbb{P}} : \mathbb{P} \to \mathbb{P} + \mathbb{Q} \leftarrow \mathbb{Q} : i_{\mathbb{Q}}$ are the inclusion functors; similarly one express the action of w on morphisms.

Moreover w^* preserves colimits on each argument (though not on both arguments at the same time).

Proposition 7.2.9 *Let $\mathbb{P}$ and $\mathbb{Q}$ be two small categories, the functor*

$$w^* : \widehat{\mathbb{P}} \times \widehat{\mathbb{Q}} \longrightarrow \widehat{\mathbb{P} \times \mathbb{Q}}$$

defined as above preserves colimits in each argument, i.e., if $Y \in \mid \widehat{\mathbb{Q}} \mid$, the functor w^*_Y defined by

$$\widehat{\mathbb{P}} \xrightarrow{(-,Y)} \widehat{\mathbb{P}} \times \widehat{\mathbb{Q}} \xrightarrow{w^*} \widehat{\mathbb{P} \times \mathbb{Q}}$$

preserves colimits in $\widehat{\mathbb{P}}$, and similarly, for every $X \in \mid \widehat{\mathbb{P}} \mid$, the functor w^*_X preserves colimits in $\widehat{\mathbb{Q}}$.

Proof: To prove it, it is enough to show that for every pair (P,Q) and $X \in \mid \widehat{\mathbb{P}} \mid$,

$$w^*_Y(X)(P,Q) = X(P) \times Y(Q) \cong \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(w^*_Y \mathrm{y}_{\mathbb{P}})(X)(P,Q) \ .$$

That is, since colimit preserving functors between presheaf categories correspond to left Kan extension along their Yoneda embeddings, we want to show that w^*_Y is the left Kan extension of its restriction to the representables. We express left Kan extensions using the coend formula and use preservation of coends by product [56] and the Density Formula of Section 4.1:

$$\mathrm{Lan}_{\mathrm{y}_{\mathbb{P}}}(w^*_Y \mathrm{y}_{\mathbb{P}})(X)(P,Q) \ \cong \ \int^{P'} X(P') \times \mathrm{y}_{\mathbb{P}}(P)(P') \times Y(Q)$$

$$\begin{aligned} &\cong (\int^{P'} X(P') \times \mathrm{y}_{\mathbb{P}}(P)(P')) \times Y(Q) \\ &\cong X(P) \times Y(Q) \ . \end{aligned}$$

□

Since colimit preserving functors preserve open maps (Corollary 4.6.6), as a consequence of the proposition above we have the following corollary:

Corollary 7.2.10 *Let $\mathbb{P}$ and $\mathbb{Q}$ be two small categories, the functor*

$$w^* : \widehat{\mathbb{P}} \times \widehat{\mathbb{Q}} \longrightarrow \widehat{\mathbb{P} \times \mathbb{Q}}$$

defined as above preserves (surjective) open maps, i.e., if $f : X \to Y$ and $g : W \to Z$ are two (surjective) open maps in $\widehat{\mathbb{P}}$ and $\widehat{\mathbb{Q}}$ respectively, then $w^*(f, g)$ is a (surjective) $\mathbb{P} \times \mathbb{Q}$-open map.

Observe moreover that the action of w^* can be derived from the monoidal structure of **Prof**. In fact we have that the following diagram commutes

$$\begin{tikzcd} \widehat{\mathbb{P}} \times \widehat{\mathbb{Q}} \arrow[rr, "w^*"] \arrow[dd, "\cong"'] & & \widehat{\mathbb{P} \times \mathbb{Q}} \\ & & \mathbf{Prof}(\mathbf{1}, \mathbb{P} \times \mathbb{Q}) \arrow[u, "\cong"'] \\ \mathbf{Prof}(\mathbf{1}, \mathbb{P}) \times \mathbf{Prof}(\mathbf{1}, \mathbb{Q}) \arrow[rr, "\otimes"'] & & \mathbf{Prof}(\mathbf{1} \times \mathbf{1}, \mathbb{P} \times \mathbb{Q}) \arrow[u, "\cong"'] \end{tikzcd} \ .$$

Since w^* can be realised in terms of the monoidal structure of **Prof**, it lifts pointwise to $\mathbf{Prof}^{\mathcal{I}}$.

Finally observe that w^* arises as a left Kan extension as well.

Proposition 7.2.11 *Let $\mathbb{P}$ and $\mathbb{Q}$ be two small categories, then*

$$w^*_{\mathbb{P},\mathbb{Q}} = \mathrm{Lan}_{w_{\mathbb{P},\mathbb{Q}}}(\mathrm{y}_{\mathbb{P}\times\mathbb{Q}}) \ .$$

7.3 The equation

We derive an indexed family of π-calculus path object $\mathbb{P}$ in $\mathbf{Prof}^{\mathcal{I}}$ by solving the following equations:

$$\begin{aligned} \mathbb{P} &= \mathbb{P}_\perp + \mathit{Out} + \mathit{In} \\ \mathit{Out} &= (\mathbb{N} \otimes \mathbb{N} \otimes \mathbb{P}_\perp) + (\mathbb{N} \otimes (\delta\mathbb{P})_\perp) && (7.4) \\ \mathit{In} &= \mathbb{N} \otimes (\mathbb{N} \multimap \mathbb{P})_\perp && (7.5) \end{aligned}$$

Unfolding, the four components of $\mathbb{P}$ represent silent action, free output, bound output and input respectively. We give the solution to this equation in three stages: first we describe recursively at each set s the corresponding path category $\mathbb{P}(s)$; then we specify inductively the profunctor arrow that connects $\mathbb{P}(s)$ to $\mathbb{P}(s')$ for any injective function $i : s \to s'$ and finally we describe, again inductively, the coherence isomorphisms.

From the descriptions of the constructors δ and $\multimap$, we can think of the family $\mathbb{P}(-)$ as being recursively described by

$$\mathbb{P}(s) \;=\; \mathbb{P}(s)_\perp + s \times s \times \mathbb{P}(s)_\perp + s \times \mathbb{P}(s+1)_\perp + s \times (s \times \mathbb{P}(s) + \mathbb{P}(s+1))_\perp \,.$$

The category $\mathbb{P}(s)$ is thus a poset, in fact a forest of trees.

- There are four kinds of root: $\tau.$, $x!y.$, $x!{*}.$ and $x?$ for any $x, y \in s$.
- Above these in the order relation we find respectively: $\tau.p$, $x!y.p$, $x!{*}.p'$, $x?(y{\mapsto}p)$ and $x?({*}{\mapsto}p')$, where the last two lie above $x?$ and where p is an object of $\mathbb{P}(s)$ while p' is an object of $\mathbb{P}(s+1)$.

The arrows of $\mathbb{P}(s)$ are then given by the set of rules in Figure 7.2.

$$\frac{}{\tau \leq_s \tau.p} \qquad \frac{}{x!y \leq_s x!y.p} \qquad \frac{}{x!{*} \leq_s x!{*}.p'} \qquad \frac{}{x? \leq_s x?(y{\mapsto}p)} \qquad \frac{}{x? \leq_s x?({*}{\mapsto}p)}$$

$$\frac{p \leq_s q}{\tau.p \leq_s \tau.q} \qquad \frac{p \leq_s q}{x!y.p \leq_s x!y.q} \qquad \frac{p' \leq_{s+1} q'}{x!{*}.p' \leq_s x!{*}.q'}$$

$$\frac{p \leq_s q}{x?(y{\mapsto}p) \leq_s x?(y{\mapsto}q)} \qquad \frac{p' \leq_{s+1} q'}{x?({*}{\mapsto}p') \leq_s x?({*}{\mapsto}q')}$$

Figure 7.2: The partial order $\mathbb{P}(s)$

Prefixings To give an inductive (on the structure of the objects) definition of the action of $\mathbb{P}(-)$ on morphisms we define first *prefixing* functors that extend that prefixing notation we used in the definition of the objects of $\mathbb{P}(s)$.

Definition 7.3.1 *Assume a fixed finite set s and suppose $x, y \in s$. As usual for us (cf. Section 4.4), write $l_* : \widehat{\mathbb{C}} \to \widehat{\mathbb{C}_\perp}$ for the embedding that takes a presheaf X over $\mathbb{C}$ to the rooted presheaf $\lfloor X \rfloor$ over $\mathbb{C}_\perp$.*

- *For α one of τ or $x!y$, let $In_\alpha : \mathbb{P}(s)_\perp \hookrightarrow \mathbb{P}(s)$ be the inclusion of $\mathbb{P}(s)_\perp$ in the appropriate component. Define a prefixing functor*

$$\alpha. \stackrel{\text{def}}{=} In_{\alpha,!} l_* : \widehat{\mathbb{P}(s)} \longrightarrow \widehat{\mathbb{P}(s)} \,.$$

- *Let $In_{x!*} : \mathbb{P}(s+1)_\perp \hookrightarrow \mathbb{P}(s)$ be the inclusion of $\mathbb{P}(s+1)_\perp$ in the appropriate component. Define a prefixing functor*

$$x!{*} \stackrel{\text{def}}{=} In_{x!*,!} l_* : \widehat{\mathbb{P}(s+1)} \longrightarrow \widehat{\mathbb{P}(s)} \,.$$

- *Let* $In_{x?} : (s \times \mathbb{P}(s) + \mathbb{P}(s+1))_\perp \to \mathbb{P}(s)$, *be the inclusion functor sending*

$$(y, p) \mapsto x?(y{\mapsto}p) \qquad \textit{and} \qquad p' \mapsto x?(*{\mapsto}p') \ .$$

Define

$$x? \stackrel{\text{def}}{=} In_{x?,!}l_* : (s \times \widehat{\mathbb{P}(s) + \mathbb{P}}(s+1)) \longrightarrow \widehat{\mathbb{P}(s)} \ .$$

Corollary 4.6.6, Proposition 4.4.1 and Proposition 4.6.8 are sufficient to deduce that all these functors preserve surjective open maps.

Proposition 7.3.2 *The prefixing functors of Definition 7.3.1 preserves surjective open maps, hence open map bisimulation.*

Moreover we also have the following elementary characterisation that shows that they are indeed doing what is expected.

Proposition 7.3.3

- *The functor*

$$\alpha. : \widehat{\mathbb{P}(s)} \to \widehat{\mathbb{P}(s)}$$

has action on objects described by,

$$\alpha.X(p) \cong \begin{cases} \{\star\} & \textit{if } p = \alpha. \\ X(p') & \textit{if } p = \alpha.p' \\ \emptyset & \textit{otherwise.} \end{cases}$$

- *The functor*

$$x! : \widehat{\mathbb{P}(s+1)} \to \widehat{\mathbb{P}(s)}$$

has action on objects described by,

$$x!*.Y(p) \cong \begin{cases} \{\star\} & \textit{if } p = x!* \\ Y(p') & \textit{if } p = x!*.p' \\ \emptyset & \textit{otherwise.} \end{cases}$$

- *The functor*

$$x? : (s \times \widehat{\mathbb{P}(s) + \mathbb{P}}(s+1)) \longrightarrow \widehat{\mathbb{P}(s)}$$

has action on objects described by,

$$x?\langle F, X\rangle(p) = \begin{cases} \{\star\} & \textit{if } p = x? \\ F(y)(p') & \textit{if } p = x?(y{\mapsto}p') \\ Y(p') & \textit{if } p = x?(*{\mapsto}p') \\ \emptyset & \textit{otherwise,} \end{cases}$$

where we represent presheaves over $(s \times \mathbb{P}(s) + \mathbb{P}(s+1))$ *as pairs* $\langle F, Y\rangle$ *with* F *a profunctor* $F : s \to\!\!\!\!+ \mathbb{P}(s)$ *and* Y *a presheaf* $Y \in |\widehat{\mathbb{P}(s+1)}|$.

Proof: We show the proof only for the first set of functors. The calculation for the other cases proceed analogously. Recall that,

$$(\alpha.X)(p) \stackrel{\text{def}}{=} In_{\alpha,!}l_*X(p) \cong \int^q \lfloor X \rfloor(q) \times (In_\alpha q)(p) \ .$$

We can consider three cases:

1. If $p = \alpha$ then the chain of bijections above continues as
$$\begin{array}{ll} \cong & \int^q \lfloor X \rfloor(q) \quad (\text{since } (In_\alpha q)(\alpha) \cong \{\star\}) \\ \cong & \{\star\} \qquad\qquad\quad (\text{since } \lfloor X \rfloor \text{ is rooted}). \end{array}$$
2. If $p = \alpha.p'$ then the chain continues as
$$\begin{array}{lll} \cong & \int^q \lfloor X \rfloor(q) \times \mathrm{y}_{\mathbb{P}(s)_\perp}(q)(\lfloor p' \rfloor) & (\text{since } \alpha.p' = In_\alpha \lfloor p' \rfloor) \\ \cong & \lfloor X \rfloor(\lfloor p' \rfloor) & (\text{by the Density Formula}) \\ = & X(p') & (\text{by full and faithfulness of } l_*). \end{array}$$
3. In any other case $(In_\alpha q)(p) = \emptyset$ for every $q \in \mid \mathbb{P}(s)_\perp \mid$ and so the coend is the empty set as well.

$\square$

NOTATION: In fact for presheaves X over $s \times \mathbb{P}(s) + \mathbb{P}(s+1)$ we shall often move, without giving notice, back an forth between the representation as pairs $\langle F, Y \rangle = \langle \lambda\, x.X^x, X^* \rangle$ with $F : s \to \widehat{\mathbb{P}(s)}$ and $Y \in \mid \widehat{\mathbb{P}(s+1)} \mid$ defined by

$$F(x)(p) \stackrel{\text{def}}{=} X(x,p) \stackrel{\text{def}}{=} X^x(p) \qquad \text{and} \qquad Y(p') \stackrel{\text{def}}{=} X(p') \stackrel{\text{def}}{=} X^*(p') \ ,$$

and as sums

$$\sum_{x \in s} (x \mapsto X^x) + (* \mapsto X^*) \ .$$

We can use the prefixing functors above to give a description of the action of $\mathbb{P}(-)$ on morphisms, $i : s \to s'$. We work by induction on the structure of the objects in $\mathbb{P}(s)$. In the base cases minimal paths in $\mathbb{P}(s)$ go to the same in $\mathbb{P}(s')$, regarded via Yoneda as presheaves:

$$\mathbb{P}(i)(\tau.) = \tau. \quad \mathbb{P}(i)(x!y.) = i(x)!i(y). \quad \mathbb{P}(i)(x!*_s.) = i(x)!*_{s'}. \quad \mathbb{P}(i)(x?) = i(x)?$$

The inductive steps are:

$$\begin{array}{rr} \mathbb{P}(i)(\tau.p) = \tau.\mathbb{P}(i)(p) & \mathbb{P}(i)(x!y.p) = i(x)!i(y).\mathbb{P}(i)(p) \\ \mathbb{P}(i)(x!*_s.p') = i(x)!*_{s'}.\mathbb{P}(i+1)(p') & \mathbb{P}(i)(x?(y \mapsto p)) = i(x)?(i(y) \mapsto \mathbb{P}(i)(p)) \\ & \mathbb{P}(i)(x?(* \mapsto p')) = i(x)?((\mathbb{N} \multimap \mathbb{P})(i)(* \mapsto p')) \ . \end{array}$$

In the last of these we use the non-trivial action of $(\mathbb{N} \multimap \mathbb{P})(i)$ from (7.3) to 'fill in' input behaviour on receiving names from $(s' \setminus Im(i))$.

Concerning the coherence isomorphisms, it is clear form the definitions that

$$\mathbb{P}(1_s) = \mathrm{y}_{\mathbb{P}(s)} \ .$$

If $i : s \to s'$ and $j : s' \to s''$ the coherence isomorphism

$$\mathbb{P}_{i,j} : \mathbb{P}(ji) \overset{\sim}{\Longrightarrow} \mathbb{P}(j)\mathbb{P}(i)$$

is defined inductively as follows. On the base cases it is just equality, the inductive steps are:

$$\begin{array}{ll}
(\mathbb{P}_{i,j})_{\tau.p} = \tau.((\mathbb{P}_{i,j})_p) & (\mathbb{P}_{i,j})_{x!y.p} = ji(x)!ji(y).((\mathbb{P}_{i,j})_p) \\
(\mathbb{P}_{i,j})_{x!*_s.p'} = ji(x)!*_{s''}.((\mathbb{P}_{j+1,i+1})_{p'}) & (\mathbb{P}_{i,j})_{x?(y\mapsto p)} = ji(x)?(ji(y) \mapsto (\mathbb{P}_{i,j})_p) \\
 & (\mathbb{P}_{i,j})_{x?(*\mapsto p')} = ji(x)?(\mathbb{N}\!\!\multimap\!\!\mathbb{P})_{i,j})_{p'} \ .
\end{array}$$

7.3.1 A decomposition result

As in Chapter 5 we wish to take advantage of decomposition results to define transition relations for presheaves that we shall use to deduce, in the denotational semantics, that late bisimulation for terms coincides with open map bisimulation of their denotation.

Proposition 7.3.4 *For every set s the category $\mathbb{P}(s)$ is isomorphic to*

$$\mathbb{P}(s)_\perp + \sum_{x,y\in s} \mathbb{P}(s)_\perp + \sum_{x\in s} \mathbb{P}(s+1)_\perp + \sum_{x\in s} (s \times \mathbb{P}(s) + \mathbb{P}(s+1))_\perp \ .$$

Proof: The proof is a trivial consequence of the following (obvious) general fact. For any category $\mathbb{C}$ and any set s, there is an isomorphism of categories

$$\sum_{x\in s} \mathbb{C} \cong s \times \mathbb{C} \ ,$$

where in the expression $s \times \mathbb{C}$, the set s is identified with the discrete category of objects the elements of s. $\square$

We have already seen (Chapter 5) that a presheaf X over a "lifted" category $\mathbb{C}_\perp$ can be decomposed as

$$X \cong \sum_{x\in X(\perp)} \lfloor X_{|x} \rfloor \ .$$

Moreover, via the isomorphism

$$\prod_{i\in I} \widehat{\mathbb{C}_i} \cong \widehat{\sum_{i\in I} \mathbb{C}_i} \ ,$$

for $(\mathbb{C}_i)_{i\in I}$ a family of small categories indexed by a set I, one has that a presheaf X over $\widehat{\sum_{i\in I} \mathbb{C}_i}$ can be written (omitting the obvious inclusion functor) as the sum $\sum_{i\in I} X_i$, where each X_i is a presheaf over $\mathbb{C}_i$. Combining these with Proposition 7.3.4 and recalling the characterisation of the prefixing functors (Proposition 7.3.3) we have the following:

Theorem 7.3.5 (Decomposition of Presheaves) *Let $X \in \widehat{\mathbb{P}(s)}$. Then X is isomorphic to*

$$\sum_{i\in X(\tau.)} \tau.X_i + \sum_{x,y\in s} \sum_{j\in X(x!y.)} x!y.X_j + \sum_{x\in s} \sum_{k\in X(x!*.)} x!*.X_k + \sum_{x\in s} \sum_{l\in X(x?)} x?\langle \lambda y.X_l^y, X_l^* \rangle \ .$$

Since $\mathbb{P}$ is defined recursively and the objects of $\mathbb{P}(s)$ are described by simultaneous induction, one can derive an inductive description of the resumptions categories, $(p/\mathbb{P}(s))^+$, associated to any path object (cf. Definition 5.1.5).

Proposition 7.3.6 *Let s be a finite set, $x, y \in s$ and $p \in\mid \mathbb{P}(s) \mid$; then the following holds:*

Base cases:

- *If $p = \tau$ then $(p/\mathbb{P}(s))^+ \cong \mathbb{P}(s)$.*
- *If $p = x!y$ then $(p/\mathbb{P}(s))^+ \cong \mathbb{P}(s)$.*
- *If $p = x!*$ then $(p/\mathbb{P}(s))^+ \cong \mathbb{P}(s+1)$.*
- *If $p = x?$ then $(p/\mathbb{P}(s))^+ \cong (s \times \mathbb{P}(s) + \mathbb{P}(s+1))$.*

Inductive steps:

- *If $p = \tau.p'$ then $(p/\mathbb{P}(s))^+ \cong (p'/\mathbb{P}(s))^+$.*
- *If $p = x!y.p'$ then $(p/\mathbb{P}(s))^+ \cong (p'/\mathbb{P}(s))^+$.*
- *If $p = x!*.p'$ then $(p/\mathbb{P}(s))^+ \cong (p'/\mathbb{P}(s+1))^+$.*
- *If $p = x?(y \mapsto p')$ then $(p/\mathbb{P}(s))^+ \cong (p'/\mathbb{P}(s))^+$.*
- *If $p = x?(* \mapsto p')$ then $(p/\mathbb{P}(s))^+ \cong (p'/\mathbb{P}(s+1))^+$.*

Thus given a presheaf X over $\mathbb{P}(s)$ and an element $x \in X(p)$, $X_{|x}$ is either a presheaf over $\mathbb{P}(s+n)$, for some (unique) $n \in \omega$, or a presheaf over $(s+n) \times \mathbb{P}(s+n) + \mathbb{P}(s+n+1)$. As in the Decomposition Theorem, in the latter case we shall write $(X_{|x})^a$ for the component of $X_{|x}$ associated with $a \in s+n$ and $(X_{|x})^*$ for the component at $\mathbb{P}(s+n+1)$.

As a consequence of Theorem 5.1.7 we have the following preservation property for resumptions with respect to the Decomposition Theorem.

Proposition 7.3.7 *Let X, Y be two presheaves over $\mathbb{P}(s)$ with $f : X \to Y$ a surjective open map between them. Then the following "restrictions" of f are surjective open:*

$$\begin{array}{rl}
f_{|i} : X_{|i} \to Y_{|i'} & \textit{where } i \in X(\tau.) \textit{ and } i' = f_{\tau.}(i) \\
f_{|j} : X_{|j} \to Y_{|j'} & \textit{where } j \in X(x!y.) \textit{ and } j' = f_{x!y.}(j) \\
f_{|k} : X_{|k} \to Y_{|k'} & \textit{where } k \in X(x!*.) \textit{ and } k' = f_{x!*.}(k) \\
(f_{|l})^a : (X_{|l})^a \to (Y_{|l'})^a & \textit{where } l \in X(x?),\ l' = f_{x?}(l) \textit{ and } a \in s \\
(f_{|l})^* : (X_{|l})^* \to (Y_{|l'})^* & \textit{where } l \in X(x?) \textit{ and } l' = f_{x?}(l),
\end{array}$$

where $(f_{|l})^a$ and $(f_{|l})^*$ are the components of $f_{|l}$ according to the isomorphism

$$(s \times \widehat{\mathbb{P}(s) + \mathbb{P}}(s+1)) \cong \prod_{a\in s} \widehat{\mathbb{P}(s)} \times \widehat{\mathbb{P}(s+1)} \ .$$

Using the Decomposition Theorem, we can also characterise the action of $\mathbb{P}(i)_!$ on presheaves.

Lemma 7.3.8 *Let $i : s \to s'$ be an injective function between finite sets and $X \in |\mathbb{P}(s)|$. Then the following hold:*

1. *If $X \cong \tau.Y$ then $\mathbb{P}(i)_!(X) \cong \tau.\mathbb{P}(i)_!(Y)$.*
2. *If $X \cong x!y.Y$ then $\mathbb{P}(i)_!(X) \cong i(x)!i(y).\mathbb{P}(i)_!(Y)$.*
3. *If $X \cong x!*.Y$ then $\mathbb{P}(i)_!(X) \cong i(x)!*.\mathbb{P}(i+1)_!(Y)$.*
4. *If $X \cong x?\langle \lambda\, y.\, X^y, X^* \rangle$ then $\mathbb{P}(i)_!(X) \cong i(x)?\langle \lambda\, w.\, \overline{X}^w, \mathbb{P}(i+1)_!(X^*)\rangle$, where for every $w \in s'$, $\overline{X}^w = \mathbb{P}(i)_!(X^y)$ if $w = i(y)$ and it is $\mathbb{P}([i, w])_!(X^*)$ otherwise.*

Proof:

1.
$$\begin{array}{rcll}
\mathbb{P}(i)_!(X) & = & \int^P \tau.Y(P)\,.\,\mathbb{P}(i)(P) & \text{(by definition)}\\
& = & \int^{P=In_\tau Q} \lfloor Y \rfloor(Q)\,.\,(\tau.\mathbb{P}(i)_!(l^*Q)) & \text{(since all the other paths give empty contribution)}\\
& \cong & \int^Q \lfloor Y \rfloor(Q)\,.\,(\tau.\mathbb{P}(i)_!(l^*Q)) & \\
& \cong & \tau.(\int^Q \lfloor Y \rfloor(Q)\,.\,\mathbb{P}(i)_!(l^*Q) & \text{(since the colimit is connected and } \tau. \text{ preserves connected colimits)}\\
& \cong & \tau.(\mathbb{P}(i)_! l^* \lfloor Y \rfloor) & \\
& = & \tau.(\mathbb{P}(i)_!(Y)) &
\end{array}$$

2. A similar calculation to that above.
3. Idem.
4.
$$\begin{array}{rcl}
\mathbb{P}(i)_!(X) & = & \int^P x?\langle \lambda\, y.\, X^y, X^*\rangle(P)\,.\,\mathbb{P}(i)(P)\\
& & \quad \text{(by definition)}\\
& = & \int^{P=In_{x?}R} \langle \lambda\, y.\, X^y, X^*\rangle(R)\,.\,i(x)?(\mathbb{N} \multimap \mathbb{P})(i)(R)\\
& & \quad \text{(since all the other paths give empty contribution)}\\
& = & \int^R \lfloor \langle \lambda\, y.\, X^y, X^*\rangle \rfloor(R)\,.\,i(x)?(\mathbb{N} \multimap \mathbb{P})(i)_!(l^*R)\\
& \cong & i(x)? \int^R \lfloor \langle \lambda\, y.\, X^y, X^*\rangle \rfloor(R)\,.\,(\mathbb{N} \multimap \mathbb{P})(i)_!(l^*R)\\
& & \quad \text{(since the colimit is connected and } i(x)? \text{ preserves connected colimits)}\\
& \cong & i(x)?(\mathbb{N} \multimap \mathbb{P})(i)_! l^* \lfloor \langle \lambda\, y.\, X^y, X^*\rangle \rfloor\\
& \cong & i(x)?(\mathbb{N} \multimap \mathbb{P})(i)_! \langle \lambda\, y.\, X^y, X^*\rangle\\
& \cong & i(x)?\langle \lambda\, w.\, \overline{X}^w, \mathbb{P}(i+1)_!(X^*)\rangle\\
& & \quad \text{(by Lemma 7.2.8).}
\end{array}$$

□

As an immediate consequence since left Kan extensions preserve sums we have the following:

Theorem 7.3.9 *Let $f : s \to s'$ be an injective function between finite sets and let X be a presheaf over $\mathbb{P}(s)$ with decomposition*

$$\sum_{i \in X(\tau.)} \tau.X_i + \sum_{x,y \in s} \sum_{j \in X(x!y.)} x!y.X_j + \sum_{x \in s} \sum_{k \in X(x!*.)} x!*.X_k + \sum_{x \in s} \sum_{l \in X(x?)} x?\langle \lambda y.X_l^y, X_l^* \rangle \ ,$$

then $\mathbb{P}(f)_!(X)$ *is isomorphic to the following presheaf:*

$$\sum_{i\in X(\tau.)} \tau.\mathbb{P}(f)_!(X_i) + \sum_{x,y\in s}\sum_{j\in X(x!y.)} f(x)!f(y).\mathbb{P}(f)_!(X_j)$$
$$+\sum_{x\in s}\sum_{k\in X(x!*.)} f(x)!*.\mathbb{P}(f+1)_!(X_k) + \sum_{x\in s}\sum_{l\in X(x?)} f(x)?\langle \lambda w.\overline{X}_l^w, \mathbb{P}(f+1)(X_l^*)\rangle\ ,$$

where $\overline{X}_l^w$ *is defined as in Lemma 7.3.8 above.*

7.3.2 Transition relations for presheaves and indexed late bisimilarity for $\mathbb{P}$

We can use the decomposition result to define indexed transition relations for presheaves over the $\mathbb{P}(s)$'s. As in Chapter 5 we could use the elements of the presheaves to induce some extra information on our transition arrows. Still as with **CCS**, it turns out that this is not necessary here.[2]

Definition 7.3.10 *Given a presheaf* X *over* $\mathbb{P}(s)$*, we say that*

- $X \xrightarrow{\tau} X'$ *if it exists* $i \in X(\tau)$ *such that* $X' = X_{|i}$.
- $X \xrightarrow{x!y} X'$ *if it exists* $j \in X(x!y)$ *such that* $X' = X_{|j}$.
- $X \xrightarrow{x!*} X'$ *if it exists* $k \in X(x!*)$ *such that* $X' = X_{|k}$.
- $X \xrightarrow{x?} \langle F, X'\rangle$ *if it exists* $l \in X(x?)$ *such that* $\langle F, X'\rangle = \langle \lambda y.X_l^y, X_l^*\rangle$.

Given the transition relation, it is natural now to define *late* bisimulation relations for presheaves.

Definition 7.3.11 *A* $\mathbb{P}$-late bisimulation *is a family* $(R_s)_{s\in\mathcal{I}}$ *of symmetric binary relations on presheaves in* $\widehat{\mathbb{P}(s)}$ *such that for any finite name set* s *and any two presheaves* X, Y *over* $\mathbb{P}(s)$*, if* $X\ R_s\ Y$ *then*

$$\begin{aligned}
X \xrightarrow{\tau} X' &\Rightarrow \exists Y'.\ Y \xrightarrow{\tau} Y'\ \&\ X'\ R_s\ Y' \\
X \xrightarrow{x!y} X' &\Rightarrow \exists Y'.\ Y \xrightarrow{x!y} Y'\ \&\ X'\ R_s\ Y' \\
X \xrightarrow{x!*} X' &\Rightarrow \exists Y'.\ Y \xrightarrow{x!*} Y'\ \&\ X'\ R_{s+1}\ Y' \\
X \xrightarrow{x?} \langle F, X'\rangle &\Rightarrow \exists \langle G, Y'\rangle.\ Y \xrightarrow{x?} \langle G, Y'\rangle\ \&\ X'\ R_{s+1}\ Y' \\
&\qquad\qquad\qquad\qquad \&\ \forall y\in s.\ F(y)\ R_s\ G(y)\ .
\end{aligned}$$

We say that $X, Y \in \widehat{\mathbb{P}(s)}$ *are* $\mathbb{P}$-late bisimilar *iff* $X\ R_s\ Y$ *for some* $\mathbb{P}$*-late bisimulation* $(R_s)_{s\in\mathcal{I}}$.

Lemma 7.3.12 $\mathbb{P}$*-late bisimilarity is an equivalence relation.*

[2] Chapter 8 will provide with a language where we conjecture that this extra information is vital in order to characterise operationally open map bisimulation.

Using Proposition 7.3.7 we can show that this $\mathbb{P}$-late bisimilarity corresponds exactly to open map bisimilarity.

Lemma 7.3.13 *Suppose X and Y are presheaves over $\mathbb{P}(s)$. Then:*

(i) *If $f : X \to Y$ is a surjective $\mathbb{P}(s)$-open map then X and Y are $\mathbb{P}$-late bisimilar.*

(ii) *If $X\ R_s\ Y$ for some $\mathbb{P}$-late bisimulation $(R_s)_{s\in\mathcal{I}}$ then X and Y are related by a span of surjective open maps.*

Proof:

1. Define $(R_s)_{s\in\mathcal{I}}$ as follows:
   $A\ R_s\ B$ iff there exists a surjective $\mathbb{P}(s)$-open map from A to B or from B to A. That this is a late bisimulation follows directly from Proposition 7.3.7, moreover by assumption $X\ R_s\ Y$.
2. Suppose $(R_s)_{s\in\mathcal{I}}$ is a $\mathbb{P}$-late bisimulation, then we define for every pair of presheaves A, B such that $A\ R_s\ B$ for some s a sub-presheaf of their product $Z^{A,B} \hookrightarrow A \times B$ such that the two projections on A and B are surjective $\mathbb{P}(s)$-open maps.
   The definition of the $Z^{A,B}$'s is given inductively on the structure of the path objects and goes as follows:

$$\begin{array}{rcl}
Z^{A,B}(\tau) & = & \{(i,i')|A_{|i}\ R_s\ B_{|i'}\} \\
Z^{A,B}(x!y) & = & \{(j,j')|A_{|j}\ R_s\ B_{|j'}\} \\
Z^{A,B}(x!*) & = & \{(k,k')|A_{|k}\ R_{s+1}\ B_{|k'}\} \\
Z^{A,B}(x?) & = & \{(l,l')|\forall y\,.\,A^y_{|l}\ R_s\ B^y_{|l'} \text{ and } A^*_{|l}\ R_{s+1}\ B^*_{|l'}\} \\
Z^{A,B}(\tau.p) & = & \{(a,b)|(a,b)\in Z^{A_{|i},B_{|i'}}(p), \\
 & & \text{where } (i,i') = (A\times B)(\tau \le \tau.p)(a,b)\} \\
Z^{A,B}(x!y.p) & = & \{(a,b)|(a,b)\in Z^{A_{|j},B_{|j'}}(p), \\
 & & \text{where } (j,j') = (A\times B)(x!y \le x!y.p)(a,b)\} \\
Z^{A,B}(x!*.p) & = & \{(a,b)|(a,b)\in Z^{A_{|k},B_{|k'}}(p), \\
 & & \text{where } (k,k') = (A\times B)(x!* \le x!*.p)(a,b)\} \\
Z^{A,B}(x?(y\mapsto p)) & = & \{(a,b)|(a,b)\in Z^{A^y_{|l},B^y_{|l'}}(p), \\
 & & \text{where } (l,l') = (A\times B)(x? \le x?(y\mapsto p))(a,b)\} \\
Z^{A,B}(x?(*\mapsto p)) & = & \{(a,b)|(a,b)\in Z^{A^*_{|l},B^*_{|l'}}(p), \\
 & & \text{where } (l,l') = (A\times B)(x? \le x?(*\mapsto p))(a,b)\}
\end{array}$$

We need to show first of all that $Z^{A,B}$ is a presheaf, i.e., that for every two paths, p, q, such that $p \le q$ and for every element $(a,b) \in Z^{A,B}(q)$, then

$$(A\times B)(p \le q)(a,b) \in Z^{A,B}(p)\ .$$

But this is a straightforward inductive proof on the rules that define the partial order relation for path objects (Figure 7.2) as the following case example shows.

$$\frac{p' \le q'}{p = \tau.p' \le \tau.q' = q}\ :$$

If $(a,b) \in Z^{A,B}(q)$ then by definition $(a,b) \in Z^{A_{|i},B_{|i'}}(q')$ for

$$(i,i') = A \times B(\tau \le \tau.q')(a,b)\ .$$

Then by the inductive hypothesis the following holds

$$A_{|i} \times B_{|i'}(p' \le q')(a,b) \in Z^{A_{|i},B_{|i'}}(p')\ ,$$

but by definition of $A_{|i}$ and $B_{|i'}$ (Definition 5.1.5),

$$A_{|i} \times B_{|i'}(p' \le q')(a,b) = A \times B(p \le q)(a,b)\ .$$

Thus, $A \times B(p \le q)(a,b) \in Z^{A_{|i},B_{|i'}}(p')$ hence by definition of $Z^{A,B}$,

$$A \times B(p \le q)(a,b) \in Z^{A,B}(p)\ .$$

Now we shall check that the projections $A \overset{\pi_A}{\leftarrow} Z^{A,B} \overset{\pi_B}{\rightarrow} B$ are surjective $\mathbb{P}(s)$-open maps. We begin with surjectivity and by symmetry concentrate only on one of the two projections. The proof goes by induction on the structure of the path objects and again we exemplify it considering only one case (the others being rather similar). Let $p = x?(y \mapsto q)$ be a path object. We want to show that if $a \in A(p)$, then there exists a $z = (a,b) \in Z^{A,B}(p)$. By definition if $a \in A(p)$ then there exists $l \in A(x?)$ such that $a \in A^y_{|l}$. Since $(R_s)_{s\in\mathcal{I}}$ is a late bisimulation, then there exists $l' \in B(x?)$ such that $A^y_{|l}\ R_s\ B^y_{|l'}$. By inductive hypothesis, $\pi_{A^y_{|l}} : Z^{A^y_{|l},B^y_{|l'}} \to A^y_{|l}$ is surjective, hence there exists $z = (a,b) \in Z^{A^y_{|l},B^y_{|l'}}(q)$ and by definition this means that $z \in Z^{A,B}(x?(y \mapsto q))$ too.

The proof that the projection is $\mathbb{P}(s)$-open is instead done by induction on the rules defining the partial order relation between paths as the following case example shows.

$$\frac{p \le q}{x!y.p \le x!y.q}\ :$$

We want to deduce that the following square is a quasi pullback,

$$\begin{tikzcd}
Z^{A,B}(x!y.q) \overset{\mathrm{def}}{=} \sum_{(j,j')\in A\times B(x!y)} Z^{A_{|j},B_{|j'}}(q) \arrow[r, "(\pi_A)_{x!y.q}=\sum(\pi_{A_{|j}})_q"] \arrow[d, "Z^{A,B}(x!y.p\le x!y.q)"'] & \sum_{j\in A(x!y)} A_{|j}(q) = A(x!y.q) \arrow[d, "A(x!y.p\le x!y.q)"] \\
Z^{A,B}(x!y.p) \overset{\mathrm{def}}{=} \sum_{(j,j')\in A\times B(x!y)} Z^{A_{|j},B_{|j'}}(p) \arrow[r, "(\pi_A)_{x!y.p}=\sum(\pi_{A_{|j}})_p"'] & \sum_{j\in A(x!y)} A_{|j}(p) = A(x!y.p)\ .
\end{tikzcd}$$

But by inductive hypothesis each "summand" in the diagram above, that is each square like the following,

$$\begin{tikzcd}
Z^{A_{|j},B_{|j'}}(q) \arrow[r, "(\pi_{A_{|j}})_q"] \arrow[d, "Z^{A_{|j},B_{|j'}}(p\le q)"'] & A_{|j}(q) \arrow[d, "A_{|j}(p\le q)"] \\
Z^{A_{|j},B_{|j'}}(p) \arrow[r, "(\pi_{A_{|j}})_p"'] & A_{|j}(p)\ ,
\end{tikzcd}$$

is a quasi pullback, thus the first diagram is a quasi pullback too. □

Combining Lemma 7.3.12 and 7.3.13 gives:

Theorem 7.3.14 *Two presheaves X and Y over $\mathbb{P}(s)$ are $\mathbb{P}$-late bisimilar if and only if they are connected by a span of surjective open maps.*

Moving to a larger set of free names does not affect $\mathbb{P}$-late bisimilarity since colimit preserving functors preserve (surjective) open maps (Corollary 4.6.6).

Proposition 7.3.15 (Weakening) *If $X, Y \in \widehat{\mathbb{P}(s)}$ are $\mathbb{P}$-late bisimilar then so are $\mathbb{P}(i)_!(X)$ and $\mathbb{P}(i)_!(Y)$ for any injection $i : s \to s'$.*

Moving to smaller name sets is a little more complicated. For any $i : s \to s'$ in $\mathcal{I}$ define $e_i : \mathbb{P}(s) \to \mathbb{P}(s')$ by induction as follows:

$$
\begin{aligned}
e_i(\tau) &= \tau & e_i(x!y) &= i(x)!i(y) \\
e_i(x!*_s) &= i(x)!*_{s'} & e_i(x?) &= i(x)? \\[4pt]
e_i(\tau.p) &= \tau.e_i(p) & e_i(x!y.p) &= i(x)!i(y).e_i(p) \\
e_i(x!*_s.p) &= i(x)!*_{s'}.e_{i+1}(p) & e_i(x?(y \mapsto p)) &= i(x)?(i(y) \mapsto e_i(p)) \\
& & e_i(x?(*_s \mapsto p')) &= i(x)?(*_{s'} \mapsto e_{i+1}(p)) \ .
\end{aligned}
$$

This differs from $\mathbb{P}(i)$ in having a much simpler action on input of unknowns $x?(*{\mapsto}p')$. Even so $e_i^\star$, which again by Corollary 4.6.6 preserves open maps, turns out to be a left inverse to $\mathbb{P}(i)_!$.

Lemma 7.3.16 *Let $i : s \to s'$ be an injective function between finite sets, then for any presheaf $X \in \mid \mathbb{P}(s) \mid$,*

$$e_i^* \mathbb{P}(i)_!(X) \cong X \ .$$

Proof: Observe first of all that $e_i^* \mathbb{P}(i) = \mathrm{y}_{\mathbb{P}(s)}$ as it is immediately proved by induction on the structure of the paths. By the preservation property of left-adjoint functors with respect to left Kan extensions [17] we then have the following chain of natural isomorphisms:

$$
\begin{aligned}
e_i^* \mathbb{P}(i)_! &\cong e_i^* \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}(s)}}(\mathbb{P}(i)) \\
&\cong \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}(s)}}(e_i^* \mathbb{P}(i)) \\
&\cong \mathrm{Lan}_{\mathrm{y}_{\mathbb{P}(s)}}(\mathrm{y}_{\mathbb{P}(s)}) \\
&\cong 1_{\widehat{\mathbb{P}(s)}} \ .
\end{aligned}
$$

□

This allows us to prove the following result.

Proposition 7.3.17 (Strengthening) *For $X, Y \in \widehat{\mathbb{P}(s)}$ and $i : s \to s'$ in $\mathcal{I}$, if $\mathbb{P}(i)_!(X)$ and $\mathbb{P}(i)_!(Y)$ are $\mathbb{P}$-late bisimilar over $\mathbb{P}(s')$ then so are X and Y over $\mathbb{P}(s)$.*

Proof: By the lemma above $X \cong e_i^*\mathbb{P}(i)_!X$ that is open map bisimilar to $e_i^*\mathbb{P}(i)_!Y$ since e_i^* being colimit preserving preserves open map bisimulation. Again by the lemma above one has that $e_i^*\mathbb{P}(i)_!Y \cong Y$ and hence X is open map bisimilar to Y. □

These results suggest that we could have imposed similar uniformity constraints on the family $(R_s)_{s\in\mathcal{I}}$ in Definition 7.3.11: we conjecture that without loss of generality we can require that $\mathbb{R} \rightarrowtail\!\!\!\mapsto (\mathbb{P} \,\&\, \mathbb{P})$ be a pointwise discrete cartesian subobject in $\mathbf{Prof}^{\mathcal{I}}$.

7.4 Constructions

7.4.1 Restriction

We define here the operator that will be used to interpret name restriction in π-calculus processes. It arises as a family of profunctor arrows indexed by finite sets s and their elements:

$$\nu_{y\in s} : \mathbb{P}(s) \mathrel{\rightarrow\!\!\!\!\!\!+} \mathbb{P}(s - \{y\}) \ .$$

We define the $\nu_{y\in s}$ simultaneously for all s by induction on the structure of the paths. So for each path in $\mathbb{P}(s)$, according to its structure:

$$
\begin{aligned}
\nu_{y\in s}(\tau.) &= \tau. \\
\nu_{y\in s}(x!z.) &= \begin{cases} x!z. & \text{if } x, z \neq y \\ x!*_{s-\{y\}}. & \text{if } x \neq y \text{ and } z = y \\ \emptyset & \text{otherwise} \end{cases} \\
\nu_{y\in s}(x!*_s.) &= \begin{cases} x!*_{s-\{y\}}. & \text{if } x \neq y \\ \emptyset & \text{otherwise} \end{cases} \\
\nu_{y\in s}(x?) &= \begin{cases} x? & \text{if } x \neq y \\ \emptyset & \text{otherwise} \end{cases} \\
\nu_{y\in s}(\tau.p) &= \tau.\nu_{y\in s}(p) \\
\nu_{y\in s}(x!z.p) &= \begin{cases} x!z.\nu_{y\in s}(p) & \text{if } x, z \neq y \\ x!*_{s-\{y\}}.\mathbb{P}(b_{s,y})(p) & \text{if } x \neq y \text{ and } z = y \\ \emptyset & \text{otherwise} \end{cases} \\
\nu_{y\in s}(x!*_s.p') &= \begin{cases} x!*_{s-\{y\}}.\nu_{y\in s+1}(p') & \text{if } x \neq y \\ \emptyset & \text{otherwise} \end{cases} \\
\nu_{y\in s}(x?(z \mapsto p)) &= \begin{cases} x?(z \mapsto \nu_{y\in s}(p)) & \text{if } x, z \neq y \\ x? & \text{if } x \neq y = z \\ \emptyset & \text{otherwise} \end{cases} \\
\nu_{y\in s}(x?(*_s \mapsto p')) &= \begin{cases} x?(*_{s-\{y\}} \mapsto \nu_{y\in s+1}(p')) & \text{if } x \neq y \\ \emptyset & \text{otherwise} \end{cases}
\end{aligned}
$$

where $b_{s,y} : s \to (s - \{y\}) \cup \{*_{s-\{y\}}\}$ is the bijection that renames y to $*_{s-\{y\}}$. Using the Decomposition Theorem we can characterise the action of the restriction operator on presheaves.

Lemma 7.4.1 *Let s be a finite sets and let $y \in s$, then the following hold:*

1. *If $X = \alpha.Y \in |\widehat{\mathbb{P}(s)}|$ with $\alpha \in \{\tau\} \cup \{x!z \mid x \neq y \neq z\} \cup \{x!* \mid x \neq y\}$ then*
$$\nu_{y\in s,!}X \cong \alpha.\nu_{y\in s,!}Y \ .$$
2. *If $X = x!y.Y \in |\widehat{\mathbb{P}(s)}|$ with $x \neq y$ then $\nu_{y\in s,!}X \cong x!*.\mathbb{P}(b_{s,y})_!Y$.*
3. *If $X = x?(z \mapsto Y) \in |\widehat{\mathbb{P}(s)}|$ with $x \neq y \neq z$ then $\nu_{y\in s,!}X \cong x?(z \mapsto \nu_{y\in s,!}Y)$.*
4. *If $X = x?(y \mapsto Y) \in |\widehat{\mathbb{P}(s)}|$ with $x \neq y$ then $\nu_{y\in s,!}X \cong x?$.*
5. *If $X = x?(* \mapsto Y) \in |\widehat{\mathbb{P}(s)}|$ with $x \neq y$ then $\nu_{y\in s,!}X \cong x?(z \mapsto \nu_{y\in s+1,!}Y)$.*
6. *If $X = y!z.Y$ or $X = y?(z \mapsto Y)$ or $X = y?(* \mapsto Y)$, then $\nu_{y\in s,!}X = \emptyset$.*
7. *If $X = x?\langle\lambda\ z.Y^z, Y^*\rangle$, with $x \neq y$ then $\nu_{y\in s,!}X \cong x?\langle\lambda\ z.\nu_{y\in s,!}\overline{Y}^z, \nu_{y\in s+1,!}Y^*\rangle$ where $\overline{Y}^y = \emptyset$ and $\overline{Y}^z = Y^z$ for $z \neq y$.*
8. *If $X = y?\langle\lambda\ z.Y^z, Y^*\rangle$ then $\nu_{y\in s,!}X = \emptyset$.*

Proof:

1.
$$\begin{array}{rcll}
\nu_{y\in s,!}X & \cong & \int^p X(p)\,.\,\nu_{y\in s}(p) & \\
& = & \int^p (\alpha.Y)(p)\,.\,\nu_{y\in s}(p) & \\
& = & \int^{p=In_\alpha q}(\alpha.Y)(p)\,.\,\nu_{y\in s}(p) & \text{(since the other paths give empty contribution)} \\
& = & \int^q \lfloor Y\rfloor(q)\,.\,(\alpha.(\nu_{y\in s,!}(l^*q)) & \text{(since } In_\alpha q = \alpha.(l^*q)) \\
& \cong & \alpha.(\int^q \lfloor Y\rfloor(q)\,.\,\nu_{y\in s,!}(l^*q) & \text{(since } \alpha. \text{ preserves connected colimits)} \\
& \cong & \alpha.(\int^p Y(p)\,.\,\nu_{y\in s}(p)) & \text{(since } q=\bot \text{ contributes with the empty presheaf)} \\
& \cong & \alpha.(\nu_{y\in s,!}Y) & \text{(by definition)}
\end{array}$$

2.
$$\begin{array}{rcll}
\nu_{y\in s,!}X & \cong & \int^p X(p)\,.\,\nu_{y\in s}(p) & \\
& = & \int^p (x!y.Y)(p)\,.\,\nu_{y\in s}(p) & \\
& = & \int^{p=In_{x!y}q}(x!y.Y)(p)\,.\,\nu_{y\in s}(p) & \text{(since the other paths give empty contribution)} \\
& = & \int^q \lfloor Y\rfloor(q)\,.\,x!*.\mathbb{P}(b_{s,y})_!(l^*q) & \text{(since } In_{x!y}q = x!y.(l^*q)) \\
& \cong & x!*.(\int^q \lfloor Y\rfloor(q)\,.\,\mathbb{P}(b_{s,y})_!(l^*q) & \text{(since } x!*. \text{ preserves connected colimits)} \\
& \cong & x!*.(\mathbb{P}(b_{s,y})_! l^*\lfloor Y\rfloor) & \\
& = & x!*.(\mathbb{P}(b_{s,y})_! Y) & \text{(since } l^*l_* = 1_{\mathbb{P}(s)})
\end{array}$$

3. $$\begin{array}{rcl} \nu_{y\in s,!}X & \cong & \int^p X(p)\,.\,\nu_{y\in s}(p) \\ & = & \int^p (x?(z\mapsto Y))(p)\,.\,\nu_{y\in s}(p) \\ & = & \int^q (x?(z\mapsto Y))(x?(z\mapsto l^*q)\,.\,\nu_{y\in s}(x?(z\mapsto l^*q)) \\ & & \quad \text{(since the other paths give empty contribution)} \\ & = & \int^q \lfloor Y\rfloor(q)\,.\,(x?(z\mapsto \nu_{y\in s})(l^*q) \\ & & \quad \text{(since } In_x?(z\mapsto \lfloor p\rfloor = x?(z\mapsto p) \text{ and } x?(z\mapsto \emptyset = x?) \\ & \cong & x?(\int^q \lfloor Y\rfloor(q)\,.\,(z\mapsto \nu_{y\in s}(l^*q)) \\ & & \quad \text{(since } x? \text{ preserves connected colimits)} \\ & \cong & x?(z\mapsto \nu_{y\in s,!}l^*\lfloor Y\rfloor) \\ & = & x?(z\mapsto \nu_{y\in s,!}Y) \\ & & \quad \text{(since } l^*l_* = 1_{\mathbb{P}(s)}) \end{array}$$

4. Similar as above with the extra restriction that only the component $p = x?$ of the induced colimit gives a non-empty contribution.
5. Similar as above where simply, by definition, one moves from $\nu_{y\in s}$ to $\nu_{y\in s+1}$.
6. Here clearly no component gives a non-empty contribution.
7. This is a consequence of the fact that $\langle\lambda\ z.Y^z, Y^*\rangle$ can be seen as a notation for the presheaf over $s\times\mathbb{P}(s)+\mathbb{P}(s+1)$, $\sum_{z\in s}(z\mapsto Y^z)+(*\mapsto Y^*)$.
8. Again it is immediately seen that every component of the induced colimit is the empty presheaf.

□

Using the Lemma above and the Decomposition Theorem we can now characterise the action of ν on a general presheaf as follows:

Theorem 7.4.2 *Let $X \in|\widehat{\mathbb{P}(s)}|$ have the following decomposition*

$$\sum_{i\in X(\tau.)}\tau.X_i + \sum_{x,z\in s}\sum_{j\in X(x!z.)} x!z.X_j + \sum_{x\in s}\sum_{k\in X(x!*.)} x!*.X_k + \sum_{x\in s}\sum_{l\in X(x?)} x?\langle\lambda z.X_l^z, X_l^*\rangle\ .$$

then $\nu_{y\in s,!}X$ is isomorphic to the presheaf

$$\begin{array}{l}\displaystyle\sum_{i\in X(\tau.)}\tau.\nu_{y\in s,!}X_i + \sum_{\substack{x,z\in s\\ x\neq y\neq z}}\ \sum_{j\in X(x!z.)} x!z.\nu_{y\in s,!}X_j + \sum_{\substack{x\in s\\ x\neq y}}\ \sum_{j'\in X(x!y.)} x!*.\mathbb{P}(b_{s,y})_!X_{j'} \\ \displaystyle\quad + \sum_{\substack{x\in s\\ x\neq y}}\ \sum_{k\in X(x!*.)} x!*.\nu_{y\in s+1,!}X_k + \sum_{\substack{x\in s\\ x\neq y}}\ \sum_{l\in X(x?)} x?\langle\lambda z.\nu_{y\in s,!}\overline{X}_l^z, \nu_{y\in s+1,!}X_l^*\rangle\ .\end{array}$$

with $\overline{X}_l^y=\emptyset$ and $\overline{X}_l^z = X_l^z$ in any other case.

Proof: The simple proof just uses the fact that sums preserve coends to reduce to the cases that are handled by the Lemma 7.4.1. □

Using the above results we can show that the family of ν's satisfies the following naturality property:

Theorem 7.4.3 *For any injective function $i : s \to s'$, the following square commutes (up to a natural isomorphism):*

$$\begin{tikzcd} \mathbb{P}(s) \arrow[r, "\nu_{y\in s}", "\shortmid"{marking}] \arrow[d, "\mathbb{P}(i)"', "\shortmid"{marking}] & \mathbb{P}(s-\{y\}) \arrow[d, "\mathbb{P}(i')", "\shortmid"{marking}] \\ \mathbb{P}(s') \arrow[r, "\nu_{i(y)\in s'}", "\shortmid"{marking}] & \mathbb{P}(s'-\{i(y)\}) \end{tikzcd}$$

where $i' : (s - \{y\}) \to (s' - \{i(y)\})$ is the restriction of i.

Proof: We prove this fact by structural induction on the paths, $p \in |\mathbb{P}(s)|$.

Base Cases: If $p \in \{\tau\} \cup \{x!z \mid x \neq y \neq z\} \cup \{x? \mid x \neq y\}$, then by definition

$$\mathbb{P}(i')(\nu_{y\in s}(p)) = \mathbb{P}(i)(p) = \nu_{i(y)\in s'}(\mathbb{P}(i)(p)) \ .$$

If $p \in \{x!*_s \mid x \neq y\}$, then by definition

$$\mathbb{P}(i')(\nu_{y\in s}(x!*_s)) = i(x)!*_{s'-\{i(y)\}} = \nu_{i(y)\in s'}(i(x)!*_{s'}) = \nu_{i(y)\in s'}(\mathbb{P}(i)(x!*_s)) \ .$$

Finally, in any other case both the composites return the empty presheaf.

Inductive Step: If $p = \tau.p'$ then

$$\begin{array}{rcll} \mathbb{P}(i')(\nu_{y\in s}p) &=& \mathbb{P}(i')(\tau.\nu_{y\in s}p') & \text{(by definition of } \nu) \\ &=& \tau.\mathbb{P}(i')(\nu_{y\in s}p') & \text{(by definition of } \mathbb{P}) \\ &=& \tau.\nu_{i(y)\in s'}(\mathbb{P}(i)p') & \text{(by inductive hypothesis)} \\ &=& \nu_{i(y)\in s'}(\mathbb{P}(i)\tau.p') & \text{(by definition of } \nu \text{ and } \mathbb{P}). \end{array}$$

A similar argument proves the cases

$$p \in \{x!z.p' \mid x \neq y \neq z\} \cup \{x!*_s \mid x \neq y\} \cup \{x?(z \mapsto p') \mid x \neq y \neq z\} \ .$$

If $p \in \{x?(*_s \mapsto p') \mid x \neq y\}$ then

$$\begin{array}{rcl} \mathbb{P}(i')(\nu_{y\in s}p) &=& \mathbb{P}(i')_!(x?(* \mapsto \nu_{y\in s+1}p')) \\ &=& i(x)?(\mathbb{N} \multimap \mathbb{P})(i')_!(* \mapsto \nu_{y\in s+1}p') \\ && \qquad \text{(by definition of } \mathbb{P}(i') \text{ and since } i(x) = i'(x)) \\ &\cong& i(x)?(\sum_{z\notin Imi'}(z \mapsto \mathbb{P}([i', z])_!\nu_{y\in s}p') + (* \mapsto \mathbb{P}(i'+1)_!\nu_{y\in s}p')) \\ && \qquad \text{(by definition of } \mathbb{N} \multimap \mathbb{P}) \\ &\cong& i(x)?(\sum_{z\notin Imi'}(z \mapsto \nu_{i(y)\in s',!}\mathbb{P}([i,z])p') + (* \mapsto \nu_{i(y)\in s',!}\mathbb{P}(i+1)p') \\ && \qquad \text{(by inductive hypothesis)} \\ &=& i(x)?(\sum_{z\notin Imi'}(z \mapsto \nu_{i(y)\in s',!}\mathbb{P}([i,z])p') + (* \mapsto \nu_{i(y)\in s',!}\mathbb{P}(i+1)p') \\ && +(i(y) \mapsto \emptyset) \\ && \qquad \text{(since we just added the empty presheaf)} \\ &\cong& \nu_{i(y)\in s',!}(i(x)?\langle \lambda z \ . \ \mathbb{P}([i,z])p', \mathbb{P}(i+1)p' \rangle) \\ && \qquad \text{(by point 7 of Lemma 7.4.1)} \\ &\cong& \nu_{i(y)\in s',!}(i(x)?(\mathbb{N} \multimap \mathbb{P})(* \mapsto p')) \\ &=& \nu_{i(y)\in s',!}\mathbb{P}(i)(x?(* \mapsto p')) \end{array}$$

□

In particular we can observe that the family $(\nu_{*\in s+1})_s$ defines a pseudo natural transformation $\nu : \delta(\mathbb{P}) \dot{\to} \mathbb{P}$. The important feature of the definition of ν is that it correctly turns free output into bound output, as summarised in this result directly obtainable from Lemma 7.4.1:

Lemma 7.4.4 *Let X be a presheaf over $\mathbb{P}(s)$ and let $x, y \in s$. If $X \xrightarrow{x!y} X'$, then*

$$\nu_{y\in s,!}(X) \xrightarrow{x!*_{s-\{y\}}} \mathbb{P}(b_{s,y})_!(X') \ .$$

Another thing which is worth observing is that if y, z are two different elements of a set s, then

$$\nu_{z\in s-\{y\},!} \circ \nu_{y\in s} \cong \nu_{y\in s-\{z\},!} \circ \nu_{z\in s}.$$

This suggests a definition of ν as a contravariant pseudo functor from $\mathcal{I}$ to **Prof** with $\nu(s) = \mathbb{P}(s)$ and $\nu(i) : \mathbb{P}(s') \nrightarrow \mathbb{P}(s)$ the restriction, in any order, of the elements of $(s' \setminus Im(i))$.

7.4.2 Parallel composition

We turn now to parallel composition and as in Chapter 5 we break the definition of the parallel composition functors

$$|_s : \widehat{\mathbb{P}(s)} \times \widehat{\mathbb{P}(s)} \to \widehat{\mathbb{P}(s)}$$

into the components

$$\widehat{\mathbb{P}(s)} \times \widehat{\mathbb{P}(s)} \xrightarrow{\quad l_* \times l_* \quad} \widehat{\mathbb{P}(s)_\perp} \times \widehat{\mathbb{P}(s)_\perp} \xrightarrow{w^*} \widehat{\mathbb{P}(s)_\perp \times \mathbb{P}(s)_\perp} \xrightarrow{(||_s)_!} \widehat{\mathbb{P}(s)} \ ,$$

where the last functor is colimit preserving. Hence from previous (general) results about colimit and connected colimit preserving functors we shall be able to conclude that $|_s$ preserves open map bisimulation.

Definition 7.4.5 *For every finite set s, with elements indicated with the letters x, y, z, define the symmetric profunctor*

$$||_s : \mathbb{P}(s)_\perp \times \mathbb{P}(s)_\perp \longrightarrow\!\!\!\!\!\!\!+ \;\mathbb{P}(s)$$

by simultaneous induction, for all sets s, on the structure of the path objects represented using the 'In' functors to reduce the number of different cases to be considered, as follows *(omitting the obvious Yoneda embeddings):*

$$\begin{array}{rcl}
\bot||_s\bot & = & \emptyset \\
\lfloor p\rfloor||_s\bot & = & p \\
\lfloor In_\alpha(q)\rfloor||_s\lfloor In_{\alpha'}(r)\rfloor & = & \alpha.(\lfloor q\rfloor||_s\lfloor In_{\alpha'}(r)\rfloor)+\alpha'.(\lfloor In_\alpha(q)\rfloor||\lfloor r\rfloor) \\[2ex]
\lfloor In_\alpha(q)\rfloor||_s\lfloor In_{x'!*}(r)\rfloor & = & \alpha.(\lfloor q\rfloor||_s\lfloor In_{x'!*}(r)\rfloor)+x'!*.(\lfloor \mathbb{P}(i)(In_\alpha(q))\rfloor(||_{s+1})_!\lfloor r\rfloor) \\[2ex]
\lfloor In_\tau(q)\rfloor||_s\lfloor In_{x?}(r)\rfloor & = & \tau.(\lfloor q\rfloor||_s\lfloor In_{x?}(r)\rfloor \\
 & & +x?\langle\lambda y\ .\ \lfloor In_\tau(q)\rfloor||_s\lfloor r(y)\rfloor, \lfloor\mathbb{P}(i)(In_\tau(q))\rfloor(||_{s+1})_!\lfloor r(*)\rfloor\rangle \\[2ex]
\lfloor In_{x!y}(q)\rfloor||_s\lfloor In_{x'?}(r)\rfloor & = & \begin{cases}
x!y.(\lfloor q\rfloor||_s\lfloor In_{x'?}(r)\rfloor) \\
+x'?\langle\lambda z\ .\ \lfloor In_{x!y}(q)\rfloor||_s\lfloor r(z)\rfloor, \lfloor\mathbb{P}(i)(In_{x!y}(q))\rfloor(||_{s+1})_!\lfloor r(*)\rfloor\rangle \\
\hfill \textit{if } x\neq x' \\
\tau.(\lfloor q\rfloor||_s\lfloor r(y)\rfloor)+x!y.(\lfloor q\rfloor||_s\lfloor In_{x'?}(r)\rfloor) \\
+x'?\langle\lambda z\ .\ \lfloor In_{x!y}(q)\rfloor||_s\lfloor r(z)\rfloor, \lfloor\mathbb{P}(i)(In_{x!y}(q))\rfloor(||_{s+1})_!\lfloor r(*)\rfloor\rangle \\
\hfill \textit{otherwise}
\end{cases} \\[2ex]
\lfloor In_{x!*}(q)\rfloor||_s\lfloor In_{x'?}(r)\rfloor & = & \begin{cases}
x!*.(\lfloor q\rfloor(||_{s+1})_!\lfloor\mathbb{P}(i)(In_{x'?}(r)))\rfloor \\
+x'?\langle\lambda z\ .\ \lfloor In_{x!*}(q)\rfloor||_s\lfloor r(z)\rfloor, \lfloor\mathbb{P}(i)(In_{x!*}(q))\rfloor(||_{s+1})_!\lfloor r(*)\rfloor\rangle \\
\hfill \textit{if } x\neq x' \\
\tau.(\nu_{*\in s+1,!}(\lfloor q\rfloor||_{s+1}\lfloor r(*)\rfloor))+x!*.(\lfloor q\rfloor(||_{s+1})_!\lfloor\mathbb{P}(i)(In_{x'?}(r))\rfloor) \\
+x'?\langle\lambda z\ .\ \lfloor In_{x!*}(q)\rfloor||_s\lfloor r(z)\rfloor, \lfloor\mathbb{P}(i)(In_{x!*}(q))\rfloor(||_{s+1})_!\lfloor r(*)\rfloor\rangle \\
\hfill \textit{otherwise}
\end{cases} \\[2ex]
\lfloor In_{x?}(q)\rfloor||_s\lfloor In_{y?}(r)\rfloor & = & x?\langle\lambda z\ .\ \lfloor q(z)\rfloor||_s\lfloor In_{y?}(r)\rfloor, \lfloor q(*)\rfloor(||_{s+1})_!\lfloor\mathbb{P}(i)(In_{y?}(r))\rfloor\rangle \\
 & & +y?\langle\lambda z\ .\ \lfloor In_{x?}(q)\rfloor||_s\lfloor r(z)\rfloor, \lfloor\mathbb{P}(i)(In_{x?}(q))\rfloor(||_{s+1})_!\lfloor r(*)\rfloor\rangle
\end{array}$$

for $\alpha,\alpha'\in\{\tau\}\cup\{x!y\mid x,y\in s\}$ and $i:s\hookrightarrow s+1$ the obvious inclusion. For $In_{x?}(r)$, we define, $r(y)=\emptyset$ when $r=\bot$ or $r=\lfloor(z\mapsto p)\rfloor$ for $z\neq y$, while $r(y)=p$ if $r=\lfloor(y\mapsto p)\rfloor$ and similarly we define $r(*)$.
The action on morphisms, i.e., on the lesser or equal than relation, is inductively determined as well in the obvious way.

Definition 7.4.6 (Parallel Composition) *For every finite set s define the parallel composition functor*

$$|_s \stackrel{\text{def}}{=} (||_s)_! \circ w_{\mathbb{P}(s)_\bot,\mathbb{P}(s)_\bot}\circ(l_*\times l_*)\ .$$

Using again the Decomposition Theorem, with proofs analogous to those of Chapter 5 for **CCS** (Lemma 5.1.13 and Proposition 5.1.14), we can characterise the parallel composition as follows.

Theorem 7.4.7 *Let X and Y be two presheaves over $\mathbb{P}(s)$ with the respective decompositions indexed by i,j,k,l and i',j',k',l'. Then $X|_sY$, is isomorphic to the following* *inductively defined presheaf:*

$$\begin{aligned}
&\sum_{i\in I}\tau.(X_i|_sY)+\sum_{x,y\in s}\sum_{j\in J_{x!y}}x!y.(X_j|_sY)+\sum_{x\in s}\sum_{k\in K_x}x!*_s.(X_k|_{s+1}\mathbb{P}(i)_!(Y))\\
&\quad+\sum_{x\in s}\sum_{l\in L_x}x?\langle\lambda y.(X_l^y|_sY),X_l^*|_{s+1}\mathbb{P}(i)_!(Y)\rangle\\
&+\sum_{i'\in I'}\tau.(X|_sY_{i'})+\sum_{x,y\in s}\sum_{j'\in J'_{x!y}}x!y.(X|_sY_{j'})+\sum_{x\in s}\sum_{k'\in K'_x}x!*_s.(\mathbb{P}(i)_!(X)|_{s+1}Y_{k'})\\
&\quad+\sum_{x\in s}\sum_{l'\in L'_x}x?\langle\lambda y.(X|_sY_{l'}^y),\mathbb{P}(i)_!(X)|_{s+1}Y_{l'}^*\rangle\\
&+\sum_{x,y\in s}\sum_{j\in J_{x!y}}\sum_{l'\in L'_x}\tau.(X_j|_sY_{l'}^y)+\sum_{x\in s}\sum_{k\in K_x}\sum_{l'\in L'_x}\tau.\nu_{*\in s+1,!}(X_k|_{s+1}Y_{l'}^*)\\
&+\sum_{x,y\in s}\sum_{j'\in J'_{x!y}}\sum_{l\in L_x}\tau.(X_l^y|_sY_{j'})+\sum_{x\in s}\sum_{k'\in K'_x}\sum_{l\in L_x}\tau.\nu_{*\in s+1,!}(X_l^*|_{s+1}Y_{k'})\,,
\end{aligned}$$

where $i:s\to s+1$ is the obvious inclusion function.

As we have already announced since $|_s$ arises as the composite of open map preserving functors, we have the following congruence property:

Theorem 7.4.8 *Let X,Y,Z,W be presheaves over $\mathbb{P}(s)$. If maps $f:X\to Z$ and $g:Y\to W$ are surjective open, then so is $f|_sg:X|_sY\to Z|_sW$.*

Proof: By definition $f|_sg=(||_s)_!(w^*(l_*f\times l_*g)$, but by Proposition 4.4.4, l_*f and l_*g are surjective open; by Proposition 2.2.7, $l_*f\times l_*g$ is surjective open; by Corollary 7.2.10 $w^*(l_*f\times l_*g)$ is surjective open and finally by Corollary 4.6.6, $(||_s)_!(w^*(l_*f\times l_*g)$ is surjective open. □

Using the characterisation of Theorem 7.4.7 it is not difficult to see, in analogy with what done for the restriction (cf. Theorem 7.4.3), that $|_s$ is pseudo natural in s, i.e., for every $i:s\to s'$ the following square commutes up to a natural isomorphism:

$$\begin{tikzcd}
\widehat{\mathbb{P}(s)}\times\widehat{\mathbb{P}(s)} \arrow[r, "|_s"] \arrow[d, "\mathbb{P}(i)_!\times\mathbb{P}(i)_!"'] & \widehat{\mathbb{P}(s)} \arrow[d, "\mathbb{P}(i)_!"] \\
\widehat{\mathbb{P}(s')}\times\widehat{\mathbb{P}(s')} \arrow[r, "|_{s'}"'] & \widehat{\mathbb{P}(s')}\ .
\end{tikzcd}$$

7.4.3 Replication

As we saw from the structural congruences of the π-calculus, given a process P, its replicated version $!P$ is characterised by putting

$$!P\equiv P\,|\,!P\ .$$

A different way of achieving the same effect would have been to introduce in the operational semantics 'recursive' rules like

$$\frac{P\,|\,!P \xrightarrow{\alpha} Q}{!P \xrightarrow{\alpha} Q}\ .$$

A natural way of giving the analogous presheaf operations

$$!_s : \widehat{\mathbb{P}(s)} \to \widehat{\mathbb{P}(s)}$$

is by means of an initial fixed point construction. Recall in fact that

$$|_s : \widehat{\mathbb{P}(s)} \times \widehat{\mathbb{P}(s)} \to \widehat{\mathbb{P}(s)}$$

preserves connected colimits in each argument separately. In particular then this holds for colimits of ω-chains.

Definition 7.4.9 *Let X be a presheaf over $\mathbb{P}(s)$. Define*

$$|_s^X : \widehat{\mathbb{P}(s)} \to \widehat{\mathbb{P}(s)}$$

as $|_s^X(Y) = X|_sY$ and $|_s^X(f) = 1_X|_sf$.

For what we said, the functor $|_s^X$ preserves colimits of ω-chains. Recall also that $\widehat{\mathbb{P}(s)}$ is cocomplete, hence any ω-chain of presheaves in $\widehat{\mathbb{P}(s)}$ has a colimiting cone. So the following definition is justified.

Definition 7.4.10 *Let X be a presheaf, define*

$$!X \stackrel{\text{def}}{=} \text{colim}\ |_s^{X,\omega}\ ,$$

where $|_s^{X,\omega} : \omega \to \widehat{\mathbb{P}(s)}$ is defined by

- $|_s^{X,\omega}(0) = \emptyset$, $|_s^{X,\omega}(n+1) = |_s^X(|_s^{X,\omega}n)$,
- $|_s^{X,\omega}(0 \le 1) = 0_X$, *the unique arrow from the empty presheaf to X while for any $n\rangle 0$, $|_s^{X,\omega}(n \le n+1)$ is defined inductively as $|_s^X(|_s^{X,\omega}(n-1 \le n))$.*

Since $|_s^X$ preserves colimits of ω-chains, $!X$ is a fixed point for $|_s^X$, i.e.,

$$!X \cong X|_s!X\ .$$

As usual with colimits (cf. [56]) a choice of $!X$, for every presheaf X, induces uniquely a functor, $!_s : \widehat{\mathbb{P}(s)} \to \widehat{\mathbb{P}(s)}$. Moreover since $\mathbb{P}(i)_!$ preserves colimits, $!$ is natural in s.

Proposition 7.4.11 *For every injective function $i : s \to s'$, with s and s' finite sets, the following square commutes up to a natural isomorphism:*

$$\begin{tikzcd}
\widehat{\mathbb{P}(s)} \arrow[r, "!_s"] \arrow[d, "\mathbb{P}(i)_!"'] & \widehat{\mathbb{P}(s)} \arrow[d, "\mathbb{P}(i)_!"] \\
\widehat{\mathbb{P}(s')} \arrow[r, "!_{s'}"'] & \widehat{\mathbb{P}(s')}\ .
\end{tikzcd}$$

Proof:

$$\begin{array}{rcll} \mathbb{P}(i)_!!_s & = & \mathbb{P}(i)_!\mathrm{colim}\ |_s^{-,\omega} & \text{(by definition)} \\ & \cong & \mathrm{colim}\ \mathbb{P}(i)_!|_s^{-,\omega} & \text{(since } \mathbb{P}(i)_! \text{ preserves colimits)} \\ & \cong & \mathrm{colim}\ |_s^{\mathbb{P}(i)_!(-),\omega} & \text{(by naturality of } |). \end{array}$$

□

7.5 The interpretation

We now have all the ingredients for a compositional semantics of π-calculus terms as (indexed) presheaves.

Following [126], we give the interpretation to process terms in two steps. First we associate a process P with free names in s to a presheaf $(\![P]\!)_s \in |\widehat{\mathbb{P}(s)}|$. Then later, in the full interpretation, we take account of all possible name substitutions by giving a process P with free names s a denotation as a natural transformation:

$$[\![P]\!] : N^{|s|} \stackrel{\cdot}{\longrightarrow} \mathbb{P} .$$

Definition 7.5.1 *Let s be a set of names. For π-calculus processes whose free names lie in s we inductively define:*

$$\begin{array}{lll} (\![0]\!)_s = \emptyset & (\![P+Q]\!)_s = (\![P]\!)_s + (\![Q]\!)_s & (\![[x=x]P]\!)_s = (\![P]\!)_s \\ (\![\overline{x}y.P]\!)_s = x!y.(\![P]\!)_s & (\![P \mid Q]\!)_s = (\![P]\!)_s |_s (\![Q]\!)_s & (\![[x=y]P]\!)_s = \emptyset \ \textit{if}\ (x \neq y) \\ (\![!P]\!)_s = !_s((\![P]\!)_s) & (\![\nu x\, P]\!)_s = \nu_{x \in s+\{x\}}((\![P]\!)_{s+\{x\}}) & \end{array}$$

$$(\![x(y).P]\!)_s = x?\langle F, Y\rangle \quad \textit{where } F(z) = (\![P[z/y]]\!)_s \textit{ for any } z \in s, \\ \textit{and } Y = (\![P[*_s/y]]\!)_{s+1}.$$

The following "Substitution Lemma" is fundamental to being able to index the interpretation of process terms with respect to all possible substitution of free names.

Lemma 7.5.2 (Substitution Lemma) *Let $i : s \to s'$ be an injective function between finite sets, with $\mathbf{x} = \langle x_1, x_2, \ldots, x_{|s|}\rangle$ the names in s. Then for any process P with free names in s,*

$$\mathbb{P}(i)_!((\![P]\!)_s) \cong (\![P[i(\mathbf{x})/\mathbf{x}]]\!)_{s'} .$$

Proof: See Appendix C. □

The free names of a process may be bound differently in different contexts.

Definition 7.5.3 *Let P be a process with $|s|$ free names. We define the interpretation $[\![P]\!]$ as the natural transformation $[\![P]\!] : \mathbb{N}^{|s|} \stackrel{\cdot}{\longrightarrow} \mathbb{P}$, defined as follows:*

$$\begin{array}{rrcl} [\![P]\!]_{s'} : & \overbrace{s' \times s' \cdots \times s'}^{|s|-times} & \longrightarrow\!\!\!\!\!\!+ & \mathbb{P}(s') \\ & \langle a_1, a_2, \ldots, a_{|s|}\rangle & \longmapsto & (\![P[\mathbf{a}/\mathbf{x}]]\!)_{s'} \end{array}$$

Thus the denotation of a process with free names s carries an environment $N^{|s|}$ as a parameter. The proof that this is indeed a natural transformation depends on Lemma 7.5.2 which establishes that the $(\![-]\!)$-interpretation respects name substitution. We can now show our major result, that bisimulation between processes in the π-calculus coincides with that obtained in the model via open maps.

The first two theorems establish a bisimulation between a process P with free names in s and its denotation $(\![P]\!)_s$.

Theorem 7.5.4 *Let P be a process whose free names lie in s. Then*

- $P \xrightarrow{\overline{x}y} Q$ *implies* $\exists X$ *with* $(\![P]\!)_s \xrightarrow{x!y} X$ *and* $X \cong (\![Q]\!)_s$
- $P \xrightarrow{\overline{x}(y)} Q$ *implies* $\exists X$ *with* $(\![P]\!)_s \xrightarrow{x!*_s} X$ *and* $X \cong (\![Q[*_s/y]]\!)_{s+1}$
- $P \xrightarrow{x(y)} Q$ *implies* $(\![P]\!)_s \xrightarrow{x?} \langle F, Y\rangle$ *with* $F(z) \cong (\![Q[z/y]]\!)_s$ *and* $Y \cong (\![Q[*_s/y]]\!)_{s+1}$
- $P \xrightarrow{\tau} Q$ *implies* $\exists X$ *with* $(\![P]\!)_s \xrightarrow{\tau} X$ *and* $X \cong (\![Q]\!)_s$.

Proof: See Appendix C. □

The following lemma is crucial in establishing the "converse" of Theorem 7.5.4 (Theorem 7.5.6, below).

Lemma 7.5.5 *Let P be any process term with free names in s. Then the following four facts hold:*

1. *If* $(\![!P]\!)_s \xrightarrow{x!y} X$ *then there exists a process term Q such that* $P \xrightarrow{\overline{x}y} Q$ *and* $X \cong (\![Q|!P]\!)_s$.
2. *If* $(\![!P]\!)_s \xrightarrow{x!*_s} X$ *then there exist a process term Q and a fresh new name y such that* $P \xrightarrow{\overline{x}(y)} Q$ *and* $X \cong (\![Q[*_s/y]|!P]\!)_{s+1}$.
3. *If* $(\![!P]\!)_s \xrightarrow{x?} \langle F, Y\rangle$ *then there exist a process term Q and a fresh new name y such that* $P \xrightarrow{x(y)} Q$ *and for every* $z \in s$, $F(z) \cong (\![Q[z/y]|!P]\!)_s$ *and* $Y \cong (\![Q[*_s/y]|!P]\!)_{s+1}$.
4. *If* $(\![!P]\!)_s \xrightarrow{\tau} X$ *then (at least) one of the following holds:*
   - *There exists a process term Q such that* $P \xrightarrow{\tau} Q$ *and* $X \cong (\![Q|!P]\!)_s$.
   - *There exist process terms Q and R and names* $x, y, z \in s$ *such that*

     $$P \xrightarrow{x(y)} Q \qquad \text{and} \qquad P \xrightarrow{\overline{x}z} R$$

     and $X \cong (\![Q[z/y]|R|!P]\!)_s$.
   - *There exist process terms Q and R and a name $x \in s$ such that*

     $$P \xrightarrow{x(*_s)} Q \qquad \text{and} \qquad P \xrightarrow{\overline{x}(*_s)} R$$

     and $X \cong (\![\nu *_s(Q|R)|!P]\!)_s$.

Proof: The proof is by induction on the structure of P. It uses obviously the fact that

$$(\![!P]\!)_s \stackrel{\text{def}}{=} !_s((\![P]\!)_s) ,$$

i.e., that the denotation of $!P$ is obtained as the colimit of an ω-chain in a presheaf category, where colimits are computed pointwise. We spell out the part of the proof that concerns the τ move, as this is the most delicate bit since it might involve communication of two copies of the process. Suppose then that $(\![!P]\!)_s \xrightarrow{\tau} X$. Then by definition, there exists $x \in (\![!P]\!)_s(\tau)$ and hence there exists an n (that we can choose to be the least such) and an $x' \in (\![\overbrace{(P|P|P|\ldots|P)}^{n \text{ times}}]\!)_s(\tau)$, that is sent to x in the colimiting diagram. By definition this means that there exists an X' such that

$$\overbrace{(P|P|P|\ldots|P)}^{n \text{ times}} \xrightarrow{\tau} X'$$

and that moreover X is the colimit of the chain

$$X' \longrightarrow X'|_s(\![P]\!)_s \longrightarrow (X'|_s(\![P]\!)_s)|_s(\![P]\!)_s \longrightarrow ((X'|_s(\![P]\!)_s)|_s(\![P]\!)_s)|_s(\![P]\!)_s \longrightarrow \cdots .$$

Because of the characterisation of parallel composition of Theorem 7.4.7 and since n is chosen least, we have that (at least) one of the following cases must hold:

1. There exists a presheaf X'' such that

$$(\![P]\!)_s \xrightarrow{\tau} X'' \qquad \text{and} \qquad X' \cong (\![\overbrace{(P|P|P|\ldots|P)}^{n-1 \text{ times}}]\!)_s|_s X'' .$$

2. There exist names $x, z \in s$, presheaf X'' and a pair $\langle F, Y\rangle$ such that

$$(\![\overbrace{(P|P|P|\ldots|P)}^{n-1 \text{ times}}]\!)_s \xrightarrow{x!z} X'' \qquad \text{and} \qquad (\![P]\!)_s \xrightarrow{x?} \langle F, Y\rangle$$

   and $X' \cong X''|_s F(z)$.

3. There exist a name $x \in s$, presheaf X'' and a pair $\langle F, Y\rangle$ such that

$$(\![\overbrace{(P|P|P|\ldots|P)}^{n-1 \text{ times}}]\!)_s \xrightarrow{x!*_s} X'' \qquad \text{and} \qquad (\![P]\!)_s \xrightarrow{x?} \langle F, Y\rangle$$

   and $X' \cong X''|_{s+1} Y$.

4. The symmetric of the point 2 and 3 above.

In case of 1, then we obviously can also deduce, since parallel composition is symmetric that

$$X' \cong X''|_s(\![\overbrace{(P|P|P|\ldots|P)}^{n-1 \text{ times}}]\!)_s .$$

In particular we then have the following chains of commutative diagrams (writing P^n for the P in parallel with itself n times):

$$\begin{tikzcd}
\llparenthesis P^{n-1} \rrparenthesis_s |_s X'' \arrow[r] \arrow[d] & (\llparenthesis P^{n-1} \rrparenthesis_s |_s X'')|_s \llparenthesis P \rrparenthesis_s \arrow[r] \arrow[d] & (((\llparenthesis P^{n-1} \rrparenthesis_s |_s X'')|_s \llparenthesis P \rrparenthesis_s) \llparenthesis P \rrparenthesis_s \arrow[r] \arrow[d] & \cdots \\
X''|_s \llparenthesis P^{n-1} \rrparenthesis_s \arrow[r] & X''|_s \llparenthesis P^{n} \rrparenthesis_s \arrow[r] & X''|_s \llparenthesis P^{n+1} \rrparenthesis \arrow[r] & \cdots ,
\end{tikzcd}$$

where the vertical arrows are always isomorphisms obtained by application of the symmetry and associativity of $|_s$. For what said above, the colimit of the upper chain is X, while that of the lower chain is $X''|_s \llparenthesis !P \rrparenthesis_s$, since $|_s$ preserves connected colimits on each argument. Hence

$$X \cong X''|_s \llparenthesis !P \rrparenthesis_s \ .$$

We now have the inductive hypothesis saying that there exists a Q such that $P \xrightarrow{\tau} Q$ and $X'' \cong \llparenthesis Q \rrparenthesis_s$, hence

$$X \cong \llparenthesis Q \rrparenthesis_s |_s \llparenthesis !P \rrparenthesis_s = \llparenthesis Q|!P \rrparenthesis_s \ .$$

In case 2, we know from the decomposition result of parallel composition that there must exist an X''' such that $\llparenthesis P \rrparenthesis_s \xrightarrow{x!z} X'''$ and

$$X'' \cong ((\llparenthesis P^{j_1} \rrparenthesis_s |_s X''')|_s \llparenthesis P^{j_2} \rrparenthesis_s \ ,$$

with $j_1 + j_2 = n - 2$. Again because of symmetry and associativity of $|_s$ we can assume that such a P is the first in the list, i.e., we have a chain of commutative diagrams whose vertical arrows are isomorphisms:

$$\begin{tikzcd}
((\llparenthesis P^{j_1} \rrparenthesis_s |_s X''')|_s \llparenthesis P^{j_2} \rrparenthesis_s \arrow[r] \arrow[d] & (((\llparenthesis P^{j_1} \rrparenthesis_s |_s X''')|_s \llparenthesis P^{j_2} \rrparenthesis_s) \llparenthesis P \rrparenthesis_s \arrow[r] \arrow[d] & \cdots \arrow[r] \arrow[d] & \cdots \arrow[d] \\
X'''|_s \llparenthesis P^{n-2} \rrparenthesis_s \arrow[r] & X'''|_s \llparenthesis P^{n-1} \rrparenthesis_s \arrow[r] & X'''|_s \llparenthesis P^{n} \rrparenthesis_s \arrow[r] & \cdots .
\end{tikzcd}$$

Then, parallel composing with $F(z)$, we have

$$\begin{tikzcd}
(((\llparenthesis P^{j_1} \rrparenthesis_s |_s X''')|_s \llparenthesis P^{j_2} \rrparenthesis_s)|_s F(z) \arrow[r] \arrow[d] & ((((\llparenthesis P^{j_1} \rrparenthesis_s |_s X''')|_s \llparenthesis P^{j_2} \rrparenthesis_s) \llparenthesis P \rrparenthesis_s)|_s F(z) \arrow[r] \arrow[d] & \cdots \arrow[d] \\
(X'''|_s \llparenthesis P^{n-2} \rrparenthesis_s)|_s F(z) \arrow[r] & (X'''|_s \llparenthesis P^{n-1} \rrparenthesis_s)|_s F(z) \arrow[r] & \cdots .
\end{tikzcd}$$

Now by definition, the colimit of the upper chain is X. Again by symmetry and associativity we also have the following chain of commutative diagrams whose vertical arrows are isomorphisms:

$$\begin{tikzcd}
(X'''|_s \llparenthesis P^{n-2} \rrparenthesis_s)|_s F(z) \arrow[r] \arrow[d] & (X'''|_s \llparenthesis P^{n-1} \rrparenthesis_s)|_s F(z) \arrow[r] \arrow[d] & (X'''|_s \llparenthesis P^{n} \rrparenthesis) F(z) \arrow[r] \arrow[d] & \cdots \\
(X'''|_s F(z))|_s \llparenthesis P^{n-2} \rrparenthesis_s \arrow[r] & (X'''|_s F(z))|_s \llparenthesis P^{n-1} \rrparenthesis_s \arrow[r] & (X'''|_s F(z))|_s \llparenthesis P^{n} \rrparenthesis_s \arrow[r] & \cdots .
\end{tikzcd}$$

The upper chain of this diagram is the lower chain of the above one and the lower chain has colimit given by $(X'''|_s F(z))|_s (\![!P]\!)_s$. Hence $X \cong (X'''|_s F(z))|_s (\![!P]\!)_s$. By inductive hypothesis, there exist Q and y such that $P \xrightarrow{x(y)} Q$ and $(\![Q[z/y]]\!)_s \cong F(z)$ and there exists R such that $P \xrightarrow{\overline{x}z} R$ and $(\![R]\!)_s \cong X'''$. Therefore

$$\begin{array}{rcl} X & \cong & (X'''|_s F(z))|_s (\![!P]\!)_s \\ & \cong & ((\![R]\!)_s|_s (\![Q[z/y]]\!)_s)|_s (\![!P]\!)_s \\ & \cong & ((\![Q[z/y]]\!)_s|_s (\![R]\!)_s)|_s (\![!P]\!)_s \\ & \cong & (\![Q[z/y]|R|!P]\!)_s \ . \end{array}$$

The other cases are treated similarly to this last one. □

Theorem 7.5.6 *Let P be a process whose free names lie in s. Then*

- $(\![P]\!)_s \xrightarrow{x!y} X$ *implies* $\exists Q$ *with* $P \xrightarrow{\overline{x}y} Q$ *and* $(\![Q]\!)_s \cong X$
- $(\![P]\!)_s \xrightarrow{x!*_s} X$ *implies* $\exists Q, y$ *with* $P \xrightarrow{\overline{x}(y)} Q$ *and* $(\![Q[*_s/y]]\!)_{s+1} \cong X$
- $(\![P]\!)_s \xrightarrow{x?} \langle F, Y\rangle$ *implies* $\exists Q, y$ *with* $P \xrightarrow{x(y)} Q$ *and* $(\![Q[*_s/y]]\!)_{s+1} \cong Y$ *and* $F(z) \cong (\![Q[z/y]]\!)_s$
- $(\![P]\!)_s \xrightarrow{\tau} X$ *implies* $\exists Q$ *with* $P \xrightarrow{\tau} Q$ *and* $(\![Q]\!)_s \cong X$.

Proof:[Sketch] The proof of this is by structural induction on the structure of the process term P and it depends mainly on the characterisation Theorems 7.4.2 and 7.4.7. We omit the details but for the treatment of replication that exemplify the most complicated case, i.e., when $(\![!P]\!)_s \xrightarrow{\tau} X$. In this case there are three possibilities as shown in Lemma 7.5.5:

1. $(\![P]\!)_s \xrightarrow{\tau} X'$ with $X \cong X'|_s(\![!P]\!)_s$. Then by inductive hypothesis $P \xrightarrow{\tau} R$ and $(\![R]\!)_S \cong X'$. By the structural congruence $!P \equiv P\,|\,!P$, one has that $!P \xrightarrow{\tau} R\,|\,!P$ and
$$(\![R\,|\,!P]\!)_S \stackrel{\text{def}}{=} (\![R]\!)_s|_s(\![!P]\!)_s \cong X'|_s(\![!P]\!)_s \cong X \ .$$
2. $(\![P]\!)_s \xrightarrow{x?} \langle F, Y\rangle$ and $(\![P]\!)_s \xrightarrow{x!y} X'$ for some $x, y \in s$ and $X \cong (F(y)|_s X')|_s(\![!P]\!)_s$. By inductive hypothesis,
$$P \xrightarrow{x(w)} Q,\ P \xrightarrow{\overline{x}y} R,\ (\![Q[y/w]]\!)_s \cong F(y) \,\text{and}\, (\![R]\!)_s \cong X' \ .$$
Again the structural congruence defining $!P$ implies (by application of the rule COM) that
$$!P \xrightarrow{\tau} (Q[y/w]|R)|!P$$
and the above induces that
$$(\![Q[y/w]|R)|!P]\!)_s \cong (F(y)|_s X')|_s(\![!P]\!)_s \ .$$
3. The last case deals with a bound output, and it is treated as the one right above where one uses the rule CLOSE instead of COM.

$\square$

Using these results we can relate the (late) bisimulation for processes with the late bisimulation for presheaves.

Lemma 7.5.7 *Let P and Q be two π-calculus processes with free names in s. Then P is bisimilar to Q (Definition 7.1.1) if and only if $(\!\![P]\!\!)_s$ is late bisimilar to $(\!\![Q]\!\!)_s$ (Definition 7.3.11).*

Proof: If $(R_s)_{s\in\mathcal{I}}$ is a late bisimulation such that $(\!\![P]\!\!)_s \, R_s \, (\!\![Q]\!\!)_s$. Define $\mathcal{S}$ to be the relation on closed process terms given by

$$A \, \mathcal{S} \, B \text{ iff } \exists\, s \, \exists X, Y \in \widehat{\mathbb{P}(s)} \; (\!\![A]\!\!)_s \cong X \, R_s \, Y \cong (\!\![B]\!\!)_s \ .$$

Theorem 7.5.4 and 7.5.6 now immediately implies that $\mathcal{S}$ is a bisimulation that moreover includes the pair (P, Q) by definition.
Vice versa, if $\mathcal{S}$ is a bisimulation such that $P \, \mathcal{S} \, Q$, define the family of R_s's as follows:

$$A \, R_s \, B \text{ iff } \exists P', Q' \text{ such that } fn(P') \cup fn(Q') \subseteq s \text{ and } P' \, \mathcal{S} \, Q' \text{ and } A \cong (\!\![P']\!\!)_s \text{ and } B \cong (\!\![Q']\!\!)_s.$$

Again from Theorem 7.5.4 and 7.5.6 one immediately deduces that $(R_s)_{s\in\mathcal{I}}$ is a late bisimulation and by definition one has that $(\!\![P]\!\!)_s \, R_s \, (\!\![Q]\!\!)_s$. $\square$

Note that in the second part of the above proof we have actually obtained a family of "large" relations in the sense that each R_s will not typically be a (small) set but a proper class since for each pair of presheaves in the relation we also have all its isomorphic pairs. This can be reduced by restricting to consider only the presheaves reachable from the interpretation of P and Q, but this was an unnecessary complication for the purpose of the proof above.

The Lemma 7.5.7 and the Theorem 7.3.14 induce the following correspondence between bisimulation in the π-calculus and open map bisimulation.

Theorem 7.5.8 *Let P and Q be two π-calculus processes with free names in s. Then P is late bisimilar to Q if and only if $(\!\![P]\!\!)_s$ and $(\!\![Q]\!\!)_s$ are connected by a span of surjective open maps.*

Suppose now that P is a π-calculus process with free names s_P. Then for any larger set of names s, an injection $i : s_P \to s$ induces a natural transformation $\pi^{s_P,i} : \mathbb{N}^{|s|} \stackrel{.}{\to} \mathbb{N}^{|s_P|}$ that projects $|s|$-tuples of names to $|s_P|$-tuples. When i is simply an inclusion and no confusion arises we write this as π^{s_P}. With the above notation the Substitution Lemma induces the following:

Theorem 7.5.9 *Let P and Q be two π-calculus processes with free names s_P and s_Q respectively. Take $s_{P,Q}$ to be the union $s_P \cup s_Q$. Then P is late equivalent (bisimulation congruent) to Q if and only if for any finite set s and any $|s_{P,Q}|$-tuple $\mathbf{a}$ of elements of s, $[\![P]\!]_s \pi_s^{s_P}(\mathbf{a})$ and $[\![Q]\!]_s \pi_s^{s_Q}(\mathbf{a})$ are connected by a span of surjective open maps.*

Note that it is sufficient here to take s to be exactly the free names $s_{P,Q}$ of the two processes. We can also present this result using the 2-categorical setting of our model:

Corollary 7.5.10 *Let P and Q be two π-calculus processes with free names s_P and s_Q respectively. Then P is late equivalent to Q if and only if $[\![P]\!] \circ \pi^{s_P}$ and $[\![Q]\!] \circ \pi^{s_Q}$ are connected by a span of modifications whose components are surjective open maps.*

7.6 Late *vs.* early

We have given the π-calculus here in its late version, where a process $x(y).P$ carries out input in two stages: it first synchronises with another process that is prepared to send on channel x; then, later, the transmitted value is substituted for y in the body of P. There is an alternative *early* semantics where these two steps happen together and processes synchronise on (channel,value) pairs. The operational consequences of this choice are discussed in [88, 89]. There is a corresponding early bisimulation '$\dot{\sim}_E$' and early equivalence '$\sim_E$', which are both strictly coarser than their late forms.

We can follow these late and early alternatives in our denotational semantics. In presheaf models, synchronisation points are marked by lifting $(-)_\perp$ in the equation for the path category. An early version of (7.5) would be

$$In_E \;=\; \mathbb{N} \otimes (\mathbb{N} \multimap \mathbb{P}_\perp)\;. \tag{7.6}$$

This means that instead of paths $x?$, $x?(y{\mapsto}p)$ and $x?(*{\mapsto}p')$ we now have $x?y$, $x?*$, $x?y.p$ and $x?*.p'$. Still the action of In_E on functions $i : s \to s'$ will be driven by the function space $\multimap$, i.e.,

$$\mathbb{N} \otimes (\mathbb{N} \multimap \mathbb{P}_\perp)(i)(x?*.p') = \sum_{z \notin Im\, i} i(x)?z.\mathbb{P}([i,z])(p') + i(x)?*.\mathbb{P}(i+1)(p')\;.$$

Solving this new equation in $\mathbf{Prof}^{\mathcal{I}}$ gives an object $\mathbb{P}_E$, but observe that, unlike before, now, not all path objects will be denoted by process terms. In fact, for example, path objects like $x?y.p$ should be the denotation of a process term that can receive an input uniquely along the channel x and only if the input is y. In the terminology of [89] $x?y.p$ wants to act as a process which is capable of performing only a *free input* action xy and this is clearly not expressible in the language.

The definition of the restriction (Section 7.4.1) is easily adapted in the relevant clauses by

$$\begin{array}{rcl}
\nu_{y\in s}(x?z) & = & \left\{\begin{array}{ll} x?z & \text{if } x \neq y \neq z \text{ and } z \in s+1 \\ \emptyset & \text{otherwise} \end{array}\right. \\
\nu_{y\in s}(x?z.p) & = & \left\{\begin{array}{ll} x?z\nu_{y\in s}p & \text{if } x \neq y \neq z \neq * \text{ and } z \in s+1 \\ x?*\nu_{y\in s+1}p & \text{if } x \neq y \text{ and } z = * \\ \emptyset & \text{otherwise}\;. \end{array}\right.
\end{array}$$

Slightly longer is the treatment of parallel composition (that includes more cases than before).

Definition 7.6.1 *For every finite set s, with elements indicated with the letters x, y, z, define the symmetric profunctor*

$$||_s^E : \mathbb{P}_E(s)_\perp \times \mathbb{P}_E(s)_\perp \dashrightarrow \mathbb{P}_E(s)$$

by simultaneous induction, for all sets s, on the structure of the path objects as in Figure 7.3 (omitting the obvious Yoneda embeddings).

As before the parallel composition for presheaves is obtained by precomposing with w^* and $l_* \times l_*$.

Definition 7.6.2 (Parallel Composition) *For every finite set s define the parallel composition functor*

$$|_s^E \stackrel{\mathrm{def}}{=} (||_s^E)_! \circ w_{\mathbb{P}_E(s)_\perp, \mathbb{P}_E(s)_\perp} \circ (l_* \times l_*) \ .$$

Replication is defined again using a least fixed point construction, iterating the parallel composition of a presheaf with itself. We do not go through all the details, the interpretation and the results analogous to Theorem 7.5.4 and 7.5.6 for establishing the correspondence of the early bisimulation with open map bisimulation in $\mathbb{P}_E$; instead we prefer to remark on the existence of an arrow in $\mathbf{Prof}^{\mathcal{I}}$ mapping the late interpretation onto the early one (cf. Section 5.2).

Definition 7.6.3 *For every finite set s, define the functor $k_s : \mathbb{P}_E(s) \to \mathbb{P}(s)$ inductively as follows:*

$$k_s(\tau) = \tau \qquad k_s(x!y) = x!y \qquad k_s(x!*) = x!*$$
$$k_s(x?y) = x? \qquad k_s(x?*) = x?$$

$$k_s(\tau.p) = \tau.k_s(p) \qquad k_s(x!y.p) = x!y.k_s(p) \qquad k_s(x!*.p') = x!*.k_{s+1}(p')$$
$$k_s(x?y.p) = x?(y \mapsto k_s(p)) \quad k_s(x?*.p') = x?(* \mapsto k_{s+1}p') \ .$$

Clearly this family of functors induce two pseudo natural transformations $k_! : \mathbb{P}_E \dot{\to} \mathbb{P}$ and

$$k^* : \mathbb{P} \dot{\to} \mathbb{P}_E \tag{7.7}$$

that are pointwise described by, $(k_s)_!$ and $(k_s)^*$, respectively. Being arrows in $\mathbf{Prof}^{\mathcal{I}}$, both the pseudo natural transformations preserves open map bisimulation pointwise, moreover k^* maps the late interpretation onto the early one.

Theorem 7.6.4 *If we write $(\![\cdot]\!)^E$ for the interpretation of π-calculus processes in $\mathbb{P}_E$, then for any finite set s and any process term P with free names in s, the following hold:*

$$k_s^*(\![P]\!)_s \cong (\![P]\!)_s^E \ .$$

Proof:[Sketch] The proof is a straightforward induction on the structure of P once we have shown that for any s the following three things hold:

$$
\begin{array}{rcl}
\bot ||_s^E \bot & = & \emptyset \\
\lfloor p \rfloor ||_s^E \bot & = & p \\
\lfloor In_\alpha(q) \rfloor ||_s^E \lfloor In_{\alpha'}(r) \rfloor & = & \alpha.(\lfloor q \rfloor ||_s^E \lfloor In_{\alpha'}(r) \rfloor) + \alpha'.(\lfloor In_\alpha(q) \rfloor ||^E \lfloor r \rfloor) \\
\\
\lfloor In_\alpha(q) \rfloor ||_s^E \lfloor In_{x'!*}(r) \rfloor & = & \alpha.(\lfloor q \rfloor ||_s^E \lfloor In_{x'!*}(r) \rfloor) + x'!*.(\lfloor \mathbb{P}_E(i)(In_\alpha(q)) \rfloor (||_{s+1}^E)_! \lfloor r \rfloor) \\
\\
\lfloor In_\tau(q) \rfloor ||_s^E \lfloor In_{x?y}(r) \rfloor & = & \tau.(\lfloor q \rfloor ||_s^E \lfloor In_{x?y}(r) \rfloor + x?y.(\lfloor In_\tau(q) \rfloor ||_s^E \lfloor r \rfloor) \\
\\
\lfloor In_\tau(q) \rfloor ||_s^E \lfloor In_{x?*}(r) \rfloor & = & \tau.(\lfloor q \rfloor ||_s^E \lfloor In_{x?*}(r) \rfloor + x?*.(\lfloor \mathbb{P}_E(i) In_\tau(q) \rfloor ||_{s+1}^E \lfloor r \rfloor) \\
\\
\lfloor In_{x!y}(q) \rfloor ||_s^E \lfloor In_{x'?y'}(r) \rfloor & = & \left\{ \begin{array}{ll}
x!y.(\lfloor q \rfloor ||_s^E \lfloor In_{x'?y'}(r) \rfloor) & \\
+x'?y'.(\lfloor In_{x!y}(q) \rfloor ||_s^E \lfloor r \rfloor) & \text{if } x \neq x' \text{ or } y \neq y' \\
\tau.(\lfloor q \rfloor ||_s^E \lfloor r \rfloor) & \\
+x!y.(\lfloor q \rfloor ||_s^E \lfloor In_{x'?y'}(r) \rfloor) & \\
+x'?y'.(\lfloor In_{x!y}(q) \rfloor ||_s^E \lfloor r \rfloor) & \text{otherwise}
\end{array} \right. \\
\\
\lfloor In_{x!y}(q) \rfloor ||_s^E \lfloor In_{x'?*}(r) \rfloor & = & x!y.(\lfloor q \rfloor ||_s^E \lfloor In_{x'?*}(r) \rfloor) + x'?*.(\lfloor \mathbb{P}_E(i) In_{x!y}(q) \rfloor (||_{s+1}^E)_! \lfloor r \rfloor) \\
\\
\lfloor In_{x!*}(q) \rfloor ||_s^E \lfloor In_{x'?y'}(r) \rfloor & = & x!*.(\lfloor q \rfloor (||_{s+1}^E)_! \mathbb{P}_E(i) \lfloor In_{x'?y'}(r) \rfloor) + x'?y'.(\lfloor In_{x!*}(q) \rfloor ||_s^E \lfloor r \rfloor) \\
\\
\lfloor In_{x!*}(q) \rfloor ||_s^E \lfloor In_{x'?*}(r) \rfloor & = & \left\{ \begin{array}{ll}
x!*.(\lfloor q \rfloor (||_{s+1}^E)_! \lfloor \mathbb{P}_E(i)(In_{x'?*}(r))) \rfloor & \\
+x'?*.(\lfloor \mathbb{P}_E(i) In_{x!*}(q) \rfloor ||_{s+1}^E \lfloor r \rfloor) & \text{if } x \neq x' \\
\tau.(\nu_{*\in s+1,!}(\lfloor q \rfloor ||_{s+1}^E \lfloor r \rfloor)) & \\
+x!*.(\lfloor q \rfloor (||_{s+1}^E)_! \lfloor \mathbb{P}_E(i)(In_{x'?*}(r)) \rfloor) & \\
+x'?*.(\lfloor \mathbb{P}_E(i)(In_{x!*}(q)) \rfloor (||_{s+1}^E)_! \lfloor r \rfloor) & \text{otherwise}
\end{array} \right. \\
\\
\lfloor In_{x?y}(q) \rfloor ||_s^E \lfloor In_{x'?y'}(r) \rfloor & = & x?y.(\lfloor q \rfloor ||_s^E \lfloor In_{x'?y'}(r) \rfloor) + x'?y'.(\lfloor In_{x?y} q \rfloor ||_s^E \lfloor r \rfloor) \\
\\
\lfloor In_{x?y}(q) \rfloor ||_s^E \lfloor In_{x'?*}(r) \rfloor & = & x?y.(\lfloor q \rfloor ||_s^E \lfloor In_{x'?*}(r) \rfloor) + x'?*.(\lfloor \mathbb{P}_E(i) In_{x?y} q \rfloor (||_{s+1}^E)_! \lfloor r \rfloor) \\
\\
\lfloor In_{x?*}(q) \rfloor ||_s^E \lfloor In_{x'?*}(r) \rfloor & = & x?*.(\lfloor q \rfloor (||_{s+1}^E)_! \lfloor \mathbb{P}_E(i) In_{x'?*}(r) \rfloor) \\
& & +x'?*.(\lfloor \mathbb{P}_E(i) In_{x?*} q \rfloor (||_{s+1}^E)_! \lfloor r \rfloor)
\end{array}
$$

for $\alpha, \alpha' \in \{\tau\} \cup \{x!y \mid x, y \in s\}$, $i : s \hookrightarrow s + 1$ the obvious inclusion.
The action on morphisms, i.e., on the lesser or equal than relation, is inductively determined as well in the obvious way.

Figure 7.3: 'Early' parallel composition

1. $k_s^* \nu_{y \in s+\{y\}} \cong \nu^E_{y \in s+\{y\}} k^*_{s+\{y\}}$
2. $k_s^*(||_s)_! \cong (||^E_s)_!(k^*_\perp \mathrm{y}_{\mathbb{P}(s)_\perp} \times k^*_\perp \mathrm{y}_{\mathbb{P}(s)_\perp})_!$
3. $k_s^* !_s \cong !^E_s k^*_s$,

where, naturally, the upper index E stands for the corresponding semantical construct in $\mathbb{P}_E$. To prove (1), since all the functors are colimit preserving, it is enough to check it for the representables and this is done by induction on the structure of the paths. The base cases are all trivial but for $p = x?$, $x \neq y$ that requires unwinding the definition of k^* in a non completely trivial case. In fact

$$\begin{aligned}
k_s^* \nu_{y \in s+\{y\}}(x?) &= k_s^*(x?) \\
&= \sum_{w \in s+1} x?w \\
&= \sum_{w \in s+1} \nu^E_{y \in s+\{y\}} x?w \\
&= \nu^E_{y \in s+\{y\},!}(\sum_{w \in s+1} x?w + x?y) \\
&= \nu^E_{y \in s+\{y\},!}(k^*_{s+\{y\}}(x?)) \ .
\end{aligned}$$

Similarly for the inductive step the only "interesting" case (even if it does not make use of the inductive hypothesis) is given by $x?(y \mapsto p)$ that goes as follows:

$$\begin{aligned}
k_s^* \nu_{y \in s+\{y\}}(x?(y \mapsto p)) &= k_s^*(x?) \\
&= \sum_{w \in s+1} x?w \\
&= \sum_{w \in s+1} \nu^E_{y \in s+\{y\}} x?w \\
&= \nu^E_{y \in s+\{y\},!}(\sum_{w \in s+1} x?w + x?y.p)) \\
&= \nu^E_{y \in s+\{y\},!}(k^*_{s+\{y\}}(x?(y \mapsto p))) \ .
\end{aligned}$$

Also for point (2) is enough to check things on the representables and the inductive proof is a tedious verification that all the definitions agrees. Finally point (3) follows directly from point (2) since ! is always defined in term of the (replicated versions of) diagonals an parallel composition. □

We can even move the synchronisation point for output: clause (7.4) has a later variant

$$Out_L = \mathbb{N} \otimes (\mathbb{N} \otimes \mathbb{P})_\perp \ . \tag{7.8}$$

It turns out that this makes no difference to process bisimilarity, but it does correspond closely to the presentation style of [84]. There processes synchronise on channel names alone, $P \xrightarrow{\overline{x}} C$ or $P \xrightarrow{x} F$, becoming *concretions* C (name-process pairs, $\mathbb{N} \otimes \mathbb{P}$) and *abstractions* F (name-to-process functions, $\mathbb{N} \multimap \mathbb{P}$) respectively. Actual communication is represented by the application of abstractions to concretions $F \bullet C$.

We believe that the domain models of the π-calculus in [32, 126] do not cover the early version, chiefly because rearrangements like equation (7.6) are harder to express. There the domain equation for processes uses the Plotkin powerdomain to mark synchronisation; while our equation for paths uses the much simpler lifting operation.

7.7 Other π-calculi

The development of our model has been purely denotational, with no operational manipulation of processes through expansion laws or the like. As a consequence, there are no *required* operators in the language, and the model remains valid for any subset of the π-calculus. Even so, particular sublanguages may fit simpler equations. For example, the asynchronous π-calculus of [18] constrains output to the form $\overline{x}y.0$, suggesting the clause

$$Out_A = \mathbb{N} \otimes \mathbb{N} \tag{7.9}$$

to replace (7.4). The πI-calculus of [117] allows only bound output $\overline{x}(y).P$, equivalent to $\nu y(\overline{x}y.P)$ in the original π-calculus. Every communication now passes a fresh name, and we would replace (7.5) and (7.4) with

$$In_I = Out_I = \mathbb{N} \otimes \delta\mathbb{P} . \tag{7.10}$$

Moreover the morphism $\mathbb{P}(i)$ now arises from the category map e_i introduced just before Proposition 7.3.17, with the restrictions $\nu(i)$ from the end of Section 7.4.1 being $e_i^\star$, the left inverse, and now also right adjoint, to $\mathbb{P}(i)_!$. This gives some support to Sangiorgi's claim that the πI-calculus is a simpler, more symmetric version of the π-calculus.

These examples show the flexibility of our approach by drawing on the rich categorical structure of $\mathbf{Prof}^{\mathcal{I}}$. As ever in category theory, this also leads us to look at the maps *between* models: we hope to find further morphisms like (7.7), from 'late' to 'early', that might tie together the wide selection of customised π-calculi proposed in recent years.

As we have already mentioned we do not have a general function space in $\mathbf{Prof}^{\mathcal{I}}$. Where it does exist we are no longer be constrained to passing just names or other ground values. Process-passing systems like CHOCS [130] or even the full higher-order π-calculus of [116] could then fit into our framework. The difficulties here lie not just in writing plausible equations, but also in extracting their operational content to see if the semantics and bisimilarities that arise fit any existing scheme as we shall see in the next chapter.

Chapter 8

Higher Order Processes

In this chapter we begin the investigation of presheaf models for process passing languages. Our motivation is that of understanding operationally the bisimulation induced on the language terms by open maps. Unfortunately we shall not quite reach our objective here, but we believe that the results that we are presenting are a first step towards it. We shall concentrate on a **CCS**-like process calculus. By this we mean that we assume the existence of a fixed set of channels and we discard the possibility of communicating and creating new channels as in the π-calculus (cf. Chapter 7). This restriction is imposed on us since, with our present knowledge, a treatment of higher order and name passing at the same time would require considering function spaces in the indexed category $\mathbf{Prof}^{\mathcal{I}}$. But as we said in Chapter 7, we do not believe that for all $\mathbb{C} : \mathcal{I} \to \mathbf{Prof}$ there exists a functor $\mathbb{C} \multimap -$ right adjoint to $\mathbb{C} \otimes -$.

Moreover, since the function space that we are using is of a "linear" kind, we further constrain our process term with a linearity condition that ensures that if in a term there are several occurrences of the same variable, at any particular time, only one of these can be active.

The operational reading of this constraint, roughly speaking, is that when a process is received as input, it can be run at most once.

The main difficulty with respect to the previous case studies (cf. Chapters 5 and 7), lies in defining the operational semantics in a way that takes into account the elements of the denoted presheaves. The reason for this is that the elements of presheaves seem crucial in characterising operationally open map bisimulation for abstractions. In this chapter we shall work towards an identification of the elements of the presheaf denotation with the derivation trees according to the operational semantics.

8.1 The 2-category Conn

We begin the chapter with a brief introduction to the 2-category of presheaf categories and connected colimit preserving functors to motivate our choice of function space (to model abstractions) and tensor (to model concretions). This is based on the experience accumulated in Chapters 5 and 7 which suggests that all the main operations used to model term constructors are connected colimit preserving (possibly on each argument separately, in case of many sorted operations like parallel composition). Then if we wish to represent higher order communication as a form of function application as we hinted in Section 5.2 and 7.6 for the "ground values" case (cf. [84, 85, 118]), it is natural to look in our setting at this 2-category and at its relation with our main category of domains, **Cocont**.

Definition 8.1.1 (Conn) *Define* **Conn** *to be the* 2*-category of:*

Objects: *Small categories,* $\mathbb{C}, \mathbb{D}, \mathbb{E} \dots$ *.*

Arrows: *Connected colimit preserving functors,* $F : \widehat{\mathbb{C}} \to \widehat{\mathbb{D}}$*.*

2**-cells:** *Natural transformations.*

Recall that already in Chapter 4 we remarked on the equivalence of categories

$$\mathbf{Conn}(\mathbb{C}, \mathbb{D}) \simeq \mathbf{Cocont}(\mathbb{C}_\perp, \mathbb{D})$$

In fact **Conn** arises as the co-Kleisli 2-category for the 2-comonad on **Cocont** induced by lifting $(-)_\perp$.

Proposition 8.1.2 *The* 2*-functor*[1]

$$(-)_\perp : \mathbf{Cocont} \longrightarrow \mathbf{Cocont}$$

equipped with counit, ε*, and comultiplication,* δ*, as defined below is a* 2*-comonad.*

- *The* 2*-natural transformation* $\varepsilon : (-)_\perp \to 1_{\mathbf{Cocont}}$ *is given, for any small category* $\mathbb{C}$*, by* $l^*_{\mathbb{C}} : \widehat{\mathbb{C}_\perp} \to \widehat{\mathbb{C}}$*, where* $l : \mathbb{C} \to \mathbb{C}_\perp$ *is the obvious inclusion*
- *The comultiplication* $\delta : (-)_\perp \to ((-)_\perp)_\perp$ *is given by* $\pi^*_{\mathbb{C}}$ *where* $\pi_{\mathbb{C}} : (\mathbb{C}_\perp)_\perp \to \mathbb{C}_\perp$ *is the projection functor that identifies the two bottom elements.*

Proposition 8.1.3 Conn *is pseudo equivalent to the co-Kleisli category*[2] *for* $((-)_\perp, \delta, \varepsilon)$*.*

Proof: The co-Kleisli category $(coKl_\perp)$ has the same objects of **Cocont**, arrows the colimit preserving functors $F : \widehat{\mathbb{C}_\perp} \to \widehat{\mathbb{D}}$ and 2-cells the natural transformations. But now the equivalence

$$\mathbf{Conn}(\mathbb{C}, \mathbb{D}) \simeq \mathbf{Cocont}(\mathbb{C}_\perp, \mathbb{D})$$

immediately induces the pseudo equivalence

$$\mathbf{Conn} \simeq coKl_\perp$$

that we are seeking. □

So we have an embedding of **Cocont** into **Conn** that locally induces a reflection

$$\mathbf{Cocont}(\mathbb{C}, \mathbb{D}) \underset{\longrightarrow}{\overset{\longleftarrow}{\perp}} \mathbf{Conn}(\mathbb{C}, \mathbb{D}) \ .$$

[1] While generally the functors like $\otimes$ or $(-)^*$ are pseudo, $(-)_\perp$ can be easily made into a strict one.

[2] We drop the '2' (and pseudo) terminology from now on.

The category **Conn** does not have all the completeness properties of **Cocont** in particular is not algebraically complete (hence it is not algebraically compact, either). Still it has the symmetric monoidal closed structure that we need for our 'function application'.

Proposition 8.1.4 **Conn** *has (infinite) products, terminal objects and it is symmetric monoidal closed.*

Proof: Products and terminal objects are inherited from **Cocont**. For the monoidal structure, define

$$(- \otimes^c -) \stackrel{\text{def}}{=} ((-)_\perp \otimes (-)_\perp)^+ \ ,$$

i.e., $\mathbb{C} \otimes^c \mathbb{D}$ is the category $\mathbb{C}_\perp \times \mathbb{D}_\perp$ to which the initial object has been removed.

The function space is given by

$$(- \multimap^c -) = (-)_\perp \multimap - \ .$$

So on objects, $\mathbb{C}$ and $\mathbb{D}$, $\mathbb{C} \multimap^c \mathbb{D} = (\mathbb{C}_\perp)^{\mathrm{op}} \times \mathbb{D}$.

Concerning the adjunction situation, observe that for any three small categories, $\mathbb{C}, \mathbb{D}, \mathbb{E}$, the following equivalences of categories hold:

$$\begin{array}{rcl}
\mathbf{Conn}(\mathbb{C} \otimes^c \mathbb{D}, \mathbb{E}) & = & \mathbf{Conn}((\mathbb{C}_\perp \times \mathbb{D}_\perp)^+, \mathbb{E}) \\
& \simeq & \mathbf{Cocont}(((\mathbb{C}_\perp \times \mathbb{D}_\perp)^+)_\perp, \mathbb{E}) \\
& \cong & \mathbf{Cocont}(\mathbb{C}_\perp \times \mathbb{D}_\perp, \mathbb{E}) \\
& \simeq & \mathbf{Cocont}(\mathbb{C}_\perp, \mathbb{D}_\perp \multimap \mathbb{E}) \\
& = & \mathbf{Cocont}(\mathbb{C}_\perp, \mathbb{D} \multimap^c \mathbb{E}) \\
& \simeq & \mathbf{Conn}(\mathbb{C}, \mathbb{D} \multimap^c \mathbb{E}) \ .
\end{array}$$

□

We cannot in general solve recursive domain equations directly in **Conn**, we shall find solutions in **Cocont**.

8.2 An equation for higher order processes

Our concern here is the semantics of a process language in which processes themselves can be sent and received along a fixed set of channels Ch. We derive the category of paths by solving the equation

$$\mathbb{P} = \mathbb{P}_\perp + \sum_{a \in Ch} \mathbb{C}_\perp + \sum_{a \in Ch} \mathbb{F}_\perp \qquad \mathbb{C} = \mathbb{P} \otimes^c \mathbb{P} \qquad \mathbb{F} = (\mathbb{P} \multimap^c \mathbb{P}) \ .$$

The three components of $\mathbb{P}$ represent paths beginning with a silent action, an output on a channel, resuming as a concretion ($\mathbb{C}$), and an input from a channel, resuming as an abstraction ($\mathbb{F}$). Our choice of path for abstractions narrows us to a *linear* process-passing language, one where the input process can be run at most once to yield a single (computation) path; we shall see later how this affects the process language. The fact that the input path is lifted (cf. definition of $\multimap^c$) implies that the process received as input can be ignored and need not be run.

Many variations on these choices of path categories are possible of course: we might instead have taken $\mathbb{F}$ to be $(!\mathbb{P} \multimap \mathbb{P})$ which would allow the input process to be copied and used arbitrarily; we might have exploited the fully higher-order nature of **Conn** to give a rather abstract account of the concretions and abstractions and their interaction along the lines of that proposed in [118].

As usual, since the ! is not involved, the solutions to the equations are partial orders that we regard as categories. For convenience in future definitions by cases we first give an explicit description of the objects of $\mathbb{P}_\perp$, $\mathbb{C}_\perp$ and $\mathbb{F}_\perp$ as the terms

$$\begin{array}{lcl} p & ::= & \perp \mid \tau.p \mid a!c \mid a?f \\ c & ::= & \langle p\rangle q \\ f & ::= & (p \mapsto p') \mid (- \mapsto \perp) \text{ where } p' \neq \perp. \end{array}$$

We shall write t for a general path of form p, c or f.

The morphisms are then the simple partial-order relations induced by the following rules:

$$\frac{}{\perp \leq_p p} \qquad \frac{}{(- \mapsto \perp) \leq_f f}$$

$$\frac{p \leq_p q}{\tau.p \leq_p \tau.q} \qquad \frac{p \leq_p p' \quad q \leq_p q'}{\langle p\rangle q \leq_c \langle p'\rangle q'} \qquad \frac{p' \leq_p p \quad q \leq_p q'}{(p \mapsto q) \leq_f (p' \mapsto q')}$$

$$\frac{c \leq_c c'}{a!c \leq_p a!c'} \qquad \frac{f \leq_f f'}{a?f \leq_f a?f'}$$

The categories $\mathbb{P}, \mathbb{C}$ and $\mathbb{F}$ are then the subcategories of $\mathbb{P}_\perp$, $\mathbb{C}_\perp$ and $\mathbb{F}_\perp$ obtained by removing the initial objects $\perp$, $\langle\perp\rangle\perp$ and $(- \mapsto \perp)$.

Within the set of all paths it is convenient for future use to distinguish the set of *atomic* paths.

Definition 8.2.1 *Define the* atomic paths *as those path objects that are non bottom (i.e., neither $\perp$, $\langle\perp\rangle\perp$ or $(- \mapsto \perp)$) and expressed by the following grammar:*

$$\begin{array}{lcl} \alpha_p & ::= & \perp \mid \tau.\perp \mid a!\langle\perp\rangle\perp \mid a?(- \mapsto \perp) \\ \alpha_c & ::= & \langle\alpha_p\rangle\perp \mid \langle\perp\rangle\alpha_p \\ \alpha_f & ::= & (p \mapsto \alpha_p) \mid (- \mapsto \perp) \end{array}$$

We shall write α for a general atomic path of form α_p, α_c or α_f.

NOTATION: From now on we shall write $\tau.$ for $\tau.\perp$ and similarly, $a!$ for $a!\langle\perp\rangle\perp$ and $a?$ for $a?(- \mapsto \perp)$.

When α_p is any one of $\tau.$, $a!$ or $a?$, we shall write $\alpha_p.t$ for $\tau.t$, $a!t$ and $a?t$ respectively, provided t is appropriate.

8.3 An higher order process language

We build our process language around the path objects of $\mathbb{P}$, $\mathbb{C}$, $\mathbb{F}$. Assume a set of variables $Vars$ over processes, whose elements will typically be written $x, y, z, \cdots$. We define the syntactic categories of processes (P), concretions (C) and abstractions (F):

$$\begin{aligned}
P &::= \mathbf{Nil} \mid \tau.P \mid a!C \mid a?F \mid \textstyle\sum_{i\in I} P_i \mid [p \leq P']P \mid (P \mid P') \\
&\qquad\qquad\qquad\qquad \mid rep(P) \mid F \bullet P \mid x \\
C &::= \langle P \rangle P' \\
F &::= (x)P
\end{aligned}$$

We shall write T for a general term of the form P, C or F. Above, in the sum terms we assume that I is any set, in tests we have made use of a coercion that identifies process paths such as p with a closed process term and we have a replication $rep(P)$. Replication would have been subsumed by a general recursive definition of processes allowing for example $rec\, x.(P \mid x)$. Such recursive definitions, provided made consistent with the linearity condition on terms below, pose no significant problems. As mentioned a path p can be regarded as a special term $\overline{p}$ of matching type:

- $\overline{\bot} = \mathbf{Nil}$, $\overline{\tau.p} = \tau.\overline{p}$, $\overline{a!} = a!\langle\mathbf{Nil}\rangle\mathbf{Nil}$, $\overline{a?} = a?(x)\mathbf{Nil}$, $\overline{\langle p\rangle p'} = \langle\overline{p}\rangle\overline{p'}$,
- $\overline{(p \mapsto p')} = (x)[\overline{p}{\leq}x]\overline{p'}$, $\overline{a!c} = a!\overline{c}$, $\overline{a?f} = a?\overline{f}$.

Henceforth, we will write p for $\overline{p}$, identifying a path with its corresponding term. As usual we have for any term T the set of free variables in $FV(T)$ and we think of terms as being defined up to α-conversion. We impose the following *linearity condition* on the multiple presence of free variables. It will ensure that at most one occurrence of a variable can be active at any one time.

Definition 8.3.1 *A term T is inductively defined to be* linear *if each of its proper sub terms is linear and in the case where*

- *T is of the form $U \mid V$, $\langle U\rangle V$, $[p \leq U]V$ or $U \bullet V$, then $FV(U) \cap FV(V) = \emptyset$,*
- *T is of the form $rep(P)$, then $FV(P) = \emptyset$.*

A substitution such as $P[Q_1/x_1, \ldots, Q_k/x_k]$ is defined as usual, using α-conversion to avoid the unwanted capture of free variables.

Proposition 8.3.2 *Substitution respects linear terms in the sense that: if $P, Q_1, \ldots, Q_k$ are linear terms with pairwise disjoint sets of free variables, then $P[Q_1/x_1, \ldots, Q_k/x_k]$ is linear.*

8.3.1 Operational semantics

We introduce now the basic operational semantics for the calculus. This is given in Figure 8.1. For the purpose of the testing we consider the possibility of checking for "path capabilities", where the path can be arbitrary.

We could at the cost of a considerably more complicated syntax, with "mixed concretions" like $\langle F\rangle P$ and worse like $\langle\langle F\rangle P\rangle F$, have introduced path transitions associated with all kinds of paths.

Atomic transitions:

Prefixings

$$\frac{}{\tau.P \xrightarrow{\tau} P} \quad \frac{}{a!C \xrightarrow{a!} C} \quad \frac{}{a?F \xrightarrow{a?} F}$$

Sums and Tests

$$\frac{P_j \xrightarrow{\alpha} T \quad j \in I}{\sum_{i\in I} P_i \xrightarrow{\alpha} T} \quad \frac{Q \xrightarrow{p} \quad P \xrightarrow{\alpha} T}{[p \le Q]P \xrightarrow{\alpha} T}$$

Parallel Composition

$$\frac{P \xrightarrow{a?} F \quad Q \xrightarrow{a!} \langle R\rangle S}{P \mid Q \xrightarrow{\tau} F \cdot R \mid S} \quad \frac{Q \xrightarrow{a!} \langle R\rangle S \quad P \xrightarrow{a?} F}{Q \mid P \xrightarrow{\tau} S \mid F \cdot R}$$

$$\frac{P \xrightarrow{a?} (x)P'}{P \mid Q \xrightarrow{a?} (x)(P' \mid Q)} \quad \frac{P \xrightarrow{a?} (x)P'}{Q \mid P \xrightarrow{a?} (x)(Q \mid P')}$$

$$\frac{P \xrightarrow{a!} \langle P'\rangle P''}{P \mid Q \xrightarrow{a!} \langle P'\rangle(P'' \mid Q)} \quad \frac{P \xrightarrow{a!} \langle P'\rangle P''}{Q \mid P \xrightarrow{a!} \langle P'\rangle(Q \mid P'')}$$

$$\frac{P \xrightarrow{\tau} P'}{P \mid Q \xrightarrow{\tau} P' \mid Q} \quad \frac{P \xrightarrow{\tau} P'}{Q \mid P \xrightarrow{\tau} Q \mid P'}$$

Replication and Application

$$\frac{P \mid rep(P) \xrightarrow{\alpha} T}{rep(P) \xrightarrow{\alpha} T} \quad \frac{P[Q/x] \xrightarrow{\alpha} T}{(x)P \bullet Q \xrightarrow{\alpha} T}$$

Concretions and Abstractions

$$\frac{P \xrightarrow{\alpha} T}{\langle P\rangle Q \xrightarrow{\langle\alpha\rangle\perp} T} \quad \frac{P \xrightarrow{\alpha} T}{\langle Q\rangle P \xrightarrow{\langle\perp\rangle\alpha} T} \quad \frac{F \cdot p \xrightarrow{\alpha} T}{F \xrightarrow{(p\mapsto\alpha)} T}$$

Path capabilities:

$$\frac{P \xrightarrow{\alpha} U}{P \xrightarrow{\alpha}} \qquad \frac{P \xrightarrow{\alpha} T \quad T \xrightarrow{t}}{P \xrightarrow{\alpha.t}}$$

$$\frac{P \xrightarrow{p} \quad Q \xrightarrow{q}}{\langle P\rangle Q \xrightarrow{\langle p\rangle q}} \qquad \frac{P \xrightarrow{p}}{\langle P\rangle Q \xrightarrow{\langle p\rangle\perp}} \qquad \frac{Q \xrightarrow{q}}{\langle P\rangle Q \xrightarrow{\langle\perp\rangle q}}$$

$$\frac{F \cdot p \xrightarrow{q}}{F \xrightarrow{p\mapsto q}}$$

The path t in a capability $T \xrightarrow{t}$ need not be simple because of the extra rule introducing non-simple concretion paths of the form $\langle p\rangle q$.

Figure 8.1: The basic operational semantics

As one expects, the operational semantics respects linear terms:

Proposition 8.3.3 *If $T \xrightarrow{\alpha} U$ and T is linear, then so is U.*

From now on we will always assume that terms are linear.

8.4 Presheaf semantics

The purpose of this section is to provide the terms of the language with a compositional semantics in terms of presheaves and to prove various facts about it, notably a Substitution Lemma (Lemma 8.4.14) relating application in the model with substitution in the language, a "soundness" result (Theorem 8.4.16) for the presheaf semantics and a stability-like (cf. [13]) property of the denotations of open terms. First steps toward an operational characterisation of open map bisimulation are also made. They lead to a characterisation of open map bisimulation, for a fragment of the process calculus corresponding to a form of λ-calculus as applicative bisimulation [2].

8.4.1 Transition relations for presheaves

Closed terms of the language will denote presheaves over $\mathbb{P}$. As we know (cf. Chapter 5) presheaves, X, can be given a transition relation

$$X \xrightarrow{p}_d Y \text{ if } d \in X(p) \text{ and } Y = X_{|d}$$

taking from a presheaf over $\mathbb{P}$ to a presheaf over the *resumption* category $(p/\mathbb{P})^+$.

In the operational semantics we allow in the case of concretions the possibility of choosing which branch of the concretion we wish to observe. We also allow for testing path capabilities of processes. This motivates the following "adjustment" of the above definition of transition for presheaves.

Definition 8.4.1 *For presheaves X, C, F over $\mathbb{P}, \mathbb{C}, \mathbb{F}$ respectively, define the following transition relations:*

- $X \xrightarrow{\alpha} Y$ *if there exists $d \in X(\alpha)$ such that* $X \xrightarrow{\alpha}_d Y$.
- $C \xrightarrow{\langle\alpha\rangle -} Y$ *if* $C(-, \perp) \xrightarrow{\alpha} Y$.
- $C \xrightarrow{\langle -\rangle\alpha} Y$ *if* $C(\perp, -) \xrightarrow{\alpha} Y$.
- $F \xrightarrow{\perp\mapsto\alpha}_d Y$ *if* $F(\perp, -) \xrightarrow{\alpha}_d Y$.
- $F \xrightarrow{p\mapsto\alpha}_Y$ *if* $F(p, -) \xrightarrow{\alpha} Y$.
- $X \xrightarrow{p}$ *if* $X(p) \neq \emptyset$.
- $C \xrightarrow{c}$ *if* $C(c) \neq \emptyset$.
- $F \xrightarrow{f}$ *if* $F(f) \neq \emptyset$.

Observations can be "concatenated":

Proposition 8.4.2 *For every presheaf X and Z over $\mathbb{P}$ and F over $\mathbb{F}$ the following hold.*

1. *If* $X \xrightarrow{\tau} Y$ *and* $Y \xrightarrow{p}$ *then* $X \xrightarrow{\tau.p}$.
2. *If* $X \xrightarrow{a!} C$ *and* $C \xrightarrow{c}$ *then* $X \xrightarrow{a!c}$.
3. *If* $X \xrightarrow{a?} F$ *and* $F \xrightarrow{f}$ *then* $X \xrightarrow{a?f}$.
4. *If* $F \xrightarrow{p\mapsto\alpha} Y$ *and* $Y \xrightarrow{t}$ *then* $F \xrightarrow{p\mapsto\alpha.t}$.

8.4.2 Constructions

As preparation for the denotational semantics of processes we present the basic constructions that we shall need.

Prefixings: We are accustomed by now to defining prefixings by precomposing embeddings such as

$$In_{\tau!} : \widehat{\mathbb{P}_\perp} \hookrightarrow \widehat{\mathbb{P}}$$

with the lifting operator l_*. We have, as before, the characterisations for

$$\tau. \stackrel{\text{def}}{=} In_{\tau!}l_* \qquad a! \stackrel{\text{def}}{=} (In_{a!})_! l_* \qquad \text{and} \qquad a? \stackrel{\text{def}}{=} (In_{a?})_! l_* \ .$$

For every presheaf X over $\mathbb{P}$,

$$\begin{array}{rccl} \tau.X : & \mathbb{P}^{op} & \to & \mathbf{Set} \\ & p & \mapsto & \left\{ \begin{array}{cl} \{\star\} & \text{if } p = \tau. \\ X(p') & \text{if } p = \tau.p' \\ \emptyset & \text{otherwise} \end{array} \right. \end{array}$$

For every channel a, presheaf C over $\mathbb{C}$ and presheaf F over $\mathbb{F}$,

$$\begin{array}{rcl} a!C : \mathbb{P}^{op} & \to & \mathbf{Set} \\ p & \mapsto & \left\{ \begin{array}{cl} \{\star\} & \text{if } p = a! \\ C(c) & \text{if } p = a!c \\ \emptyset & \text{otherwise} \end{array} \right. \end{array} \qquad \begin{array}{rcl} a?F : \mathbb{P}^{op} & \to & \mathbf{Set} \\ p & \mapsto & \left\{ \begin{array}{cl} \{\star\} & \text{if } p = a? \\ F(f) & \text{if } p = a?f \\ \emptyset & \text{otherwise} \end{array} \right. \end{array}$$

As usual, by definition, following Corollary 4.6.6 and Proposition 4.6.8, open map bisimulation is preserved by the prefixings:

Theorem 8.4.3 *The prefixing functors, $\tau.$, $a!$ and $a?$ preserve surjective open maps and hence bisimulation.*

Using the prefixing operators we can express a decomposition result for presheaves over $\mathbb{P}$.

Theorem 8.4.4 *Let X be a presheaf over $\mathbb{P}$, then X is isomorphic to*

$$\sum_{i\in I} \tau.(X_{|i}) + \sum_{a\in Ch} \sum_{j\in X(a!)} a!(X_{|j}) + \sum_{a\in Ch} \sum_{k\in X(a?)} a?(X_{|k}) \ .$$

Concretions:

Concretions are modelled as presheaves over $\mathbb{C} = (\mathbb{P}_\perp \otimes \mathbb{P}_\perp)^+$.

Let $X, Y \in \widehat{\mathbb{P}}$, Define $\langle X\rangle Y$ as the result of the application to (X, Y) of the functor

$$\widehat{\mathbb{P}} \times \widehat{\mathbb{P}} \xrightarrow{l_* \times l_*} \widehat{\mathbb{P}_\perp} \times \widehat{\mathbb{P}_\perp} \xrightarrow{w^*_{\mathbb{P}_\perp, \mathbb{P}_\perp}} \widehat{\mathbb{P}_\perp \times \mathbb{P}_\perp} \xrightarrow{l^*} \mathbb{C} \ ,$$

i.e., for every pair $\langle r\rangle s \in |\mathbb{C}|$,

$$\langle X\rangle Y(\langle r\rangle s) = \lfloor X\rfloor(r) \times \lfloor Y\rfloor(s) \ .$$

Of course, from what we know well by now, it is clear that if X and W are $\mathbb{P}$-open bisimilar presheaves and the same is true for Y and Z, then $\langle X\rangle Y$ is $\mathbb{C}$-open bisimilar to $\langle W\rangle Z$.

Remark: The functor $\langle -\rangle -$ derives from the tensor operator $\otimes^c$. In fact it corresponds to

$$\otimes^c : \mathbf{Conn}(\mathbf{0}, \mathbb{P}) \times \mathbf{Conn}(\mathbf{0}, \mathbb{P}) \to \mathbf{Conn}((\mathbf{0}_\perp \otimes \mathbf{0}_\perp)^+, (\mathbb{P}_\perp \otimes \mathbb{P}_\perp)^+) = \mathbf{Conn}(\mathbf{0}, \mathbb{C}) \ ,$$

via the isomorphism $\mathbf{Conn}(\mathbf{0}, \mathbb{P}) \cong \mathbf{Prof}(\mathbf{1}, \mathbb{P}) \cong \widehat{\mathbb{P}}$, where $\mathbf{0}$ is the empty category (no objects, no arrows), while $\mathbf{1} = \mathbf{0}_\perp$ is the terminal category (one object, one arrow).

Observations tested in either branches of concretions can be paired:

Proposition 8.4.5 *For any two presheaves over $\mathbb{P}$, X and Y the following hold:*

1. *If $X \xrightarrow{p}$ and $Y \xrightarrow{q}$ then $\langle X\rangle Y \xrightarrow{\langle p\rangle q}$.*
2. *If $X \xrightarrow{p}$ then $\langle X\rangle Y \xrightarrow{\langle p\rangle \perp}$.*
3. *If $Y \xrightarrow{p}$ then $\langle X\rangle Y \xrightarrow{\langle \perp\rangle p}$.*

Abstractions: Let $H : \mathbb{P}_\perp \to \widehat{\mathbb{P}}$ be a functor. Define $\widetilde{H}$ to be the presheaf over $(\mathbb{P}_\perp \multimap \mathbb{P})$ defined so that

$$\widetilde{H}(p \mapsto q) = (Hp)q$$

on a path in $\mathbb{F}$; its action on morphisms is inherited in the obvious way from H. Similarly, if G is a presheaf over $(\mathbb{P}_\perp \multimap \mathbb{P})$, define $\widetilde{G}$ to be the corresponding functor $\mathbb{P}_\perp \to \widehat{\mathbb{P}}$.

Let $h : F \to G$ be a map in $\widehat{\mathbb{P} \multimap \mathbb{Q}}$. As we saw in Chapter 4 (Section 4.6) h is $(\mathbb{P} \multimap \mathbb{Q})$-open if and only if,
(i) h_p is $\mathbb{Q}$-open, for every $p \in |\mathbb{P}|$ and (ii) h_q is $\mathbb{P}^{\mathrm{op}}$-open, for every $q \in |\mathbb{Q}|$.
The first condition is to be expected; it directly entails that bisimilar profunctors are pointwise bisimilar. The second condition (ii) is peculiar at first sight. As we shall see later, it is this condition that leads us to take a closer look at the elements of presheaves. We shall think of elements as representing the derivation trees of the operational semantics. Accordingly we think of elements of $\widetilde{F}(p)q$ as derivations associated with F doing a q-transition for input p. When such derivations are associated with minimum inputs (see Theorem 8.4.20 below) the effect of (ii) is to ensure that derivations matched by a

bisimulation are enabled at the same minimal inputs. Condition (ii) is also important in extending (i) to hold for presheaves over $\mathbb{P}$ as input (cf. Corollary 4.6.7), i.e., condition (i) and (ii) imply that for any presheaf $X \in |\widehat{\mathbb{P}}|$, $(\widetilde{F})_!(X)$ is $\mathbb{Q}$-open map bisimilar to $(\widetilde{G})_!(X)$.

Parallel Composition: In order to define a parallel composition operator

$$| : \widehat{\mathbb{P}} \times \widehat{\mathbb{P}} \longrightarrow \widehat{\mathbb{P}}$$

we begin by defining a colimit preserving functor

$$\|_! : \widehat{\mathbb{P}_\perp \times \mathbb{P}_\perp} \to \widehat{\mathbb{P}}$$

out of which the functor $|$ will be obtained as

$$\widehat{\mathbb{P}} \times \widehat{\mathbb{P}} \xrightarrow{l_* \times l_*} \widehat{\mathbb{P}_\perp} \times \widehat{\mathbb{P}_\perp} \xrightarrow{w^*_{\mathbb{P}_\perp, \mathbb{P}_\perp}} \widehat{\mathbb{P}_\perp \times \mathbb{P}_\perp} \xrightarrow{\|_!} \widehat{\mathbb{P}} \,.$$

In particular, being a composition of surjective open map preserving functors, $|$ will preserve open map bisimulation. To define a colimit preserving functor from $\widehat{\mathbb{P}_\perp \times \mathbb{P}_\perp}$ to $\widehat{\mathbb{P}}$ is enough to say what the functor does on the representables. So we define

$$\| : \mathbb{P}_\perp \times \mathbb{P}_\perp \longrightarrow \widehat{\mathbb{P}}$$

inductively on the structure of the paths in $\mathbb{P}$. This calls for simultaneous inductive definitions of a whole family of 'parallel composition functors',

$$\begin{array}{ll}
{}^{\mathbb{P}}\|^{\mathbb{P}} : \mathbb{P}_\perp \times \mathbb{P}_\perp \to \widehat{\mathbb{P}} & \\
{}^{\mathbb{F}}\|^{\mathbb{C}} : \mathbb{F}_\perp \times \mathbb{C}_\perp \to \widehat{\mathbb{P}} & {}^{\mathbb{C}}\|^{\mathbb{F}} : \mathbb{C}_\perp \times \mathbb{F}_\perp \to \widehat{\mathbb{P}} \\
{}^{\mathbb{F}}\|^{\mathbb{P}} : \mathbb{F}_\perp \times \mathbb{P}_\perp \to \widehat{\mathbb{F}} & {}^{\mathbb{P}}\|^{\mathbb{F}} : \mathbb{P}_\perp \times \mathbb{F}_\perp \to \widehat{\mathbb{F}} \\
{}^{\mathbb{C}}\|^{\mathbb{P}} : \mathbb{C}_\perp \times \mathbb{P}_\perp \to \widehat{\mathbb{C}} & {}^{\mathbb{P}}\|^{\mathbb{C}} : \mathbb{P}_\perp \times \mathbb{C}_\perp \to \widehat{\mathbb{C}} \,,
\end{array}$$

to take account of all possible matching situation that can be encountered. In the sequel we shall omit the upper indices when no confusion arises. For convenience we introduce some notation.

NOTATION: In the sequel we shall use the following notation.

- If $l^{\mathbb{P}} : \mathbb{P} \to \mathbb{P}_\perp$ is the obvious inclusion functor, write $j^{\mathbb{P}}$ for the functor

  $$\mathrm{Lan}_{l^{\mathbb{P}}}(\mathrm{y}_{\mathbb{P}}) : \mathbb{P}_\perp \to \widehat{\mathbb{P}} \,,$$

  i.e., the functor that sends every non bottom path object $\lfloor p \rfloor$ to $\mathrm{y}_{\mathbb{P}}(p)$ and $\perp$ to the empty presheaf, $\emptyset$. Observe that $j^{\mathbb{P}}$ is the restriction at $\mathbb{P}_\perp$ of $(l^{\mathbb{P}})^* : \widehat{\mathbb{P}_\perp} \to \widehat{\mathbb{P}}$ which, in turn, is the value of the counit for the lifting comonad of **Cocont** at $\mathbb{P}$ (cf. Proposition 8.1.2).
- Similarly for $l^{\mathbb{C}} : \mathbb{C} \to \mathbb{C}_\perp$ and $l^{\mathbb{F}} : \mathbb{F} \to \mathbb{F}_\perp$, write $j^{\mathbb{C}}$ and $j^{\mathbb{F}}$ for the corresponding Kan extensions, $\mathrm{Lan}_{l^{\mathbb{C}}}(\mathrm{y}_{\mathbb{C}})$ and $\mathrm{Lan}_{l^{\mathbb{F}}}(\mathrm{y}_{\mathbb{F}})$.

- When no confusion arises we shall often write j for the appropriate one of $j^{\mathbb{P}}$, $j^{\mathbb{C}}$ or $j^{\mathbb{F}}$.
- If X is a presheaf over and p an object of $\mathbb{P}$, write $p \mapsto X$ for the presheaf over $\mathbb{F}$ defined by
$$p \mapsto X(q,r) = \begin{cases} X(r) & \text{if } p \leq q \\ \emptyset & \text{otherwise} \end{cases}$$

For any of the functors above, define now
$$\perp \parallel \perp = \emptyset \ .$$

For $p \in \mathbb{P}$, define
$$p^{\mathbb{P}}\|^{\mathbb{P}} \perp = \perp \parallel p = j(p) \ .$$

In any other case
$$a.t \parallel b.u = \begin{cases} a.(t \parallel b.u) + b.(a.t \parallel u) + \tau.(t \parallel u) & \text{if } (a = a? \ \& \ b = a!) \text{ or} \\ & \quad (a = a! \ \& \ b = a?) \\ a.(t \parallel b.u) + b.(a.t \parallel u) & \text{otherwise.} \end{cases}$$

For abstractions and concretions
$$\begin{array}{l} (- \mapsto \perp)^{\mathbb{F}}\|^{\mathbb{C}} \ \langle s \rangle r = \perp \parallel r \\ (p \mapsto q) \parallel \langle s \rangle r = \begin{cases} \lfloor q \rfloor \parallel r & \text{if } p \leq s \\ \perp \parallel r \text{ otherwise,} \end{cases} \\ \langle s \rangle r^{\mathbb{C}}\|^{\mathbb{F}} \ (- \mapsto \perp) = r \parallel \perp \\ \langle s \rangle r \parallel (p \mapsto q) = \begin{cases} r \parallel \lfloor q \rfloor & \text{if } p \leq s \\ r \parallel \perp \text{ otherwise.} \end{cases} \end{array}$$

For abstractions and paths, we use the notation $(p \mapsto X)$ that we introduced above.
$$\begin{array}{l} (- \mapsto \perp)^{\mathbb{F}}\|^{\mathbb{P}} \ r = (- \mapsto (\perp^{\mathbb{P}}\|^{\mathbb{P}} \ r)) \\ ((p \mapsto q) \parallel r) = (p \mapsto (\lfloor q \rfloor \parallel r)) \\ r^{\mathbb{P}}\|^{\mathbb{F}} \ (- \mapsto \perp = (- \mapsto (r^{\mathbb{P}}\|^{\mathbb{P}} \ \perp)) \\ r \parallel (p \mapsto q) = (p \mapsto (r \parallel \lfloor q \rfloor)) \end{array}$$

Finally concretions and paths compose as follows
$$(\langle s \rangle r)^{\mathbb{C}}\|^{\mathbb{P}} \ p = \langle j(s) \rangle (r \parallel p) \quad \text{and} \quad p \parallel (\langle s \rangle r) = \langle j(s) \rangle (p \parallel r)$$

The action on morphisms (i.e., the partial order relation is defined inductively in a similar way).

Definition 8.4.6 *For* $\mathbb{A}, \mathbb{B} \in \{\mathbb{P}, \mathbb{C}, \mathbb{F}\}$, *define* ${}^{\mathbb{A}}|^{\mathbb{B}} : \widehat{\mathbb{A}} \times \widehat{\mathbb{B}} \to \widehat{\mathbb{D}}$ *as the composite*
$$\widehat{\mathbb{A}} \times \widehat{\mathbb{B}} \xrightarrow{l_* \times l_*} \widehat{\mathbb{A}_\perp} \times \widehat{\mathbb{B}_\perp} \xrightarrow{w^*_{\mathbb{A}_\perp, \mathbb{B}_\perp}} \widehat{\mathbb{A}_\perp \times \mathbb{B}_\perp} \xrightarrow{{}^{\mathbb{A}}\|^{\mathbb{B}}_!} \widehat{\mathbb{D}} \ ,$$
where $\mathbb{D} \in \{\mathbb{P}, \mathbb{C}, \mathbb{F}\}$ *is uniquely determined by* $\mathbb{A}$ *and* $\mathbb{B}$ *according to the "typing" of the parallel composition functors.*

Since ${}^{\mathbb{A}}|^{\mathbb{B}}$ is the composite of open map bisimulation preserving functors we have the following congruence property:

Theorem 8.4.7 *For every $\mathbb{A}, \mathbb{B} \in \{\mathbb{P}, \mathbb{C}, \mathbb{F}\}$ if X is $\mathbb{A}$-open bisimilar to W and Y is $\mathbb{B}$-open bisimilar to Z, then $X^{\mathbb{A}}|^{\mathbb{B}}Y$ is $\mathbb{D}$-open map bisimilar to $W^{\mathbb{A}}|^{\mathbb{B}}Z$.*

Observe now that the Yoneda embeddings $\mathrm{y}_{\mathbb{F}_\perp \times \mathbb{P}_\perp}$ and $\mathrm{y}_{\mathbb{F}_\perp \times \mathbb{C}_\perp}$ (and similarly their symmetric counterpart) can be factorised as follows:

$$\mathrm{y}_{\mathbb{F}_\perp \times \mathbb{P}_\perp} = w^*_{\mathbb{F}_\perp, \mathbb{P}_\perp} \circ (l^{\mathbb{F}}_* \times l^{\mathbb{P}}_*) \circ (j^{\mathbb{F}} \times j^{\mathbb{P}}) \tag{8.1}$$

$$\mathrm{y}_{\mathbb{F}_\perp \times \mathbb{C}_\perp} = w^*_{\mathbb{F}_\perp, \mathbb{C}_\perp} \circ (l^{\mathbb{F}}_* \times l^{\mathbb{C}}_*) \circ (j^{\mathbb{F}} \times j^{\mathbb{C}}) \ . \tag{8.2}$$

Hence we have the following "characterisation" of $||$'s that we shall need later.

Proposition 8.4.8 *1. The functor ${}^{\mathbb{F}}||^{\mathbb{P}}$ is naturally isomorphic to ${}^{\mathbb{F}}|^{\mathbb{P}} \circ (j^{\mathbb{F}} \times j^{\mathbb{P}})$.*

2. The functor $||^{\mathbb{F},\mathbb{C}}$ is naturally isomorphic to $|^{\mathbb{F},\mathbb{C}} \circ (j^{\mathbb{F}} \times j^{\mathbb{C}})$.

And similarly for the symmetric versions.

Using the same kind of proof for the analogous results in Chapter 5 (Lemma 5.1.13), we can characterise inductively the action of ${}^{\mathbb{A}}|^{\mathbb{B}}$.

Lemma 8.4.9 *If $X, Y \in |\widehat{\mathbb{P}}|$, $C \in |\widehat{\mathbb{C}}|$ and $F \in |\widehat{\mathbb{F}}|$ the the following hold (omitting the upper indices):*

1. *$\tau.X|\tau.Y \cong \tau.(X|\tau.Y) + \tau.(\tau.X|Y)$.*
2. *$\tau.X|a?F \cong \tau.(X|a?F) + a?(\tau.X|F)$.*
3. *$\tau.X|a!C \cong \tau.(X|a!C) + a!(\tau.X|C)$.*
4. *If $C = \langle Z_1 \rangle Z_2$ then $X|C \cong \langle Z_1 \rangle (X|Z_2)$.*
5. *$\widetilde{(F|X)}$ is isomorphic to the functor $\widetilde{F(-)}|X$, i.e., the Currying of the functor*

$$\mathbb{P}_\perp \xrightarrow{\widetilde{F}} \widehat{\mathbb{P}} \xrightarrow{\langle -, X \rangle} \widehat{\mathbb{P}} \times \widehat{\mathbb{P}} \xrightarrow{|} \widehat{\mathbb{P}} \ .$$

6. *$a?F|a!C \cong a?(F|a!C) + a!(a?F|C) + \tau.(F|C)$.*
7. *If $C = \langle Z_1 \rangle Z_2$ then $F|C \cong \widetilde{F}_!(Z_1)|Z_2$.*

Analogous results hold for the symmetric operations.

Proof: In case 1, 2, 3 and 6 the proof goes exactly as in Lemma 5.1.13. The proof of 4 goes as follows:

$$\begin{aligned}
X|C &= \textstyle\int^{p,q,r} \lfloor X \rfloor(p) \times \lfloor Z_1 \rfloor(q) \times \lfloor Z_2 \rfloor(r) \,.\, (p \parallel (\langle q \rangle r)) \\
&= \textstyle\int^{p,q,r} \lfloor X \rfloor(p) \times \lfloor Z_1 \rfloor(q) \times \lfloor Z_2 \rfloor(r) \,.\, (\langle j(q) \rangle p \parallel r) \\
&\cong \textstyle\int^{q} \lfloor Z_1 \rfloor(q) \,.\, (\langle j(q) \rangle \int^{p,r} \lfloor X \rfloor(p) \times \lfloor Z_2 \rfloor(r) \,.\, (p \parallel r)) \\
&\quad \text{(by Fubini and since } \langle - \rangle - \text{ preserves connected colimits in each argument)} \\
&\cong \textstyle\int^{q} \lfloor Z_1 \rfloor(q) \,.\, (\langle j(q) \rangle (X|Z_2)) \\
&\cong \langle \textstyle\int^{q} \lfloor Z_1 \rfloor(q) \,.\, j(q) \rangle (X|Z_2) \\
&\quad \text{(since } \langle - \rangle - \text{ preserves connected colimits in each argument)} \\
&\cong \langle Z_1 \rangle (X|Z_2) \ .
\end{aligned}$$

Concerning 5 we have that

$$F|X \stackrel{\text{def}}{=} \int^{f,q} \lfloor F \rfloor(f) \times \lfloor X \rfloor(q) \,.\, f \parallel q \ .$$

Because of Proposition 8.4.8 and since coends in a presheaf category can be computed pointwise we have that, for every $p \in |\mathbb{P}_\perp|$ the following holds:

$$\widetilde{F|X}(p) = (F|X)(p, -) \cong \int^{f,q} \lfloor F \rfloor(f) \times \lfloor X \rfloor(q) \,.\, (j(f)(p)|j(q)) \ .$$

Now since $|$ preserves connected colimits in each argument this is further transformable to

$$(\int^{f} \lfloor F \rfloor(f) \,.\, j(f)(p))|(\int^{q} \lfloor X \rfloor q \,.\, j(q))$$

which by the density formula, applied twice, is just

$$\widetilde{F}(p)|X \ .$$

That the actions on morphisms agree as well is immediately obtained from a similar calculation involving morphisms. Point 7 is proved similarly as follows:

$$\begin{array}{rcl}
F|C & = & \int^{f,p,q} \lfloor F \rfloor(f) \times \lfloor Z_1 \rfloor(p) \times \lfloor Z_2 \rfloor(q) \,.\, f \parallel \langle p \rangle q \\
& & \qquad \text{(by definition)} \\
& \cong & \int^{f,p,q} \lfloor F \rfloor(f) \times \lfloor Z_1 \rfloor(p) \times \lfloor Z_2 \rfloor(q) \,.\, j(f)|j(\langle p \rangle q) \\
& & \qquad \text{(by Proposition 8.4.8)} \\
& \cong & \int^{f,p,q} \lfloor F \rfloor(f) \times \lfloor Z_1 \rfloor(p) \times \lfloor Z_2 \rfloor(q) \,.\, j(f)(p)|j(q) \\
& & \qquad \text{(by definition)} \\
& \cong & (\int^{f,p} \lfloor F \rfloor(f) \times \lfloor Z_1 \rfloor(p) \,.\, j(f)(p))|(\int^{q} \lfloor Z_2 \rfloor(q) \,.\, j(q)) \\
& & \qquad \text{(since } | \text{ preserves connected colimits in each argument)} \\
& \cong & \widetilde{F}_!(\lfloor Z_1 \rfloor)|Z_2 \\
& & \qquad \text{(by definition of } \widetilde{F}_! \text{ and from the density formula).}
\end{array}$$

□

Using the above lemma one easily proves the following theorem, using the fact that coends distribute over sums.

Theorem 8.4.10 *Let X and Y be two presheaves over $\mathbb{P}$ with decompositions (cf. Theorem 8.4.4)*

$$X \cong \sum_{i \in X(\tau)} \tau.(X_{|i}) + \sum_{a \in Ch} \sum_{j \in X(a!)} a!(\langle X_{|j,1} \rangle X_{|j,2}) + \sum_{a \in Ch} \sum_{k \in X(a?)} a?(X_{|k})$$

and

$$Y \cong \sum_{i' \in Y(\tau)} \tau.(Y_{|i'}) + \sum_{a \in Ch} \sum_{j' \in Y(a!)} a!(\langle Y_{|j',1} \rangle Y_{|j',2}) + \sum_{a \in Ch} \sum_{k' \in Y(a?)} a?(Y_{|k'}) \ .$$

Then $X \mid Y$ is isomorphic to

$$\sum_{(i,*)} \tau.(X_i \mid Y) + \sum_{(a!j,*)} a!\langle X_{j,1}\rangle(X_{j,2} \mid Y) + \sum_{(a?k,*)} a?(X_k(-) \mid Y)$$
$$\sum_{(*,i')} \tau.(X \mid Y_{i'}) + \sum_{(*,a!j')} a!\langle Y_{j',1}\rangle(X \mid Y_{j',2}) + \sum_{(*,a!k')} a?(X \mid Y_{k'}(-))$$
$$+ \sum_{(a!j,a?k')} \tau.(X_{j,2} \mid (Y_{k'} \bullet X_{j,1})) + \sum_{(a?k,a!j')} \tau.((X_k \bullet Y_{j',1}) \mid Y_{j',2}) \,.$$

If a path object t is observed in a parallel composition $X|Y$, i.e., if there exists $d \in (X|Y)(t)$, we can extract how much of X and Y had been used to produce the observation d. We begin with two auxiliary results (Lemma 8.4.11 and Corollary 8.4.12) which show that in the case of the parallel composition of two (lifted) path objects, it is possible to describe precisely how much of each path was used in producing an observation of a chosen shape.

Lemma 8.4.11 *Let t, u be two lifted path terms, for which $t \parallel u$ is defined. Let v be a path and suppose that*

$$z \in (t \parallel u)(v) \ .$$

Then there exists a unique pair $(t', u') \leq (t, u)$ and element $z' \in (t' \parallel u')(v)$ such that

1. *for all $v' \neq v$, if $v \leq v'$, $(t' \parallel u')(v') = \emptyset$, i.e., all of t' and u' was used to observe v.*
2. *$((t' \leq t) \parallel (u' \leq u))_v(z') = z$.*

Proof: The proof is an easy, though tedious induction on the structure of v. □

The fact that the triple (t', u', z') is uniquely determined immediately entails the following corollary:

Corollary 8.4.12 *With the same hypothesis as in Lemma 8.4.11. If t'', u'' and z'' are such that*

$$((t'' \leq t) \parallel (u'' \leq u))_v(z'') = z \ ,$$

then $t' \leq t''$, $u' \leq u''$ and

$$((t' \leq t'') \parallel (u' \leq u''))_v(z') = z'' \ .$$

Recall (see Chapter 4) that a coend formula in **Set**, like

$$\int^C X(C) \times F(C)(D) \ ,$$

for $X : \mathbb{C}^{\mathrm{op}} \to \mathbf{Set}$ and $F : \mathbb{C} \to \widehat{\mathbb{D}}$ defines a set given by

$$\coprod_{c \in |\mathbb{C}|} X(C) \times F(C)(D) / \sim \ ,$$

where $\sim$ is the equivalence relation generated by $\sim^\circ$ that is defined as

$$(x', y') \in X(C') \times F(C')(D) \sim^\circ (x, y) \in X(C) \times F(C)(D)$$

if there exists $f : C' \to C$ in $\mathbb{C}$ such that

$$x' = X(f)(x) \qquad \text{and} \qquad F(f)_D(y') = y\ .$$

Using this notation the following lemma is an immediate consequence of Lemma 8.4.11 and Corollary 8.4.12.

Lemma 8.4.13 *Let X and Y be two presheaves for which $|$ is defined*[3]*. Let t be any path term of the appropriate type then for any r, s lifted paths and any triple*

$$(x, y, z) \in \lfloor X \rfloor(r) \times \lfloor Y \rfloor(s) \times (r \parallel s)(t)$$

there exist r' and s' and a (necessarily unique) triple

$$(x', y', z') \in \lfloor X \rfloor(r') \times \lfloor Y \rfloor(s') \times (r' \parallel s')(t)$$

such that

1. *$(x', y', z') \sim^\circ (x, y, z)$.*
2. *For every r'', s'', x'', y'' and z'' such that $(x'', y'', z'') \sim (x, y, z)$, then $r' \leq r''$, $s' \leq s''$ and $(x', y', z') \sim^\circ (x'', y'', z'')$.*

Proof: Use the Lemma 8.4.11 to determine r', s' and z'. Define

$$x' = \lfloor X \rfloor(r' \leq r)(x) \qquad \text{and} \qquad y' = \lfloor Y \rfloor(s' \leq s)\ .$$

Corollary 8.4.12 ensures that the second condition is met. □

We shall use Lemma 8.4.13 later in proving a property of open terms (Theorem 8.4.20) reminiscent of Berry's stability condition on functions [13].

Replication: Following what we did in Chapter 7, we shall define the denotation of a replicated process $rep(P)$, $[\![rep(P)]\!]$ as the least fixed point of the functor

$$-|[\![P]\!] : \widehat{\mathbb{P}} \to \widehat{\mathbb{P}}\ ,$$

where $[\![P]\!]$ is the denotation of P. So, for every presheaf X over $\mathbb{P}$ define

$$-|X : \widehat{\mathbb{P}} \to \widehat{\mathbb{P}}\ ,$$

to be the functor that sends objects Y to $Y|X$ and morphisms f to $f|1_X$. If $0_X : \emptyset \to X$ is the unique morphism form the initial presheaf to X, define $!X$ to be a colimit for the chain

$$X = \emptyset|X \xrightarrow{0_x|1_X} X|X \xrightarrow{(0_x|1_X)|1_X} (X|X)|X \xrightarrow{\qquad} \cdots \qquad .$$

By parametricity (Theorem 4.1.5), this induces a unique functor

$$! : \widehat{\mathbb{P}} \to \widehat{\mathbb{P}}\ .$$

[3] I.e., two presheaves over $\mathbb{P}$ or one over $\mathbb{P}$ and the other over $\mathbb{C}$ or ... .

8.4.3 Denotational semantics

Suppose P is a process term with free variables within $x_1, \cdots, x_n$ (possibly the empty list). The denotation of P in this context is a functor

$$[\![P[\vec{x}]]\!] : {\mathbb{P}_\perp}^n \to \widehat{\mathbb{P}},$$

Similarly a concretion or abstraction term with free variables within $x_1, \cdots, x_n$ are denoted by functors from ${\mathbb{P}_\perp}^n$ to $\widehat{\mathbb{C}}$ or $\widehat{\mathbb{F}}$ respectively. For conciseness and readability we give the semantic definition only on objects. As usual the definition is by structural induction on terms T:

$$\begin{array}{lr}
[\![\mathbf{Nil}[\vec{x}]]\!]\vec{p} = \emptyset, \text{ the empty presheaf} & [\![\tau.P[\vec{x}]]\!]\vec{p} = \tau.([\![P[\vec{x}]]\!]\vec{p}) \\
[\![a!C[\vec{x}]]\!]\vec{p} = a!([\![C[\vec{x}]]\!]\vec{p}) & [\![a?F[\vec{x}]]\!]\vec{p} = a?([\![F[\vec{x}]]\!]\vec{p}) \\
[\![\sum_{i\in I} P_i[\vec{x}]]\!]\vec{p} = \sum_{i\in I} [\![P_i[\vec{x}]]\!]\vec{p} & [\![[q \leq Q]P[\vec{x}]]\!]\vec{p} = [\![Q[\vec{x}]]\!]\vec{p}(q) \,.\, [\![P[\vec{x}]]\!]\vec{p} \\
[\![P \mid Q[\vec{x}]]\!]\vec{p} = [\![P[\vec{x}]]\!]\vec{p} \mid [\![Q[\vec{x}]]\!]\vec{p} & [\![rep(P)[\vec{x}]]\!]\vec{p} = rep([\![P[\vec{x}]]\!]\vec{p}) \\
[\![F \bullet P[\vec{x}]]\!]\vec{p} = ([\![F[\vec{x}]]\!]\vec{p})_!(\lfloor [\![P[\vec{x}]]\!]\vec{p} \rfloor) & [\![x_i[\vec{x}]]\!]\vec{p} = p_i \\
[\![\langle P\rangle Q[\vec{x}]]\!]\vec{p} = \langle [\![P[\vec{x}]]\!]\vec{p}\rangle [\![Q[\vec{x}]]\!]\vec{p} & [\![(y)P[\vec{x}]]\!]\vec{p} = [\![P[\vec{x}, y]]\!]\vec{p}
\end{array}$$

Above we have used $S \,.\, X$, where S is a set and X is a presheaf, to stand for the copower of X, the sum of X with itself S times as in most of the coend formulas that we have seen in this thesis; the copower could also be written as $\Sigma_{s\in S}X$.

We now observe the important fact that linear application amounts to substitution. This rests on the property that terms being linear ensures that viewed denotationally they preserve connected colimits in their free variables.

Lemma 8.4.14 (Substitution Lemma) *Let T be a term of the language and P be a process term. Let $\vec{x}$ be a set of variables including those free in P and suppose that $fv(T) \subseteq [\vec{x}, y]$. Then*

$$([\![T[\vec{x}, y]]\!]\vec{p})_!(\lfloor [\![P[\vec{x}]]\!]\vec{p} \rfloor) \cong [\![T[P/y][\vec{x}]]\!]\vec{p} \;.$$

Proof: The proof is by induction on the structure of T. For readability let's assume

$$A = \lfloor [\![P[\vec{x}]]\!]\vec{p} \rfloor \;.$$

The base case $T = \mathbf{Nil}$ is trivial, the other base case is $T = z$. Then we have two possibilities either $z = x_i$ for some x_i component of $\vec{x}$ or $z = y$. In the first case $[\![T[\vec{x}, y]]\!]\vec{p})_!$ is the functor that constantly returns p_i and $[\![T[P/y][\vec{x}]]\!]\vec{p} = [\![x_i[\vec{x}]]\!]\vec{p} = p_i$. In the other case,

$$\begin{array}{rcl}
([\![T[\vec{x}, y]]\!]\vec{p})_! A & = & ([\![y[\vec{x}, y]]\!]\vec{p})_! A \\
& = & \int^q A(q) \,.\, j(q) \\
& = & l^*(\int^q A(q) \,.\, \mathrm{y}_{\mathbb{P}_\perp}(q) \quad (\text{since } j = l^* \mathrm{y}_{\mathbb{P}_\perp}) \\
& = & l^*(A) \\
& = & [\![P[\vec{x}]]\!]\vec{p} \\
& = & [\![y[P/y][\vec{x}]]\!]\vec{p} \;.
\end{array}$$

The other possibilities are treated as follows:

- If $T = \tau.Q$:

$$\begin{array}{rcll}
([\![T[\vec{x},y]]\!]\vec{p})_!A & = & \int^q A(q)\,.\,(\tau.([\![Q[\vec{x},y]]\!]\vec{p}r)) & \\
& \cong & \tau.(\int^q A(q)\,.\,[\![Q[\vec{x},y]]\!]\vec{p}) & \text{(since } \tau. \text{ preserves} \\
& & & \text{connected colimits)} \\
& \cong & \tau.(([\![Q[\vec{x},y]]\!]\vec{p})_!A) & \\
& \cong & \tau.([\![Q[P/y][\vec{x}]]\!]\vec{p}) & \text{(by inductive hypothesis)} \\
& = & [\![\tau.Q[P/y][\vec{x}]]\!]\vec{p}\,. &
\end{array}$$

- If $T = a!C$ or $T = a?F$ one uses a similar argument as above, using the fact that $a!$ and $a?$ preserves connected colimits.
- If $T = \langle Q\rangle R$ with y free in Q and not in R:

$$\begin{array}{rcl}
([\![T[\vec{x},y]]\!]\vec{p})_!A & = & \int^q A(q)\,.\,\langle[\![Q[\vec{x},y]]\!]\vec{p}r\rangle[\![R[\vec{x}]]\!]\vec{p} \\
& \cong & \langle\int^q A(q)\,.\,[\![Q[\vec{x},y]]\!]\vec{p}r\rangle[\![R[\vec{x}]]\!]\vec{p} \\
& & \text{(since } \langle-\rangle- \text{ preserves connected colimits in each argument)} \\
& \cong & \langle([\![Q[\vec{x},y]]\!]\vec{p})_!A\rangle[\![R[\vec{x}]]\!]\vec{p} \\
& \cong & \langle[\![Q[P/y][\vec{x}]]\!]\vec{p}\rangle[\![R[\vec{x}]]\!]\vec{p} \\
& & \text{(by inductive hypothesis)} \\
& = & [\![(\langle Q\rangle R)[P/y][\vec{x}]]\!]\vec{p}\,.
\end{array}$$

  The case when y is free in R and not in Q is treated similarly.
- If $T = (z)Q$:

$$\begin{array}{rcll}
([\![T[\vec{x},y]]\!]\vec{p})_!A & = & \int^q A(q)\,.\,[\![Q[\widetilde{\vec{x},y,z}]]\!]\vec{p}r & \\
& \cong & \int^q A(q)\,.\,\widetilde{[\![Q[\vec{x}},y,z]]\!]\vec{p}r & \text{(since Currying preserve colimits} \\
& & & \text{being an equivalence)} \\
& \cong & [\![Q[\widetilde{P/y]}[\vec{x},z]]\!]\vec{p} & \text{(by inductive hypothesis)} \\
& \cong & [\![(z)Q[P/y][\vec{x}]]\!]\vec{p}\,. &
\end{array}$$

- If $T = F \bullet Q$ and y is not free in Q:

$$\begin{array}{rcl}
([\![T[\vec{x},y]]\!]\vec{p})_!A & = & \int^q A(q)\,.\,\int^r \lfloor[\![Q[\vec{x}]]\!]\vec{p}\rfloor r\,.\,[\![F[\widetilde{\vec{x},y]}]\!]\vec{p}qr \\
& \cong & \int^r \lfloor[\![Q[\vec{x}]]\!]\vec{p}\rfloor r\,.\,\int^q A(q)\,.\,[\![F[\widetilde{\vec{x},y]}]\!]\vec{p}qr \\
& & \qquad \text{(by Fubini and since copowers distribute over coends)} \\
& \cong & \int^r \lfloor[\![Q[\vec{x}]]\!]\vec{p}\rfloor r\,.\,[\![F[\widetilde{P/y]}[\vec{x}]]\!]\vec{p}r \\
& & \qquad \text{(by inductive hypothesis)} \\
& \cong & [\![(F \bullet Q)[P/y][\vec{x}]]\!]\vec{p}\,.
\end{array}$$

  The case when y is not free in F is treated similarly.
- If $T = rep(P)$ then the property trivially holds, since P has no free variables (as it was for **Nil**).

- If $T = [r \le Q]R$ and y is not a free variable of R then:

$$\begin{array}{rcl}
(\llbracket T[\vec{x}, y]\rrbracket\vec{p})_!A & = & \int^q A(q)\,.\,((\llbracket Q[\vec{x}, y]\rrbracket\vec{p}q)(r)\,.\,\llbracket R[\vec{x}]\rrbracket\vec{p}) \\
& \cong & \int^q (A(q) \times (\llbracket Q[\vec{x}, y]\rrbracket\vec{p}q)(r))\,.\,,\llbracket R[\vec{x}]\rrbracket\vec{p} \\
& & \qquad\text{(by definition of copower)} \\
& \cong & \llbracket Q[P/y][\vec{x}]\rrbracket\vec{p})(r))\,.\,\llbracket R[\vec{x}]\rrbracket\vec{p} \\
& & \qquad\text{(since coends are calculated pointwise)} \\
& = & \llbracket([r \le Q]R)[P/y][\vec{x}]\rrbracket\vec{p}\ .
\end{array}$$

If y is not a free variable of Q:

$$\begin{array}{rcl}
(\llbracket T[\vec{x}, y]\rrbracket\vec{p})_!A & = & \int^q A(q)\,.\,((\llbracket Q[\vec{x}]\rrbracket\vec{p})(r)\,.\,\llbracket R[\vec{x}, y]\rrbracket\vec{p}q) \\
& \cong & (\llbracket Q[\vec{x}]\rrbracket\vec{p})(r)\,.\,\int^q (A(q)\,.\,,\llbracket R[\vec{x}, y]\rrbracket\vec{p}q \\
& & \qquad\text{(since copowers distribute over coends)} \\
& \cong & (\llbracket Q[\vec{x}]\rrbracket\vec{p})(r)\,.\,\llbracket R[P/y][\vec{x}]\rrbracket \\
& & \qquad\text{(by inductive hypothesis)} \\
& = & \llbracket([r \le Q]R)[P/y][\vec{x}]\rrbracket\vec{p}\ .
\end{array}$$

- If $T = Q|R$ and y is not a free variable of R one uses, as for the $\langle - \rangle -$ case, that the functor $(-|-)$ preserves connected colimits in each argument.
- If $T = \sum_{i \in I} P_i$ then:

$$\begin{array}{rcll}
(\llbracket T[\vec{x}, y]\rrbracket\vec{p})_!A & = & \int^q A(q)\,.\,\sum_{i \in I} \llbracket P_i[\vec{x}, y]\rrbracket\vec{p}q & \\
& \cong & \sum_{i \in I} \int^q A(q)\,.\,\llbracket P_i[\vec{x}, y]\rrbracket\vec{p}q & \text{(since sums preserve coends)} \\
& \cong & \sum_{i \in I} \llbracket P_i[P/y][\vec{x}]\rrbracket\vec{p} & \text{(by inductive hypothesis)} \\
& = & \llbracket \sum_{i \in I} P_i[P/y][\vec{x}]\rrbracket\vec{p}\ . &
\end{array}$$

□

An immediate consequence of the Lemma 8.4.14 is the following characterisation of the denotation of the linear application of the language.

Corollary 8.4.15 *Let P, Q be process terms with free variable amongst $\vec{x}$. Then,*

$$\llbracket (y)Q \bullet P[\vec{x}]\rrbracket\vec{p} \cong \llbracket Q[P/y][\vec{x}]\rrbracket\vec{p}\ .$$

Proof: The proof is given by the following calculation:

$$\begin{array}{rcll}
\llbracket (y)Q \bullet P[\vec{x}]\rrbracket\vec{p} & = & (\llbracket (y)Q[\vec{x}, y]\rrbracket\vec{p})_!(\lfloor \llbracket P[\vec{x}]\rrbracket\vec{p} \rceil) & \text{(by definition)} \\
& \cong & \llbracket Q[P/y][\vec{x}]\rrbracket\vec{p} & \text{(by Lemma 8.4.14)}\ .
\end{array}$$

□

8.4.4 A soundness result

We now use the transition relations on presheaves to show the "soundness" of the operational semantics with respect to the presheaf semantics.

Theorem 8.4.16 *Let T be any closed term of the language. For every path term t, if $T \xrightarrow{t}$ then $[\![T]\!] \xrightarrow{t}$ and if t is atomic and $T \xrightarrow{t} U$ then there exists an X such that $[\![T]\!] \xrightarrow{t} X$ with $X \cong [\![U]\!]$.*

Proof: The proof goes by induction on the rules.
Prefixings: If

$$\frac{}{T = \alpha.U \xrightarrow{\alpha} U}$$

then $[\![T]\!] = \alpha.[\![U]\!] \xrightarrow{\alpha} [\![U]\!]$.
Sums: If

$$\frac{P_j \xrightarrow{\alpha} U \quad j \in I}{T = \sum_{i\in I} P_i \xrightarrow{\alpha} U}$$

then $[\![T]\!] = \sum_{i\in I} [\![P_i]\!]$ and since by inductive hypothesis $[\![P_i]\!] \xrightarrow{\alpha} X \cong [\![U]\!]$ then

$$[\![T]\!] \xrightarrow{\alpha} X' \cong X \cong [\![U]\!]$$

by definition.
Tests: If

$$\frac{Q \xrightarrow{p} \qquad P \xrightarrow{\alpha} U}{T = [p \leq Q]P \xrightarrow{\alpha} U}$$

then $[\![T]\!] = [\![Q]\!](p) \,.\, [\![P]\!]$. By inductive hypothesis

$$[\![P]\!] \xrightarrow{\alpha} X \cong [\![U]\!]$$

and $[\![Q]\!](p) \neq \emptyset$ since $[\![Q]\!] \xrightarrow{p}$. Hence, by definition of copower there must exists an $X' \cong X$ such that $[\![T]\!] \xrightarrow{\alpha} X'$.
Parallel composition: If

$$\frac{P \xrightarrow{a?} F \quad Q \xrightarrow{a!} \langle R\rangle S}{T = P \mid Q \xrightarrow{\tau} F \bullet R \mid S}$$

then by inductive hypothesis $P \xrightarrow{a?} A \cong [\![F]\!]$ and $Q \xrightarrow{a!} \langle X\rangle Y \cong \langle [\![R]\!]\rangle [\![S]\!]$. Because of the characterisation result of Theorem 8.4.10, we know that

$$[\![P|Q]\!] \xrightarrow{\tau} X \cong \widetilde{A}_! X|Y \cong \widetilde{[\![F]\!]}_!([\![R]\!])|[\![S]\!] = [\![F \bullet R]\!]|[\![S]\!] = [\![F \bullet R|S]\!] \ .$$

If

$$\frac{P \xrightarrow{a?} (x)P'}{T = P \mid Q \xrightarrow{a?} (x)(P' \mid Q)}$$

then by inductive hypothesis $P \xrightarrow{a?} A \cong [\![(x)P']\!]$ and hence again by the characterisation Theorem 8.4.10

$$(\![P|Q]\!) \xrightarrow{a?} X \cong A|[\![Q]\!] \cong [\![(x)P']\!]|[\![Q]\!] \cong [\![(x)(P'|Q)]\!] \ ,$$

where the last passage is justified by point 5 of Lemma 8.4.9. The other cases concerning parallel composition are treated in a similar way to the two above.
Replication: If

$$\frac{P \mid rep(P) \xrightarrow{\alpha} U}{T = rep(P) \xrightarrow{\alpha} U}$$

then by inductive hypothesis $[\![P|rep(P)]\!] \xrightarrow{\alpha} X \cong [\![U]\!]$. But, by construction

$$[\![T]\!] \cong [\![P|rep(P)]\!]$$

hence there must exists an X' such that

$$[\![T]\!] \xrightarrow{\alpha} X' \cong X \ .$$

Application: If

$$\frac{P[Q/x] \xrightarrow{\alpha} U}{T = (x)P \bullet Q \xrightarrow{\alpha} U}$$

then by inductive hypothesis $[\![P[Q/x]]\!] \xrightarrow{\alpha} X \cong [\![U]\!]$. By Corollary 8.4.15, there must exists an X' isomorphic to X such that $[\![(x)P \bullet Q]\!] \xrightarrow{\alpha} X'$.
Concretions: If

$$\frac{P \xrightarrow{\alpha} U}{T = \langle P\rangle Q \xrightarrow{\langle\alpha\rangle -} U}$$

then by inductive hypothesis $[\![P]\!] \xrightarrow{\alpha} X \cong [\![U]\!]$ and hence by Definition 8.4.1,

$$[\![T]\!] = \langle [\![P]\!]\rangle [\![Q]\!] \xrightarrow{\langle\alpha\rangle -} X \ .$$

Abstractions: If

$$\frac{F \bullet p \xrightarrow{\alpha} U}{T = F \xrightarrow{(p \mapsto \alpha)} U}$$

then by inductive hypothesis $[\![F \bullet p]\!] \xrightarrow{\alpha} X \cong [\![U]\!]$. Recall now that

$$[\![F]\!](p \mapsto \alpha) = [\![F \bullet p]\!](\alpha) \ ,$$

hence $[\![T]\!] \xrightarrow{p \mapsto \alpha} X$.
Path Capabilities: If

$$\frac{P \xrightarrow{\alpha} U \quad U \xrightarrow{t}}{T = P \xrightarrow{\alpha.t}}$$

then by inductive hypothesis $[\![T]\!] = [\![P]\!] \xrightarrow{\alpha} X \cong [\![U]\!]$ and $[\![U]\!] \xrightarrow{t}$ and therefore, $X \xrightarrow{t}$, too. By Proposition 8.4.2, then

$$[\![T]\!] \xrightarrow{\alpha.t} \ .$$

The other cases are trivial. □

The converse result, showing that

> if $[\![T]\!] \xrightarrow{t}$, then $T \xrightarrow{t}$ and if t is atomic and $[\![T]\!] \xrightarrow{t} X$, then there exists U such that $T \xrightarrow{t} U$ with $[\![U]\!] \cong X$

is not so easily obtainable. A plain induction on the structure of the terms will encounter difficulties in treating the application $(F \bullet P)$ and parallel composition $(P|Q)$ cases. A more refined strategy is needed. This problem is related to the unproven conjecture (see Section 8.4.5 below) stating the existence of bijections between elements of a presheaf, denotation of a term T, and derivation trees in the operational semantics of T.

8.4.5 Toward a characterisation of open map bisimulation

We say two presheaves are *open-map bisimilar* iff they are related by a span of surjective open maps. This induces a relation between closed terms; closed terms T and U are related iff their denotations $[\![T]\!]$ and $[\![U]\!]$ are open-map bisimilar in either $\widehat{\mathbb{P}}$, $\widehat{\mathbb{C}}$ or $\widehat{\mathbb{F}}$, depending on the type of T and U. It is this relation on terms we wish to characterise. Because of the intertwined definition of $\mathbb{P}$, $\mathbb{C}$ and $\mathbb{F}$, an operational characterisation of $\mathbb{P}$-open map bisimilarity would necessary involve a characterisation of $\mathbb{C}$- and $\mathbb{F}$-open map bisimilarity, too. It is the characterisation of the latter which poses most problems. In fact using the decomposition results of Theorem 8.4.4 it is easy to prove the following:

Proposition 8.4.17 *Let X and Y be two presheaves over $\mathbb{P}$, then X is $\mathbb{P}$-open bisimilar to Y if and only if the following three conditions (and their symmetric counterpart) hold:*

1. *For every X', if $X \xrightarrow{\tau} X'$ then there exists Y' such that $Y \xrightarrow{\tau} Y'$ and X' is $\mathbb{P}$-open bisimilar to Y'.*
2. *For every C, if $X \xrightarrow{a!} C$ then there exists D such that $Y \xrightarrow{a!} D$ and C is $\mathbb{C}$-open bisimilar to D.*
3. *For every F, if $X \xrightarrow{a?} F$ then there exists G such that $Y \xrightarrow{a?} G$ and F is $\mathbb{F}$-open bisimilar to G.*

Before proving the proposition we introduce, for the purpose of this section, some abbreviations for open map bisimilarity.

NOTATION: In the reminder of this section we shall write $X \sim_{\mathbb{P}} Y$ to mean that the presheaf X is $\mathbb{P}$-open bisimilar to the presheaf Y. Similarly we write $C \sim_{\mathbb{C}} D$ to mean that C is $\mathbb{C}$-open bisimilar to D and $F \sim_{\mathbb{F}} G$ to mean that F is $\mathbb{F}$-open bisimilar to G.

Proof:[of Proposition 8.4.17]

"Only if": We need to define a span of surjective open maps

$$X \xleftarrow{f} Z \xrightarrow{g} Y \ .$$

For every $i \in X(\tau)$, define $Y_{\tau,i} = \{i' \in Y(\tau) \mid X_{|i} \sim_{\mathbb{P}} Y_{|i'}\}$. By assumption, $Y_{\tau,i}$ is never empty. For every $i' \in Y_{\tau,i}$, let $X_{|i} \xleftarrow{f^{\tau,i,i'}} Z^{\tau,i,i'} \xrightarrow{g^{\tau,i,i'}} Y_{|i'}$ be a span of surjective open maps that needs to exist by assumption. Similarly for $a!$ and $a?$ choose spans

$X_{|j} \overset{f^{a!,j,j'}}{\longleftarrow} Z^{a!,j,j'} \overset{g^{a!,j,j'}}{\longrightarrow} Y_{|j'}$ and $X_{|k} \overset{f^{a?,k,k'}}{\longleftarrow} Z^{a?,k,k'} \overset{g^{a?,k,k'}}{\longrightarrow} Y_{|k'}$ for $j' \in Y_{a!,j}$ and $k' \in Y_{a?,k}$ where $Y_{a!,j}$ and $Y_{a?,k}$ are defined similarly to $Y_{\tau,i}$. Define now the presheaf $Z : \mathbb{P}^{\mathrm{op}} \to \mathbf{Set}$ as follows on objects:

$$\begin{array}{rcl} Z(\tau) & = & \{(i,i') \in X(\tau) \times Y(\tau) | i' \in Y_{\tau,i}\} \\ Z(a!) & = & \{(j,j') \in X(a!) \times Y(a!) | j' \in Y_{a!,j}\} \\ Z(a?) & = & \{(k,k') \in X(a?) \times Y(a?) | k' \in Y_{a?,k}\} \\ Z(\tau.p) & = & \displaystyle\sum_{i \in X(\tau)} \sum_{i' \in Y_{\tau,i}} Z^{\tau,i,i'}(p) \\ Z(a!c) & = & \displaystyle\sum_{j \in X(a!)} \sum_{j' \in Y_{a!,j}} Z^{a!,j,j'}(c) \\ Z(a?f) & = & \displaystyle\sum_{k \in X(a?)} \sum_{k' \in Y_{a?,k}} Z^{a?,k,k'}(f) \ . \end{array}$$

The action on morphisms is given by the action of the presheaves $Z^{\tau,i,i'}$, $Z^{a!,j,j'}$ and $Z^{a?,k,k'}$ but for the cases

$$\tau \le \tau.p, \quad a! \le a!c, \quad \text{and} \quad a? \le a?f$$

that are determined in the obvious way. The natural transformations f and g are determined by projections on the first and second components on the base cases and by the open maps $f^{\tau,i,i'}$, $g^{\tau,i,i'}$, ... , in the other cases. Because of the assumptions it is immediately deducible that f and g are both surjective and $\mathbb{P}$-open.
"If": This direction is trivial. One can compare it with Proposition 7.3.7.

□

We have already seen that for any two presheaves X and Y over a product category $\mathbb{A} \times \mathbb{B}$, if X is $\mathbb{A} \times \mathbb{B}$-open bisimilar to Y then for every object $A \in |\mathbb{A}|$, $X(A,-)$ is $\mathbb{B}$-open bisimilar to $Y(A,-)$ and for every object $B \in |\mathbb{B}|$, $X(-,B)$ is $\mathbb{A}$-open bisimilar to $Y(-,B)$. The converse does not hold in general. It does so in the special case represented by the denotations of concretions.

Proposition 8.4.18 *Let X, Y, Z and W be presheaves over $\mathbb{P}$, then $\langle X \rangle Y$ is $\mathbb{C}$-open bisimilar to $\langle Z \rangle W$ if and only if X is $\mathbb{P}$-open bisimilar to Z and Y is $\mathbb{P}$-open bisimilar to W.*

Proof: One direction we have already commented on; the other we can prove directly or invoke what we already said in Section 8.4.2 where the functor $\langle - \rangle -$ was defined and shown to preserve open map bisimulation for general reasons. □

For the denotation of concretions, $\mathbb{C}$-open bisimilarity is then reduced to $\mathbb{P}$-open bisimilarity. For *abstractions*, i.e., presheaves over

$$\mathbb{F} = \mathbb{P}_\perp \multimap \mathbb{P} = (\mathbb{P}_\perp)^{\mathrm{op}} \times \mathbb{P} \ ,$$

things do not work so simply. Clearly $F \sim_{\mathbb{F}} G$ implies that $\widetilde{F}(p) \sim_{\mathbb{P}} \widetilde{G}(p)$, for every $p \in\mid \mathbb{P} \mid$. Some uniformity constraints need to be imposed on a family of spans of surjective $\mathbb{P}$-open maps $(\widetilde{F}(p) \leftarrow Z_p \rightarrow \widetilde{G}(p))_{p\in|\mathbb{P}_\perp|}$ in order to be able to glue them into a 'single' span of surjective F-open maps

$$F \leftarrow Z \rightarrow G \ .$$

In order to match up these constraints we are, apparently unavoidably, led to consider elements of the presheaves explicitly.

Proposition 8.4.19 *Let F and G be two presheaves over $\mathbb{F}$. A family of spans of surjective $\mathbb{P}$-open maps*

$$(\widetilde{F}(p) \stackrel{f_p}{\leftarrow} Z_p \stackrel{g_p}{\rightarrow} \widetilde{G}(p))_{p\in|\mathbb{P}_\perp|}$$

induces a span of surjective $\mathbb{F}$-open maps

$$F \stackrel{f}{\leftarrow} Z \stackrel{g}{\rightarrow} G$$

if and only if for every $p \leq p'$ in $\mathbb{P}_\perp$ we have

1. *$(q,x) \sim_{Z_p} (q,y)$ implies $(q, \widetilde{F}(p \leq p')_q(x)) \sim_{Z_{p'}} (q, \widetilde{G}(p \leq p')_q(y))$.*
2. *$(q, \widetilde{F}(p \leq p')_q(x)) \sim_{Z_{p'}} (q, y')$ implies that there exists $y \in \widetilde{G}(p)q$ such that*

$$y' = \widetilde{G}(p \leq p')_q(y) \qquad \text{and} \qquad (q,x) \sim_{Z_p} (q,y) \ .$$

3. *$(q,x') \sim_{Z_{p'}} (q, \widetilde{G}(p \leq p')_q(y))$ implies that there exists $x \in \widetilde{F}(p)q$ such that*

$$x' = \widetilde{F}(p \leq p')_q(x) \qquad \text{and} \qquad (q,x) \sim_{Z_p} (q,y) \ .$$

Where $(q,x) \sim_{Z_p} (q,y)$ is an abbreviation for "there exists $z \in Z(p)$, such that $(f_p)_q(z) = x$ and $(g_p)_q(z) = y$".

Proof:[Sketch] For simplicity, we can, without loss of generality assume that all the Z_p's are in fact pointwise subsets of the products $\widetilde{F}(p) \times \widetilde{G}(p)$, i.e., every $z \in Z_p(q)$ is in fact a pair $(x,y) \in \widetilde{F}(p)(q) \times \widetilde{G}(p)(q)$. Define then $Z : \mathbb{F}^{\mathrm{op}} \rightarrow \mathbf{Set}$ as $Z(p \mapsto q) = Z_p(q)$, with

$$Z(p \leq p', q' \leq q)(x,y) = Z'_p(q' \leq q)(\widetilde{F}(p \leq p')_q(x), \widetilde{G}(p \leq p')_q(y))$$

for any pair $(x,y) \in Z(p,q)$. Condition 1 ensures that Z is well defined, while conditions 2 and 3 provide the quasi-pullback conditions for the projections to be $\mathbb{F}$-open. □

To characterise $\mathbb{F}$-open bisimilarity for abstractions in the language, we should then annotate the transitions of the operational semantics with expressions accounting for the extra information required by the conditions 1, 2 and 3 in the proposition above. We conjecture the existence of a tight correspondence between derivation trees associated with a closed process term T and path t and elements of $[\![T]\!](t)$. We believe in fact, though we lack a proof, that there are bijections

$$[\![T]\!](t) \cong \{d \mid d \text{ is a derivation tree for } T \xrightarrow{t}\} \ ,$$

for every term T and path t. Our hope is to describe the functorial action of presheaves and open map bisimulation via annotations to the operational semantics. Even without this, there is already room for some improvement on the conditions 2 and 3 of Proposition 8.4.19 based on a condition satisfied by presheaves denoting terms of the language which reminds of Berry's stability condition for functions [13].

Theorem 8.4.20 *Let T be a term with free variables $\vec{x} = x_1, \ldots, x_n$. If $d \in ([\![T[\vec{x}]]\!]\vec{p})(q)$, for $\vec{p} = p_1, \ldots, p_n$ a vector of (lifted) paths, then there exists a pointwise smaller vector $\vec{p^0} \leq \vec{p}$ and element $d^0 \in ([\![T[\vec{x}]]\!]\vec{p^0})(q)$ such that*

1. $d = ([\![T[\vec{x}]]\!](\vec{p^0} \leq \vec{p}))_q(d^0)$
2. *$\vec{p^0}$ is the least vector satisfying 1, i.e., if $\vec{p^1} \leq \vec{p}$ and $d^1 \in ([\![T[\vec{x}]]\!]\vec{p^0})(q)$ is such that $d = ([\![T[\vec{x}]]\!](\vec{p^1} \leq \vec{p}))_q(d^1)$ then $\vec{p^0} \leq \vec{p^1}$ and $d^1 = ([\![T[\vec{x}]]\!](\vec{p^0} \leq \vec{p^1}))_q(d^0)$.*

Proof: The proof is by induction on the structure of the term T. It makes essential use of the linearity constraint on terms. Most cases are trivial. We concentrate on the two that require looking into the coend definition of the semantics operations.
Let $T = P|Q$ and let $\vec{y}$ and $\vec{z}$ be the free variables of P and Q, respectively. By the linearity constraint we know that $fv(P) \cap fv(Q) = \emptyset$. Let $\vec{p^P}$ and $\vec{p^Q}$ the corresponding splitting of the vector $\vec{p}$ in the hypothesis. We then have:

$$[\![T]\!]\vec{p}(q) = \int^{r,s} \lfloor [\![P]\!]\vec{p^P} \rfloor(r) \times \lfloor [\![Q]\!]\vec{p^Q} \rfloor(s) \times (r \parallel s)(q) \ .$$

Since $d \in [\![T]\!]\vec{p}(q)$ there must exists r and s and a triple

$$(x, y, z) \in \lfloor [\![P]\!]\vec{p^P} \rfloor(r) \times \lfloor [\![Q]\!]\vec{p^Q} \rfloor(s) \times (r \parallel s)(q)$$

such that $d = [(x, y, z)]_\sim$ (cf. Section 4.1). By Lemma 8.4.13 we know that within the equivalence class $[(x, y, z)]_\sim$, we can minimise r and s. Let r' and s' be the result of such minimisation, with (x', y', z') the corresponding triple of elements. If $r = \perp$, then $s \neq \perp$, since $\perp \parallel \perp$ is the empty presheaf. So $s = \lfloor p_s \rfloor$, for some path $p_s \in |\mathbb{P}|$. Define $(\vec{p^P})^0 = \vec{\perp}$, the constantly bottom vector and $(\vec{p^Q})^0$ the one obtained by applying the inductive hypothesis to $([\![Q]\!]\vec{p^Q})(p_s)$. Let y^0 be the corresponding element such that $([\![Q]\!]((\vec{p^Q})^0 \leq \vec{p^Q}))_{p_s}(y^0) = y'$ and define $d^0 = [(*, y^0, z')]_\sim$ in

$$([\![P|Q]\!]\vec{\perp}, (\vec{p^Q})^0)(q) = \int^{r,s} \lfloor [\![P]\!]\vec{\perp} \rfloor(r) \times \lfloor [\![Q]\!](\vec{p^Q})^0 \rfloor(s) \times (r \parallel s)(q) \ .$$

If $r' \neq \perp$ and $s' = \perp$ just do the symmetric thing and if they are both non bottom, minimise separately both $\vec{p^P}$ and $\vec{p^Q}$.
The other "complicated" case is when $T = (x)P \bullet Q$. In this case

$$([\![T]\!]\vec{p})(q) = \int^{r} \lfloor [\![Q]\!]\vec{p^Q} \rfloor(r) \times ([\![P]\!]\vec{p^P}, r)(q) \ ,$$

where again we have split the free variables of T in two disjoint sets. Now $d \in [(x, y)]_\sim$, with $(x, y) \in \lfloor [\![Q]\!]\vec{p^Q} \rfloor(r) \times ([\![P]\!]\vec{p^P}, r)(q)$ for some r. Let $y' \in [\![P]\!]((\vec{p^P})^0, r^0)(q)$ be the minimal representative for y that we can find by the inductive hypothesis. If $r^0 = \bot$, then define $(p^{\vec{Q}})^0 = \vec{\bot}$, and take $d^0 = [(*, y')]$, otherwise apply the inductive hypothesis again on $([\![Q]\!]p^{\vec{Q}})(p_{r^0} \leq p_r))(x)$, for $r^0 = \lfloor p_{r^0} \rfloor$ and $r = \lfloor p_r \rfloor$, to obtain $(p^{\vec{Q}})^0 \leq p^{\vec{Q}}$ and

$$x' \in ([\![Q]\!]p^{\vec{Q}})(p_{r^0}) = (\lfloor [\![Q]\!] \rfloor p^{\vec{Q}})(r^0) \ .$$

define $d^0 = [(x', y')]_\sim$. □

NOTATION: In the situation of Theorem 8.4.20 we say that $\vec{p^0}$ is *minimum for* (q, d).

We can use Theorem 8.4.20 to simplify conditions 2 and 3 of Proposition 8.4.19 by examining only *minimum inputs.*

Corollary 8.4.21 *Let F and G be two presheaves over $\mathbb{F}$. A family of spans of surjective $\mathbb{P}$-open maps*

$$(\widetilde{F}(p) \stackrel{f_p}{\leftarrow} Z_p \stackrel{g_p}{\rightarrow} \widetilde{G}(p))_{p \in \mathbb{P}_\bot|}$$

induce a span of surjective $\mathbb{F}$-open maps

$$F \stackrel{f}{\leftarrow} Z \stackrel{g}{\rightarrow} G$$

if and only if

1. *for every $p \leq p'$ in $\mathbb{P}_\bot$ then $(q, x) \sim_{Z_p} (q, y)$ implies*

$$(q, \widetilde{F}(p \leq p')_q(x)) \sim_{Z_{p'}} (q, \widetilde{G}(p \leq p')_q(y)) \ .$$

2. *for every path p, $(q, x) \sim_{Z_p} (q, y')$ and p minimum for (q, x) implies that p is minimum for (q, y) and vice versa if p is minimum for (q, y) then it is minimum for (q, x), too.*

8.4.6 Applicative bisimulation recovered

The terms of the process language include as a fragment a form of λ-calculus, on which bisimulation from open maps can be characterised more simply than for the full process language. The characterisation is in terms of a relation of applicative bisimulation [2].

We restrict the syntactic categories to *deterministic* terms, linear terms with the following syntax:

$$\begin{array}{lcl} P & ::= & \mathbf{Nil} \mid \tau.P \mid a?F \mid [p \leq P']P \mid F \bullet P \mid x \\ F & ::= & (x)P \end{array}$$

The operational semantics restricts to this fragment of the language. As presheaves, these deterministic terms are subobjects of the *terminal object.*

Proposition 8.4.22 *In a presheaf category $\widehat{\mathbb{C}}$ a terminal object $1_{\mathbb{C}}$ is given by the presheaf that takes any object C of $\mathbb{C}$ to the singleton set $\{*\}$.*

A presheaf X over $\mathbb{C}$ is a subobject of $1_{\mathbb{C}}$ if at every object C of $\mathbb{C}$, $X(C)$ is either the empty set or a singleton set. We write $X \hookrightarrow 1_{\mathbb{C}}$ to mean that X is a subobject of $1_{\mathbb{C}}$.

Proposition 8.4.23 *Let T be any term in the restricted language with free variables in $\vec{x}$. Then for every vector of matching length of closed terms possibly in the full language, $\vec{P}$, if every P_i in $\vec{P}$ is such that $[\![P_i]\!]$ is a subobject of the terminal $1_{\mathbb{P}}$, then $[\![T[\vec{P}/\vec{x}]]\!]$ is a subobject of $1_{\mathbb{T}}$.*

Proof: The proof is by induction on the structure of T.

- If $T = \mathbf{Nil}$, $[\![T[\vec{P}/\vec{x}]]\!] \stackrel{\text{def}}{=} \emptyset \hookrightarrow 1_{\mathbb{P}}$.
- If $T = \tau.Q$, then by inductive hypothesis, $[\![Q[\vec{P}/\vec{x}]]\!] \hookrightarrow 1_{\mathbb{P}}$, then

$$[\![T[\vec{P}/\vec{x}]]\!] = \tau.[\![Q[\vec{P}/\vec{x}]]\!] \hookrightarrow 1_{\mathbb{P}} \quad \text{(cf. definition of } \tau.) \ .$$

- If $T = a?F$, then by inductive hypothesis $[\![F[\vec{P}\vec{x}]]\!] \hookrightarrow 1_{\mathbb{F}}$, then

$$[\![T[\vec{P}/\vec{x}]]\!] = a?([\![F[\vec{P}\vec{x}]]\!] \hookrightarrow 1_{\mathbb{P}}$$

- If $T = [p \leq Q]R$ then by inductive hypothesis, $[\![Q[\vec{P}/\vec{x}]]\!](p)$ is either the empty set or a singleton set. In the first case $[\![T[\vec{P}/\vec{x}]]\!] = \emptyset \hookrightarrow 1_{\mathbb{P}}$. Otherwise, by inductive hypothesis $[\![T[\vec{P}/\vec{x}]]\!] \cong [\![R[\vec{P}/\vec{x}]]\!]$ that is a subobject of $1_{\mathbb{P}}$ by inductive hypothesis.
- If $T = (y)Q \bullet R$, then

$$[\![T[\vec{P}/\vec{x}]]\!] \cong [\![Q[\vec{P}]]\!]_!(\lfloor [\![R[\vec{P}/\vec{x}]]\!] \rfloor) \cong [\![Q[\vec{P}/\vec{x}][R[\vec{P}/\vec{x}]/y]]\!] \ ,$$

  where the last isomorphism depends on Corollary 8.4.15. But, $[\![Q[\vec{P}/\vec{x}][R[\vec{P}/\vec{x}]/y]]\!]$ is known to be a subobject of $1_{\mathbb{P}}$ by inductive hypothesis.
- If $T = y$, $[\![T[\vec{P}/\vec{x}]]\!] = [\![P_y]\!]$ that satisfies the condition by assumption.
- If $T = (y)Q$, then for every $q \mapsto r$ object of $\mathbb{F}$,

$$[\![T[\vec{P}/\vec{x}]]\!](q \mapsto r) \cong [\![Q[\vec{P}/\vec{x}][q/y]]\!](r) \ .$$

  Representables are subobjects of the terminal because $\mathbb{P}$, $\mathbb{C}$ and $\mathbb{F}$ are partial orders, hence $[\![Q[\vec{P}/\vec{x}][q/y]]\!](r)$ is either the empty set or a singleton set by inductive hypothesis.

□

For terms in this fragment, at each path object, there is only at most one possible way of deriving a transition. Open map bisimulation between closed deterministic terms can be characterised using relations of a familiar form between terms:

Theorem 8.4.24 *Let T and U be closed terms in the restricted language. Their denotations as presheaves $[\![T]\!]$ and $[\![U]\!]$ are open-map bisimilar iff there are symmetric relations $\mathcal{R}^{\mathbb{P}}$ between closed process terms and $\mathcal{R}^{\mathbb{F}}$ between closed abstractions, relating the terms (i.e. so $T\ \mathcal{R}^{\mathbb{P}}\ U$ or $T\ \mathcal{R}^{\mathbb{F}}\ U$), such that:*

- *Whenever $P\ \mathcal{R}^{\mathbb{P}}\ Q$,* $P \xrightarrow{\tau} P' \Rightarrow \exists Q'.\ Q \xrightarrow{\tau} Q' \ \&\ P'\ \mathcal{R}^{\mathbb{P}}\ Q'$, *and*
  $P \xrightarrow{a?} F \Rightarrow \exists G.\ Q \xrightarrow{a?} G \ \&\ F\ \mathcal{R}^{\mathbb{F}}\ G$.
- *Whenever $(x)Q\ \mathcal{R}^{\mathbb{F}}\ (y)R$, then $Q[P/x]\ \mathcal{R}^{\mathbb{P}}\ R[P/y]$, for all closed process terms P in the restricted language.*

8.5 Some remarks

In this last chapter we have presented a presheaf model for a linear higher-order process language. It comes automatically equipped with the result that open map bisimulation is a congruence. We outlined the problems we have encountered so far in trying to give a characterisation of open map bisimulation based on the operational semantics.We still hope to obtain purely operational characterisation of open map bisimulation. We presently lack informative examples and counterexamples to probe variations in the way we think the characterisation can be set up. We expect and hope to obtain a fruitful operational reading of a broad range of presheaf semantics in which the elements of presheaf denotations correspond to derivations in an operational semantics.

Chapter 9

Conclusion

9.1 Summary

Building mainly on the work of Winskel and Nielsen [141] on categorical *models for concurrency* and Joyal, Nielsen and Winskel [64] on *open map bisimulation* in this thesis we have studied presheaf categories as models for concurrency with a built-in notion of bisimulation. Our research has developed in two directions. First we have refined the axiomatisation of models implicit in [141] and proposed the notion of presheaf models for **CCS**-like languages. One advantage of these models over the more traditional ones is that, for general reasons, bisimulation from open maps is a congruence. This was shown to be useful in proving similar congruence results in traditional models using known embeddings of the latter in presheaf ones. Our example was a presheaf model generalising event structures, so we could prove that strong history preserving bisimulation for event structures is a congruence with respect to the general process language **Proc** of [141]. Further, in a similar way, the refinement for event structures proposed in [41] was shown to preserve strong history preserving bisimulation. Crucial for proving all this was the result (Proposition 3.2.5) asserting that colimit preserving functors between presheaf categories preserves open map bisimulation.

The second direction we took was that of considering the 2-category **Cocont** of presheaf categories and colimit preserving functors as a category of non-deterministic domains as suggested in [138]. This was done for the purpose of describing presheaf categories, appropriate for modelling specific process languages, as initial solutions to recursive domain equations. The connection with domain theory has been made formal by developing, in the vein of axiomatic domain theory [30, 105, 35], suitably generalised versions of the classical notions and results. Notably Theorem 6.1.9, leading to a limit/colimit coincidence result, generalises Theorem 2 of [125]. The generalisation requires moving from order enriched categories to 2-categories and, following some established folklore, moving from *embedding-projection pairs* to the more general *adjoint pairs*. On the same line as [30], we defined axiomatically a class of *pseudo* algebraically compact [35] 2-categories, which included **Cocont**. Further, for the 2-categories in the class, parametric properties of *free* algebras were proved. All this has been used not just

to formalise our intuitions about **Cocont** but also to study open map bisimulation from a domain theoretical point of view. We established induction and coinduction principles of recursively defined domains [100, 99, 31] and we used these to give a domain theoretical characterisation of strong bisimulation for *arbitrary* trees. Proposition 3.2.5 mentioned above has been generalised to a proof that the horizontal composition of open 2-cells in **Cocont** preserve open map bisimulation and used to show that also connected colimit preserving functors between presheaf categories preserve it. All the functors needed to model process constructors fell into this latter class; hence for presheaf models we could immediately deduce the congruence property of bisimulation.
We tested our approach with several examples ranging from **CCS** [82], **CCS** with late value passing (as was done in [138]), π-calculus [87, 88] to a form of **CCS** with "linear" process passing. In the first three examples we have shown that our abstract notion of bisimulation corresponded to the usual one on process terms. In the latter we have exploited the monoidal closed structure of **Cocont** to provide a denotational semantics to a higher order process language. Beside the congruence results, a highlight was a *Substitution Lemma* (Lemma 8.4.14) proving that the application in the model corresponded to substitution in the language. For a fragment of the language corresponding to a form of λ-calculus, open map bisimulation was shown to coincide with applicative bisimulation [2]. What we did not succeed in doing was giving an operational characterisation of the bisimulation induced by open maps for the full language and this will be a subject of future work (see below).

9.2 Further research

There are several lines for future research, extending the work presented here. We briefly outline here some of the possibilities as well as some connections with related research.

9.2.1 Higher dimensional transition systems (hdts)

In a joint work with Vladimiro Sassone [23] we introduced a new category of models for concurrency building on previous intuitions of Pratt [112] and van Glabbeek [40]. Open maps were used to give an abstract characterisation of the notion of bisimulation that we had devised. The step further that we are making [24] is that of harnessing the machinery presented in Chapter 3 of this thesis for the purpose of the semantics of concurrent processes as hdts. In this way we expect to obtain automatically congruence results for bisimulation. Hdts seem also appropriate for modelling so-called coordination languages (LINDA-like) [19, 27].

More speculatively, we expect to obtain "geometric realisation functors" [76] that should help clarifying the relationship between our approach and the closely related work of Goubault and others on higher dimensional automata [44, 43, 45, 29].

9.2.2 Higher order process languages

As we saw in Chapter 8, the monoidal closed structure (or even the cartesian closed one obtained by means of an exponential, !) of **Prof** can be used to describe a path category suitable for higher order process languages. Avoiding the use of the exponential we have designed a *linear* process language to test our model with. One of our hopes was that of being able to operationally characterise the bisimulation relation induced on terms by open maps. This requires the capability of expressing the structure of the derivation trees in the operational semantics in a way that allows the reading off the functorial aspects of presheaves (cf. Section 8.4.5). As we wrote we do not have a definite answer to this problem yet, though there are some promising conjectures. It is also worth noticing that the linearity constraint of our language is consistent with a similar one in Cardelli and Gordon's Calculus of Ambients [20] as noted by Winskel [139].

Another interesting problem is that of using the monoidal closed structure of **Prof** to combine higher order features with non-interleaving models. First steps have been made by Hildebrandt, Panangaden and Winskel in [54] where a profunctor based model of non-deterministic dataflow networks is developed.

Modelling higher order process calculi that include name passing features might require a more "advanced" use of enriched categories [65]. We say more about it in Section 9.2.6 below.

9.2.3 A metalanguage for process constructors

Recently Glynn Winskel [140] has begun developing a metalanguage for process constructors, e.g., parallel composition, analogous to those used in domain theory for continuous functions (see [137]). Any functor described by the metalanguage will be connected colimit preserving and hence open map bisimulation preserving. Since ω-chains are connected colimits, any endofunctor described by the metalanguage will have a "least" fixed point. With respect to the problem of characterising operationally open map bisimulation for higher order processes, one hope is that the structure of the meta-language can be used to provide a direct reading of the operational characterisation of the defined functors. The metalanguage should be useful in proving once and for all characterisation results like those of Proposition 5.1.14, Theorem 7.4.7 and Theorem 8.4.10, for parallel composition which have a lot in common.

9.2.4 Weak bisimulation and hiding

In this thesis we concentrated on strong bisimulation. In the conclusions of [64], it is suggested that weak bisimulation could be reduced to strong bisimulation via a monad, in a way that imitates Milner's approach [82] that we recall below. In [92] Nielsen and Cheng have tackled the question for transition systems by allowing more morphisms (the "weak" simulations) while keeping those of the path category (finite strings with the prefix ordering) fixed. Seeking a structured approach to this, Fiore (private communication) has reduced it, in the presheaf setting, to consider "quotienting" functors,

$$q : \mathbb{P} \to \mathbb{Q} \ ,$$

and, using the category of elements construction, "quotient preserving" morphisms. A monad (W_q, η, μ), depending on q, is definable on $\widehat{\mathbb{P}}$ for general reasons. In the case of transition systems, it is known from Milner [82] that it is possible to "reduce" weak bisimulation to strong bisimulation. By this we mean that given two transition systems, T_1 and T_2 on a set of labels, $L \cup \{\tau\}$, one can transform them into, say T_1' and T_2', in such a way that T_1 is weakly bisimilar to T_2 if and only if T_1' is strongly bisimilar to T_2'. In fact this construction is functorial in the category of transition systems [142]. Composing with the embedding from synchronisation trees to transition systems and the unfolding functor from transition systems to synchronisation trees yields an endofunctor on synchronisation trees, that is in fact the underlying functor of a monad. It turns out that this corresponds to the endofunctor

$$W_q : (\widehat{L \cup \{\tau\}})^+ \to (\widehat{L \cup \{\tau\}})^+$$

for $q : (L \cup \{\tau\})^+ \to L^*$ the functor that removes all the occurrences of the letter τ from any string. This extends to give a general treatment of weak bisimulation. We are pursuing this line of research in collaboration with Fiore and Winskel. Points to be addressed include:

- When obvious quotienting functors for categories of models exist, as above, we should explore the induced notion of weak bisimulation. This is particularly interesting when done for models for which there is no clear cut understanding of what weak bisimulation should mean, e.g., event structures and timed transition systems.
- Find abstract considerations to determine how to define quotienting functors, q, when the models are defined as initial solution of recursively defined equations. For instance, in all the examples treated in this thesis, path categories were obtained as solution to equations that looked like the following

  $$\mathbb{P} = \mathbb{P}_\perp + F(\mathbb{P}) \ ,$$

  where $\mathbb{P}_\perp$ stand for the possibility of observing an internal communication and then proceeds while $F(\mathbb{P})$ was, in general, a more complicated expression having to do with situations involving inputs and outputs. If $b : F(\mathbb{B}) \mathrel{-\!\!\mapsto} \mathbb{B}$ is an initial solution to the equation that does not allow observing silent actions, i.e., $\mathbb{P} = F(\mathbb{P})$, then one can define an algebra

  $$\mathbb{B}_\perp + F(\mathbb{B}) \mathrel{-\!\!\mapsto} \mathbb{B}$$

  by copairing b with the profunctor that sends $\perp$ to the initial presheaf and any other object to the corresponding representable (we used to call this j in previous chapters). If $a : \mathbb{A}_\perp + F(\mathbb{A}) \multimap \mathbb{A}$ is an initial solution for the bigger equation, by initiality there exists a universal profunctor

$$q : \mathbb{A} \multimap \mathbb{B}$$

such that the following square of profunctors commute (up to natural isomorphism, of course):

$$\begin{tikzcd}
\mathbb{A}_\perp + F(\mathbb{A}) \arrow[r, "a"] \arrow[d, "q_\perp + F(q)"'] & \mathbb{A} \arrow[d, "q"] \\
\mathbb{B}_\perp + F(\mathbb{B}) \arrow[r, "{[j,b]}"'] & \mathbb{B} \, .
\end{tikzcd}$$

We believe that we can use this fact, to deduce suitable quotienting functors q.

9.2.5 Action calculi

We have presented a categorical formalism for the semantics of process calculi in order to capture abstractly the notion of bisimulation. In *action calculi* [86] one instead looks for the basic ingredients needed to define classes of communicating systems (actions) that can be composed with each other. A reduction semantics is given for the actions. In order to reason about the behaviour of different actions one seeks a way of extracting transition systems out of the reduction semantics. Recently Sewell [123] has made progresses in this direction by classifying reduction semantics in terms of their properties as rewriting systems. We hope instead to be able to employ the open maps paradigm by finding ways of deriving path categories out of action calculi and vice versa.

9.2.6 Beyond presheaves

Presheaf categories are obtained by freely completing *small* categories with all colimits of *small* diagrams. This results (always but for one trivial case) in a non-small category. Often in process languages one naturally constrains the size of the model since there is often little reason to consider processes that cannot correspond to physical machines. For these languages the presheaf construction is overgenerous. Recently we have investigated in a joint work with John Power and Glynn Winskel [22] more restricted forms of completions that, while keeping the spirit and especially the possibility of abstract congruence results of the presheaf approach, can be represented as endofunctor of **Cat**. These constructions has been characterised axiomatically in terms of KZ-monads [68]. The categories of non-deterministic domains, analogous to **Cocont** are then the Kleisli categories of the considered KZ-monads. Just like the Yoneda embedding with respect to presheaves, the unit of the monads provide a full embedding of a category into its "completion" and hence a canonical choice of path category. Open map bisimulation is preserved by the arrows of the Kleisli, i.e., the algebra maps between the free algebras of the monad. A recent observation, that needs to be fully checked, suggests the possibility of characterising (bounded in size) event structures over a set of labels L as

$T(\mathbf{Pom}_L)$ for one such KZ-monad, T.[1] Moreover this same T, when applied to partial order categories, like L^+, returns exactly the (bounded in size) free completion of the category under all colimits of the specified size. This would mean that by means of such a 2-monad T both interleaving and non-interleaving models can be characterised by the same kind of completion.

More speculatively we envisage the possibility of moving from **Prof**, (the bicategorical equivalent of **Cocont**) to $\mathcal{V}$-**Prof**, for suitably complete and cocomplete $\mathcal{V}$'s [65]. For instance, we have noticed (see Chapter 7) that in our present setting it seems highly unlikely that it will be possible to provide semantics to process languages that combine higher order features with name passing, e.g., the Higher order π-calculus [116]. By looking for a model in $\mathbf{Set}^{\mathcal{I}}$-**Prof** rather than $\mathbf{Prof}^{\mathcal{I}}$, we overcome the mathematical difficulties due to the lack of general function spaces in $\mathbf{Prof}^{\mathcal{I}}$. It is left to verify that equations solved in $\mathbf{Set}^{\mathcal{I}}$-**Prof** give meaningful solutions! Alternatively one might consider enriching over categories of sets with probability distributions on their elements to be able to cope with probabilistic/Markov processes [70, 16].

Following suggestions of Martin Hyland, we are also currently investigating the possibility of considering extensions of the category of name sets, $\mathcal{I}$. This to allow the possibility of having meaningful solutions for equations involving both higher-order and name passing features already in $\mathbf{Cat}^{\mathcal{I}}$.

At the same speculative level we should also mention the possibility, suggested in [64], of incorporating *fairness* constraints in the models by moving from presheaves to sheaves.

Finally, it is not clear to us how our approach relates to the abstract understanding of bisimulation provided by coalgebras [4, 114]. The hope is that the recent work and ongoing research of Turi and Plotkin [131, 132] will help provide the missing links.

[1] Recall that $\mathbf{Pom}_L$ is the category of (finite) pomsets over L.

Appendix A

Basic Definitions of Enriched Category Theory

A.1 Enriched categories

In this appendix we review some concepts from enriched category theory that we needed from Chapter 4. In particular we will concentrate on 2-categories.

Definition A.1.1 (Monoidal categories) *A* monoidal category $\mathcal{V}$ *is a 6-tuple*

$$(\mathcal{V}_0, \otimes, I, a, l, r)$$

where $\mathcal{V}_0$ is a category, $\otimes : \mathcal{V}_0 \times \mathcal{V}_0 \to \mathcal{V}_0$ is a functor, $I \in |\mathcal{V}_0|$ is an object, a, l and r are families of natural isomorphisms, $a_{v,w,x} : (v \otimes w) \otimes x \to v \otimes (w \otimes x)$, $l_v : I \otimes v \to v$, $r_v : v \otimes I \to v$ subject to the coherence axioms:

$$\begin{tikzcd}
((v \otimes w) \otimes x) \otimes y \arrow[r, "a_{v\otimes w,x,y}"] \arrow[d, "a_{v,w,x}\otimes 1"'] & (v \otimes w) \otimes (x \otimes y) \arrow[r, "a_{v,w,x\otimes y}"] & v \otimes (w \otimes (x \otimes y)) \\
(v \otimes (w \otimes x)) \otimes y \arrow[rr, "a_{v,w\otimes x,y}"] & & v \otimes ((w \otimes x) \otimes y) \arrow[u, "1\otimes a_{w,x,y}"']
\end{tikzcd}$$

$$\begin{tikzcd}
(v \otimes I) \otimes w \arrow[rr, "a_{v,I,w}"] \arrow[dr, "r_v\otimes 1"'] & & v \otimes (I \otimes w) \arrow[dl, "1\otimes l_w"] \\
& v \otimes w\ . &
\end{tikzcd}$$

A monoidal category is symmetric *if in addition it has a family of natural isomorphisms:*

$$\sigma_{v,w} : v \otimes w \xrightarrow{\simeq} w \otimes v$$

such that $\sigma_{w,v}\sigma_{v,w} = 1_{v\otimes w}$ and satisfying two coherence axioms with respect to associativity and identity (cf. [65], P. 29).

It is plenty of examples of symmetric monoidal categories, since any category with finite products is one. An interesting example of a symmetric monoidal structure not given by the product, related to our thesis by the analogy of Section 4.2.1 is given by **Rel**, where the cartesian product of sets does not correspond to the categorical product that is given by the disjoint union.

Definition A.1.2 ($\mathcal{V}$-categories) *Let $\mathcal{V} = (\mathcal{V}_0, \otimes, I, a, l, r)$ be a monoidal category. A $\mathcal{V}$-category, $\mathcal{C}$ consists of*

- *a class of objects $|\mathcal{C}|$*
- *for any two objects $C, D \in |\mathcal{C}|$, a* hom-set *object of $\mathcal{V}$, $\mathcal{C}(C, D)$*
- *for any object $C \in |\mathcal{C}|$ an arrow in $\mathcal{V}$, $j_C : I \to \mathcal{C}(C, C)$*
- *for any three objects $C, D, E \in |\mathcal{C}|$ a* composition law*, i.e., an arrow in $\mathcal{V}$,*

$$c_{C,D,E} : \mathcal{C}(D, E) \otimes \mathcal{C}(C, D) \to \mathcal{C}(C, E)$$

subject to the coherence laws (omitting some indices) given by commutativity of the following diagrams:

$$\begin{tikzcd}
(\mathcal{C}(E,F) \otimes \mathcal{C}(D,E)) \otimes \mathcal{C}(C,D) \arrow[rr, "a"] \arrow[d, "c_{D,E,F}\otimes 1"'] & & \mathcal{C}(E,F) \otimes (\mathcal{C}(D,E) \otimes \mathcal{C}(C,D)) \arrow[d, "1\otimes c_{C,D,E}"] \\
\mathcal{C}(D,F) \otimes \mathcal{C}(C,D) \arrow[dr, "c_{C,D,F}"] & & \mathcal{C}(E,F) \otimes \mathcal{C}(C,E) \arrow[dl, "c_{C,E,F}"'] \\
& \mathcal{C}(C,F) &
\end{tikzcd}$$

$$\begin{tikzcd}
\mathcal{C}(D,D) \otimes \mathcal{C}(C,D) \arrow[r, "c_{C,D,D}"] & \mathcal{C}(C,D) & \mathcal{C}(C,D) \otimes \mathcal{C}(C,C) \arrow[l, "c_{C,C,D}"'] \\
I \otimes \mathcal{C}(C,D) \arrow[u, "j_D\otimes 1"] \arrow[ur, "l"'] & & \mathcal{C}(C,D) \otimes I \arrow[u, "1\otimes j_C"'] \arrow[ul, "r"]
\end{tikzcd}$$

Example A.1.3
- *Any locally small category is a* **Set***-category where the monoidal structure for* **Set** *is given by the cartesian product.*
- *Any cartesian closed category $\mathcal{C}$ enriches over itself by taking as hom-set objects $\mathcal{C}(C, D)$ the exponential (D^C). In particular* **Cat** *is a* **Cat***-category.*
- *The category* **Rel** *of sets an relations is a* **Poset***-category where* **Poset** *is the category of partial ordered sets and monotone functions. Again the monoidal structure is given by the product.*

Definition A.1.4 ($\mathcal{V}$-functors) *Given two $\mathcal{V}$-categories $\mathcal{C}$ and $\mathcal{D}$ a $\mathcal{V}$-functor*

$$F : \mathcal{C} \to \mathcal{D}$$

is given by the following data:

- *A function $F : |\mathcal{C}| \to |\mathcal{D}|$*

- *For any pair of objects $C, D \in |\mathcal{C}|$ an arrow in $\mathcal{V}$, $F_{C,D} : \mathcal{C}(C,D) \to \mathcal{D}(FC,FD)$*

such that the following diagrams in $\mathcal{V}$ commute:

$$\begin{tikzcd}
\mathcal{C}(D,E) \otimes \mathcal{C}(C,D) \arrow[r, "c_{C,D,E}"] \arrow[d, "F_{D,E} \otimes F_{C,D}"'] & \mathcal{C}(C,E) \arrow[d, "F_{C,E}"] \\
\mathcal{D}(FD,FE) \otimes \mathcal{D}(FC,FD) \arrow[r, "c_{FC,FD,FE}"'] & \mathcal{D}(FC,FE)
\end{tikzcd}$$

$$\begin{tikzcd}
& \mathcal{C}(C,C) \arrow[dd, "F_{C,C}"] \\
I \arrow[ur, "j_C"] \arrow[dr, "j_{FC}"'] & \\
& \mathcal{D}(FC,FC)\ .
\end{tikzcd}$$

Definition A.1.5 ($\mathcal{V}$-natural transformations) *If $F, G : \mathcal{C} \to \mathcal{D}$ are two $\mathcal{V}$-functors a* $\mathcal{V}$-natural transformation *$\alpha : F \stackrel{\cdot}{\longrightarrow} G$ consists of a family of arrows indexed by the objects of $\mathcal{C}$*

$$\alpha_C : I \to \mathcal{D}(FC, GC)$$

such that the following diagram commutes

$$\begin{tikzcd}
& I \otimes \mathcal{C}(C,D) \arrow[r, "\alpha_D \otimes F_{C,D}"] & \mathcal{D}(FD,GD) \otimes \mathcal{D}(FC,FD) \arrow[dr, "c_{FC,FD,GD}"] & \\
\mathcal{C}(C,D) \arrow[ur, "l^{-1}"] \arrow[dr, "r^{-1}"'] & & & \mathcal{D}(FC,GD) \\
& \mathcal{C}(C,D) \otimes I \arrow[r, "G_{C,D} \otimes \alpha_C"'] & \mathcal{D}(GC,GD) \otimes \mathcal{D}(FC,GC)\ . \arrow[ur, "c_{FC,GC,GD}"'] &
\end{tikzcd}$$

A.2 2-Categories

A very special class of $\mathcal{V}$-categories is given by **Cat**-categories, or, ignoring size problems **CAT**-categories. These go under the name of 2-categories. To fix some terminology we reformulate the definition of what a 2-category is, referring to the general definition of $\mathcal{V}$-categories for the coherence axioms.

Definition A.2.1 (2-Categories) *A* 2-category*, $\mathcal{K}$ is given by*

- *a collection of objects, $|\mathcal{K}|$,*
- *for any two objects $K, L \in |\mathcal{K}|$ a category $\mathcal{K}(K,L)$,*
- *for any object $K \in |\mathcal{K}|$ an* identity arrow*, $1_K \in |\mathcal{K}(K,K)|$*

- *for any three objects a composition law (a functor)*

$$c_{K,L,M} : \mathcal{K}(L, M) \times \mathcal{K}(K, L) \to \mathcal{K}(K, M) \ ,$$

satisfying the instantiation of the coherence diagrams of Definition A.1.2.

The objects of $\mathcal{K}(K, L)$ are called arrows and indicated as $f : K \to L$, while the arrows of $\mathcal{K}(K, L)$ are called 2-cells and indicated with the double arrow notation,

$$\alpha : f \Longrightarrow g \ .$$

Given two 2-cells, $\begin{tikzcd} K \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Downarrow\alpha"] & L \end{tikzcd}$ and $\begin{tikzcd} L \arrow[r, bend left, "f'"] \arrow[r, bend right, "g'"'] \arrow[r, phantom, "\Downarrow\beta"] & M \end{tikzcd}$, we write $\beta\alpha$ for $c_{K,L,M}(\beta, \alpha)$ and talk of the *horizontal* composition of α and β. The composition of 2-cells as arrows of, say $\mathcal{K}(K, L)$, is called *vertical* and will be written by interposing a $\cdot$ between the two 2-cells, e.g., $\alpha' \cdot \alpha$, for $\alpha' : g \Rightarrow h$. Given a 2-cell $\begin{tikzcd} K \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Downarrow\alpha"] & L \end{tikzcd}$ and an arrow $f' : L \to M$ we write $f'\alpha$ for $1_{f'}\alpha$; similarly if $f'' : M \to K$. This operation is often called the *whiskering* of α with f'.

As a consequence of the definition of 2-category we have the following property.

Proposition A.2.2 (Interchange Law) *Let $\mathcal{K}$ be a 2-category and let* $\begin{tikzcd} K \arrow[r, bend left, "f"] \arrow[r, bend right, "g"'] \arrow[r, phantom, "\Downarrow\alpha"] & L \end{tikzcd}$ *,* $\begin{tikzcd} K \arrow[r, bend left, "g"] \arrow[r, bend right, "h"'] \arrow[r, phantom, "\Downarrow\beta"] & L \end{tikzcd}$ *,* $\begin{tikzcd} L \arrow[r, bend left, "f'"] \arrow[r, bend right, "g'"'] \arrow[r, phantom, "\Downarrow\varphi"] & M \end{tikzcd}$ *and* $\begin{tikzcd} L \arrow[r, bend left, "g'"] \arrow[r, bend right, "h'"'] \arrow[r, phantom, "\Downarrow\psi"] & M \end{tikzcd}$ *be four 2-cells, then*

$$(\psi \cdot \varphi)(\beta \cdot \alpha) = (\psi\beta) \cdot (\varphi\alpha) \ .$$

This means that if one writes a diagram like

$$\begin{tikzcd} K \arrow[r, "g" description] \arrow[r, bend left=60, "f"] \arrow[r, bend right=60, "h"'] \arrow[r, phantom, shift left=3, "\Downarrow\alpha"] \arrow[r, phantom, shift right=3, "\Downarrow\beta"] & L \arrow[r, "g'" description] \arrow[r, bend left=60, "f'"] \arrow[r, bend right=60, "h'"'] \arrow[r, phantom, shift left=3, "\Downarrow\varphi"] \arrow[r, phantom, shift right=3, "\Downarrow\psi"] & M \end{tikzcd}$$

it defines a 2-cell from $f'f$ to $h'h$ in a unique way. A diagram like this one above is called a *pasting* diagram. The interchange law is the basis for a more general coherence result which asserts that any pasting diagram defines uniquely a 2-cell [67, 108].

Concerning applications of the notion of pasting we can give as an example the definition of an adjoint pair in a 2-category. This will specialise in the case of **Cat** to the usual notion of adjoint pair of functors.

Definition A.2.3 (Adjoints in 2-categories) *If $\mathcal{K}$ is a 2-category, a pair of arrows* $\begin{tikzcd} K \arrow[r, bend left, "f"] & L \arrow[l, bend left, "g"] \end{tikzcd}$ *is an adjoint pair, $f \dashv g$, if there exist 2-cells*

$$\eta : 1_K \Longrightarrow gf \qquad \text{and} \qquad \varepsilon : fg \Longrightarrow 1_L$$

such that the following equalities hold:

$$\begin{tikzcd}
K \arrow[rr, "1"] \arrow[dr, "f"'] & \arrow[d, phantom, "\Downarrow\eta"] & K \arrow[dr, "f"] & \\
& L \arrow[ur, "g"] \arrow[rr, "1"'] & \arrow[u, phantom, "\Downarrow\varepsilon"] & L
\end{tikzcd}
=
\begin{tikzcd}
K \arrow[d, bend left, "f"] \arrow[d, bend right, "f"'] \arrow[d, phantom, "\Swarrow 1"] \\
L
\end{tikzcd}$$

$$\begin{tikzcd}
& K \arrow[rr, "1"] \arrow[dr, "f"] \arrow[d, phantom, "\Downarrow\varepsilon"] & \arrow[d, phantom, "\Downarrow\eta"] & K \\
L \arrow[ur, "g"] \arrow[rr, "1"'] & \, & L \arrow[ur, "g"'] &
\end{tikzcd}
=
\begin{tikzcd}
& K \\
L \arrow[ur, bend left, "g"] \arrow[ur, bend right, "g"'] \arrow[ur, phantom, "\Searrow 1"] &
\end{tikzcd} .$$

As we said 2-categories are special kind of enriched categories. The instantiation of the definitions of $\mathcal{V}$-functor and $\mathcal{V}$-natural transformation yields the notion of 2-functor and 2-natural transformation. Still among 2-categories other more general classes of functors and natural transformations can be considered and the practice shows that often this more general notions are the one that naturally appears. We have already seen in Section 1.4.1 the notion of a pseudo-functor from a category $\mathcal{B}$ to **CAT**. We now make this concept precise by defining pseudo-functors between 2-categories.

Definition A.2.4 (Pseudo-functors) *Let $\mathcal{K}$ and $\mathcal{L}$ be two 2-categories. A* pseudo-functor *$F : \mathcal{K} \to \mathcal{L}$ is given by*

- *a function $F : |\mathcal{K}| \to |\mathcal{L}|$*
- *for any two objects $K, L \in |\mathcal{K}|$ a functor, $F_{K,L} : \mathcal{K}(K, L) \to \mathcal{L}(FK, FL)$*
- *for any object $K \in |\mathcal{K}|$ an isomorphic 2-cell, $\varphi_K : 1_{F(K)} \overset{\sim}{\Longrightarrow} F_{K,K}(1_K)$*
- *for any triple of objects of $\mathcal{K}$, K, L, M, a natural isomorphism*

$$\varphi_{K,L,M} : c_{FK,FL,FM} \circ (F_{L,M} \times F_{K,L}) \overset{\cdot}{\longrightarrow} F_{K,M} \circ c_{K,L,M}$$

$$\begin{tikzcd}
\mathcal{K}(L,M) \times \mathcal{K}(K,L) \arrow[rr, "c_{K,L,M}"] \arrow[d, "(F_{L,M} \times F_{K,L})"'] & \arrow[d, phantom, "\overset{\varphi_{K,L,M}}{\cong}"] & \mathcal{K}(K,M) \arrow[d, "F_{K,M}"] \\
\mathcal{L}(FL,FM) \times \mathcal{L}(FK,FL) \arrow[rr, "c_{FK,FL,FM}"'] & \, & \mathcal{L}(FK,FM)
\end{tikzcd} .$$

satisfying the coherence conditions given by commutativity of the following diagrams for any triple of arrows of $\mathcal{K}$

$$K \xrightarrow{f} L \xrightarrow{g} M \xrightarrow{h} N \;\; :$$

$$\begin{tikzcd}
F(f)F(1_K) \arrow[dr, Rightarrow, "\varphi_{1_K,f}"'] & Ff \arrow[l, Rightarrow, "F(f)\varphi_K"'] \arrow[r, Rightarrow, "\varphi_L F(f)"] \arrow[d, Rightarrow, "1_{Ff}"] & F(1_L)F(f) \arrow[dl, Rightarrow, "\varphi_{f,1_L}"] \\
& F(f) &
\end{tikzcd}$$

$$\begin{tikzcd}
FhFgFf \arrow[r, Rightarrow, "\varphi_{g,h}Ff"] \arrow[d, Rightarrow, "Fh\varphi_{f,g}"'] & F(hg)Ff \arrow[d, Rightarrow, "\varphi_{f,hg}"] \\
FhF(gf) \arrow[r, Rightarrow, "\varphi_{gf,h}"'] & F(hgf)
\end{tikzcd} ,$$

where for sake of readability we wrote $\varphi_{f,g}$ *instead than* $(\varphi_{K,L,M})_{f,g}$ *and similarly for the other occurrences of* φ. *If instead of isomorphic* 2*-cells,* φ_K*, one has equality of arrows and instead of the natural isomorphisms,* $\varphi_{K,L,M}$ *one has commutativity of the square*

$$\begin{tikzcd}
\mathcal{K}(L,M) \times \mathcal{K}(K,L) \arrow[r, "c_{K,L,M}"] \arrow[d, "(F_{L,M}\times F_{K,L})"'] & \mathcal{K}(K,M) \arrow[d, "F_{K,M}"] \\
\mathcal{L}(FL,FM) \times \mathcal{L}(FK,FL) \arrow[r, "c_{FK,FL,FM}"'] & \mathcal{L}(FK,FM)
\end{tikzcd} ,$$

F *is said to be a* 2-functor. *In this case, the coherence conditions are trivially satisfied.*

Indexed categories as defined in Definition 1.4.8 are examples of pseudo functors, since any category, $\mathcal{C}$, can be regarded as a 2-category whose hom-categories, $\mathcal{C}(A,B)$ are discrete categories.

When considering 2-categories different notions of limits arise as generalisations of the usual one (cf. [65] or the first section of [14] for a quick review). We will concentrate on what we call (consistently with our terminology) *pseudo-limits*. Note that elsewhere (e.g., [127]) these are called bilimits (where *bi* stands for bicategorical) and the prefix pseudo is reserved for a stricter class of bilimits.

We shall actually be interested only in particular kinds of pseudo-limits of which we shall give explicit definitions when needed. We have already given in Section 1.5 the definition of *pseudo-initial* object as an illustrative example of the change of perspective that occur in moving form categories to 2-categories. We repeat it here below:

Definition A.2.5 (Pseudo-initial object) *An object* 0 *of a* 2*-category* $\mathcal{K}$ *is* pseudo-initial *if for every object* K*,* $\mathcal{K}(0,K)$ *is equivalent to the category* $\mathbf{1}$ *with only one object and one morphism. In other words* 0 *is pseudo-initial if for every object* K*, there exists an arrow* $0_K : 0 \to K$ *and for every pair of arrows* $f, g : 0 \to K$ *there exists a unique* 2*-cell,* $\alpha : f \Longrightarrow g$*.*

We have omitted the definition of pseudo-natural transformations (strong transformations in [127]) and modifications. They are needed in defining pseudo-limits in general but we decided to give an explicit description of the data required in our particular examples rather than giving the general definitions here.

A.3 Bicategories

We conclude this appendix by recalling the existence of the notion of bicategory. Roughly speaking a bicategory is a 2-category where the horizontal composition is associative only *up to isomorphism*. This means that the diagrams of Definition A.1.2, when specialised to $\mathcal{V} = \mathbf{CAT}$ will now commute only up to natural isomorphisms and that these isomorphism satisfy some coherence conditions, too. We refer to [9] for the precise definition of what a bicategory is. The definition of pseudo-functors, pseudo-natural transformations and modifications lifts from 2-categories to bicategories [127]. In this setting, pseudo-functors are often called *homomorphisms* [9, 127], while pseudo-natural transformations go under the name of *strong transformations* [127].

In our practice bicategories seem to arise more naturally than 2-categories. They are often more difficult to work with because of the extra coherence conditions that one has to carry along. Fortunately there are coherence results that permit us to "strictify" a bicategory into a 2-category without losing its relevant properties [107, 106, 109].

Appendix B

Some proofs for Chapter 6

B.1 Theorem 6.4.1

> Let $\mathcal{K}$ and $\mathcal{L}$ be two pseudo ω**Cat**-algebraically complete 2-categories and
>
> $$T : \mathcal{K} \times \mathcal{L} \to \mathcal{L}$$
>
> be a pseudo-ω**Cat**-functor. For any $A \in |\mathcal{K}|$, write μT_A for the object part of a chosen pseudo-initial algebra for the pseudo endofunctor, $T_A : \mathcal{L} \to \mathcal{L}$, defined by freezing the first component to always be, the object A, or 1_A or 1_{1_A}. Then the mapping $A \to \mu T_A$ extends canonically to a pseudo ω**Cat**-functor, $\mu T_{(-)} : \mathcal{K} \to \mathcal{L}$.

To prove the theorem we first need a couple of lemmas as well as a brief excursion into the theory of lax algebra morphisms [15, 107] as was also done in [30, Chapter 7] to prove a similar result for **Cpo**-categories.

Definition B.1.1 *If $T : \mathcal{K} \to \mathcal{K}$ is a pseudo functor and $f : TA \to A$ and $g : TB \to B$ are two algebras for T, a* lax morphism *from f to g is given by a pair (h, H), with $h : A \to B$ and $H : hf \Rightarrow gTh$. If H is a pseudo cell, we say that (h, H) is a* pseudo morphism*.*

Define Lax-T-alg *to be the 2-category of T-algebras, lax morphisms and 2-cells given by $\alpha : (h, H) \Rightarrow (k, K)$ where α is a 2-cell of $\mathcal{K}$, $\alpha : h \Rightarrow k$ with*

$$(gT\alpha) \cdot H = K \cdot (\alpha f) \ .$$

Lemma B.1.2 *Let $\mathcal{K}$ be an ω**Cat**-category, $T : \mathcal{K} \to \mathcal{K}$ a pseudo ω**Cat**-functor and $a : TA \to A$ a pseudo initial algebra for T. Let $x : TX \to X$ be another algebra and $(\mathrm{it}(x), \iota(x)) : a \to x$ be a pseudo morphism given by the universal property of $a : TA \to A$. Then $(\mathrm{it}(x), \iota(x))$ is terminal in* Lax-T-alg(a, x).

Proof: By the pseudo Lambek lemma we know that a is an equivalence, i.e., there exists $\overline{a}$, $\sigma : 1_A \cong a\overline{a}$ and $\tau : \overline{a}a \cong 1_{TA}$. Moreover σ and τ can be chosen as to be unit and counit for the adjunction $\overline{a} \dashv a$ [15]. Let $(h, H) : a \to x$ be a lax-T-algebra morphism. Consider the chain in $\mathcal{K}(A, X)$ given by

$$h \xRightarrow{(H\overline{a})\cdot(h\sigma)} x(Th)\overline{a} \xRightarrow{xT((H\overline{a})\cdot(h\sigma))\overline{a}} xT(x(Th)\overline{a})\overline{a} \Longrightarrow \cdots \ . \tag{B.1}$$

Formally the chain is inductively defined by

Objects: $U_0 = h$, $U_{n+1} = x(TU_n)\overline{a}$

Arrows: $f_0 = (H\overline{a}) \cdot (h\sigma)$, $f_{n+1} = x(Tf_n)\overline{a}$

Let $(U, \varphi_n : U_n \Rightarrow U)$ be a colimiting cone for the chain. First of all we show that

$$U \cong \mathrm{it}(x) \ .$$

Since a is pseudo initial, it is enough to show that there exists a pseudo cell

$$u : Ua \cong xTU \ .$$

Consider the following calculation

$$\begin{array}{rcll} Ua & \cong & (\mathrm{colim}\ _{n\geq 0} f_n)a & \\ & \cong & (\mathrm{colim}\ _{n\geq 0} f_{n+1}a) & \\ & \cong & (\mathrm{colim}\ _{n\geq 0} xTf_n\overline{a}a) & \text{(by definition of } f_{n+1}) \\ & \cong & (\mathrm{colim}\ _{n\geq 0} xTf_n) & \text{(since } \overline{a}a \cong 1_{TA}) \\ & \cong & x(\mathrm{colim}\ _{n\geq 0} Tf_n) & \\ & \cong & xT(U) \ . & \end{array}$$

This suggests that we can derive the isomorphism u by inspecting the universal 2-cells between two different colimiting cones for the same ω-chain. Consider in fact the chain

$$x(TU_0)\overline{a}a \xRightarrow{x(Tf_0)\overline{a}a} x(TU_1)\overline{a}a \xRightarrow{x(Tf_1)\overline{a}a} x(TU_2)\overline{a}a \Longrightarrow \cdots \ , \tag{B.2}$$

obtained in either of the following two ways:

1. By precomposing the chain B.1 with a and dropping the first item of it.
2. By applying T to the chain B.1 and postcomposing the resulting chain with x and precomposing it with $\overline{a}a$.

Thus, the chain B.2 above has colimiting cones given by

$$(Ua, \varphi_{n+1}a : x(TU_n)\overline{a}a = U_{n+1}a \Rightarrow Ua)$$

and by

$$(xTU, x(T\varphi_n)\tau : x(TU_n)\overline{a}a \Rightarrow xTU) \ .$$

Hence there exists a unique $u : Ua \overset{\sim}{\Longrightarrow} xTU$ such that for every $n \geq 1$,

$$u \cdot (\varphi_n a) = xT\varphi_{n-1}\tau \ . \tag{B.3}$$

Notice that

$$u \cdot (\varphi_0 a) = xT(\varphi_0) \cdot H \tag{B.4}$$

since

$$\begin{array}{rcll} u \cdot (\varphi_0 a) & = & u \cdot (\varphi_1 a) \cdot (f_0 a) & \\ & = & x(T\varphi_0)\tau \cdot (H\overline{a}a) \cdot (h\sigma a) & \text{(by the property (B.3) for } n = 1\text{)} \\ & = & xT(\varphi_0) \cdot H \cdot (ha\tau) \cdot (h\sigma a) & \text{(by the interchange law)} \\ & = & xT(\varphi_0) \cdot H & \text{(by a triangular identity).} \end{array}$$

By the universal property of pseudo initial algebras, there exists a unique (necessarily pseudo) 2-cell, $\varphi : U \Rightarrow \mathrm{it}(x)$ such that

$$\iota(x) \cdot (\varphi a) = (xT\varphi) \cdot u \ . \tag{B.5}$$

Let $\overline{\varphi} : h \Rightarrow \mathrm{it}(x)$ be defined as

$$\overline{\varphi} \stackrel{\text{def}}{=} \varphi \cdot \varphi_0 \ .$$

To conclude that $\overline{\varphi} : (h, H) \Rightarrow (\mathrm{it}(x), \iota(x))$ we need to show that

$$(xT\overline{\varphi}) \cdot H = \iota(x) \cdot (\overline{\varphi}a) \tag{B.6}$$

$$\begin{array}{rcll} (xT\overline{\varphi}) \cdot H & = & (xT(\varphi \cdot \varphi_0)) \cdot H & \text{(by definition of } \overline{\varphi}\text{)} \\ & = & (xT\varphi) \cdot (xT\varphi_0) \cdot H & \text{(since } T \text{ is locally a functor)} \\ & = & (xT\varphi) \cdot u \cdot (\varphi_0 a) & \text{(by the equality (B.4))} \\ & = & \iota(x) \cdot (\varphi a) \cdot (\varphi_0 a) & \text{(by the property (B.5))} \\ & = & \iota(x) \cdot (\overline{\varphi}a) \ . & \end{array}$$

Finally we should show that $\overline{\varphi}$ is the unique 2-cell satisfying equation (B.6). Assume then the existence of $\alpha : (h, H) \Rightarrow (\mathrm{it}(x), \iota(x))$ such that

$$(xT\alpha) \cdot H = \iota(x) \cdot (\alpha a) \ . \tag{B.7}$$

We want to show that $\alpha = \overline{\varphi}$. Observe that α generates a cone $(\mathrm{it}(x), \alpha_n : U_n \Rightarrow \mathrm{it}(x))$ inductively as follows:

$$\alpha_0 \stackrel{\text{def}}{=} \alpha \ , \qquad \alpha_{n+1} \stackrel{\text{def}}{=} (\mathrm{it}(x)\sigma^{-1}) \cdot (\iota(x)^{-1}\overline{a}) \cdot (x(T\alpha_n)\overline{a}) \ .$$

In fact (proof by induction),

$$\begin{array}{rcll} \alpha_1 f_0 & = & (\mathrm{it}(x)\sigma^{-1}) \cdot (\iota(x)^{-1}\overline{a}) \cdot (x(T\alpha)\overline{a}) \cdot (H\overline{a}) \cdot (h\sigma) & \text{(by definition)} \\ & = & (\mathrm{it}(x)\sigma^{-1}) \cdot (\alpha a\overline{a}) \cdot (h\sigma) & \text{(by equation (B.7))} \\ & = & \alpha \cdot (h\sigma^{-1}) \cdot (h\sigma) & \text{(by the interchange law)} \\ & = & \alpha \stackrel{\text{def}}{=} \alpha_0 \ , & \end{array}$$

while for $n \geq 1$,

$$\begin{array}{rcll} \alpha_{n+1} f_n & = & (\mathrm{it}(x)\sigma^{-1}) \cdot (\iota(x)^{-1}\overline{a}) \cdot (xT\alpha_n\overline{a}) \cdot (xTf_{n-1}\overline{a}) & \text{(by definition)} \\ & = & (\mathrm{it}(x)\sigma^{-1}) \cdot (\iota(x)^{-1}\overline{a}) \cdot (xT\alpha_{n-1}\overline{a}) & \text{(by inductive hypothesis)} \\ & = & \alpha_n & \text{(by definition) .} \end{array}$$

Since $(\mathrm{it}(x), \overline{\varphi}_n \stackrel{\mathrm{def}}{=} \varphi \cdot \varphi_n : U_n \Rightarrow \mathrm{it}(x))$ is a colimiting cone there exists a unique $\overline{\alpha} : \mathrm{it}(x) \Rightarrow \mathrm{it}(x)$ such that for every n:

$$\overline{\alpha} \cdot \overline{\varphi}_n = \alpha_n \; . \tag{B.8}$$

Our goal is to show that $\overline{\alpha}$ is equal to $1_{\mathrm{it}(x)}$. In fact if this is the case, the following chain of equalities holds

$$\alpha = \alpha_0 = \overline{\alpha \varphi_0} = \overline{\varphi}_0 = \overline{\varphi}$$

and we are done. To show

$$\overline{\alpha} = 1_{\mathrm{it}(x)}$$

we shall use the universal property of pseudo initial algebras and prove that

$$\iota(x) \cdot (\overline{\alpha} a) = (xT\overline{\alpha}) \cdot \iota(x) \; . \tag{B.9}$$

Since there exists a unique 2-cell satisfying the equation (B.9) above and $1_{\mathrm{it}(x)}$ does so, this will imply that $\overline{\alpha}$ is equal to $1_{\mathrm{it}(x)}$. We deduce that equation (B.9) holds using the universal property of colimiting cones, namely we shall prove that for any n,

$$\iota(x) \cdot (\overline{\alpha} a) \cdot (\overline{\varphi}_n a) = (xT\overline{\alpha}) \cdot \iota(x) \cdot (\overline{\varphi}_n a) \; . \tag{B.10}$$

In fact, since $(\mathrm{it}(x)a, \overline{\varphi}_n a : U_n a \Rightarrow \mathrm{it}(x)a)$ is a colimiting cone, to prove that the equation (B.10) above holds for every n, is the same as proving that (B.9) holds. The proof is again by induction. We will use that

$$a\tau = \sigma^{-1} a \; , \tag{B.11}$$

which follows from the triangular identity $(a\tau) \cdot (\sigma a) = 1_a$, since σ is invertible.

Base Case:

$$\begin{aligned}
\iota(x) \cdot (\overline{\alpha} a) \cdot (\overline{\varphi}_0 a) &= \iota(x) \cdot (\alpha a) && \text{(by (B.8) and definition of } \alpha_0) \\
&= (xT\alpha) \cdot H && \text{(by equation (B.7))} \\
&= (xT\overline{\alpha}) \cdot (xT\overline{\varphi}_0) \cdot H && \text{(by (B.8) and definition of } \alpha_0) \\
&= (xT\overline{\alpha}) \cdot \iota(x) \cdot (\overline{\varphi}_0 a) && \text{(by equation (B.6) and since } \overline{\varphi}_0 \stackrel{\mathrm{def}}{=} \overline{\varphi}).
\end{aligned}$$

Inductive Step:

$$\begin{array}{rcl}
\iota(x)\cdot(\overline{\alpha}a)\cdot(\overline{\varphi}_{n+1}a) & = & \iota(x)\cdot(\alpha_{n+1}a) \\
& & \qquad\text{(by definition)} \\
& = & \iota(x)\cdot(\mathrm{it}(x)\sigma^{-1}a)\cdot(\iota(x)^{-1}\overline{a}a)\cdot(xT\alpha_n\overline{a}a) \\
& & \qquad\text{(by definition)} \\
& = & \iota(x)\cdot(\mathrm{it}(x)a\tau)\cdot(\iota(x)^{-1}\overline{a}a)\cdot(xT\alpha_n\overline{a}a) \\
& & \qquad\text{(by the equality (B.11))} \\
& = & \iota(x)\cdot\iota(x)^{-1}\cdot(xT\alpha_n\tau) \\
& & \qquad\text{(by the interchange law)} \\
& = & xT\alpha_n\tau \\
& = & xT(\overline{\alpha}\cdot\varphi\cdot\varphi_n)\tau \\
& & \qquad\text{(by (B.8) and definition of } \overline{\varphi}_n\text{)} \\
& = & (xT\overline{\alpha})\cdot(xT\varphi)\cdot(x(T\varphi_n)\tau) \\
& & \qquad\text{(by the interchange law)} \\
& = & (xT\overline{\alpha})\cdot(xT\varphi)\cdot u\cdot(\varphi_{n+1}a) \\
& & \qquad\text{(by equation (B.3))} \\
& = & (xT\overline{\alpha})\cdot\iota(x)\cdot(\varphi a)\cdot(\varphi_{n+1}a) \\
& & \qquad\text{(by equation (B.5))} \\
& = & (xT\overline{\alpha})\cdot\iota(x)\cdot(\overline{\varphi}_{n+1}a) \\
& & \qquad\text{(by definition).}
\end{array}$$

□

Lemma B.1.3 *Let $T : \mathcal{K} \to \mathcal{K}$ be a pseudo ω**Cat**-functor. Let $a : TA \to A$ be a pseudo initial algebra for T. If X is any object of $\mathcal{K}$, for every arrow $x : TX \to X$, write $(\mathrm{it}(x), \iota(x))$ for a chosen pseudo algebra morphism that exists since a is pseudo initial. Then the mapping*

$$\begin{array}{ccc} |\mathcal{K}(TX,X)| & \longrightarrow & |\mathcal{K}(A,X)| \\ x & \longmapsto & \mathrm{it}(x)\ , \end{array}$$

extends canonically to a functor, say F, that preserves colimits of ω-chains.

Proof: Let $\alpha : x \Rightarrow y$ be an arrow, i.e., a 2-cell, in $\mathcal{K}(TX, X)$. The pair

$$(\mathrm{it}(x), (\alpha T\mathrm{it}(x))\cdot\iota(x))$$

is a lax T-algebra morphism,

$$(\mathrm{it}(x), (\alpha T\mathrm{it}(x))\cdot\iota(x)) : a \to y\ .$$

By the Lemma B.1.2, $(\mathrm{it}(y), \iota(y))$ is terminal in Lax-T-alg(a, y), hence there exists a unique 2-cell,

$$\alpha_T : \mathrm{it}(x) \Rightarrow \mathrm{it}(y)$$

such that $(yT\alpha_T)\cdot(\alpha T\mathrm{it}(x))\cdot\iota(x)) = \iota(y)\cdot(\alpha_T a)$. Define $F(\alpha) = \alpha_T$. With this definition F is a functor, in fact:

- $$\begin{aligned}((xT1_{\mathrm{it}(x)})\cdot(1_xT\mathrm{it}(x))\cdot\iota(x)) &= \iota(x)\\ &= \iota(x)\cdot(1_{\mathrm{it}(x)}a)\ ,\end{aligned}$$

  hence $F(1_x)=1_{\mathrm{it}(x)}$.
- While if $\alpha : x \Rightarrow y$ and $\beta : y \Rightarrow z$, with

$$(yT\alpha_T)\cdot(\alpha T\mathrm{it}(x))\cdot\iota(x)) = \iota(y)\cdot(\alpha_T a) \tag{B.12}$$

$$(zT\beta_T)\cdot(\beta T\mathrm{it}(y))\cdot\iota(y)) = \iota(z)\cdot(\beta_T a)\ , \tag{B.13}$$

  we have the following

$$\begin{aligned}(zT(\beta_T\alpha_T))\cdot((\beta\alpha)T\mathrm{it}(x))\cdot\iota(x) &= (zT\beta_T)\cdot(zT\alpha_T)\cdot(\beta T\mathrm{it}(x))\cdot(\alpha T\mathrm{it}(x))\cdot\iota(x)\\ &\qquad\text{(since T is locally a functor)}\\ &= (zT\beta_T)\cdot(\beta T\mathrm{it}(y))\cdot(yT\alpha_T)\cdot(\alpha T\mathrm{it}(x))\cdot\iota(x)\\ &\qquad\text{(by the interchange law)}\\ &= (zT\beta_T)\cdot(\beta T\mathrm{it}(y))\cdot\iota(y)\cdot(\alpha_T a)\\ &\qquad\text{(by (B.12))}\\ &= \iota(z)\cdot(\beta_T a)\cdot(\alpha_T a)\\ &\qquad\text{(by (B.13))}\\ &= \iota(z)\cdot((\beta_T\alpha_T)a)\ ,\end{aligned}$$

  hence $F(\beta\alpha)=F(\beta)F(\alpha)$.

We are left with showing that F preserves colimiting cones of ω-chains. Suppose then that a chain

$$x_0 \xRightarrow{\alpha_0} x_1 \xRightarrow{\alpha_1} x_2 \xRightarrow{\alpha_2} x_3 \xRightarrow{\alpha_3} \cdots$$

in $\mathcal{K}(TX,X)$ has a colimiting cone given by $(x,\beta_n : x_n \Rightarrow x)$. We want to show that the cone

$$(\mathrm{it}(x),\beta_{n,T} : \mathrm{it}(x_n) \Rightarrow \mathrm{it}(x))$$

is colimiting for the chain

$$\mathrm{it}(x_0) \xRightarrow{\alpha_{0,T}} \mathrm{it}(x_1) \xRightarrow{\alpha_{1,T}} \mathrm{it}(x_2) \xRightarrow{\alpha_{2,T}} \mathrm{it}(x_3) \xRightarrow{\alpha_{3,T}} \cdots \qquad .$$

Suppose $(f,\varphi_n : \mathrm{it}(x_n) \Rightarrow f)$ is a colimiting cone for the chain above. We shall show first of all that $\mathrm{it}(x)\cong f$. To start observe that for any n the following square commutes by the defining property of $\alpha_{n,T}$:

$$\begin{tikzcd}
\mathrm{it}(x_n)a \arrow[r, Rightarrow, "\alpha_{n,T}a"] \arrow[d, Rightarrow, "\iota(x_n)"'] & \mathrm{it}(x_{n+1})a \arrow[d, Rightarrow, "\iota(x_{n+1})"] \\
x_nT\mathrm{it}(x_n) \arrow[r, Rightarrow, "\alpha_nT\alpha_{n,T}"'] & x_{n+1}T\mathrm{it}(x_{n+1})
\end{tikzcd} \qquad .$$

Hence the chain given by

$$\mathrm{it}(x_n)a \xrightarrow{\alpha_{n,T}a} \mathrm{it}(x_{n+1})a$$

has colimiting cones given both by

$$(xTf, (\beta_n T(\varphi_n)) \cdot \iota(x_n) : \mathrm{it}(x_n)a \Rightarrow xTf)$$

(since T preserves colimits of ω-chains) and by

$$(fa, \varphi_n a : \mathrm{it}(x_n)a \Rightarrow fa) \ .$$

Therefore there exists a pseudo cell $\iota(f) : fa \overset{\sim}{\Rightarrow} xTf$ such that

$$\iota(f) \cdot (\varphi_n a) = (\beta_n T(\varphi_n)) \cdot \iota(x_n) \ . \tag{B.14}$$

Moreover, since $\iota(f)$ is an isomorphism, by the universal property of pseudo initial algebras there exists a unique (pseudo) cell $\varphi : \mathrm{it}(x) \Rightarrow f$, such that

$$\iota(f) \cdot (\varphi a) = (xT\varphi) \cdot \iota(x) \ . \tag{B.15}$$

To conclude that the cone of $\beta_{n,T}$'s is colimiting is enough to show that for any n,

$$\varphi \cdot \beta_{n,T} = \varphi_n \ .$$

To prove that this equality holds we use Lemma B.1.2, i.e., we use the fact that $(f, \iota(f))$ is terminal in Lax-T-alg(a, x). So we show that both φ_n and $\varphi \cdot \beta_{n,T}$ are morphisms in Lax-T-alg(a, x) from $(\mathrm{it}(x_n), (\beta_n T(\mathrm{it}(x_n))) \cdot \iota(x_n))$ to $(f, \iota(f))$ hence they have to be necessarily equal. Thus we have to show that the equalities

$$\iota(f) \cdot (\varphi_n a) = (xT\varphi_n) \cdot (\beta_n T(\mathrm{it}(x_n))) \cdot \iota(x_n) \tag{B.16}$$

and

$$\iota(f) \cdot ((\varphi \cdot \beta_{n,T})a) = (xT(\varphi \cdot \beta_{n,T})) \cdot (\beta_n T(\mathrm{it}(x_n))) \cdot \iota(x_n) \tag{B.17}$$

hold for any n. Concerning equation (B.16) we have the following chain of equalities:

$$\begin{array}{rcll}
\iota(f) \cdot (\varphi_n a) & = & (\beta_n T(\varphi_n)) \cdot \iota(x_n) & \text{(by equation (B.14))} \\
& = & (xT\varphi_n) \cdot (\beta_n T(\mathrm{it}(x_n))) \cdot \iota(x_n) & \text{(by the interchange law).}
\end{array}$$

While for the equation (B.17) we have the following:

$$\begin{array}{rcl}
\iota(f) \cdot ((\varphi \cdot \beta_{n,T})a) & = & \iota(f) \cdot (\varphi a) \cdot (\beta_{n,T} a) \\
& = & (xT\varphi) \cdot \iota(x) \cdot (\beta_{n,T} a) \\
& & \qquad \text{(by equation (B.15))} \\
& = & (xT\varphi) \cdot (xT\beta_{n,T}) \cdot (\beta_n T\mathrm{it}(x_n)) \cdot \iota(x_n) \\
& & \qquad \text{(by the property defining } \beta_{n,T}) \\
& = & (xT(\varphi \cdot \beta_{n,T})) \cdot (\beta_n T\mathrm{it}(x_n)) \cdot \iota(x_n) \ .
\end{array}$$

□

Proof:[of Theorem 6.4.1] The pseudo functor $\mu T_{(-)} : \mathcal{K} \to \mathcal{L}$ is defined using the properties of pseudo initial algebras. In fact, if $a : A \to B$ is an arrow in $\mathcal{K}$, let $i_A : T_A(\mu T_A) \to \mu T_A$ and $i_B : T_B(\mu T_B) \to \mu T_B$ be the chosen pseudo initial algebras. Then

$$T(a, 1_{(\mu T_B)}) : T_A(\mu T_B) = T(A, \mu T_B) \longrightarrow T(B, \mu T_B) = T_B(\mu T_B) \ .$$

Consider the algebra

$$T_A(\mu T_B) \xrightarrow{T(a,1_{(\mu T_B)})} T_B(\mu T_B) \xrightarrow{i_B} \mu T_B \ .$$

By the universal property there exists a pair $(\mu T_a, \mu T_\alpha)$ as in the following square:

$$\begin{tikzcd}
T_A(\mu T_A) \arrow[r, "i_A"] \arrow[d, "T_A(\mu T_a)"'] \arrow[dr, phantom, "\overset{\cong}{\mu T_\alpha}"] & \mu T_A \arrow[d, "\mu T_a"] \\
T_A(\mu T_B) \arrow[r, "i_B T(a,1_{(\mu T_B)})"'] & \mu T_B \ .
\end{tikzcd}$$

Given a choice of the pair $(\mu T_a, \mu T_\alpha)$ (for every $a : A \to B$), the action on 2-cells is canonically determined by Lemma B.1.2 as in Lemma B.1.3. Moreover the coherence isomorphisms are uniquely determined by the universal property of pseudo initial algebras and Lemma B.1.3 ensures that the defined pseudo functor is ω**Cat**. $\square$

Appendix C

Two proofs for Chapter 7

C.1 Lemma 7.5.2 [Substitution Lemma]

Let $i : s \to s'$ be an injective function between finite sets, with $\mathbf{x} = \langle x_1, x_2, \ldots, x_{|s|}\rangle$ the names in s. Then for any process P with free names in s,

$$\mathbb{P}(i)_!(\llbracket P \rrbracket_s) \cong \llbracket P[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} \,.$$

Proof: By induction on the structure of P.

Base Case: $P = 0$, trivial

Inductive Step:

$P = Q + R$**:**
$$\begin{aligned}
\mathbb{P}(i)_!(\llbracket Q + R \rrbracket_s) &= \mathbb{P}(i)_!(\llbracket Q \rrbracket_s + \llbracket R \rrbracket_s) \\
&\qquad\text{(by definition of } \llbracket \cdot \rrbracket_s\text{)} \\
&\cong \mathbb{P}(i)_!(\llbracket Q \rrbracket_s) + \mathbb{P}(i)_!(\llbracket R \rrbracket_s) \\
&\qquad\text{(since } \mathbb{P}(i)_! \text{ is colimit preserving)} \\
&\cong \llbracket Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} + \llbracket R[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} \\
&\qquad\text{(by inductive hypothesis)} \\
&\cong \llbracket (Q + R)[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} \\
&\qquad\text{(by definition)}
\end{aligned}$$

$P = [y = y]Q$**:**
$$\begin{aligned}
\mathbb{P}(i)_!(\llbracket [y = y]Q \rrbracket_s) &= \mathbb{P}(i)_!(\llbracket Q \rrbracket_s) \\
&\qquad\text{(by definition of } \llbracket \cdot \rrbracket_s\text{)} \\
&\cong \llbracket Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} \\
&\qquad\text{(by inductive hypothesis)} \\
&\cong \llbracket [iy = iy]Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} \\
&\qquad\text{(by definition of } \llbracket \cdot \rrbracket_{s'}\text{)} \\
&= \llbracket ([y = y]Q)[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'}
\end{aligned}$$

$P = \overline{x}y.Q$:

$$\begin{array}{rcll}
\mathbb{P}(i)_!(\llbracket \overline{x}y.Q \rrbracket_s) & = & \mathbb{P}(i)_!(x!y.\llbracket Q \rrbracket_s) & \text{(by definition of } \llbracket \cdot \rrbracket_s) \\
& \cong & i(x)!i(y).(\mathbb{P}(i)_!(\llbracket Q \rrbracket_s)) & \text{(by definition of } \mathbb{P}(i)) \\
& \cong & i(x)!i(y).\llbracket Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} & \text{(by inductive hypothesis)} \\
& \cong & \llbracket \overline{i(x)}i(y).Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} & \text{(by definition of } \llbracket \cdot \rrbracket_{s'}) \\
& = & \llbracket (\overline{x}y.Q)[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} &
\end{array}$$

$P = Q \mid R$:

$$\begin{array}{rcl}
\mathbb{P}(i)_!(\llbracket Q \mid R \rrbracket_s) & = & \mathbb{P}(i)_!(\llbracket Q \rrbracket_s |_s \llbracket R \rrbracket_s) \\
& & \quad \text{(by definition of } \llbracket \cdot \rrbracket_s) \\
& \cong & \mathbb{P}(i)_!(\llbracket Q \rrbracket_s) |_{s'} \mathbb{P}(i)_!(\llbracket R \rrbracket_s) \\
& & \quad \text{(by naturality of } |) \\
& \cong & \llbracket Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} |_{s'} \llbracket R[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} \\
& & \quad \text{(by inductive hypothesis)} \\
& \cong & \llbracket (Q \mid R)[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} \\
& & \quad \text{(by definition)}
\end{array}$$

$P = [x = y]Q$ with $x \neq y$: Trivial

$P = !Q$:

$$\begin{array}{rcll}
\mathbb{P}(i)_!(\llbracket !Q \rrbracket_s) & = & \mathbb{P}(i)_!(!_s \llbracket Q \rrbracket_s) & \text{(by definition of } \llbracket \cdot \rrbracket_s) \\
& \cong & !_{s'}(\mathbb{P}(i)_! \llbracket Q \rrbracket_s) & \text{(by naturality of } !) \\
& \cong & !_{s'} \llbracket Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} & \text{(by inductive hypothesis)} \\
& \cong & \llbracket !Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} & \text{(by definition)}
\end{array}$$

$P = \nu y\, Q$:

$$\begin{array}{rcll}
\mathbb{P}(i)_!(\llbracket \nu y\, Q \rrbracket_s) & = & \mathbb{P}(i)_!(\nu_{y \in s+\{y\},!} \llbracket Q \rrbracket_{s+\{y\}}) & \text{(by definition of } \llbracket \cdot \rrbracket_s) \\
& \cong & \nu_{y \in s'+\{y\}}(\mathbb{P}(i+y)_! \llbracket Q \rrbracket_{s+\{y\}}) & \text{(by naturality of } \nu) \\
& \cong & \nu_{y \in s'+\{y\}} \llbracket Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} & \text{(by inductive hypothesis)} \\
& \cong & \llbracket \nu\ y\ Q[i(\mathbf{x})/\mathbf{x}] \rrbracket_{s'} & \text{(by definition)}
\end{array}$$

$P = x(y).Q$**:**

$$\begin{array}{rcl}
\mathbb{P}(i)_!(\llbracket x(y).Q\rrbracket_s) & = & \mathbb{P}(i)_!(x?\langle\lambda z\,.\,\llbracket Q[z/y]\rrbracket_s, \llbracket Q[*/y]\rrbracket_{s+1}\rangle) \\
& & \qquad\qquad\text{(by definition of } \llbracket\cdot\rrbracket) \\
& \cong & i(x)?(\sum_{z\in s}(i(z)\mapsto \mathbb{P}(i)_!\llbracket Q[z/y]\rrbracket_s) \\
& & +\sum_{w\notin Im\, i}(w\mapsto \mathbb{P}([i,w])_!\llbracket Q[*/y]\rrbracket_{s+1}) \\
& & +(*\mapsto \mathbb{P}(i+1)_!\llbracket Q[*/y]\rrbracket_{s+1})) \\
& & \qquad\qquad\text{(by point (4) of Lemma 7.3.8)} \\
& \cong & i(x)?(\sum_{z\in s}(i(z)\mapsto \llbracket Q[z/y][i(\mathbf{x})/\mathbf{x}]\rrbracket_{s'}) \\
& & +\sum_{w\notin Im\, i}(w\mapsto \llbracket Q[*/y][[i,w](\mathbf{x},*)/(\mathbf{x},*)]\rrbracket_{s'}) \\
& & +(*\mapsto \llbracket Q[*/y][(i+1)(\mathbf{x},*)/(\mathbf{x},*)]\rrbracket_{s'+1})) \\
& & \qquad\qquad\text{(by inductive hypothesis)} \\
& = & i(x)?(\sum_{i(z)\in s'}(i(z)\mapsto \llbracket Q[i(\mathbf{x})/\mathbf{x}][i(z)/y]\rrbracket_{s'}) \\
& & +\sum_{w\notin Im\, i}(w\mapsto \llbracket Q[i(\mathbf{x})/\mathbf{x}][w/y]\rrbracket_{s'}) \\
& & +(*\mapsto \llbracket Q[(i+1)(\mathbf{x})/\mathbf{x}][*/y]\rrbracket_{s'+1})) \\
& = & i(x)?(\sum_{v\in s'}(v\mapsto \llbracket Q[i(\mathbf{x})/\mathbf{x}][v/y]\rrbracket_{s'}) \\
& & +(*\mapsto \llbracket Q[(i+1)(\mathbf{x})/\mathbf{x}][*/y]\rrbracket_{s'+1})) \\
& \cong & i(x)?\langle\lambda v\,.\,\llbracket Q[i(\mathbf{x})/\mathbf{x}][v/y]\rrbracket_{s'}, \llbracket Q[(i+1)(\mathbf{x})/\mathbf{x}][*/y]\rrbracket_{s'+1}\rangle \\
& \cong & \llbracket i(x)(y).Q[i(\mathbf{x})/\mathbf{x}]\rrbracket_{s'} \\
& & \qquad\qquad\text{(by definition of } \llbracket\cdot\rrbracket) \\
& = & \llbracket (x(y).Q)[i(\mathbf{x})/\mathbf{x}]\rrbracket_{s'}
\end{array}$$

□

C.2 Theorem 7.5.4

Let P be a process whose free names lie in s. Then

1. $P \xrightarrow{\overline{x}y} Q$ implies $\exists X$ with $\llbracket P\rrbracket_s \xrightarrow{x!y} X$ and $X \cong \llbracket Q\rrbracket_s$
2. $P \xrightarrow{\overline{x}(y)} Q$ implies $\exists X$ with $\llbracket P\rrbracket_s \xrightarrow{x!*_s} X$ and $X \cong \llbracket Q[*_s/y]\rrbracket_{s+1}$
3. $P \xrightarrow{x(y)} Q$ implies $\llbracket P\rrbracket_s \xrightarrow{x?} \langle F, Y\rangle$ with $F(z) \cong \llbracket Q[z/y]\rrbracket_s$ and $Y \cong \llbracket Q[*_s/y]\rrbracket_{s+1}$
4. $P \xrightarrow{\tau} Q$ implies $\exists X$ with $\llbracket P\rrbracket_s \xrightarrow{\tau} X$ and $X \cong \llbracket Q\rrbracket_s$.

Proof: The proof is a simple rule induction on the rules defining the operational semantics of the π-calculus (Figure 7.1). We exemplify in all details only some of the cases.

1. The only axiom allowing this kind of transition is the rule OUT according to which $P = \overline{x}y.Q$, hence by Definition 7.5.1 we have that $\llbracket P\rrbracket_s = x!y.\llbracket Q\rrbracket_s \xrightarrow{x!y} \llbracket Q\rrbracket_s$.

   In case of application of the rule SUM, $P = P_1 + P_2$ and $P_1 \xrightarrow{\overline{x}y} Q$, hence by inductive hypothesis $\llbracket P_1\rrbracket_s \xrightarrow{x!y} X \cong \llbracket Q\rrbracket_s$ and by definition

$$\llbracket P_1 + P_2\rrbracket_s = \llbracket P_1\rrbracket_s + \llbracket P_2\rrbracket_s \xrightarrow{x!y} X \cong \llbracket Q\rrbracket_s\ .$$

In case of application of PAR, $P = P_1 \,|\, P_2$ and $P_1 \xrightarrow{\overline{x}y} Q$, hence by inductive hypothesis $(\![P_1]\!)_s \xrightarrow{x!y} X \cong (\![Q]\!)_s$, i.e., there exists a $j \in (\![P]\!)_s(x!y)$ such that $X = ((\![P]\!)_s)_{|j}$. By Definition 7.5.1, $(\![P_1 \,|\, P_2]\!)_s = (\![P_1]\!)|_s(\![P_2]\!)_s$ and by the characterisation of Theorem 7.4.7, $j \in (\![P_1]\!)_s|_s(\![P_2]\!)_s$ and $((\![P_1]\!)_s|_s(\![P_2]\!)_s)_{|j} \cong X|_s(\![P_2]\!)_s$. Hence

$$(\![P_1 \,|\, P_2]\!)_s \xrightarrow{x!y} ((\![P_1]\!)_s|_s(\![P_2]\!)_s)_{|j} \cong X|_s(\![P_2]\!)_s \cong (\![Q]\!)_s|_s(\![P_2]\!)_s = (\![Q \,|\, P_2]\!)_s \ .$$

In case of application of RES, $P = \nu z\, P_1$, $P_1 \xrightarrow{\overline{x}y} Q_1$ with $z \notin \{x, y\}$ and $Q = \nu z\, Q_1$. By inductive hypothesis,

$$(\![P_1]\!)_{s+\{z\}} \xrightarrow{x!y} X \cong (\![Q_1]\!)_{s+\{z\}} \ .$$

Again by the characterisation of $\nu_{z \in s+\{z\}}$ given by Theorem 7.4.2 one has that if there exists a $j \in (\![P_1]\!)_s(x!y)$ such that $X = ((\![P_1]\!)_{s+\{z\}})_{|j}$ then

$$j \in \nu_{z \in s+\{z\}}((\![P_1]\!)_{s+\{z\}})(x!y) = (\![\,\nu z\, P_1]\!)_s(x!y)$$

too and $((\![\,\nu z\, P_1]\!)_s)_{|j} \cong \nu_{z \in s+\{z\}}(X) \cong \nu_{z \in s+\{z\}}((\![Q_1]\!)_{s+\{z\}}) = (\![\,\nu z\, Q_1]\!)_s$.

2. Similar as above for the cases given by SUM, PAR and RES. In case of application of the rule OPEN this means that $P = \nu y\, P_1$ and $P_1 \xrightarrow{\overline{x}y} Q$. Hence by inductive hypothesis, $(\![P_1]\!)_{s+\{y\}} \xrightarrow{x!y} X \cong (\![Q]\!)_{s+\{y\}}$. Again by the characterisation of Theorem 7.4.2 one has that

$$\nu_{y \in s+\{y\}}((\![P_1]\!)_{s+\{y\}} \xrightarrow{x!*} \mathbb{P}(b_{s,y})_! X \cong \mathbb{P}(b_{s,y})_!((\![Q]\!)_{s+\{y\}}) \ .$$

But by the Substitution Lemma (Lemma 7.5.2)

$$\mathbb{P}(b_{s,y})_!((\![Q]\!)_{s+\{y\}}) \cong (\![Q[*/y]]\!)_{s+1} \ .$$

3. Similar as above for the case SUM, PAR, RES. In case of application of the axiom IN the property is immediately derived from the definition as it was for OUT.
4. Similar as above for SUM, PAR and RES. In case of application of the rule COM, this means that $P = P_1 \,|\, P_2$, $P_1 \xrightarrow{x(y)} Q_1$, $P_2 \xrightarrow{\overline{x}z} Q_2$ and $Q = Q_1[z/y] \,|\, Q_2$. By inductive hypothesis we have that

$$\begin{aligned} &(\![P_1]\!)_s \xrightarrow{x?} \langle F, Y\rangle \cong \langle \lambda\, z.\, (\![Q_1[z/y]]\!)_s, (\![Q_1[*/y]]\!)_{s+1}\rangle \\ &(\![P_2]\!)_s \xrightarrow{x!z} X \cong (\![Q_2]\!)_s \ . \end{aligned}$$

By the characterisation of Theorem 7.4.7 we then have that

$$(\![P_1 \,|\, P_2]\!)_s \cong (\![P_1]\!)_s|_s(\![P_2]\!)_s \xrightarrow{\tau} W \cong F(z)|_s X \cong (\![Q_1[z/y]]\!)_s|_s(\![Q_2]\!)_s \ .$$

In case of application of the CLOSE rule we have that $P = P_1 \,|\, P_2$, $P_1 \xrightarrow{x(y)} Q_1$, $P_2 \xrightarrow{\overline{x}(y)} Q_2$ and $Q = \nu y(Q_1 \,|\, Q_2)$. By inductive hypothesis we have that

$$\begin{aligned} &(\![P_1]\!)_s \xrightarrow{x?} \langle F, Y\rangle \cong \langle \lambda\, z.\, (\![Q_1[z/y]]\!)_s, (\![Q_1[*/y]]\!)_{s+1}\rangle \\ &(\![P_2]\!)_s \xrightarrow{x!*} X \cong (\![Q_2[*/y]]\!)_{s+1} \ . \end{aligned}$$

Hence by the Theorem 7.4.7 we then have that

$$(\![P_1 \mid P_2]\!)_s \cong (\![P_1]\!)_s|_s(\![P_2]\!)_s \xrightarrow{\tau} W \cong \nu_{*\in s+1}(Y|_{s+1}X)\ .$$

To conclude, observe that the following chain of isomorphisms hold:

$$\begin{array}{rcl}
(\![\,\nu y\,Q]\!)_s & = & \nu_{y\in s+\{y\}}(\![Q]\!)_{s+\{y\}} \\
& & \qquad\qquad \text{(by definition of } (\![\,\cdot\,]\!)) \\
& \cong & \nu_{y\in s+\{y\}}(\mathbb{P}(b_{s,y})_!(\![Q[*/y]]\!)_{s+1}) \\
& & \qquad\qquad \text{(by the Substitution Lemma)} \\
& \cong & \nu_{*\in s+1}(\![Q[*/y]]\!)_{s+1} \\
& & \qquad\qquad \text{(by naturality of } \nu \text{ (Theorem 7.4.3))} \\
& \cong & \nu_{*\in s+1}((\![Q_1[*/y]]\!)_{s+1}|_{s+1}(\![Q_2[*/y]]\!)_{s+1}) \\
& \cong & \nu_{*\in s+1}(Y|_{s+1}X)\ .
\end{array}$$

□

Bibliography

[1] Samson Abramsky. On semantic foundations for applicative multiprogramming. In *ICALP '83, Tenth Colloquium on Automata, Languages and Programming*, volume 154 of *Lecture Notes in Computer Science*, pages 1–14. Springer-Verlag, 1983.

[2] Samson Abramsky. The lazy lambda calculus. In *Research topics in Functional Programming*, pages 65–117. Addison Wesley, 1990.

[3] Samson Abramsky. A domain equation for bisimulation. *Information and Computation*, 92(2):161–218, 1991.

[4] Peter Aczel and N. Mendler. A final coalgebra theorem. In D. H. Pitt et al., editor, *Proceedings of CTCS '89, International Conference on Category Theory and Computer Science*, volume 389 of *Lecture Notes in Computer Science*, pages 357–365, 1989.

[5] Jiři Adámek and Jiři Rosický. *Locally Presentable and Accessible Categories*, volume 189 of *London Mathematical Society Lecture Notes Series*. Cambridge University Press, 1994.

[6] J. C. M. Baeten and W. P. Weijland. *Process Algebra*. Cambridge University Press, 1990.

[7] Michael Barr. *$*$-autonomous categories*, volume 752 of *Lecture Notes in Mathematics*. Springer-Verlag, Berlin, 1979. With an appendix by Po Hsiang Chu.

[8] Marek Bednarczyk. Hereditary history preserving bisimulation or what is the power of the future perfect in program logics. Technical report, Polish Academy of Sciences, Gdansk, 1991.

[9] Jean Bénabou. Introduction to bicategories. In *Reports of the Midwest Category Seminar I*, volume 47 of *Lecture Notes in Mathematics*, pages 1–77. Springer-Verlag, 1967.

[10] Jean Bénabou. Les distributeurs. Rapport n° 33. Seminaires de Mathématiques Pure, Institut de Mathématiques, Université Catholique de Louvain, 1973.

[11] Jean Bénabou. Fibered categories and the foundations of naive category theory. *The Journal of Symbolic Logic*, 50(1):10–37, 1985.

[12] J. van Bentham. Correspondence theory. In M. Gabbay and Guenther, editors, *Handbook of Philosophical Logic*, volume 2, pages 167–247. Reidel, 1984.

[13] Gérard Berry, Pierre-Louis Curien, and Jean-Jacques Lévy. Full abstraction for sequential languages: the state of the art. In M. Nivat and J. Reynolds, editors, *Algebraic Semantics*, pages 89–132. Cambridge University Press, 1985.

[14] G. J. Bird, Gregory M. Kelly, A. John Power, and Ross Street. Flexible limits for 2-categories. *Journal of Pure and Applied Algebra*, 61:1–27, 1989.

[15] R. Blackwell, Gregory M. Kelly, and A. John Power. Two-dimensional monad theory. *Journal of Pure and Applied Algebra*, 59:1–41, 1989.

[16] Richard Blute, Josée Desharnais, Abbas Edalat, and Prakash Panangaden. Bisimulation for labelled Markov processes. In LICS '97 [75], pages 149–158.

[17] Francis Borceux. *Handbook of categorical algebra I*, volume 50 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, 1994.

[18] Gérard Boudol. Asynchrony and the π-calculus. Technical Report 1702, INRIA, Sophia Antipolis, 1992.

[19] Nadia Busi, Roberto Gorrieri, and Gianluigi Zavattaro. A process algebraic view of Linda coordination primitives. *Theoretical Computer Science*, 192(2):167–199, 1998.

[20] Luca Cardelli and Andrew D. Gordon. Mobile ambients. In M. Nivat, editor, *Proceedings of the First International Conference on Foundations of Software Science and Computation Structures (FoSSaCS '98)*, volume 1378 of *Lecture Notes in Computer Science*, pages 140–155. Springer-Verlag, 1998.

[21] Gian Luca Cattani, Marcelo Fiore, and Glynn Winskel. A theory of recursive domains with applications to concurrency. In *LICS 98, Proceedings of the Thirteenth Annual IEEE Symposium on Logic in Computer Science*, pages 214–225. IEEE Computer Society Press, 1998.

[22] Gian Luca Cattani, A. John Power, and Glynn Winskel. A categorical axiomatics for bisimulation. In Sangiorgi and de Simone [115], pages 581–596.

[23] Gian Luca Cattani and Vladimiro Sassone. Higher dimensional transition systems. In LICS '96 [74], pages 55–62.

[24] Gian Luca Cattani and Vladimiro Sassone. On higher dimensional transition systems. Manuscript in preparation, 1998.

[25] Gian Luca Cattani, Ian Stark, and Glynn Winskel. Presheaf models for the π-calculus. In *Proceedings of the 7th International Conference on Category Theory and Computer Science, CTCS '97*, number 1290 in Lecture Notes in Computer Science, pages 106–126. Springer-Verlag, 1997.

[26] Gian Luca Cattani and Glynn Winskel. Presheaf models for concurrency. In van Dalen and Bezem [133], pages 58–75.

[27] Paolo Ciancarini, Keld K. Jensen, and Daniel Yankelevich. On the operational semantics of a coordination language. In *Object-base Models and Languages for Concurrent Systems*, volume 924 of *Lecture Notes in Computer Science*, pages 77–106. Springer-Verlag, 1995.

[28] Brian J. Day. On closed categories of functors. In *Reports of the Midwest Category Seminar IV*, number 137 in Lecture Notes in Mathematics, pages 1–38. Springer-Verlag, 1970.

[29] Lisbeth Fajstrup, Eric Goubault, and Martin Raußen. Detecting deadlocks in concurrent systems. In Sangiorgi and de Simone [115], pages 332–347.

[30] Marcelo P. Fiore. *Axiomatic Domain Theory in Categories of Partial Maps*. Distinguished Dissertations in Computer Science. Cambridge University Press, 1996.

[31] Marcelo P. Fiore. A coinduction principle for recursive data types based on bisimulation. *Information and Computation*, 127(2):186–198, 1996.

[32] Marcelo P. Fiore, Eugenio Moggi, and Davide Sangiorgi. A fully-abstract model for the π-calculus (extended abstract). In LICS '96 [74], pages 43–54.

[33] Marcelo P. Fiore and Gordon D. Plotkin. An extension of models of axiomatic domain theory to models of synthetic domain theory. In van Dalen and Bezem [133], pages 129–149.

[34] Marcelo P. Fiore, Gordon D. Plotkin, and A. John Power. Complete cuboidal sets in axiomatic domain theory. In LICS '97 [75], pages 268–279.

[35] Peter J. Freyd. Algebraically complete categories. In A. Carboni, M.C. Pedicchio, and G. Rosolini, editors, *Category Theory*, volume 1488 of *Lecture Notes in Mathematics*, pages 131–156. Springer-Verlag, 1991.

[36] Peter J. Freyd. Remarks on algebraically compact categories. In M.P. Fourman, P.T. Johnstone, and A.M. Pitts, editors, *Applications of Categories in Computer Science*, volume 177 of *London Mathematical Society Lecture Note Series*, pages 95–106. Cambridge University Press, 1992.

[37] Peter J. Freyd and Andre Scedrov. *Categories, allegories*, volume 39 of *North-Holland Mathematical Library*. North-Holland Publishing Co., Amsterdam, 1990.

[38] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):101, 1987.

[39] Jean-Yves Girard. *Proofs and Types*, volume 7 of *Cambridge Tracts in Theoretical Computer Science*. Cambridge Univeristy Press, 1989. Translated and with appendices by P. Taylor and Y. Lafont.

[40] Rob van Glabbeek. Bisimulation for higher dimensional automata. E-mail message sent to the Concurrency mailing list on 7 July, 1991. Available at `http://theory.stanford.edu/people/rvg/hda`.

[41] Rob van Glabbeek and Ursula Goltz. Equivalence notions for concurrent systems and refinement of actions. In *Mathematical Foundations of Computer Science 1989*, number 379 in Lecture Notes in Computer Science, pages 237–248. Springer-Verlag, 1989.

[42] R. Gordon, A. John Power, and Ross Street. Coherence for tricategories. *Memoirs of the American Mathematical Society*, 117(558):vi+81, 1995.

[43] Eric Goubault. Domains of higher dimensional automata. In E. Best, editor, *Proceedings of CONCUR '93*, volume 715 of *Lecture Notes in Computer Science*, pages 293–307. Springer-Verlag, 1993.

[44] Eric Goubault. *The Geometry of Concurrency.* PhD thesis, École Polytechnique, 1995.

[45] Eric Goubault and T. Jensen. Homology of higher dimensional automata. In W. R. Cleaveland, editor, *Proceedings of CONCUR '92*, volume 630 of *Lecture Notes in Computer Science*, pages 254–268. Springer-Verlag, 1992.

[46] J. Gray and A. Scedrov, editors. *Categories in computer science and logic*, volume 92 of *Contemporary Mathematics*, Providence, RI, 1989. American Mathematical Society.

[47] John W. Gray. *Formal Category Theory: Adjointness for* 2*-Categories*, volume 391 of *Lecture Notes in Mathematics.* Springer-Verlag, 1974.

[48] Alexandre Grothendieck. *Revêtements étales et groupe fondamental*, volume 224 of *Lecture Notes in Mathematics.* Springer-Verlag, Berlin, 1971. Séminaire de Géométrie Algébrique du Bois Marie 1960–1961 (SGA 1), Dirigé par Alexandre Grothendieck. Augmenté de deux exposés de M. Raynaud.

[49] Carl A. Gunter. *Profinite Solutions for Recursive Domain Equations.* PhD thesis, University of Wisconsin at Madison, 1985.

[50] Matthew Hennessy. A fully abstract denotational semantics for the π-calculus. Technical Report 96:04, School of Cognitive and Computing Sciences, University of Sussex, 1996.

[51] Matthew Hennessy and Gordon D. Plotkin. Full abstraction for a simple parallel programming language. In J. Bečvář, editor, *Mathematical Foundations of Computer Science (MFCS) 1979*, volume 74 of *Lecture Notes in Computer Science*, pages 108–120, Berlin, 1979. Springer-Verlag.

[52] Claudio Hermida. *Fibrations Logical Predicates and Indeterminates.* PhD thesis, University of Edinburgh, 1993. Available as Technical Report DAIMI-PB 462, Computer Science Department, University of Aarhus.

[53] Claudio Hermida and Bart Jacobs. Induction and coinduction via subset types and quotient types. In P. Dybjer and R. Pollack, editors, *Informal proceedings of the Joint CLICS-TYPES Workshop on Categories and Type Theory*, 1995.

[54] Thomas T. Hildebrandt, Prakash Panangaden, and Glynn Winskel. A relational model of non-deterministic dataflow. In Sangiorgi and de Simone [115], pages 613–628.

[55] C. Anthony R. Hoare. *Communicating Sequential Processes.* Englewood Cliffs, 1985.

[56] Martin Hyland. Category theory. Notes taken by Cocky Hillhorst of Martin Hyland's course, 1995.

[57] Bart Jacobs. *Categorical Type Theory.* PhD thesis, University of Nijmegen, 1991.

[58] F. V. Jensen. Inductive inference in reflexive domains. Technical Report CSR 86-1981, Department of Computer Science, University of Edinburgh, 1981.

[59] Peter T. Johnstone. *Topos Theory*, volume 10 of *L.M.S. Mathematical Monographs.* Academic Press, 1977.

[60] Peter T. Johnstone. *Stone Spaces*, volume 3 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1982.

[61] Peter T. Johnstone. Fibered categories. Notes taken by Paul Taylor of Peter Johnstone's course, 1983.

[62] André Joyal and Ieke Moerdijk. A completeness theorem for open maps. *Annals of Pure and Applied Logic*, 70(1):51–86, 1994.

[63] André Joyal and Ieke Moerdijk. *Algebraic set theory*, volume 220 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, 1995.

[64] André Joyal, Mogens Nielsen, and Glynn Winskel. Bisimulation from open maps. *Information and Computation*, 127(2):164–185, 1996.

[65] Gregory M. Kelly. *Basic Concepts of Enriched Category Theory*, volume 64 of *London Mathematical Society Lecture Notes Series*. Cambridge University Press, 1982.

[66] Gregory M. Kelly and M. L. Laplaza. Coherence for compact closed categories. *Journal of Pure and Applied Algebra*, 19:193–213, 1980.

[67] Gregory M. Kelly and Ross Street. Review of the elements of 2-categories. In *Category Seminar Proceedings Sidney category Theory Seminar 1972/73*, pages 75–103. Springer-Verlag, 1974.

[68] Anders Kock. Monads for which structures are adjoints to units. *Journal of Pure and Applied Algebra*, 104:41–59, 1995.

[69] Joachim Lambek. A fixpoint theorem for complete categories. *Math. Zeitschr.*, 103:151–161, 1968.

[70] Kim G. Larsen and Arne Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94:1–28, 1991.

[71] F. William Lawvere. Equality in hyperdoctrines and comprehension schema as an adjoint functor. In *Applications of Categorical Algebra (Proc. Sympos. Pure Math., Vol. XVII, New York, 1968)*, pages 1–14. Amer. Math. Soc., 1970.

[72] F. William Lawvere. Metric spaces, generalized logic and closed categories. *Rend. Sem. Mat. Fis. Milano*, 43:135–166, 1973.

[73] Daniel J. Lehmann. *Categories for fixpoint semantics*. PhD thesis, University of Warwick, 1976.

[74] *LICS '96, Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1996.

[75] *LICS '97, Proceedings of the Twelfth Annual IEEE Symposium on Logic in Computer Science*. IEEE Computer Society Press, 1997.

[76] Saunders Mac Lane. *Categories for the Working Mathematician*, volume 5 of *Graduate Texts in Mathematics*. Springer-Verlag, 1971.

[77] Saunders Mac Lane and Ieke Moerdijk. *Sheaves in Geometry and Logic: A First Introduction to Topos Theory*. Springer-Verlag, 1992.

[78] Michael Makkai and Robert Paré. *Accessible Categories: The Foundations of Categorical Model Theory*, volume 104 of *Contemporary Mathematics*. American Mathematical Society, 1989.

[79] K. L. McMillan. A technique of state space search based on unfolding. *Formal Methods in System Design*, 6(1):45–65, 1995.

[80] Robin Milner. *A Calculus of Communicating Systems*, volume 92 of *Lecture Notes in Computer Science*. Springer-Verlag, 1982.

[81] Robin Milner. Calculi for synchrony and asynchrony. *Theoretical Computer Science*, 25:267–310, 1983.

[82] Robin Milner. *Communication and Concurrency.* International Series in Computer Science. Prentice Hall, 1989.

[83] Robin Milner. Functions as processes. *Mathematical Structures in Computer Science*, 20(2):119–141, 1992.

[84] Robin Milner. The polyadic π-calculus: a tutorial. In *Logic and algebra of specification (Marktoberdorf, 1991)*, volume 94 of *NATO Adv. Sci. Inst. Ser. F Comput. Systems Sci.*, pages 203–246. Springer-Verlag, Berlin, 1993.

[85] Robin Milner. The π-calculus. Lecture notes for a course, 1994.

[86] Robin Milner. Calculi for interaction. *Acta Informatica*, 33:707–737, 1996.

[87] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes. I. *Information and Computation*, 100(1):1–40, 1992.

[88] Robin Milner, Joachim Parrow, and David Walker. A calculus of mobile processes. II. *Information and Computation*, 100(1):41–77, 1992.

[89] Robin Milner, Joachim Parrow, and David Walker. Modal logics for mobile processes. *Theoretical Computer Science*, 114(1):149–171, 1993.

[90] Eugenio Moggi. Notions of computations and monads. *Information and Computation*, 93(1):55–92, 1991.

[91] M. Nielsen and E. Schmidt, editors. *ICALP '82, Nineth Colloquium on Automata, Languages and Programming*, volume 140 of *Lecture Notes in Computer Science*. Springer-Verlag, 1982.

[92] Mogens Nielsen and Allan Cheng. Observe behaviour categorically. In *Proceedings of FST&TCS 15, Fifteenth Conference on the Foundations of Software Technology and Theoretical Computer Science*, volume 1026 of *Lecture Notes in Computer Science*, pages 263–278. Springer-Verlag, 1995.

[93] Mogens Nielsen, Gordon D. Plotkin, and Glynn Winskel. Petri nets, event structrues and domains, part I. *Theoretical Computer Science*, 13:85–108, 1981.

[94] Peter W. O'Hearn and R.D. Tennent. Relational parametricity and local variables. In *Proceedings of the 20th ACM Symposium on Principles of Programming Languages*, pages 171–184. ACM, 1993.

[95] Robert Paré. Simply connected limits. *Canadian Journal of Mathematics*, XLII(4):731–746, 1990.

[96] D. M. R. Park. Concurrency and automata on infinite sequences. In *Theoretical Computer Science, 5th GL-conference*, volume 104 of *Lecture Notes in Computer Science*. Springer-Verlag, 1981.

[97] Duško Pavlović. *Predicates and Fibrations.* PhD thesis, University of Utrecht, 1990.

[98] Andrew M. Pitts. On product and change of base for toposes. *Cahiers de Topologie et Géometrie Différentielle Catégoriques*, XXVI(1):43–61, 1985.

[99] Andrew M. Pitts. A co-induction principle for recursively defined domains. *Theoretical Computer Science*, 124:195–219, 1994.

[100] Andrew M. Pitts. Relational properties of domains. *Information and Computation*, 127(2):66–90, 1996.

[101] Gordon D. Plotkin. A powerdomain construction. *SIAM Journal of Computation*, 5(3):456–487, 1976.

[102] Gordon. D. Plotkin. A structural approach to operational semantics. Technical Report DAIMI FN 19, Department of Computer Science, University of Aarhus, 1981. Reprinted in 1991.

[103] Gordon D. Plotkin. A powerdomain for countable nondeterminism. In Nielsen and Schmidt [91], pages 418–428.

[104] Gordon. D. Plotkin. Domains. Technical report, Department of Computer Science, University of Edinburgh, 1983. Includes the "Pisa notes".

[105] Gordon D. Plotkin. Algebraic completeness and compactness in an enriched setting. Invited lecture given at the Workshop on Logic, Domains, and Programming Languages. Darmstadt, 1995.

[106] A. John Power. Coherence for bicategories with finite bilimits. I. In Gray and Scedrov [46], pages 341–347.

[107] A. John Power. A general coherence result. *Journal of Pure and Applied Algebra*, 57(2):165–173, 1989.

[108] A. John Power. A 2-categorical pasting theorem. *Journal of Algebra*, 129(2):439–445, 1990.

[109] A. John Power. Why tricategories? *Information and Computation*, 120(2):251–262, 1995.

[110] A. John Power. An elementary definition of pseudo-monads. Private communication, 1998.

[111] Vaughan Pratt. Modelling concurrency with partial orders. *International Journal of Parallel Processing*, 15:33–71, 1986.

[112] Vaughan Pratt. Modelling concurrency with geometry. In *Proceedings of the 18th ACM Symposium on Principles of Programming Languages*, pages 311–322. ACM Press, 1991.

[113] A. Rabinovitch and B. Traktenbrot. Behaviour structures and nets. *Fundamenta Informatica*, 11(4):357–404, 1988.

[114] Jan Rutten and Daniele Turi. Initial algebra and final coalgebra semantics for concurrency. In J. de Bakker et al., editor, *Proceedings of the REX workshop: A decade of concurrency - Reflections and perspectives*, volume 660 of *Lecture Notes in Computer Science*, pages 530–582. Springer-Verlag, 1994.

[115] D. Sangiorgi and R. de Simone, editors. *Proceedings of the 9th International Conference on Concurrency Theory, CONCUR '98*, volume 1466 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.

[116] Davide Sangiorgi. *Expressing Mobility in Process Algebras: First-Order and Higher-Order Paradigms.* PhD thesis, University of Edinburgh, 1992.

[117] Davide Sangiorgi. π-calculus, internal mobility, and agent-passing calculi. Technical Report 2539, INRIA, Sophia Antipolis, 1995.

[118] Davide Sangiorgi. Bisimulation for higher-order process calculi. *Information and Computation*, 131(2):141–178, 1996.

[119] Vladimiro Sassone, Mogens Nielsen, and Glynn Winskel. Models for concurrency: towards a classification. *Theoretical Computer Science*, 170(1-2):297–348, 1996.

[120] Dana S. Scott. Continuous lattices. In F.W. Lawvere, editor, *Toposes, Algebraic Geometry and Logic*, volume 274 of *Lecture Notes in Mathematics*, pages 97–136. Springer-Verlag, 1972.

[121] Dana S. Scott. Domains for denotational semantics. In Nielsen and Schmidt [91], pages 577–613.

[122] R. A. G. Seely. Linear logic, *-autonomous categories and cofree algebras. In Gray and Scedrov [46], pages 371–382.

[123] Peter Sewell. From rewrite rules to bisimulation congruences. In Sangiorgi and de Simone [115], pages 269–284.

[124] Harold Simmons. The glueing construction and lax limits. *Mathematical Structures in Computer Science*, 4(4):393–431, 1994.

[125] M.B. Smyth and Gordon D. Plotkin. The category-theoretic solution of recursive domain equations. *SIAM Journal of Computing*, 11(4):761–783, 1982.

[126] Ian Stark. A fully abstract domain model for the π-calculus. In LICS '96 [74], pages 36–42.

[127] Ross Street. Fibrations in bicategories. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, XXI(2):111–160, 1980.

[128] Ross Street. Cauchy characterization of enriched categories. *Rend. Sem. Mat. Fis. Milano*, 51:217–233, 1981.

[129] Paul Taylor. The limit-colimit coincidence for categories. Manuscript, 1988.

[130] Bent Thomsen. Plain CHOCS: A second generation calculus for higher order processes. *Acta Informatica*, 30(1):1–59, 1993.

[131] Daniele Turi. *Functorial Operational Semantics and its denotational dual.* PhD thesis, The CWI, Amsterdam, 1996.

[132] Daniele Turi and Gordon Plotkin. Towards a mathematical operational semantics. In LICS '97 [75], pages 280–291.

[133] D. van Dalen and M. Bezem, editors. *Computer Science Logic. 10th International Workshop, CSL '96, Annual Conference of the European Association for Computer Science Logic. Selected Papers*, volume 1258 of *Lecture Notes in Computer Science*. Springer-Verlag, 1997.

[134] Mitchell Wand. Fixed-point constructions in order-enriched categories. *Theoretical Computer Science*, 8:13–30, 1979.

[135] Glynn Winskel. Synchronisation trees. *Theoretical Computer Science*, 34:33–82, 1985.

[136] Glynn Winskel. A category of labelled petri nets and compositional proof system. In *LICS '88, Proceedings, Third Annual Symposium on Logic in Computer Science*, pages 142–154. IEEE Computer Society Press, 1988.

[137] Glynn Winskel. *The Formal Semantics of Programming Languages.* Foundations of Computing Series. The MIT Press, 1993.

[138] Glynn Winskel. A presheaf semantics of value-passing processes (extended abstract). In U. Montanari and V. Sassone, editors, *CONCUR'96, Proceedings of the 9th International Conference on Concurrency Theory*, volume 1119 of *Lecture Notes in Computer Science*, pages 98–114. Springer-Verlag, 1996.

[139] Glynn Winskel. Ambients as presheaves. Manuscript, 1998.

[140] Glynn Winskel. A linear metalanguage for concurrency. In *Proceedings of AMAST '98*, Lecture Notes in Computer Science. Springer-Verlag, 1998. To appear.

[141] Glynn Winskel and Mogens Nielsen. Models for concurrency. In *Handbook of logic in computer science, Vol. 4*, Oxford Sci. Publ., pages 1–148. Oxford Univ. Press, 1995.

[142] Glynn Winskel and Mogens Nielsen. Models for concurrency. In A. M. Pitts and P. Dybjer, editors, *Semantics and Logics of Computation.* Cambridge University Press, 1997.

[143] R. J. Wood. Proarrows II. *Cahiers de Topologie et Géométrie Différentielle Catégoriques*, XXVI(2):135–168, 1985.

[144] Oswald Wyler. *Lecture notes on topoi and quasitopoi.* World Scientific Publishing Co. Inc., Teaneck, NJ, 1991.

Recent BRICS Dissertation Series Publications

DS-99-1 **Gian Luca Cattani.** ***Presheaf Models for Concurrency (Unrevised)*****. April 1999. PhD thesis. xiv+255 pp.**

DS-98-3 **Kim Sunesen.** ***Reasoning about Reactive Systems*****. December 1998. PhD thesis. xvi+204 pp.**

DS-98-2 **Søren B. Lassen.** ***Relational Reasoning about Functions and Nondeterminism*****. December 1998. PhD thesis. x+126 pp.**

DS-98-1 **Ole I. Hougaard.** ***The CLP(OIH) Language*****. February 1998. PhD thesis. xii+187 pp.**

DS-97-3 **Thore Husfeldt.** ***Dynamic Computation*****. December 1997. PhD thesis. 90 pp.**

DS-97-2 **Peter Ørbæk.** ***Trust and Dependence Analysis*****. July 1997. PhD thesis. x+175 pp.**

DS-97-1 **Gerth Stølting Brodal.** ***Worst Case Efficient Data Structures*****. January 1997. PhD thesis. x+121 pp.**

DS-96-4 **Torben Braüner.** ***An Axiomatic Approach to Adequacy*****. November 1996. Ph.D. thesis. 168 pp.**

DS-96-3 **Lars Arge.** ***Efficient External-Memory Data Structures and Applications*****. August 1996. Ph.D. thesis. xii+169 pp.**

DS-96-2 **Allan Cheng.** ***Reasoning About Concurrent Computational Systems*****. August 1996. Ph.D. thesis. xiv+229 pp.**

DS-96-1 **Urban Engberg.** ***Reasoning in the Temporal Logic of Actions — The design and implementation of an interactive computer system*****. August 1996. Ph.D. thesis. xvi+222 pp.**