Technical Report

UCAM-CL-TR-751
ISSN 1476-2986

Number 751

UNIVERSITY OF CAMBRIDGE
Computer Laboratory

Nominal domain theory for concurrency

David C. Turner

July 2009

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

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

This technical report is based on a dissertation submitted December 2008 by the author for the degree of Doctor of Philosophy to the University of Cambridge, Clare College.

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

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

ISSN 1476-2986

Nominal Domain Theory for Concurrency

David C. Turner

Abstract

Domain theory provides a powerful mathematical framework for describing sequential computation, but the traditional tools of domain theory are inapplicable to concurrent computation. Without a general mathematical framework it is hard to compare developments and approaches from different areas of study, leading to time and effort wasted in rediscovering old ideas in new situations.

A possible remedy to this situation is to build a denotational semantics based directly on computation paths, where a process denotes the set of paths that it may follow. This has been shown to be a remarkably powerful idea, but it lacks certain computational features. Notably, it is not possible to express the idea of names and name-generation within this simple path semantics.

Nominal set theory is a non-standard mathematical foundation that captures the notion of names in a general way. Building a mathematical development on top of nominal set theory has the effect of incorporating names into its fabric at a low level. Importantly, nominal set theory is sufficiently close to conventional foundations that it is often straightforward to transfer intuitions into the nominal setting.

Here the original path-based domain theory for concurrency is developed within nominal set theory, which has the effect of systematically adjoining name-generation to the model. This gives rise to an expressive metalanguage, Nominal HOPLA, which supports a notion of name-generation. Its denotational semantics is given entirely in terms of universal constructions on domains. An operational semantics is also presented, and relationships between the denotational and operational descriptions are explored.

The generality of this approach to including name generation into a simple semantic model indicates that it will be possible to apply the same techniques to more powerful domain theories for concurrency, such as those based on presheaves.

Acknowledgments

Firstly I would like to wholeheartedly thank my supervisor, Glynn Winskel, for all the ideas, support, encouragement, direction and freedom that he has given me over the course of my research. I hope that my work brings his vision for the future of semantics closer to reality. I would also like to thank the EPSRC for their generous support through their grant "Domain Theory for Concurrency", number GR/T22049/01, whose principal investigator was Prof. Winskel.

Sam Staton's positive influence on my work cannot be overstated, both for his detailed mathematical support and his equally thorough typographical assistance. All errors that remain are entirely my own work.

Andrew Pitts must also receive my appreciation for his auspiciously timed seminar on nominal set theory at the start of my studies, as well as his enthusiasm for all things nominal, and for his terrible puns: it is fortunate that for this work his support seems to be infinite. I regret not having capitalised more on Jamie Gabbay's equally infectious zeal for the subject; I hope that he, and all the other participants of CANS, enjoy reading this thesis as much as I have enjoyed producing it.

At the Computer Laboratory and at DPMMS I have met many other wonderful people who have helped me along my way. In particular I would like to thank Martin Hyland for pointing me in Glynn's direction in the first place; Matthew Parkinson for helping me to hammer LaTeX into shape and in particular for his macro to produce $\widehat{\text{very wide hats indeed}}$; and the Theory and Semantics Group for all our varied conversations on all manner of topics over tea and beer and games of pool.

Clare College has been a wonderfully supportive — and remarkably patient — home to me for the past decade, and the source of many valuable friendships.

My family have played a larger part in this work than perhaps they realise. I would like to thank Mum and Dad and Andy and Di and Howard and all my grandparents for giving me the encouragement, the work ethic and the environment that brought me here. I hope you can make it as far as page 10 and I hope it makes you proud.

Finally, to Tess, who is the most wonderful person in the world: without you I would have never got so far, so happily, with so much of my sanity still intact. You made it all worthwhile. This is for you.

To all of you, thank you.

— Dave Turner, Cambridge, December 2008

Contents

Chapter 1

Introduction

Nygaard and Winskel[20] motivate the study of domain theory as follows.

> Denotational semantics and domain theory of Scott and Strachey provide a *global* mathematical setting for sequential computation, and thereby place programming languages in connection with each other; connect with the mathematical worlds of algebra, topology and logic; and inspire programming languages, type disciplines and methods of reasoning.

Sequential computations typically receive some input, perform a calculation, and output the result once they have finished. In contrast, computation in the modern world is increasingly performed by interconnected collections of devices, each performing parts of computations and interacting with their neighbours and with their environment in the course of their calculations. Input may not all be received at once; output may not all be sent simultaneously; and computations do not necessarily even have a well-defined finish. In this world of concurrent, distributed computation there is no global mathematical model that serves to guide developments and unify different approaches and which provides a forum for comparing innovations. In particular, classical domain theory has failed to scale to the intricacy required to properly model concurrent computation. The result of this is that a wide variety of approaches to understanding concurrency have been developed on a more-or-less *ad hoc* basis.

Operational semantics — which studies collections of operational rules that dictate how a computation proceeds step-by-step — is an accessible and popular approach to studying concurrency[25]. Operational semantics is typically described syntactically, affecting the state of a computation by altering its symbolic representation. Operational descriptions often suggest a possible physical implementation of a computation by providing a description of the local behaviour of a process. Working at this low level of abstraction makes it easier to understand the progress of a computation, but the quantity of detail can make it harder to see the high-level situation.

On the other hand denotational semantics — which associates a computation with a mathematical object that captures the essence of the computation in its mathematical properties — starts from a much more abstract viewpoint and makes it easier to distinguish general results about computation itself from specific results about particular syntactic systems. The drawback of this approach is that the mathematics involved is often either too coarse to be a suitable tool, or too intricate to be useful to the working scientist.

Consequently, theories for concurrent computation form a disjointed landscape with unclear relationships between approaches. Particular process calculi are optimised to different tasks, and it is sometimes difficult to see which optimisations are valuable innovations that can be applied elsewhere and which are relevant only to the calculus in question. The lack of a common framework means that lessons derived from work in one area remain isolated from other areas of study.

Work by Cattani and Winskel on a semantics given in terms of presheaves[5] has the potential to provide a common denotational framework for studying concurrency. The presheaf semantics supports a rich domain theory which captures nondeterministic branching and provides a natural notion of equivalence for higher-order processes via bisimulation. Importantly, many computational features can be captured by universal constructions on presheaves which justifies the claim that the presheaf semantics may provide a broadly applicable theory of concurrent computation. This work also led to the development of a simple semantics by Nygaard and Winskel[20] where presheaves are replaced by sets of computation paths. Roughly, this semantics describes *whether* a process can perform a particular path, whereas the presheaf semantics describes *how* the path can be performed. The path semantics is therefore much coarser, but at the same time it is significantly simpler and more familiar to conventional domain theorists. Intuitions developed in the path semantics can sometimes be translated into innovations in the presheaf semantics.

A notable omission from this semantic model is that it does not support the notion of *name generation*. In the π-calculus[16], for example, names may be used to identify communication channels between processes. They may be passed from process to process to model the mobility of channels; and importantly they may be hidden to model the restriction of scope or, from another viewpoint, a process may create a new channel, complete with a unique freshly-generated identifier, at will. The original semantics of the π-calculus was given operationally, but more recent work has developed denotational models too[4, 8].

Names also crop up in the study of syntax, in the guise of placeholders for variables. A free variable in a term may be modelled by a name. Like channel names in the π-calculus it may be passed around to other parts of a term, and hidden by a binder. When performing a proof by induction over the structure of a term that contains bound variables, it is common to unbind them and assert that they are "chosen to be different" from the free variables that are already known, effectively generating fresh new names at will. This practice is normally justified by an informal assertion such as Barendregt's Variable Convention[2].

Although this is acceptable practice when human beings are manipulating syntax, it is necessary to provide a more formal explanation to allow computers to do the same manipulations. Shinwell[27, Section 1.1] and Gabbay[9, Section 33] summarise some of the approaches that have been developed and concentrate — as does this discussion — on a recently developed method based on a non-standard set theory: the theory of nominal sets as pioneered by Pitts[10]. Nominal set theory captures the common manipulations applied to names, such as binding and unbinding, in a very natural fashion and otherwise behaves much like more familiar set theories such as ZF. This is important, because dealing with names can occasionally throw up some subtleties that might be hard to pin down in a more cumbersome setting. The similarity with standard set theories is also valuable because it makes it straightforward to transfer intuitions developed in a name-free setting into the nominal setting.

Importantly, the operations of binding and generating names in nominal set theory are generalisations of — rather than dependent on — the corresponding syntactic notions. It would therefore appear plausible that nominal set theory is a suitable setting for studying the semantic as well as syntactic uses of names, such as for channel identifiers in the π-calculus. It is this observation which motivates this thesis, the aims of which are

1. to demonstrate that the nominal set theory of Pitts *et al.* can be used to *systematically* adjoin name generation to a conventional model for concurrency, and

2. to demonstrate that the path-based domain theory for concurrency of Winskel and Nygaard can be *systematically* extended with name generation.

To help with the navigation of this document the contents of each chapter are summarised below.

Chapter 2 sets out the mathematical preliminaries of the discussion, in order to fix a consistent notation and nomenclature for the rest of the dissertation. As such, it mostly consists of definitions, discussions and results that are well-known and published elsewhere.

Chapter 3 develops the promised domain theory for nondeterministic processes with names. Roughly speaking, this development takes the construction of the path-based domain theory for concurrency mentioned above and follows a parallel road within the theory of nominal sets. Importantly, all of the constructions are given by means of universal properties, which supports the claim that this work points towards a general framework for concurrency.

The path through the dissertation diverges at this point and it is possible to choose whether next to read chapters 4 and then 5, or else to read chapter 6.

Chapter 4 introduces an expressive process calculus, Nominal HOPLA, which can be used to illustrate the domain theory of chapter 3. Nominal HOPLA is closely related to the language HOPLA (a Higher-Order Process LAnguage)[20] and is inspired by the language new-HOPLA[38]. The content of chapter 4 requires only a little nominal set theory and no domain theory to appreciate, since it concentrates on a syntactic and operational description of the process calculus.

Chapter 5 links the operational semantics of chapter 4 with the domain theory of chapter 3 by giving a denotational semantics to Nominal HOPLA in terms of universal constructions, and then proving soundness and adequacy results that closely link the two styles of semantics.

Chapter 6, which depends only on chapters 2 and 3, takes a more abstract approach and develops some categorical foundations for the domain theory of chapter 3, providing support for the claim that this work is an example of a systematic procedure to incorporate name generation into a semantic model.

Finally, chapter 7 discusses a number of possible future avenues of enquiry that this thesis has made available.

Chapter 2

Preliminaries

This chapter sets out some of the mathematical prerequisites for the remainder of this dissertation. As such, it mostly consists of definitions, discussions and results that are well-known and published elsewhere, but which are collected here so that it is possible to make use of them with consistent nomenclature and notation.

Notable exceptions to this rule include lemma 2.2.1.6 and the non-implications of lemma 2.2.6.2, both of which are original work by the author. Lemma 2.2.1.6 in particular is a key insight that makes it possible to characterise the domain-theoretic notion of *isolated* elements (also known elsewhere as a *compact* or *finite* elements) within the non-standard nominal set theory, as demonstrated in lemma 3.4.3.5.

This chapter falls into two independent halves: section 2.1 introduces denotational semantics and elementary domain theory, and section 2.2 introduces the theory of nominal sets.

2.1 Domain Theory

Denotational semantics is the branch of computer science that gives meaning to pieces of syntax by assigning to each program P a mathematical object $[\![P]\!]$ that P is said to *denote*, and whose mathematical properties correspond to the computational properties of P. Domain theory is an important branch of denotational semantics, using mathematical structures called *domains* as universes of denotations. Its origins lie in work in the 1960s when Scott was searching for a setting for denotational semantics as pioneered by the work of Strachey.

Stoy[31] and Amadio and Curien[1] provide good accounts of the subject. A consequence of this work was the development of a denotational semantics for the untyped λ-calculus. This was an important development since it is clear that it is not possible to use the 'obvious' semantics where each λ-term denotes a total function, because such a semantics would have required a set D such that $D \cong D \to D$ where $\to$ is the usual set-theoretic function space. A simple cardinality argument demonstrates that the only solutions to this equation are trivial. A way around this problem was effectively to permit the model to contain 'partial' elements, corresponding to partially-defined functions, and to equip the set D with extra structure that captures the relationships between these partial elements in terms of continuous operations with respect to an appropriate topology. The resulting structured sets are generally all called domains, although this word has a variety of subtly different definitions depending on the detailed properties of the denotational semantics under study.

This section gives an overview of elementary domain theory, but much deeper treatments are available elsewhere[23, 34]. Firstly section 2.1.1 introduces two of the more common basic notions of domain, namely CPOs and ωCPOs, and the appropriate morphisms of domains that preserve their structure. Then section 2.1.2 highlights a particularly useful class of *algebraic* domains which can be obtained as the completion of a particular class of simple *isolated* elements. Section 2.1.3 discusses a number of general approaches to giving a domain theory for *nondeterministic* processes whose computational behaviour is not totally specified and which may therefore make choices at certain points. Nondeterminism can be used to describe *concurrency*, where more than one computation can be performed at once and the interactions between such computations are not necessarily precisely specified. Section 2.1.4 introduces a particularly simple domain theory for concurrency and finally section 2.1.5 gives an intuition for this domain theory in terms of the computation paths that processes can follow.

2.1.1 Complete Partial Orders and Continuous Functions

A *domain* — in the sense of domain theory — is an ordered structure with some appropriate completeness property. The elements of a domain can be viewed as representing computations, and the order on a domain represents increasing computational information, information which can be represented as a limit of approximations. A simple notion of approximation is by sequences of the form

$$d_0 \sqsubseteq d_1 \sqsubseteq \ldots \sqsubseteq d_n \sqsubseteq \ldots . \tag{2.1.1.1}$$

This gives rise to the following candidate for a suitable notion of domain.

2.1.1.2 Definition. *An ω-**complete partial order (ωCPO)** is a partial order $\langle D, \sqsubseteq_D \rangle$ that has joins of all increasing ω-chains in D.*

A more general notion of convergence is to consider approximation by directed sets (which, to avoid any ambiguity, here does not include $\varnothing$.)

2.1.1.3 Definition. *A **complete partial order (CPO)** is a partial order $\langle D, \sqsubseteq_D \rangle$ that has joins of all directed subsets of D.*

Whichever completeness property is chosen, it is chosen to capture an appropriate notion of approximation of elements of D by some kind of limiting process of the appropriate shape. It is important here to emphasise that the choice of a suitable notion of approximation is a key variable in the development of domain theory, so it is worth making the following definition.

2.1.1.4 Definition. *Write Φ_d for the property of directedness and Φ_ω for the property of being an ω-chain.*

Φ_d and Φ_ω capture the two notions of approximation that have been introduced so far. It turns out that elementary domain theory is somewhat independent of the choice of any particular notion of approximation, in which case the chosen notion of approximation can be written simply as Φ. This generality is important when working in the theory of nominal sets (the setting for much of the rest of this dissertation) since it helps to capture the salient features of the development of domain theory that are independent of any particular notions of approximation. Indeed, neither Φ_d nor Φ_ω fully capture an appropriate notion of approximation in the theory of nominal sets and it is necessary to use a subtly different definition instead. However, it is sufficient to think of Φ as being either Φ_d or Φ_ω in the following, and when introducing approximations in nominal set theory in section 3.4 the mathematics is developed from scratch.

2.1.1.5 Definition. *A Φ-**complete partial order (ΦCPO)** is a partial order $\langle D, \sqsubseteq_D \rangle$ that has joins of subsets of D that have the property Φ.*

The property Φ also gives rise to a notion of Φ-continuous function.

2.1.1.6 Definition.

$$f : \langle D, \sqsubseteq_D \rangle \to \langle E, \sqsubseteq_E \rangle \tag{2.1.1.7}$$

*is Φ-**continuous** if it is a function $f : D \to E$ that preserves*

- *the ordering $\sqsubseteq$ (i.e. it is monotone)*
- *the property of Φ-ness (i.e. if $x \subseteq D$ is Φ then so is $\{fd \mid d \in x\}$), and*
- *joins of Φ subsets of D (i.e. if $x \subseteq D$ is Φ then $f\big(\bigvee x\big) = \bigvee\{fd \mid d \in x\}$.*

The collection of ΦCPOs and Φ-continuous functions form a category **ΦCPO**. If $\Phi \in \{\Phi_d, \Phi_\omega\}$ then the category **ΦCPO** is cartesian closed. The cartesian product is given by the product of the underlying sets ordered componentwise, and the exponential is given by the space of Φ-continuous functions with the pointwise order. This is important for giving denotational semantics to a language with higher-order functions such as the λ-calculus.

2.1.1.8 Definition. *If $\mathbb{P}$ is a preorder and $x \subseteq \mathbb{P}$ then define*

$$x_\downarrow =_{\mathrm{def}} \{p' \mid \exists p \in x.p' \leq_{\mathbb{P}} p\}.$$

*Say that $x' \subseteq \mathbb{P}$ is a **lower set in** $\mathbb{P}$ if it is of the form $x_\downarrow$ for some $x \subseteq \mathbb{P}$.*

2.1.1.9 Definition. *A Φ-**ideal** in the preorder $\mathbb{P}$ is the lower set of a Φ subset of $\mathbb{P}$. In other words, it is a set of the form $x_\downarrow$ where $x \subseteq \mathbb{P}$ has the property Φ. The Φ-**ideal completion** of $\mathbb{P}$, written $\mathrm{Idl}_\Phi(\mathbb{P})$, is the set of all Φ-ideals of $\mathbb{P}$ ordered by inclusion.*

2.1.1.10 Lemma. *The Φ_d-ideal completion (respectively the Φ_ω-ideal completion) of a preorder $\mathbb{P}$ is a Φ_dCPO (respectively a Φ_ωCPO) with joins given by union.*

Proof. It is only hard to see that the union of an ω-chain of Φ_ω-ideals is itself a Φ_ω-ideal. Let $x_0 \subseteq x_1 \subseteq \ldots$ be an ω-chain of Φ_ω-ideals of $\mathbb{P}$. Suppose that each x_i is of the form $\{p_{i,0} \leq_{\mathbb{P}} p_{i,1} \leq_{\mathbb{P}} \ldots\}_\downarrow$. For each $i \in \omega$, define $j(i)$ to be the least j such that $p_{k,l} \leq_{\mathbb{P}} p_{i,j}$ for all $k, l \leq i$. Such a j certainly exists, for if $k, l \leq i$ then $x_k \subseteq x_i$ and hence $p_{k,l} \in x_i$ from which it follows that there exists j such that $p_{k,l} \leq p_{i,j}$; there are only finitely many $k, l \leq i$. Now it is the case that $\bigcup_{i \in \omega} x_i = \{p_{0,j(0)} \leq_{\mathbb{P}} p_{1,j(1)} \leq_{\mathbb{P}} \ldots\}_\downarrow$, for if $p \in \bigcup_{i \in \omega} x_i$ then $p \in x_k$ for some k and hence there exists l such that $p \leq_{\mathbb{P}} p_{k,l} \leq_{\mathbb{P}} p_{m,j(m)}$ where $m = \max(k, l)$ by definition of $j(m)$ as required. □

2.1.2 Algebraic CPOs

Certain elements of a ΦCPO D cannot be reached by the join of a succession of increasingly close proper approximations and such elements are called *isolated*. Computationally, the isolated elements of D typically correspond to those computations that can be realised in finite time. The terms *compact* and *finite* are also used in the literature, but this dissertation consistently uses the term *isolated* which is defined as follows.

2.1.2.1 Definition. *An element d of the ΦCPO $\langle D, \sqsubseteq_D \rangle$ is **Φ-isolated** if whenever $d \sqsubseteq_D \bigvee x$ for some $x \subseteq D$ satisfying Φ it follows that there exists $d' \in x$ such that $d \sqsubseteq_D d'$. The collection of all Φ-isolated elements of D is written D°.*

If a ΦCPO consists of elements that can all be approximated by its Φ-isolated elements then its structure is considerably simplified. Those ΦCPOs that consist only of such approximable elements are called *algebraic*:

2.1.2.2 Definition. *A ΦCPO D is **algebraic** if for every $d \in D$ there exists an $x \subseteq D^\circ$ which is both Φ and such that $d = \bigvee x$.*

Sometimes algebraicity also includes a constraint on the size of D — for example it may insist that that D° is countable — but this point is not dwelt upon here. Indeed, much of this dissertation works within the theory of nominal sets, and in this theory the notion of cardinality is subtle because of the failure of the Axiom of Choice. For example, lemma 2.2.6.2 demonstrates that even the idea of finiteness is delicate.

Importantly, the isolated elements of an algebraic ΦCPO determine its structure precisely, as the following lemma shows.

2.1.2.3 Lemma. *If $\langle D, \sqsubseteq_D \rangle$ is an algebraic ΦCPO then $D \cong \mathrm{Idl}_\Phi(D^\circ)$.*

Proof. If $d \in D$ then by the algebraicity of D there exists $x \subseteq D^\circ$ which is Φ and such that $d = \bigvee x$, so define $f(d) = x_\downarrow \in \mathrm{Idl}_\Phi(D^\circ)$. To see that this is well-defined let $x' \subseteq D^\circ$ be Φ and such that $d = \bigvee x'$. If $d_1 \in x'_\downarrow$ then there exists $d_2 \in x'$ such that $d_1 \sqsubseteq_D d_2$, but $d = \bigvee x'$ so that $d_1 \sqsubseteq_D d_2 \sqsubseteq d = \bigvee x$ and d_1 is Φ-isolated so there exists $d_3 \in x$ such that $d_1 \sqsubseteq_D d_3$ and hence $d_1 \in x_\downarrow$. This shows that $x'_\downarrow \subseteq x_\downarrow$ and the converse is similar.

If $I \in \mathrm{Idl}_\Phi(D^\circ)$ then by definition there exists $x \subseteq D^\circ$ which is Φ and such that $I = x_\downarrow$, so define $g(I) = \bigvee x$. To see that this is well-defined let $x' \subseteq D^\circ$ be Φ and such that $I = x'_\downarrow$. If $d \in x \subseteq I = x'_\downarrow$ then there exists $d' \in x'$ such that $d \sqsubseteq_D d'$, so that $\bigvee x \sqsubseteq_D \bigvee x'$ and the converse is similar. It is now straightforward to show that f and g define an isomorphism $D \cong \mathrm{Idl}_\Phi(D^\circ)$ as required. $\square$

Also if $f : D \to E$ is Φ-continuous and D is algebraic then f is entirely specified by its action on D° as the following lemma shows.

2.1.2.4 Lemma. *Let $\langle D, \sqsubseteq_D\rangle$ be an algebraic ΦCPO and let*

$$f, g : \langle D, \sqsubseteq_D\rangle \rightrightarrows \langle E, \sqsubseteq_E\rangle$$

be two arrows of **ΦCPO** *such that $f(d) = g(d)$ for every $d \in D^\circ$. Then $f = g$.*

Proof. Let $d \in D$, then by the algebraicity of D there exists $x \subseteq D^\circ$ which is both Φ and such that $d = \bigvee x$. Therefore

$$\begin{aligned} f(d) &= f\big(\textstyle\bigvee x\big) && && (2.1.2.5)\\ &= \textstyle\bigvee\{f(d') \mid d' \in x\} && \text{by continuity of } f\\ &= \textstyle\bigvee\{g(d') \mid d' \in x\} && \text{since } f \text{ and } g \text{ agree on } D^\circ \supseteq x\\ &= g\big(\textstyle\bigvee x\big) && \text{by continuity of } g\\ &= g(d) \end{aligned}$$

as required. $\square$

The collection of algebraic ΦCPOs and Φ-continuous functions forms a category **ΦAlg** and much of the structure of **ΦCPO** restricts to **ΦAlg**. For example, the cartesian product of a pair of algebraic ΦCPOs (as objects of **ΦCPO**) is again an algebraic ΦCPO. However if D and E are Φ-algebraic then it is not necessarily the case that the function space $D \to E$ is Φ-algebraic, and neither $\mathbf{\Phi_d Alg}$ nor $\mathbf{\Phi_\omega Alg}$ is cartesian closed. Fortunately **ΦAlg** may contain a cartesian closed subcategory that is a suitable setting for denotational semantics. In $\mathbf{\Phi_d Alg}$ one such setting is the collection of Scott domains, which are those algebraic Φ_dCPOs where every *bounded* subset has a join. Another is the collection of *strongly finite* or SFP objects[24].

A strengthening of the concept of isolation is that of primality. An element in a ΦCPO is (completely) prime if it cannot be properly decomposed as a join of any shape. The associated concept of algebraicity is that every element of the ΦCPO can be approximated entirely by primes, as follows.

2.1.2.6 Definition. *An element d of the ΦCPO $\langle D, \sqsubseteq_D\rangle$ is* ***completely prime*** *if whenever $d \sqsubseteq_D \bigvee x$ for any $x \subseteq D$ for which $\bigvee x$ exists it follows that there exists $d' \in x$ such that $d \sqsubseteq_D d'$. A ΦCPO $\langle D, \sqsubseteq_D\rangle$ is* ***prime algebraic*** *if for every $d \in D$ there exists $x \subseteq \{d' \in D \mid\ d' \text{ completely prime}\}$ such that $d = \bigvee x$.*

Sometimes completely prime elements are called simply ‘prime’.

2.1.3 Nondeterminism in Domain Theory

In order to capture a denotational semantics for nondeterministic choice, it is possible to extend the notion of a ΦCPO with a binary operation $\sqcup$ representing the nondeterministic sum of a pair of computations. There are a number of possible axiomatisations of the operation $\sqcup$ depending on the style of nondeterminism that is under study.

2.1.3.1 Definition. *A **Plotkin-nondeterministic** Φ**CPO** ($N_P\Phi CPO$) is a ΦCPO $\langle D, \sqsubseteq_D\rangle$ together with an idempotent, commutative and associative binary operation $\sqcup_D : D \times D \to D$ that is Φ-continuous in each argument.*

*A **Hoare-nondeterministic** Φ**CPO** ($N_H\Phi CPO$) is a Plotkin-nondeterministic ΦCPO $\langle D, \sqsubseteq_D, \sqcup_D\rangle$ such that additionally $d \sqsubseteq_D d \sqcup_D d'$ for all $d, d' \in D$.*

*A **Smyth-nondeterministic** Φ**CPO** ($N_S\Phi CPO$) is a Plotkin-nondeterministic ΦCPO $\langle D, \sqsubseteq_D, \sqcup_D\rangle$ such that additionally $d \sqcup_D d' \sqsubseteq_D d$ for all $d, d' \in D$.*

Intuitively, the Hoare-style nondeterminism captures a ‘may do’ semantics: a nondeterministic sum of two computations contains at least as much information as each of the summands, because it may produce the output of either. Conversely, the Smyth-style nondeterminism captures a ‘must do’ semantics: fewer guarantees can be made about the nondeterministic sum of two computations, so the sum contains less information. The Plotkin-style nondeterminism captures a mixture of both the ‘may do’ and ‘must do’ styles of semantics. When the discussion is not concerned with the differences between the styles of nondeterminism, it is enough to refer to $\mathrm{N}_P\Phi$CPOs, $\mathrm{N}_H\Phi$CPOs and $\mathrm{N}_S\Phi$CPOs simply as NΦCPOs.

The natural morphisms between NΦCPOs are Φ-continuous functions that are additionally $\sqcup$-homomorphisms, and these objects and morphisms form a category **NΦCPO** with the obvious forgetful functor $U : \mathbf{N\Phi CPO} \to \mathbf{\Phi CPO}$. Furthermore this functor has a left adjoint F associating to each ΦCPO D the object FD which is the free NΦCPO on D. The composition $UF : \mathbf{\Phi CPO} \to \mathbf{\Phi CPO}$ is usually called the *powerdomain monad* and is often written simply $\mathcal{P}$.

A similar story unfolds when attention is restricted to algebraic ΦCPOs. Nondeterministic algebraic ΦCPOs are defined similarly to more general nondeterministic CPOs, and together with Φ-continuous $\sqcup$-homomorphisms they form a category **NΦAlg** with a forgetful/free adjunction $F \dashv U : \mathbf{N\Phi Alg} \leftrightarrows \mathbf{\Phi Alg}$ as before. An advantage of restricting attention to algebraic ΦCPOs is that the powerdomain monad becomes easy to describe concretely in this setting. For example, if D is an algebraic Φ_dCPO then there is an associated preorder $!(D^\circ)$ whose elements comprise the finite subsets of D° and whose order is defined by setting $x \leq_{!(D^\circ)} x'$ iff for all $d \in x$ there exists $d' \in x'$ such that $d \sqsubseteq_D d'$. The Hoare powerdomain $\mathcal{P}_H D$ is then given as the Φ_d-ideal completion of $!(D^\circ)$. Alternatively, $\mathcal{P}_H D$ can be characterised as the collection of lower subsets of D° ordered by inclusion.

2.1.4 A Simple Domain Theory for Concurrency

Dropping some of the generality developed above, in the following consider just the notion of approximation given by directedness and just the Hoare 'may do' style of nondeterminism. Viewing nondeterminism as a computational effect (in the style of Moggi[18]) draws attention to the Kleisli category of $\mathcal{P}_H$ which can be more simply characterised as the category **PreRel** of preorders and monotone relations as follows.

2.1.4.1 Definition. *The objects of the category* **PreRel** *are preorders* $\mathbb{P}, \mathbb{Q}, \ldots$. *Its arrows* $R : \mathbb{P} \to \mathbb{Q}$ *are relations* $R \subseteq \mathbb{P} \times \mathbb{Q}$ *such that if* $p' \geq_{\mathbb{P}} p \; R \; q \geq_{\mathbb{Q}} q'$ *then* $p' \; R \; q'$. *Composition in* **PreRel** *is straightforward relational composition, and the identity relation on the preorder* $\mathbb{P}$ *is simply* $\geq_{\mathbb{P}}$.

2.1.4.2 Proposition. *If* $\mathcal{P}_H : \mathbf{\Phi_d Alg} \to \mathbf{\Phi_d Alg}$ *is the Hoare powerdomain functor described above then its Kleisli category* $\mathrm{Kl}(\mathcal{P}_H)$ *is equivalent to* **PreRel**.

Sketch Proof (Object-parts only). The equivalence comprises the functors $(-)^\circ : \mathrm{Kl}(\mathcal{P}_H) \to \mathbf{PreRel}$ and $\mathrm{Idl}_{\Phi_d} : \mathbf{PreRel} \to \mathrm{Kl}(\mathcal{P}_H)$. Lemma 2.1.2.3 demonstrates that if D is an algebraic Φ_dCPO then $D \cong \mathrm{Idl}_{\Phi_d}(D^\circ)$. For the converse, suppose that $\mathbb{P}$ is a preorder and consider elements $x \in (\mathrm{Idl}_{\Phi_d}(\mathbb{P}))^\circ$. Since $x \in \mathrm{Idl}_{\Phi_d}(\mathbb{P})$, there exists a directed $x_0 \subseteq \mathbb{P}$ such that $x = x_{0\downarrow}$ and hence $x = \bigcup_{p \in x_0} \{p\}_\downarrow$. Also since x_0 is directed in $\mathbb{P}$ it follows that $\{\{p\}_\downarrow \mid p \in x_0\}$ is directed in $\mathrm{Idl}_{\Phi_d}(\mathbb{P})$, and since x is isolated there must exist $p \in x_0$ such that $x_0 = \{p\}_\downarrow$. Thus $\mathbb{P}$ and $(\mathrm{Idl}_{\Phi_d}(\mathbb{P}))^\circ$ are isomorphic in **PreRel**. □

The use of preorders in the domain theory — rather than the more usual partial orders — does not make a significant difference to much of the theory, but it does enable an intuitive method for solving simple recursive domain equations[20].

Concretely the equivalence $\mathrm{Kl}(\mathcal{P}_H) \simeq \mathbf{PreRel}$ factorises the monad $\mathcal{P}_H$ as

$$\begin{tikzcd}
\mathbf{\Phi}_d\mathbf{Alg} \arrow[r, bend left, "!((-)^\circ)"] \arrow[r, phantom, "\perp"] & \mathbf{PreRel} \arrow[l, bend left, "\mathrm{Idl}_{\Phi_d}"]
\end{tikzcd} \tag{2.1.4.3}$$

where the functor Idl_{Φ_d} takes a preorder to its Φ_d-ideal completion and $!((-)^\circ)$ takes an algebraic domain to the preorder of finite sets of its Φ_d-isolated elements (with the order as described above).

The category $\mathrm{Kl}(\mathcal{P}_H)$ is also equivalent to the category **Lin** of preorders and join preserving (or *linear*) maps.

2.1.4.4 Definition. *The objects of* **Lin** *are preorders* $\mathbb{P}, \mathbb{Q}, \ldots$. *Its arrows* $f : \mathbb{P} \underset{\mathbf{L}}{\rightarrow} \mathbb{Q}$ *of* **Lin** *are functions* $f : \widehat{\mathbb{P}} \rightarrow \widehat{\mathbb{Q}}$ *which preserves all joins, where* $\widehat{\mathbb{P}}$ *is the free join-completion of* $\mathbb{P}$. *Concretely,* $\widehat{\mathbb{P}}$ *consists of all lower subsets of* $\mathbb{P}$ *ordered by inclusion.*

2.1.4.5 Proposition. *The category* **PreRel** *is isomorphic to the category* **Lin** *of preorders and join preserving (or linear) maps.*

Sketch Proof. The isomorphism acts as the identity on objects. If $R : \mathbb{P} \rightarrow \mathbb{Q}$ is an arrow of **PreRel** then the corresponding arrow of **Lin** is FR where if $x \in \widehat{\mathbb{P}}$ then $FRx = \{q \in \mathbb{Q} \mid \exists p \in x.p\ R\ q\}$. Conversely if $f : \mathbb{P} \rightarrow \mathbb{Q}$ is an arrow of **Lin** then the corresponding arrow of **PreRel** is Gf where if $p \in \mathbb{P}$ and $q \in \mathbb{Q}$ then $p\ Gf\ q$ iff $q \in f\{p\}_{\downarrow}$. It is straightforward to show that F and G are mutual inverses. □

Lin can be understood as a categorical model of Girard's linear logic[20]. From this viewpoint, the arrows of **Lin** correspond to *linear* processes which do not either duplicate or discard their inputs. This linearity is an important aspect of concurrency: in a distributed computation it may be difficult to copy processes. Copying may not be impossible, however, and in general it is too restrictive to constrain attention just to linear maps. Following the discipline of linear logic, copying and discarding may be controlled explicitly by considering maps whose domain is under an exponential which corresponds to a comonad on the underlying category. The adjunction $!((-)^{\circ}) \dashv \mathrm{Idl}_{\Phi_d}$ above draws attention to the comonad $!((\mathrm{Idl}_{\Phi_d}(-))^{\circ}) \cong\ !$ on **Lin**.

2.1.5 Domains of Paths

The semantic setting discussed above can be approached more directly by taking the intuition that a nondeterministic process denotes the set of computation paths that it may follow. A brief overview of this approach is below, but much more detail is available elsewhere[20]. The objects $\mathbb{P}$, $\mathbb{Q}$, ... of **Lin** take the role of domains of paths (ordered by extension) which can be used as a type system for nondeterministic processes. More precisely, nondeterministic processes of type $\mathbb{P}$ denote subsets of $\mathbb{P}$. Furthermore, within the Hoare-style 'may do' semantics if a process can perform some path then it can also perform any shorter path, so that processes of type $\mathbb{P}$ denote *lower* subsets of $\mathbb{P}$. In other words $\widehat{\mathbb{P}}$ can be thought of as a domain of meanings for processes of type $\mathbb{P}$.

As $\widehat{(-)}$ takes a preorder to its free join-completion, it is natural to consider the join-preserving (i.e. linear) maps between freely join-completed structures, but the linear maps are too restrictive a class of maps for a rich domain theory. In particular, since they preserve all joins they preserve the empty join, so that a process that receives no input cannot spontaneously perform any output. It is better to turn attention to a collection of maps that preserve some smaller class of joins: perhaps nonempty joins or perhaps only directed joins. Nonempty-join-preserving maps are also known as *affine* and form a category **Aff**, whereas directed-join-preserving maps could be called *continuous* and form a category **Cts**.

From a different viewpoint it turns out that **Lin** can be used as a setting for Girard's classical linear logic, and following the discipline of linear logic it would seem that the solution to the strictness of linearity is to consider those maps whose domain is under an exponential. **Aff** can be seen to be the coKleisli category of the comonad $(-)_\perp$: **Lin** → **Lin** which takes a preorder $\mathbb{P}$ to the order $\mathbb{P}_\perp = \mathbb{P} \uplus \{\perp\}$ where the new element $\perp$ is ordered below each element of $\mathbb{P}$. **Aff** then forms a model of affine-linear logic[12]. On the other hand, **Cts** can be seen to be the coKleisli category of the comonad ! : **Lin** → **Lin** introduced in the previous section. **Cts** then forms a model of multiplicative-exponential linear logic[3], with ! taking the role of the exponential.

It is in the setting of **Cts** that Winskel and Nygaard[20] have developed a simple domain theory that captures certain important computational features. In detail,

- **Cts** is cartesian closed and therefore supports higher-order computations.
- Its hom-sets are richly equipped with joins and therefore **Cts** supports recursively-defined processes and nondeterministic sums.
- The operation ! gives rise to a primitive observable action. A richer structure of actions is supported by means of labelling.

Perhaps of the greatest importance is that all of this structure is given by universal properties, rather than by *ad hoc* definition. Moreover, Winskel and Nygaard are led by these universal properties to the expressive metalanguage HOPLA, so named because it is a Higher-Order Process LAnguage. They demonstrate that HOPLA can simulate well-known process calculi such as higher-order CCS, and because of its denotational underpinnings it is claimed that this approach points the way towards more universal models of concurrency. That said, the domain theory based in **Cts** is no panacea. In particular because the semantics is based on a simple-minded notion of paths it equates processes that sometimes would be best kept separate. For example, the terms $t_1 =_{\mathrm{def}} a.a.\mathtt{nil}$ and

$t_2 =_{\mathrm{def}} a.\mathtt{nil}+a.a.\mathtt{nil}$ (in a CCS-like process calculus) both denote the path set $\{\epsilon, a, a.a\}$ despite the fact that t_1 and t_2 are not bisimilar. A possible remedy is as follows: $\widehat{\mathbb{P}}$ can be seen as the space of characteristic functions $\mathbb{P}^{\mathrm{op}} \to \mathbf{2}$ where $\mathbf{2}$ is the nontrivial poset on the two-element set $\{\top, \bot\}$ of truth values. Intuitively, $\top$ and $\bot$ correspond to *whether* a path was realised or not. Replacing $\mathbf{2}$ with some larger collection of truth values opens the way to thinking about *how* a path may be realised. In the case of t_2 above, the path a could be realised in two different ways, one for each component of the sum, whereas t_1 can only realise a in one way. Taking this idea sufficiently far suggests that $\mathbf{2}$ could be replaced by the category **Set**; viewing each $\mathbb{P}$ as a small category then gives a domain theory in terms of presheaves $\mathbb{P}^{\mathrm{op}} \to \mathbf{Set}$ with much finer-grained distinctions than is possible with the coarser path semantics. In fact, Cattani and Winskel[5] developed this presheaf-based semantics first, which motivated Nygaard and Winskel[19, 20] to explore the simpler path semantics.

Also absent from the path-based semantics in **Cts** is the concept of name-generation. In order to model a modern fully-fledged process algebra such as the π-calculus it is necessary to be able to model locally-scoped names and treat them as first-class data that can, for example, be passed from process to process. Winskel and Zappa Nardelli[38] developed the language new-HOPLA in which this was possible. This language was based on a sketch of a denotational semantics given in the functor category $\mathbf{Lin}^{\mathbb{I}}$ — where $\mathbb{I}$ is the category of injections between finite sets of names — following the ideas of Stark[28], Moggi[17], Fiore and Sangiorgi[8] and Oles[21] amongst others. The use of $\mathbf{Lin}^{\mathbb{I}}$ rather than **Lin** facilitates a semantics that is parametric in the 'current' set of names. However there are some technical problems in the associated domain theory, particularly regarding the existence of function spaces. This thesis aims to overcome such problems by avoiding the technical machinery of functor categories and working in a more elementary setting. The following section introduces the theory of nominal sets, which provides a suitable situation for a name-theoretic domain theory for concurrency, which ultimately leads the way to a metalanguage, dubbed Nominal HOPLA, in the same way that constructions in **Cts** and $\mathbf{Lin}^{\mathbb{I}}$ led respectively to HOPLA and new-HOPLA.

2.2 Nominal Sets

The syntactic principle of 'binding' of a variable is easy to understand. Every algebra student would agree that the theorems

$$\forall x.(5x + 2 = 12 \Rightarrow x = 2) \quad \text{and} \quad \forall y.(5y + 2 = 12 \Rightarrow y = 2)$$

express precisely the same concept, as the symbols x and y are just used to name something to which a later reference needs to be made. In some sense, it seems to be an innate ability to give something a name and later refer to that 'something' using just its name. Furthermore, such a name may only make sense in a particular context.

This concept can be seen to be surprisingly subtle when working fully formally. Computers (and the formal systems that they implement) have no such innate ability to deal with locally-scoped names, so this ability must be carefully implemented by hand. Over the years a variety of ways to deal with locally-scoped names have been created. Shinwell[27, Section 1.1] and Gabbay[9, Section 33] summarise some of these approaches and concentrate — as does this discussion — on a recently developed method based on a non-standard set theory: the theory of nominal sets as pioneered by Pitts[10]. This theory is a powerful one in the context of this thesis because it gives a purely semantic account of the idea of a bound name, which makes it possible to give a natural account of a compositional denotational semantics incorporating binding. For example, the denotation of the π-calculus term $\nu a.P$ should be built from the denotation of P via some kind of binding operator: the name a should appear 'free' in $[\![P]\!]$ but must not be free in $[\![\nu a.P]\!]$. The theory of nominal sets makes sense of the idea of a name appearing free in a semantic object such as $[\![P]\!]$ in a fashion that smoothly extends the syntactic notion of free names. It also allows the construction of the kind of binding operator that takes $[\![P]\!]$ to $[\![\nu a.P]\!]$. Importantly, the internal logic of the theory of nominal sets is very close to that of standard set theory, so the development of a domain theory for concurrency within nominal set theory is accessible to conventional domain theorists as the name-theoretic details can be ignored when they are not relevant.

The roots of nominal set theory lie in the study of syntax, so it is introduced here from this viewpoint. The theory rests on two key observations about names. The first is that the concrete names used by any piece of syntax are not relevant outside of their scope. The second is that any piece of syntax can only explicitly mention finitely many names.

From the first observation, the names used by any piece of syntax can be injectively renamed to some other collection of names without changing the underlying meaning of the syntax. Unlike the collection of injective renamings, the collection of bijective renamings forms a group, and the group structure simplifies the theory. The restriction of attention to bijective renamings is not a severe constraint. Also, since each piece of syntax only mentions finitely many names it is possible to turn attention to those bijective renamings that fix all but finitely many names. A few lemmas about such bijective renamings are developed in section 2.2.1, then section 2.2.2 introduces the theory of nominal sets and section 2.2.3 shows some examples and constructions for building new ones. The category of nominal sets is briefly explored in section 2.2.4 and section 2.2.5 demonstrates how the theory can be used to capture the idea of 'binding' a name. Finally, one of the more important differences between nominal and conventional set theories is the failure of the Axiom of Choice in the nominal setting, and section 2.2.6 explores the consequences of this.

2.2.1 Finite Automorphisms on Names

Let $\mathbb{A}$ be a fixed infinite set whose elements are to be called *names*. The letter $\mathbb{A}$ is chosen as names are also known as *atoms*: they are atomic in the sense that they have no internal structure. From the discussion above, the basis for the theory of nominal sets is that of a finite automorphism of $\mathbb{A}$, i.e. a bijection $\sigma : \mathbb{A} \to \mathbb{A}$ such that $\sigma a \neq a$ for only finitely many $a \in \mathbb{A}$. In particular this includes permutations of $\mathbb{A}$, which are represented as finite lists of transpositions of pairs of names $(a_1 b_1) \dots (a_n b_n)$ and whose action is defined by recursion on n by letting $\iota a =_{\text{def}} a$ where ι is the empty list, and

$$(a_1 b_1) \dots (a_n b_n)(a_{n+1} b_{n+1}) a =_{\text{def}} \begin{cases} (a_1 b_1) \dots (a_n b_n) b_{n+1} & a = a_{n+1} \\ (a_1 b_1) \dots (a_n b_n) a_{n+1} & a = b_{n+1} \\ (a_1 b_1) \dots (a_n b_n) a & \text{otherwise.} \end{cases} \tag{2.2.1.1}$$

It is well-known that any finite automorphism may be represented as a permutation; the following lemma strengthens this result by restricting the length of this permutation and the names that it permutes as much as possible.

2.2.1.2 Lemma. *Any finite automorphism $\sigma : \mathbb{A} \to \mathbb{A}$ may be represented as a sequence of transpositions of pairs of names $(a_1 b_1) \dots (a_n b_n)$ where either $\sigma = \mathbf{1}_{\mathbb{A}}$ (and $n = 0$) or $n < |\{a \in \mathbb{A} \mid \sigma a \neq a\}|$ and for each i, $\sigma a_i \neq a_i$ and $\sigma b_i \neq b_i$.*

Proof. The proof proceeds by strong induction on $|\{a \in \mathbb{A} \mid \sigma a \neq a\}|$. If $|\{a \in \mathbb{A} \mid \sigma a \neq a\}| = 0$ then $\sigma = \mathbf{1}_{\mathbb{A}}$. Suppose that $|\{a \in \mathbb{A} \mid \sigma a \neq a\}| > 0$ and let $a \in \mathbb{A}$ be such that $\sigma a \neq a$. In order to use the induction hypothesis, it is necessary to find an automorphism that moves strictly fewer names. Letting $a' = \sigma^{-1} a$ it will now be shown that $\sigma \circ (a a')$ is such an automorphism. To see this, notice firstly that $\sigma\big((a a')a\big) = a$. Furthermore, suppose that b is such that $\sigma\big((a a')b\big) \neq b$, then $\sigma b \neq b$ as follows. Clearly $b \neq a$; also if $b = a'$ then $\sigma b = a \neq b$; finally if $b \neq a'$ then $(a a')b = b$ so that $b \neq \sigma\big((a a')b\big) = \sigma b$ as required. Therefore $\{b \in \mathbb{A} \mid \sigma\big((a a')b\big) \neq b\} \subsetneq \{a \in \mathbb{A} \mid \sigma a \neq a\}$ so the induction hypothesis applies to $\sigma \circ (aa')$ to give a sequence $\vec{\tau}$ of transpositions. Note that $a' \neq a$ since $a \neq \sigma a$, so that $\vec{\tau}(aa')$ is as desired. □

It is sometimes useful to be able to reorder the list of transpositions that represent a particular permutation, which is possible as shown in the following lemma.

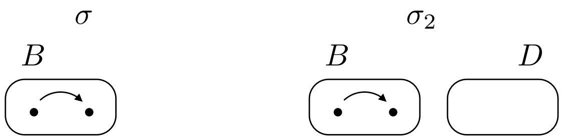
2.2.1.3 Lemma. *If σ is a permutation and a and b are names then*

$$\sigma \circ (ab) = (\sigma a \ \sigma b) \circ \sigma.$$

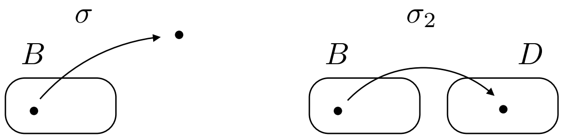
Proof. It is sufficient to show that $\sigma(ab)c = (\sigma a \ \sigma b)\sigma c$ for all names c, and for this it is sufficient to consider the three cases $c = a$, $c = b$ and $c \notin \{a, b\}$. If $c \notin \{a, b\}$ then it follows that $(ab)c = c$ and it is also the case that $\sigma c \notin \{\sigma a, \sigma b\}$ by the injectivity of σ, so that $(\sigma a \ \sigma b)\sigma c = \sigma c = \sigma(ab)c$. If $c = a$ then $(ab)c = b$ and hence $(\sigma a \ \sigma b)\sigma c = \sigma b = \sigma(ab)c$ as required. The case $c = b$ is similar. □

2.2.1.4 The Footprint Lemma. The following lemma characterises permutations in terms of their footprints on a particular set B of names. The intuition behind this lemma is as follows. Suppose that in the course of a calculation there is a bound B on the set of names that are 'known'. For example, the calculation could be a reduction of a term t in some context C and B contains the free names of t, so they cannot be permuted without affecting its context. In this example, t and C 'know' the free names. Importantly, if B contains all the names that are 'known' by a calculation then for the purposes of that particular calculation any names outside B are equivalent. Therefore it is possible to characterise a permutation σ in terms of its 'footprint' on B: the action of σ just on the names in B. Of course if $a \notin B$ but $\sigma a \in B$ then this is significant, although the precise source a of σa is not important. Similarly it is significant if $b \in B$ but $\sigma b \notin B$, although the precise destination σb of b is not. If B is finite then it is possible to specify a disjoint set D that represents a canonical collection of 'unknown' sources and destinations of the elements in B. Then from the point of view of the calculation in question the permutation σ may be represented as a permutation σ_2 such that

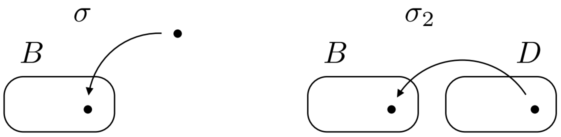
- if $b \in B$ and $\sigma b \in B$ then $\sigma_2 b = \sigma b$,



- if $b \in B$ but $\sigma b \notin B$ then $\sigma_2 b \in D$,



- if $a \notin B$ but $\sigma a \in B$ then $\sigma_2^{-1}\sigma a \in D$, and



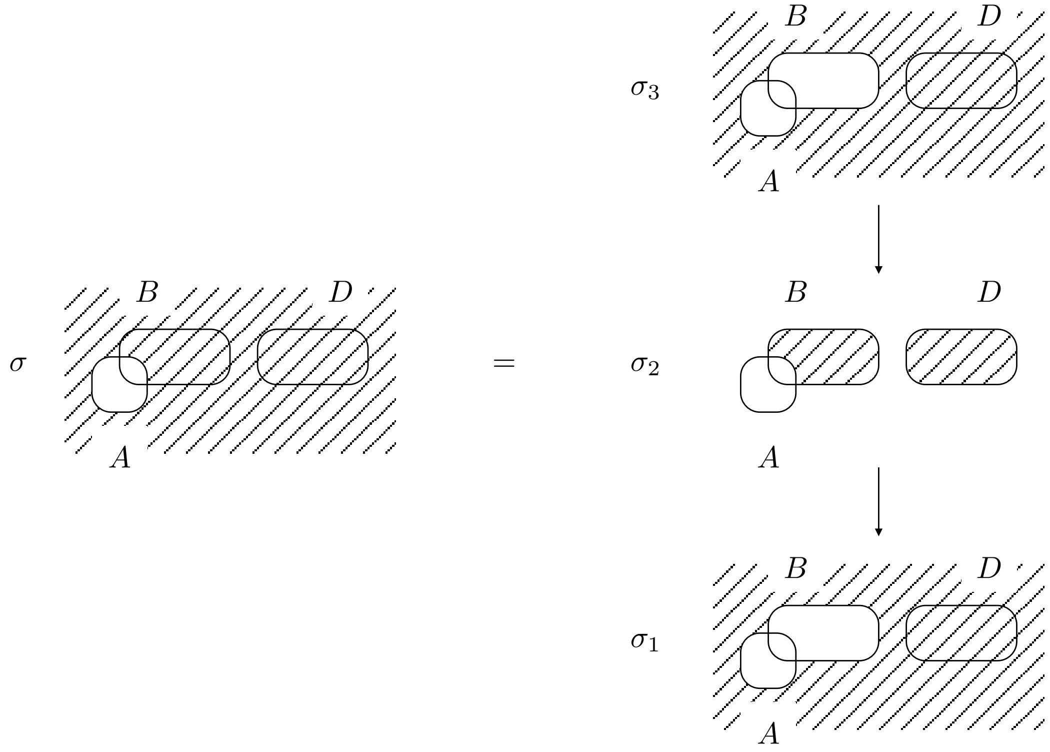
- if $e \notin B \cup D$ then $\sigma_2 e = e$.

(The symbol σ_2 is chosen for consistency with the notation in the following lemma.) Importantly a set D may be chosen that suffices for any permutation σ, and a finite such D exists, so σ_2 takes only finitely many possible values. This fact is used in the proof of 3.4.3.5.

To illustrate this, consider the simple case where $B = \{b\}$ is a singleton. For this case the choice $D = \{d\}$ for some $d \neq b$ suffices. If a permutation σ fixes b then the footprint of σ on B is represented by the identity permutation, since this is a permutation that satisfies the four properties listed above. On the other hand if $\sigma \cdot b \neq b$ then also $\sigma^{-1} \cdot b \neq b$ so that $\sigma_2 = (bd)$ represents the footprint of σ on B: it is a permutation that satisfies the four properties of a footprint representation listed above.

In this case the representing permutations were unique, but in general this is not so: if $B = \{b\}$ and D were chosen to be $\{d, d'\}$ where b, d and d' are distinct, then (bd), (bd'), (bdd') and $(bd'd)$ are all representations of a permutation σ such that $\sigma \cdot b \neq b$, and both the identity and the transposition (dd') represent any σ such that $\sigma \cdot b = b$.

The idea of representing σ in terms of a particularly constrained permutation σ_2 can be phrased in terms of a factorisation property which states that there exist σ_1, σ_2 and σ_3 such that $\sigma = \sigma_1\sigma_2\sigma_3$ and where σ_1 and σ_3 leave B fixed and σ_2 moves only $B \cup D$. The following diagram illustrates this idea: the shaded areas cover the names that may be moved. The set A in this diagram is a set of names that σ and its factors must all fix, which is a refinement of the fundamental idea that is necessary for the later application of this lemma.



Firstly it is necessary to show a well-known and elementary special case, where $B = \{b\}$ is the set of names of interest, D is chosen as the disjoint set $\{d\}$, and $\sigma = (be)$ for some fresh name e.

2.2.1.5 Proposition. *If b, d and e are distinct names then $(be) = (de)(bd)(de)$.*

Proof. It is the case that $(de)(bd) = \big((de)b\ (de)d\big)(de) = (be)(de)$ by lemma 2.2.1.3, so that $(be) = (be)(de)(de) = (de)(bd)(de)$ as required. □

The general result is now as follows.

2.2.1.6 Lemma (Footprint Lemma). *Let A, B and D be finite sets of names such that $|D| > |B|$ and such that $A \cup B$ and D are disjoint. Then for all permutations σ such that $\sigma a = a$ for all $a \in A$ there exist permutations σ_1, σ_2, σ_3 such that*

- *$\sigma = \sigma_1 \sigma_2 \sigma_3$, and*
- *$\sigma_1 a = a$ and $\sigma_3 a = a$ for all $a \in A \cup B$, and*
- *$\sigma_2 a \neq a$ only if $a \in (B \cup D) \setminus A$.*

Proof. Represent σ as a finite list of transpositions of names not in A and proceed by induction on the length of the list. If the list is empty then $\sigma = \iota$ and $\sigma_1 = \sigma_2 = \sigma_3 = \iota$ is a factorisation in the desired form. On the other hand, if the list is nonempty then write $\sigma = \sigma'(ab)$ where $a \notin A$ and $b \notin A$ and $\sigma' c = c$ for all $c \in A$. By induction there exist σ_1, σ_2 and σ_3 such that $\sigma' = \sigma_1 \sigma_2 \sigma_3$ and $\sigma_1 c = c$ and $\sigma_3 c = c$ for all $c \in A \cup B$ and $\sigma_2 c \neq c$ only if $c \in (B \cup D) \setminus A$.

If $a \notin B$ and $b \notin B$ then define $\sigma_3' = \sigma_3(ab)$. Since $\sigma_3' c = c$ for all $c \in A \cup B$, $\sigma = \sigma_1 \sigma_2 \sigma_3'$ is a factorisation in the desired form. If $a \in B$ and $b \in B$ then $\sigma_3(ab) = (ab)\sigma_3$ by lemma 2.2.1.3 so let $\sigma_2' = \sigma_2(ab)$ which has $\sigma_2' c \neq c$ only if $c \in (B \cup D) \setminus A$, so that $\sigma = \sigma_1 \sigma_2' \sigma_3$ is a factorisation in the desired form.

Otherwise without loss of generality let $a \notin B$ and $b \in B$. Then by lemma 2.2.1.3 it follows that $\sigma_3(ab) = (a'b)\sigma_3$ where $a' = \sigma_3 a$. Note that $a' \notin A \cup B$ since otherwise $a' = \sigma_3 a'$ and hence $a' = \sigma_3^{-1} a' = a \notin A \cup B$ which is a contradiction. There are two cases to consider, depending on whether $a' \in D$ or not.

If $a' \in D$ then let $\sigma_2' = \sigma_2(a'b)$. Then $\sigma_2' c \neq c$ only if $c \in (B \cup D) \setminus A$, so that $\sigma = \sigma_1 \sigma_2' \sigma_3$ is a factorisation in the desired form. On the other hand $a' \notin D$ then let $c \in D$ be such that $\sigma_2 \cdot c \in D$, which exists since by assumption $|D| > |B|$. Then $(a'b) = (a'c)(bc)(a'c)$ by proposition 2.2.1.5 so that writing $c' = \sigma_2 \cdot c \in D$ it follows that $\sigma = \sigma_1(a'c')\sigma_2(bc)(a'c)\sigma_3$ by lemma 2.2.1.3. Let $\sigma_1' = \sigma_1(a'c')$, $\sigma_2' = \sigma_2(bc)$ and $\sigma_3' = (a'c)\sigma_3$, then $\sigma = \sigma_1' \sigma_2' \sigma_3'$ is a factorisation in the desired form. □

It is sometimes useful to be able to extend an injection $i : s \rightarrowtail \mathbb{A}$ into a permutation $\sigma_i : \mathbb{A} \to \mathbb{A}$. This is always possible if s is finite, although there is never a unique such permutation.

2.2.1.7 Notation. If two sets A, B are disjoint then their union $A \cup B$ may be written as $A \mathbin{\dot\cup} B$ to emphasize its disjointness.

2.2.1.8 Lemma. *If $s \subseteq_{\text{fin}} \mathbb{A}$ and $i : s \rightarrowtail \mathbb{A}$ is an injection then there exists a finite permutation σ_i such that the restriction of σ_i to s is i.*

Proof. The proof proceeds by induction on $|s|$. If $s = \varnothing$ then any finite permutation suffices for σ_i. Now suppose that $a \notin s$ and $i : s \mathbin{\dot\cup} \{a\} \rightarrowtail \mathbb{A}$, and let $i' = i|_s : s \rightarrowtail \mathbb{A}$. By induction, there is a permutation $\sigma_{i'}$ extending i' so define $\sigma_i = (ia\ \sigma_{i'}a)\sigma_{i'}$. It is immediate that $\sigma_i a = ia$. Let $b \in s$, then $\sigma_i b = (ia\ \sigma_{i'}a)(i'b) = (ia\ \sigma_{i'}a)(ib)$. Certainly $ia \neq ib$ as i is an injection, and $ib = i'b = \sigma_{i'}b$ so that $\sigma_{i'}a \neq ib$ and hence $\sigma_i b = ib$ as required. □

2.2.2 The Theory of Nominal Sets

This section is a very short introduction to the theory of nominal sets. More details and longer discussions are available elsewhere[22].

2.2.2.1 Definition. *Write $\mathcal{G}$ for the group of finite permutations of $\mathbb{A}$.*

2.2.2.2 Definition. *A $\mathcal{G}$**-set** is a set X together with a left $\mathcal{G}$-action, written as the infix operator $\cdot_X$ or more commonly simply $\cdot$ where the set X is clear from the context.*

A $\mathcal{G}$-set is intuitively a set whose elements 'use names'. The action of a particular permutation on an element x of a $\mathcal{G}$-set is therefore the action of permuting the names that x uses. A $\mathcal{G}$-action is enough to capture the concept of 'using' names without needing to refer to syntax as follows.

2.2.2.3 Definition. *A set $s \subseteq \mathbb{A}$ is said to* ***support*** *$x \in X$ (and x is said to* ***have support*** *s) if for any permutation σ such that $\sigma a = a$ for all $a \in s$ it is the case that $\sigma \cdot x = x$.*

Intuitively s supports x if s is an upper bound for the set of names that x uses. The elements of interest use only finitely many names and in this case it is possible to find a smallest possible support.

2.2.2.4 Lemma. *If $x \in X$ has a finite support then $\bigcap\{s \subseteq_{\text{fin}} \mathbb{A} \mid s \text{ supports } x\}$ supports x.*

Proof. It is sufficient to show that if s and s' are finite supports of x then $s \cap s'$ also supports x. Let $a, b \in \mathbb{A} \setminus (s \cap s')$ and show that $(ab) \cdot x = x$ as follows. If $a, b \in \mathbb{A} \setminus s$ then $(ab) \cdot x = x$ since s supports x. Similarly if $a, b \in \mathbb{A} \setminus s'$ then $(ab) \cdot x = x$ too. Therefore without loss of generality suppose that $a \in s \setminus s'$ and $b \in s' \setminus s$. Let $c \in \mathbb{A} \setminus (s \cup s')$, then $(ac) \cdot x = x$ and $(bc) \cdot x = x$, so that $(ab) \cdot x = (ac) \cdot (bc) \cdot (ac) \cdot x = x$ as required. More generally, let σ be a permutation such that $\sigma a = a$ for all $a \in s \cap s'$, then by lemma 2.2.1.2 σ may be represented as a string of transpositions (ab) where $a, b \notin s \cap s'$. By induction, using the argument above, $\sigma \cdot x = x$ and hence $s \cap s'$ supports x as required. □

2.2.2.5 Definition. *A $\mathcal{G}$-set X is a **nominal set** if every $x \in X$ has a finite support.*

2.2.2.6 Definition. *In the light of lemma 2.2.2.4 if X is a nominal set and $x \in X$ then define* $\mathrm{supp}(x)$ *to be the smallest finite support of x, that is,*

$$\mathrm{supp}(x) =_{\mathrm{def}} \bigcap \{s \subseteq_{\mathrm{fin}} \mathbb{A} \mid s \textit{ supports } x\}.$$

2.2.2.7 Definition. *If X and Y are nominal sets and $x \in X$ and $y \in Y$ then say x **is fresh for** y and write '$x \# y$' iff* $\mathrm{supp}(x) \cap \mathrm{supp}(y) = \varnothing$.

2.2.2.8 Notation. For the remainder of this dissertation the symbol s and its relatives s', s'', s_1, s_i, etc. always refer to *finite* subsets of $\mathbb{A}$.

2.2.2.9 Lemma (Equivariance of support). *Let X be a nominal set, $x \in X$, $s \subseteq_{\mathrm{fin}} \mathbb{A}$ and $a, b \in \mathbb{A}$. Then s supports x if and only if $\{(ab) \cdot c \mid c \in s\}$ supports $(ab) \cdot x$.*

Proof. Suppose that s supports x and let σ be a permutation such that for all $c \in s$ it is the case that $\sigma \cdot (ab) \cdot c = (ab) \cdot c$. Therefore $(ab) \cdot \sigma \cdot (ab) \cdot c = c$ for all $c \in s$ so that $(ab) \cdot \sigma \cdot (ab) \cdot x = x$ since s supports x. It follows that $\sigma \cdot (ab) \cdot x = (ab) \cdot x$, and as σ was arbitrary this shows that $\{(ab) \cdot c \mid c \in s\}$ supports $(ab) \cdot x$. The converse now follows by noting that $(ab) \cdot (ab) \cdot x = x$ and $\{(ab) \cdot c \mid c \in \{(ab) \cdot d \mid d \in s\}\} = s$. □

2.2.2.10 Lemma. *Let X be a nominal set, let $x \in X$ and let $a, b \in \mathbb{A}$. If $a \in \mathrm{supp}(x)$ and $b \notin \mathrm{supp}(x)$ then $(ab) \cdot x \neq x$.*

Proof. Let $s \subseteq_{\mathrm{fin}} \mathbb{A}$ be such that s supports x and $b \notin s$, which exists by the definition of $\mathrm{supp}(x)$. Lemma 2.2.2.9 shows that $\{(ab) \cdot c \mid c \in s\}$ supports $(ab) \cdot x$, but $a \notin \{(ab) \cdot c \mid c \in s\}$. However, since $a \in \mathrm{supp}(x)$ it follows that $\{(ab) \cdot c \mid c \in s\}$ cannot support x, so that $x \neq (ab) \cdot x$ as required. □

2.2.2.11 Lemma. *Let X be a nominal set, $x \in X$, $a, b \in \mathbb{A}$ and suppose that $s \subseteq_{\mathrm{fin}} \mathbb{A}$ supports x. If $b \notin s$ and $(ab) \cdot x \neq x$ then $a \in \mathrm{supp}(x)$.*

Proof. Suppose that $a \notin \mathrm{supp}(x)$. Since $b \notin s$ it follows that $b \notin \mathrm{supp}(x)$ too, but $\mathrm{supp}(x)$ supports x so that $(ab) \cdot x = x$ which is a contradiction as required. □

2.2.3 Constructing Nominal Sets

It is worth discussing some examples of nominal sets.

2.2.3.1 Discrete nominal sets. Any set may be given the trivial $\mathcal{G}$-action $\sigma \cdot x =_{\mathrm{def}} x$ and with respect to this action every element has empty support. A set with the trivial $\mathcal{G}$-action is called a discrete nominal set. Any finite nominal set is necessarily discrete.

2.2.3.2 The set of names. The set $\mathbb{A}$ is a nominal set with permutation action given by $\sigma \cdot a =_{\mathrm{def}} \sigma a$. This is clearly a group action, and it is straightforward to show that every $a \in \mathbb{A}$ is supported by the finite set $\{a\}$ and that $a \in \mathrm{supp}(a)$ by lemma 2.2.2.11 so that $\mathrm{supp}(a) = \{a\}$ and hence $a \# b$ iff $a \neq b$.

2.2.3.3 The set of permutations of names. The group $\mathcal{G}$ is a nominal set, with permutation action given by $\sigma \cdot \sigma' =_{\mathrm{def}} \sigma\sigma'\sigma^{-1}$. It is not hard to see that this is a group action. Furthermore, $\mathrm{supp}(\sigma) = \{a \in \mathbb{A} \mid \sigma a \neq a\}$ as follows. Suppose that $\sigma' a = a$ for all $a \in \mathbb{A}$ such that $\sigma a \neq a$, then it is required to show that $\sigma' \cdot \sigma =_{\mathrm{def}} \sigma'\sigma{\sigma'}^{-1} = \sigma$. Let $a \in \mathbb{A}$ and suppose that $\sigma' a \neq a$, then ${\sigma'}^{-1} a \neq a = \sigma'{\sigma'}^{-1} a$ so that ${\sigma'}^{-1} a = \sigma{\sigma'}^{-1} a$. Therefore $(\sigma' \cdot \sigma) a = \sigma'{\sigma'}^{-1} a = a$. Also if $\sigma' a \neq a$ then $\sigma a = a$ so that $(\sigma' \cdot \sigma) a = \sigma a$ as required. On the other hand suppose that $\sigma' a = a$, then ${\sigma'}^{-1} a = a$ too. If $\sigma a = a$ then $(\sigma' \cdot \sigma) a = a = \sigma a$, and if $\sigma a \neq a$ then $\sigma\sigma a \neq \sigma a$ so that $\sigma'\sigma a = \sigma a$ and hence $(\sigma' \cdot \sigma) a = a = \sigma a$ as required. Therefore σ is supported by $\{a \in \mathbb{A} \mid \sigma a \neq a\}$. Furthermore, this is the smallest support: pick any a and b such that $\sigma a \neq a$ and $\sigma b = b$ then it is certainly the case that $(ab) \cdot \sigma \neq \sigma$ so that $a \in \mathrm{supp}(\sigma)$.

2.2.3.4 Products and sums. If X and Y are nominal sets then their product $X \times Y$ and their sum $X + Y$ are both nominal sets, with permutation actions given componentwise. The support of a pair $\langle x, y \rangle \in X \times Y$ is $\mathrm{supp}(x) \cup \mathrm{supp}(y)$.

2.2.3.5 Subsets. If X is a nominal set and $Y \subseteq X$ is closed under the action of $\mathcal{G}$ then Y is also a nominal set with respect to this action.

2.2.3.6 Nominal powersets. If X is a nominal set then its powerset $\mathcal{P}X$ has a natural $\mathcal{G}$-action given by

$$\sigma \cdot A =_{\mathrm{def}} \{\sigma \cdot x \mid x \in A\}. \tag{2.2.3.7}$$

However in general not every subset of X has finite support with respect to this action. For example, consider some $A \subseteq \mathbb{A}$ which is neither finite nor has finite complement, then this A does not have a finite support with respect to the usual $\mathcal{G}$-action on $\mathbb{A}$ defined above. To see this, suppose that A is supported by a set $s \subseteq \mathbb{A}$. If there exists $a \in A \setminus s$ and $b \in \mathbb{A} \setminus (A \cup s)$ then $(ab) \cdot A \neq A$ which contradicts that s supports A, so either $A \setminus s = \varnothing$ or $\mathbb{A} \setminus (A \cup s) = \varnothing$. If $A \setminus s = \varnothing$ then $A \subseteq s$ and hence s is infinite, whereas if $\mathbb{A} \setminus (A \cup s) = \varnothing$ then $A \cup s = \mathbb{A}$ and hence $\mathbb{A} \setminus A \subseteq s$ so that s is infinite in this case as well.

However, $\mathcal{P}X$ does contain a nominal set,

$$\mathcal{P}_{\mathrm{fs}}(X) =_{\mathrm{def}} \{A \subseteq X \mid A \text{ has finite support}\}. \tag{2.2.3.8}$$

Lemma 2.2.2.9 demonstrates that if $A \subseteq X$ is supported by s then $\sigma \cdot A$ is supported by $\{\sigma a \mid a \in s\} = \sigma \cdot s$ so that $\mathcal{P}_{\mathrm{fs}}X$ does have a well-defined $\mathcal{G}$-action. Notice that any finite subset of X is supported by the union of the supports of its elements (which is finite) and therefore the finite powerset $\mathcal{P}_{\mathrm{fin}}X$ is also a nominal set.

2.2.3.9 Nominal function spaces. If X and Y are nominal sets then the set of functions $X \to Y$ is a subset of $\mathcal{P}(X \times Y)$, where the derived $\mathcal{G}$-action amounts to

$$(\sigma \cdot f)x =_{\mathrm{def}} \sigma \cdot \big(f(\sigma^{-1} \cdot x)\big). \tag{2.2.3.10}$$

In general a function $f : X \to Y$ does not have finite support: if $A \subseteq \mathbb{A}$ is infinite and has infinite complement then its characteristic function $\chi_A : \mathbb{A} \to \mathbf{2}$ has no finite support for the same reason that A has no finite support. Again, $X \to Y$ does contain a nominal set,

$$X \to_{\mathrm{fs}} Y =_{\mathrm{def}} \{f : X \to Y \mid f \text{ has finite support}\}. \tag{2.2.3.11}$$

It is not hard to see that if s supports f then $\sigma \cdot s$ supports $\sigma \cdot f$ so that $X \to_{\mathrm{fs}} Y$ does have a well-defined $\mathcal{G}$-action.

The remainder of this discussion barely mentions the fact that any particular object that has been constructed — even subsets and functions — is finitely supported. This is justified by the following principle.

2.2.3.12 Theorem (Finite Support Principle, cf. [22, Theorem 3.5]). *Any function or relation that is defined from finitely supported functions and relations using higher-order, classical logic without choice principles, is itself finitely supported.*

A particularly special class of finitely-supported function consists of those that have empty support.

2.2.3.13 Definition. *A function $f : X \to Y$ between nominal sets is* ***equivariant*** *iff it has empty support.*

If f is equivariant then $\sigma^{-1} \cdot f = f$ so that for all $x \in X$ it follows that $fx = (\sigma^{-1} \cdot f)x = \sigma^{-1} \cdot \big(f(\sigma x)\big)$ and hence

$$\sigma \cdot (fx) = f(\sigma \cdot x). \tag{2.2.3.14}$$

The converse also holds. There are many equivariant functions. Using lemma 2.2.2.9 it is not hard to see that the support function $\mathrm{supp} : X \to \mathcal{P}_{\mathrm{fin}}\mathbb{A}$ is equivariant: $\sigma \cdot \mathrm{supp}(x) = \mathrm{supp}(\sigma \cdot x)$. The identity function $\mathbf{1}_X$ is equivariant. Composition is equivariant: $\sigma \cdot (g \circ f) = (\sigma \cdot g) \circ (\sigma \cdot f)$. The projections $X \times Y \to X$ and injections $X \to X + Y$ are equivariant. Function application is equivariant: $\sigma \cdot (fx) = (\sigma \cdot f)(\sigma \cdot x)$. The boolean operations on sets are equivariant: $\sigma \cdot (x * y) = (\sigma \cdot x) * (\sigma \cdot y)$ for $* \in \{\cup, \cap, \setminus\}$, and more generally $\sigma \cdot (\bigcup X) = \bigcup(\sigma \cdot X) = \bigcup\{\sigma \cdot x \mid x \in X\}$ and similarly for intersections.

Certain predicates also exhibit an equivariance property. For example, freshness, set membership and the subset relation are equivariant: $x \# y$ iff $(\sigma \cdot x) \# (\sigma \cdot y)$, $x \in A$ iff $\sigma \cdot x \in \sigma \cdot A$ and $A \subseteq A'$ iff $\sigma \cdot A \subseteq \sigma \cdot A'$.

A simple but useful consequence of equivariance is that an equivariant function cannot extend the support of its argument. More precisely,

2.2.3.15 Lemma. *If $f : X \to Y$ is an equivariant function between nominal sets X and Y and $x \in X$ then it follows that* $\mathrm{supp}(fx) \subseteq \mathrm{supp}(x)$. *More generally, if $f : X \to Y$ is a finitely-supported function then it follows that* $\mathrm{supp}(fx) \subseteq \mathrm{supp}(f) \cup \mathrm{supp}(x)$.

Proof. If σ is a permutation such that $\sigma a = a$ for all $a \in \mathrm{supp}(x)$ then it follows that $\sigma \cdot x = x$. Furthermore by the equivariance of f it is the case that $\sigma \cdot (fx) = f(\sigma \cdot x) = fx$ which implies that $\mathrm{supp}(x)$ supports fx as required. The general case follows by noting that function evaluation is equivariant as a function $((X \to_{\mathrm{fs}} Y) \times X) \to Y$, and the support of the pair $\langle f, x \rangle$ is given by $\mathrm{supp}(f) \cup \mathrm{supp}(x)$. □

2.2.4 A Category of Nominal Sets

2.2.4.1 Definition. *The category* **NSet** *has nominal sets for objects and equivariant functions for arrows.*

It is clear that the composition of equivariant functions is equivariant, so this does define a category. Furthermore, it has products $\times$, coproducts $+$ and exponentials $\to_{\mathrm{fs}}$ given as above, and to cut a long story short it is also a Boolean topos: the subobject classifier is the discrete nominal set on $\mathbf{2} =_{\mathrm{def}} \{\top, \bot\}$ and powerobjects are given by $\mathcal{P}_{\mathrm{fs}}$.

2.2.4.2 Lemma. *Let* $\mathbb{I}$ *be the category whose objects are finite subsets of* $\mathbb{A}$ *and whose arrows are injections between them. Then the category* **NSet** *is equivalent to the Schanuel topos* **Sch** *which is the full subcategory of* $\mathbf{Set}^{\mathbb{I}}$ *consisting of those presheaves that preserve pullbacks.*

Sketch Proof. A functor $F : \mathbf{NSet} \to \mathbf{Sch}$ is defined by

$$(FX)s =_{\mathrm{def}} \{x \in X \mid s \text{ supports } x\}, \tag{2.2.4.3}$$

where if $i : s \rightarrowtail s'$ then there is some permutation σ_i that extends i, so define $Fi(x) =_{\mathrm{def}} \sigma_i \cdot x$. Note that there are many choices for σ_i but this definition is independent of this choice, because if σ_i and σ_i' both extend i then $\sigma_i^{-1}\sigma_i' \# s$ and s supports x so that $\sigma_i^{-1} \cdot \sigma_i' \cdot x = x$ and hence $\sigma_i \cdot x = \sigma_i' \cdot x$.

A functor $G : \mathbf{Sch} \to \mathbf{NSet}$ is defined by

$$GX =_{\mathrm{def}} \mathrm{colim}(X|_{\mathbb{J}}) \tag{2.2.4.4}$$

where the colimit is taken over the restriction of X to a diagram of shape $\mathbb{J}$, which is the category whose objects are finite subsets of $\mathbb{A}$ but whose arrows are just the inclusions between them. The action of G on morphisms is given by the universal property of this colimit.

There are obvious transformations $\mathbf{1} \to FG$ and $\mathbf{1} \to GF$ which essentially take elements to their equivalence classes in the colimit. It is straightforward to show that these are natural isomorphisms and hence that F and G define an equivalence of categories as required. □

2.2.5 Binding in Nominal Sets

The syntactic notion of a binding operator is captured semantically by the following condition. Pitts[22] demonstrates that this is effectively the condition that is needed to be able to define functions by purely structural recursion over a syntactic signature with binding.

2.2.5.1 Definition. *If X is a nominal set and $f : \mathbb{A} \to X$ is a finitely supported function then f satisfies the* ***freshness condition for binders*** *(FCB) if there exists $a \in \mathbb{A}$ such that $a \# f$ and $a \# fa$.*

2.2.5.2 Lemma. *If $f : \mathbb{A} \to X$ satisfies the FCB then there exists a unique $x \in X$ such that for any name $a \# f$, $fa = x$. Furthermore, in this case* $\mathrm{supp}(x) \subseteq \mathrm{supp}(f)$ *and in particular $a \# x$.*

Proof. Since f satisfies the FCB let $a \in \mathbb{A}$ be such that $a \# f$ and $a \# fa$. Let $b \# f$, so that $(ab) \cdot f = f$. If $a = b$ then $fa = fb$ trivially, so suppose that $a \neq b$. Then $fb = ((ab) \cdot f)b = (ab) \cdot (f((ab)b)) = (ab) \cdot (fa)$. However $b \# f, a$ implies that $b \# fa$ and by assumption $a \# fa$ so that $(ab) \cdot (fa) = fa$ and hence $fb = fa$ which means that fa is the unique element of X as required.

Lemma 2.2.3.15 shows that $\mathrm{supp}(fa) \subseteq \mathrm{supp}(f, a)$, and $a \# fa$ by the argument above, so it follows that $\mathrm{supp}(fa) \subseteq \mathrm{supp}(f, a) \setminus \{a\} = \mathrm{supp}(f)$. □

2.2.5.3 Definition. *If $f : \mathbb{A} \to X$ satisfies the FCB then write* $\mathbf{fresh}\, b\, \mathbf{in}\, fb$ *for the unique $x \in X$ such that for any name $b \# f$ it is the case that $fb = x$.*

Note that if X is a discrete nominal set then any finitely-supported f satisfies the FCB. In particular if f is a finitely-supported predicate on $\mathbb{A}$ and hence a function $\mathbb{A} \to \mathbf{2}$ then it satisfies the FCB. In this case, $\mathbf{fresh}\, a\, \mathbf{in}\, fa = \text{И}a.fa$ where И is the 'new' quantifier of Pitts and Gabbay[10].

When performing calculations involving terms of the form $\mathbf{fresh}\, b\, \mathbf{in}\, fb$, it is common to pick a particular name b that is fresh for f and calculate using fb. Everything in the remainder of this discussion is finitely supported so it is always possible to simply pick a name that is asserted to be 'fresh', which means that it is fresh for everything that has already been mentioned. It is also normally clear that the f used in the term $\mathbf{fresh}\, b\, \mathbf{in}\, fb$ does satisfy the FCB, so this fact is rarely mentioned. The following lemma illustrates a simple calculation using $\mathbf{fresh} \ldots \mathbf{in} \ldots$.

2.2.5.4 Lemma. *Let $f : \mathbb{A} \to X$, $g : \mathbb{A} \to Y$ and $h : X \to Y \to Z$ be finitely-supported functions such that f and g satisfy the FCB. Then*

$$h\ (\mathbf{fresh}\, a\, \mathbf{in}\, fa)\ (\mathbf{fresh}\, b\, \mathbf{in}\, gb) = \mathbf{fresh}\, c\, \mathbf{in}\, h\ (fc)\ (gc).$$

Proof. Let d be a fresh name (i.e. $d \# f, g, h$). Then $\mathbf{fresh}\, a\, \mathbf{in}\, fa = fd$ and $\mathbf{fresh}\, b\, \mathbf{in}\, gb = gd$, and $\mathbf{fresh}\, c\, \mathbf{in}\, h\ (fc)\ (gc) = h\ (fd)\ (gd)$ as required.

Notice that as promised above it is clear that $\lambda c.h(fc)(gc)$ satisfies the FCB: if $d \# f, g, h$ then certainly $d \# \lambda c.h(fc)(gc)$ and furthermore since f and g satisfy the FCB it follows from lemma 2.2.5.2 that $d \# fd$ and $d \# gd$ so by lemma 2.2.3.15 it follows that $d \# h(fd)(gd)$ as required. □

There is a symmetric monoidal structure $\otimes$ on **NSet**. Via the equivalence of lemma 2.2.4.2 it can be characterised abstractly as the Day tensor[7] of functors $\mathbb{I} \to \mathbf{Set}$. Concretely, if X and Y are nominal sets then

$$X \otimes Y =_{\text{def}} \{\langle x, y\rangle \in X \times Y \mid x \# y\} \tag{2.2.5.5}$$

with the action of the tensor on arrows given componentwise. By an abstract argument[7] that relies on the characterisation of $\otimes$ as the Day tensor, **NSet** is $\otimes$-monoidal closed. In fact this discussion only uses this tensor as part of the functor $(-) \otimes \mathbb{A}$ and it is possible to characterise its right adjoint, written δ, as follows. There is an equivalence relation $\sim_\alpha$ on $X \times \mathbb{A}$ that captures the notion of α-equivalence as

$$\langle x, a\rangle \sim_\alpha \langle x', a'\rangle \quad \Leftrightarrow_{\text{def}} \quad \mathbf{fresh}\, b\, \mathbf{in}\, (ab) \cdot x = (a'b) \cdot x'. \tag{2.2.5.6}$$

Writing the equivalence classes as $[a].x =_{\text{def}} [\langle x, a\rangle]_{\sim_\alpha}$ it is then possible to define

$$\delta X =_{\text{def}} (X \times \mathbb{A})/{\sim_\alpha} = \{[a].x \mid a \in \mathbb{A} \text{ and } x \in X\}. \tag{2.2.5.7}$$

The permutation action is given by $\sigma \cdot ([a].x) =_{\text{def}} [\sigma a].(\sigma \cdot x)$, which is well-defined since if $[a].x = [a'].x'$ and b is a fresh name then $(ab) \cdot x = (a'b) \cdot x'$ so that $\sigma \cdot (ab) \cdot x = \sigma \cdot (a'b) \cdot x'$ and hence $(\sigma a\ b) \cdot (\sigma \cdot x) = (\sigma a'\ b) \cdot (\sigma \cdot x')$ by lemma 2.2.1.3 so that finally $[\sigma a].(\sigma \cdot x) = [\sigma a'].(\sigma \cdot x')$ as required. It follows that $\operatorname{supp}([a].x) = \operatorname{supp}(x) \setminus \{a\}$ and therefore δX is a nominal set.

If $x' \in \delta X$ and $a \# x'$ then define the **concretion** $x'@a$ as the unique $x \in X$ such that $x' = [a].x$. To see that there exists such an x note that $x' = [b].y$ for some $b \in \mathbb{A}$ and some $y \in X$, so if c is a fresh name then $(ac) \cdot (ab) \cdot y = (bc) \cdot y$ and hence $[a].((ac) \cdot y) = [b].y$. Therefore if $d \# x \in X$ then $([a].x)@d = (ad) \cdot x$.

If $f : X \to Y$ is an equivariant function and $x' \in \delta X$ then define

$$(\delta f)x' =_{\text{def}} \mathbf{fresh}\, b\, \mathbf{in}\, [b].(f(x'@b)). \tag{2.2.5.8}$$

It is straightforward to check that δ so defined is a functor on **NSet**, using the following lemma.

2.2.5.9 Lemma. *If $f : X \to Y$ is an arrow of* **NSet**, *$b \in \mathbb{A}$ and $x \in X$ then*

$$\delta f([b].x) = [b].(fx)$$

Proof.

$$\begin{aligned}\delta f([b].x) &= \mathbf{fresh}\, c\, \mathbf{in}\, [c].(f((bc)\cdot x)) \qquad (2.2.5.10)\\ &= \mathbf{fresh}\, c\, \mathbf{in}\, [c].(bc)\cdot(fx)\\ &= \mathbf{fresh}\, c\, \mathbf{in}\, (bc)\cdot([b].(fx))\\ &= \mathbf{fresh}\, c\, \mathbf{in}\, [b].(fx)\\ &= [b].(fx)\end{aligned}$$

since c is fresh and $b \# [b].(fx)$. □

The unit of the adjunction $(-)\otimes\mathbb{A} \dashv \delta$ is the transformation $\xi : \mathbf{1} \to \delta((-)\otimes\mathbb{A})$ defined by

$$\xi_X(x) =_{\mathrm{def}} \mathbf{fresh}\, a\, \mathbf{in}\, [a].\langle x, a\rangle. \qquad (2.2.5.11)$$

The counit is the concretion operator $@ : (\delta -)\otimes\mathbb{A} \to \mathbf{1}$ defined above. That the triangular identities are satisfied amounts to saying that $([a].x)@a = x$ for any name a, and $[b].(x'@b) = x'$ for a fresh name b.

To see roughly why the operation δ corresponds to 'binding', consider the set Λ of λ terms, and its quotient $\Lambda/{\sim_\alpha}$ under α-equivalence. The set Λ may be defined recursively by the rules

$$\frac{}{a}\ (a\in\mathbb{A}) \quad \frac{M \quad N}{MN} \quad \frac{M}{\lambda a.M}\ (a\in\mathbb{A}) \qquad (2.2.5.12)$$

which amounts to the statement that Λ is an initial algebra for the endofunctor $\mathbb{A} + (-)^2 + \mathbb{A}\times(-)$ on **NSet**. Notice that the components of this functor arise directly from the recursive definition of Λ. Furthermore, the initiality of Λ is the property that gives rise to the principle of structural induction on λ terms. Similarly, $\Lambda/{\sim_\alpha}$ is an initial algebra for the endofunctor $\mathbb{A}+(-)^2+\delta(-)$, and the initiality gives rise to an induction principle that can be described as 'structural-up-to-α'. It is a structural principle in the sense that it decomposes terms into their components, and it is 'up-to-α' in the sense that a decomposition of a term of the form $\lambda a.M$ picks a name a which is fresh for any other names in scope. This captures the common informal practice for inductive reasoning over syntax with binding, where bound names are always chosen to be fresh. A much more detailed discussion of this can be found elsewhere[22]. Notice that δ is a purely semantic construction, so it can be used even without any obvious notion of syntax. For this reason, the word 'binding' is used to mean 'related to the adjunction $(-)\otimes\mathbb{A} \dashv \delta$' in chapters 3 and 6, since these chapters are not concerned with syntax. This is justified by the results of chapter 5 which relates the semantic and syntactic notions of binding in the language Nominal HOPLA.

2.2.6 Choice and Nominal Sets

Nominal set theory is closely related to Fraenkel-Mostowski (FM) set theory, which was originally developed early in the 20th century to prove the independence of the Axiom of Choice (AC) from the other axioms of set theory. The relationship between the two theories is discussed further in section 3.2.1 but here it is merely worth noting that, as a consequence of its origins, AC fails to hold in FM set theory and this failure also occurs in nominal set theory. For example, AC implies that there is an injective function $f : \mathbb{N} \to \mathbb{A}$ picking distinct names for each $n \in \mathbb{N}$. Suppose that f has a finite support, s, then since f has infinite range it must be that there exist $m \neq n \in \mathbb{N}$ such that $f(m), f(n) \notin s$ and hence $(f(m)\ f(n)) \cdot f = f$. Also, $\mathbb{N}$ is a discrete nominal set so that $(f(m)\ f(n)) \cdot n = n$. However, $(f(m)\ f(n)) \cdot (f(n)) = f(m) \neq f(n)$ which is a contradiction, which shows that f cannot have a finite support so it certainly is not a function in the sense of nominal set theory.

The impact of the failure of AC is not as severe as perhaps it sounds, but it does have some subtle consequences. One of the more interesting ones is that there are a number of distinct ways of characterising the subsets of a nominal set that are, in some sense, finite.

2.2.6.1 Definitions of Finiteness

(1) The usual definition: A is finite if it bijects with a finite ordinal.

(2) Kuratowski's definition: $\varnothing$ is finite, $\{x\}$ is finite for each x, if A and B are finite then so is $A \cup B$, and no other sets are finite.

(3) A is finite if for all directed collections $(B_i)_{i \in I}$ such that $A \subseteq \bigcup B_i$ there exists $i \in I$ such that $A \subseteq B_i$.

(4) A is finite if for all increasing chains $B_0 \subseteq B_1 \subseteq \ldots \subseteq B_n \subseteq \ldots$ such that $A \subseteq \bigcup B_n$ there exists $n \in \mathbb{N}$ such that $A \subseteq B_n$.

(5) Dedekind's definition: A is finite if it has no injections into any of its proper subsets.

All of these definitions coincide in classical ZFC set theory, but the situation is more complicated within nominal set theory because everything under discussion is assumed to be finitely-supported; technically, in the internal higher-order logic of nominal set theory every quantification ranges over only finitely-supported objects. In particular, in the definition of (3)-finiteness the mapping $i \mapsto B_i$ must be finitely supported, and similarly the mapping $n \mapsto B_n$ is finitely-supported in the definition of (4)-finiteness. Also in the definition of (5)-finiteness the injections in question are all finitely-supported. For the avoidance of doubt, where the word 'finite' appears in this discussion without reference to any of these definitions, it means (1)-finite. There is another possible way to characterise finiteness that is specific to the theory of nominal sets:

(6) A is finite if $\{x \in A \mid \mathrm{supp}(x) \subseteq s\}$ is (1)-finite for each $s \subseteq_{\mathrm{fin}} \mathbb{A}$.

In fact, these 6 characterisations capture three distinct notions of finiteness as the following lemma shows. The non-implications (4)-finite $\not\Rightarrow$ (1)-finite and (5)-finite $\not\Rightarrow$ (4)-finite are the key features of this lemma. Many of the other implications are well-known general results[14, Chapter D2].

2.2.6.2 Lemma. *Let $A \subseteq X$ be a finitely-supported subset of a nominal set X. Then interpreting each of these 6 sentences in the theory of nominal sets*

$$\begin{array}{rl} & A \textit{ is (1)-finite} \\ \Leftrightarrow & A \textit{ is (2)-finite} \\ \Leftrightarrow & A \textit{ is (3)-finite} \\ \Rightarrow & A \textit{ is (4)-finite} \\ \Rightarrow & A \textit{ is (5)-finite} \\ \Leftrightarrow & A \textit{ is (6)-finite} \end{array}$$

and no other implications hold in general.

Proof.

(1)-finite ⇒ (2)-finite If A bijects with the finite ordinal n then its elements can be enumerated as $a_0, a_1, \ldots, a_{n-1}$. By a simple induction on n it follows that A is (2)-finite.

(2)-finite ⇒ (3)-finite Let $(B_i)_{i \in I}$ be a directed collection such that $A \subseteq \bigcup B_i$. If $A = \varnothing$ then any $i \in I$ is such that $A \subseteq B_i$. If $A = \{x\}$ then there must exist $i \in I$ such that $A \subseteq B_i$ since $A \subseteq \bigcup B_i$. Finally if $A = A_1 \cup A_2$ where A_1 and A_2 are (2)-finite then by induction there exist $i_1, i_2 \in I$ such that $A_1 \subseteq B_{i_1}$ and $A_2 \subseteq B_{i_2}$, and by directedness there exists $i_3 \in I$ such that $B_{i_1} \subseteq B_{i_3}$ and $B_{i_2} \subseteq B_{i_3}$ so that $A \subseteq B_{i_3}$ as required.

(3)-finite ⇒ (1)-finite Let $(B_i)_{i \in I}$ be the collection of all (1)-finite subsets of A ordered by inclusion, then this collection is directed so there exists $i \in I$ such that $A \subseteq B_i$ and hence $A = B_i$ for some (1)-finite set B_i.

(3)-finite ⇒ (4)-finite The chain $B_0 \subseteq B_1 \subseteq \ldots \subseteq B_n \subseteq \ldots$ is directed, so this case is immediate.

(4)-finite $\not\Leftarrow$ (1)-finite For example $\mathbb{A}$ is (4)-finite because if the increasing chain $B_0 \subseteq B_1 \subseteq \ldots \subseteq \mathbb{A}$ is supported by the finite set s then each element B_i of the chain must also be supported by s. However there are only finitely many such subsets of $\mathbb{A}$, namely the subsets of s and the supersets of $\mathbb{A} \setminus s$, so the chain must eventually become stationary and hence $B_n = \mathbb{A}$ for some $n \in \mathbb{N}$. On the other hand by definition $\mathbb{A}$ is not (1)-finite.

(4)-finite ⇒ (5)-finite It is simpler to show the contrapositive, so assume that A is (5)-infinite and therefore obtain a finitely supported injection $f : A \to A$ and an element $a \in A$ such that the range of f is a subset of $A \setminus \{a\}$. For each n let $B_n = \{a, f(a), f^2(a), \ldots, f^n(a)\}$. It is straightforward to show that $B_0 \subsetneq B_1 \subsetneq \ldots$ since f is injective and a is not in its range. Furthermore since f and a are finitely supported it follows that each B_n is supported by $\mathrm{supp}(f) \cup \mathrm{supp}(a)$ so that the chain $(B_n)_{n\in\mathbb{N}}$ is also finitely supported. Let $B_\omega = \bigcup B_n$ and for each n let $C_n = (A \setminus B_\omega) \cup B_n$, then certainly $A = \bigcup C_n$ and $(C_n)_{n\in\mathbb{N}}$ is finitely supported but there exists no $n \in \mathbb{N}$ such that $A = C_n$ so that A is certainly not (4)-finite.

(5)-finite $\not\Rightarrow$ (4)-finite For example $\mathcal{P}_{\mathrm{fin}}\mathbb{A}$ is (4)-infinite because if

$$B_n = \{s \subseteq_{\mathrm{fin}} \mathbb{A} \mid |s| \leq n\} \tag{2.2.6.3}$$

then $B_0 \subseteq B_1 \subseteq \ldots$ is a finitely supported chain and $\mathcal{P}_{\mathrm{fin}}\mathbb{A} = \bigcup B_n$ but there exists no $n \in \mathbb{N}$ such that $\mathcal{P}_{\mathrm{fin}}\mathbb{A} = B_n$. On the other hand suppose that $f : \mathcal{P}_{\mathrm{fin}}\mathbb{A} \to \mathcal{P}_{\mathrm{fin}}\mathbb{A}$ is a injection supported by the finite set s and let s_0 be outside the range of f. Considering the sequence that starts at s_0 and continues by setting $s_{n+1} = f(s_n)$, it is clear that this sequence never repeats as f is injective. However, by considering supports it is also the case that

$$s_{n+1} = \mathrm{supp}(s_{n+1}) = \mathrm{supp}(f(s_n)) \subseteq \mathrm{supp}(f) \cup \mathrm{supp}(s_n) \subseteq s \cup s_n \tag{2.2.6.4}$$

so that $s_n \subseteq s \cup s_0$ for all n and therefore the sequence $(s_n)_{n\in\mathbb{N}}$ must eventually repeat. This is a contradiction, so $\mathcal{P}_{\mathrm{fin}}\mathbb{A}$ is (5)-finite as required.

(5)-finite ⇒ (6)-finite Suppose for a contradiction that $\{x \in A \mid \mathrm{supp}(x) \subseteq s\}$ were infinite. This set can be wellordered (externally to the theory of nominal

sets) so that it contains an injective sequence x_0, x_1, ... Define $f : A \to A$ by

$$fx =_{\mathrm{def}} \begin{cases} x_{n+1} & \text{if } x = x_n \\ x & \text{otherwise,} \end{cases} \tag{2.2.6.5}$$

then f is supported by $\mathrm{supp}(A) \cup s$ because if $\sigma \mathrel{\#} \mathrm{supp}(A) \cup s$ then $\sigma \cdot x_n = x_n$ for all $n \in \mathbb{N}$. It is also immediate that f is injective and not surjective so that A is not (5)-finite as required.

(6)-finite ⇒ (5)-finite Suppose for a contradiction that $f : A \to A$ is finitely supported and injective but not surjective, and let x_0 be outside its range. It follows that setting $x_{n+1} = f(x_n)$ for all $n \in \mathbb{N}$ gives an injective sequence $(x_n)_{n \in \mathbb{N}}$ in A. Therefore for all n

$$\mathrm{supp}(x_{n+1}) = \mathrm{supp}(f(x_n)) \subseteq \mathrm{supp}(f) \cup \mathrm{supp}(x_n) \tag{2.2.6.6}$$

so that by induction $\mathrm{supp}(x_n) \subseteq \mathrm{supp}(f) \cup \mathrm{supp}(x_0)$ for all n. Therefore it cannot be that $\{x \in A \mid \mathrm{supp}(x) \subseteq \mathrm{supp}(f) \cup \mathrm{supp}(x_0)\}$ is finite, and hence A is not (6)-finite as required. □

2.3 Conclusion

This chapter has set out the mathematical prerequisites for the remainder of this dissertation, defining a consistent nomenclature and notation for both domain theory and for the theory of nominal sets. The next chapter concentrates on combining the ideas of these two theories to construct a nominal domain theory. For an indication that this could be harder than it might first appear, notice that (3)-finiteness and (4)-finiteness (as defined in 2.2.6.1) are very closely related to the idea of isolation in domain theory with respect to approximation by directed sets and by ω-chains respectively. This observation influences the design of an appropriate nominal notion of approximation, as demonstrated in section 3.4.2 below.

Chapter 3

Nominal Domain Theory

This chapter develops a domain theory for nondeterministic processes with names. Roughly speaking, this development takes the construction of the domain theory for concurrency that underpins HOPLA[20] and follows a parallel path within the theory of nominal sets.

In slightly greater detail, the development of the domain theory behind HOPLA runs as follows. Firstly, processes are typed by the computation paths that they may follow, and these paths may be ordered by a kind of information ordering. To incorporate nondeterminism, these 'path orders' are freely closed under joins, and this free construction draws attention to a rich category of join-preserving — or linear — maps. Computationally speaking, these linear maps use their input precisely once which is a severe restriction on their expressivity, so it is desirable to turn to 'continuous' maps which preserve only directed joins and so can express discarding or limited copying of inputs, by means of a coKleisli construction. The category of path orders and continuous maps is then rich enough to support a natural domain theory for concurrency, with many computational features given by universal constructions.

A similar story unfolds within nominal set theory, but it is complicated by a number of factors. Firstly, following the usual pattern of nominal sets, everything in sight must be finitely-supported: linear maps need only preserve finitely-supported joins, for example. As may be expected this involves little real alteration to the narrative. Secondly, if the domain theory is to make use of the name-binding constructions of nominal sets then the key adjunction $(-) \otimes \mathbb{A} \dashv \delta$ must be woven into the tale somehow. This is simple enough when dealing with basic path orders, but it takes an attention to detail to ensure that this adjunction meshes properly with the free constructions mentioned above.

Thirdly, and perhaps most surprisingly, the appropriate notion of 'continuous' here is not simply the preservation of directed joins — even finitely-supported directed joins — but of directed joins with more stringent constraints on supports. As illustrated in 3.4.2 this is the price to be paid for working in a universe without the Axiom of Choice.

In 3.1 the theory of nominal preorders is introduced, and is shown to be a straightforward combination of the theories of nominal sets and of classical preorders. In 3.2 this theory is generalised to include objects and arrows that have nontrivial finite supports, because the domain theory needs to make use of such structure. Then 3.3 builds a path-based semantics using preorders of computation paths and the induced 'linear' maps are studied. As mentioned above, linearity is too restrictive a condition on the maps and in 3.4 an appropriate notion of continuity is developed.

3.1 Nominal Preorders

The theory of nominal preorders is a straightforward reinterpretation of the classical theory of preorders within nominal set theory. In some sense the nominal structure and the order structure are orthogonal, so that many of the results of these two base theories carry through into their combination.

3.1.1 Definitions

3.1.1.1 Definition. *A **nominal preorder** is a pair $\langle \mathbb{P}, \leq_{\mathbb{P}} \rangle$ where $\mathbb{P}$ is a nominal set and $\leq_{\mathbb{P}}$ is a subset of $\mathbb{P} \times \mathbb{P}$ which is reflexive and transitive as a binary relation on $\mathbb{P}$, and which is closed under the permutation action given by $\sigma \cdot \langle p_1, p_2 \rangle =_{\text{def}} \langle \sigma \cdot p_1, \sigma \cdot p_2 \rangle$.*

As a consequence, if p_1 and p_2 are elements of a nominal preorder $\mathbb{P}$ and σ is any permutation then $p_1 \leq_{\mathbb{P}} p_2$ if and only if $(\sigma \cdot p_1) \leq_{\mathbb{P}} (\sigma \cdot p_2)$.

3.1.1.2 Definition. *A **nominal monotone function** $\mathbb{P} \to \mathbb{Q}$ is a function f between nominal preorders $\langle \mathbb{P}, \leq_{\mathbb{P}} \rangle$ and $\langle \mathbb{Q}, \leq_{\mathbb{Q}} \rangle$ such that $f(p_1) \leq_{\mathbb{Q}} f(p_2)$ whenever $p_1 \leq_{\mathbb{P}} p_2$ (i.e. it is monotone) and such that for any permutation σ, $\sigma \cdot f(p) = f(\sigma \cdot p)$ (i.e. it is equivariant).*

3.1.1.3 Definition. *The category* **NPre** *has nominal preorders for objects and nominal monotone functions for arrows.*

It is not hard to see that this structure really is a category: the composition of monotone functions is monotone and the composition of equivariant functions is equivariant.

3.1.2 The structure of NPre

The category **NPre** has products and coproducts, given as in **Pre**. In detail, if $\mathbb{P}$ and $\mathbb{Q}$ are nominal preorders then their product is the order $\langle \mathbb{P}\times\mathbb{Q}, \leq_{\mathbb{P}\times\mathbb{Q}}\rangle$ where $\mathbb{P}\times\mathbb{Q}$ is the cartesian product of the underlying nominal sets with permutation action given by $\sigma \cdot \langle p, q\rangle =_{\mathrm{def}} \langle \sigma \cdot p, \sigma \cdot q\rangle$, and $\langle p_1, q_1\rangle \leq_{\mathbb{P}\times\mathbb{Q}} \langle p_2, q_2\rangle$ iff $p_1 \leq_{\mathbb{P}} p_2$ and $q_1 \leq_{\mathbb{Q}} q_2$.

The coproduct of $\mathbb{P}$ and $\mathbb{Q}$ is given by $\langle \mathbb{P}+\mathbb{Q}, \leq_{\mathbb{P}+\mathbb{Q}}\rangle$ where $\mathbb{P}+\mathbb{Q} =_{\mathrm{def}} \mathbb{P} \uplus \mathbb{Q}$ with the permutation action given componentwise, and $\leq_{\mathbb{P}+\mathbb{Q}} =_{\mathrm{def}} \leq_{\mathbb{P}} \uplus \leq_{\mathbb{Q}}$.

The category **NPre** is also cartesian closed, where the function space $\mathbb{P} \to \mathbb{Q}$ consists of finitely-supported monotone functions from $\mathbb{P}$ to $\mathbb{Q}$, ordered pointwise.

Furthermore, **NPre** contains an object of names, $\mathbb{A}$, with the discrete ordering. It also inherits the 'fresh' tensor product of nominal sets, and concretely

$$\mathbb{P} \otimes \mathbb{A} =_{\mathrm{def}} \{\langle p, a\rangle \in \mathbb{P}\times\mathbb{A} \mid a \,\#\, p\} \tag{3.1.2.1}$$

where

$$\langle p_1, a_1\rangle \leq_{\mathbb{P}\otimes\mathbb{A}} \langle p_2, a_2\rangle \Leftrightarrow_{\mathrm{def}} a_1 = a_2 \text{ and } p_1 \leq_{\mathbb{P}} p_2. \tag{3.1.2.2}$$

Finally, **NPre** is monoidal closed with respect to this tensor. As was the case in **NSet**, the important consequence of this is the existence of a right adjoint to the functor $(-)\otimes\mathbb{A}$ which can concretely be given in terms of the α-equivalence relation on $\mathbb{A}\times\mathbb{P}$ where

$$\langle a_1, p_1\rangle \sim_\alpha \langle a_2, p_2\rangle \Leftrightarrow_{\mathrm{def}} \mathbf{fresh}\, b\, \mathbf{in}\, (a_1 b)\cdot p_1 = (a_2 b)\cdot p_2. \tag{3.1.2.3}$$

The α-equivalence classes are denoted $[a].p$ where

$$[a].p =_{\mathrm{def}} [\langle a, p\rangle]_{\sim_\alpha} \tag{3.1.2.4}$$

and the right adjoint to $(-)\otimes\mathbb{A}$ is given by the functor δ where

$$\delta\mathbb{P} =_{\mathrm{def}} \{[a].p \mid \langle a, p\rangle \in \mathbb{A}\times\mathbb{P}\} \tag{3.1.2.5}$$

and

$$[a_1].p_1 \leq_{\delta\mathbb{P}} [a_2].p_2 \Leftrightarrow_{\mathrm{def}} \mathbf{fresh}\, b\, \mathbf{in}\, (a_1 b)\cdot p_1 \leq_{\mathbb{P}} (a_2 b)\cdot p_2. \tag{3.1.2.6}$$

On arrows, if $f : \mathbb{P} \to \mathbb{Q}$ is an arrow of **NPre** then δf is defined for all $p' \in \delta\mathbb{P}$ by

$$\delta f(p') =_{\text{def}} \textbf{fresh}\, b\, \textbf{in}\, [b].f(p'@b). \tag{3.1.2.7}$$

It is not hard to see that δf is monotone if f is, and otherwise the functoriality of δ follows from the same argument as in **NSet**. Also, just as in **NSet**, the unit of the adjunction is given by the function that takes $x \in X$ to $\textbf{fresh}\, a\, \textbf{in}\, [a].\langle x, a\rangle$ and the counit is the concretion operator @ as defined in 2.2.5.

3.2 FM Preorders

As demonstrated in 2.2.4, nominal sets can be thought of as certain presheaves over $\mathbb{I}^{\text{op}}$, and in the language of presheaves the global elements of a nominal set are those (set-theoretic) elements that have empty support. In some sense the interesting elements of nominal sets are precisely those that do not have empty support, because it is these elements that have a nontrivial permutation action.

Similarly, nominal sets (and preorders) can be seen as the emptily-supported 'global' elements of a wider class of gadgets that are a little like nominal sets (and preorders respectively) but which also have a nontrivial permutation action.

For example, consider a simple type theory that uses nominal sets to interpret types. If a computation cannot produce a result with a particular name a in its support then it might be desirable to record this information in the type of that computation. This could be achieved by means of an operation $(-)^{\#a}$ which essentially deletes all elements with a in their supports from the denotation of the type. This operation certainly does not result in a nominal set in general: for example, $\mathbb{A}$ is the nominal set of all names but $\mathbb{A}^{\#a} = \mathbb{A} \setminus \{a\}$ is not a nominal set, as $b \in \mathbb{A}^{\#a}$ but $a = (ab) \cdot b \notin \mathbb{A}^{\#a}$ so $\mathbb{A}^{\#a}$ is not closed under the permutation action.

It turns out that it is worth being able to manipulate sets such as $\mathbb{A}^{\#a}$ as first-class objects. This is possible within Fraenkel-Mostowski (FM) set theory.

3.2.1 Fraenkel-Mostowski Set Theory

FM set theory is closely related to the theory of nominal sets. The cumulative hierarchy of FM sets $\mathcal{V}_{\text{FM}}$ is constructed similarly to the von Neumann hierarchy $\mathcal{V}_{\text{ZF}}$ of ZF set theory, with the following differences[10]. Firstly, the starting point includes an infinite collection of atoms as well as the empty set. The permutation action on this collection of atoms gives rise to a permutation action on all of $\mathcal{V}_{\mathrm{FM}}$ by $\in$-recursion, which gives rise to the notion of 'support' for arbitrary elements of $\mathcal{V}_{\mathrm{FM}}$. The iterative process of the construction of $\mathcal{V}_{\mathrm{FM}}$ then continues in such a way as to only include elements that have (hereditarily) finite supports.

With this construction, the $\varnothing$-supported FM sets are nominal sets where the permutation action is given by $\in$-recursion. For now it is worth pretending that these are the only nominal sets. This roughly corresponds to the pretence that conventional mathematics is developed within ZF set theory when in fact an arbitrary topos (especially a Boolean one) would be perfectly adequate most of the time. In particular the axiom of foundation is not very important to many mathematicians, and similarly the $\in$-recursive nature of the permutation action is not very important here.

In some sense $\mathcal{V}_{\mathrm{FM}}$ is a large nominal set, or 'nominal class': it has a permutation action and all of its members are finitely-supported. It only takes a small logical leap to see that this justifies the use of the language of nominal sets developed in chapter 2 to manipulate FM sets too. For example, the freshness predicate # and the construction **fresh** ... **in** ... immediately carry across to FM set theory since any FM set can be seen as a subset of the nominal 'set' $\mathcal{V}_{\mathrm{FM}}$.

The analogue of the concept of a preorder in FM set theory is a *FM-preorder*, which is a pair $\langle \mathbb{P}, \leq_{\mathbb{P}} \rangle$ where $\mathbb{P}$ and $\leq_{\mathbb{P}}$ are both FM sets such that $\leq_{\mathbb{P}}$ is a reflexive and transitive binary relation on $\mathbb{P}$. The $\in$-recursive nature of the permutation action on FM sets gives rise to a permutation action on FM-preorders, where $\sigma \cdot \mathbb{P} = \{\sigma \cdot p \mid p \in \mathbb{P}\}$ and $p \leq_{\mathbb{P}} p'$ if and only if $\sigma \cdot p \leq_{\sigma \cdot \mathbb{P}} \sigma \cdot p'$. The FM-preorder $\langle \mathbb{P}, \leq_{\mathbb{P}} \rangle$ is often referred to simply as $\mathbb{P}$, but note carefully that $\mathrm{supp}(\mathbb{P})$ is an abbreviation for $\mathrm{supp}(\mathbb{P}, \leq_{\mathbb{P}})$ (with respect to the permutation action described above, of course).

The collection of FM-preorders and finitely-supported monotone functions between their underlying sets forms a category **FMPre**, and for any FM-preorder $\mathbb{P}$ it is the case that $\mathbb{P}^{\#a}$ is also a FM-preorder (where $p_1 \leq_{\mathbb{P}^{\#a}} p_2$ is defined to be $p_1 \leq_{\mathbb{P}} p_2 \wedge p_1, p_2 \in \mathbb{P}^{\#a}$) since $\mathbb{P}^{\#a}$ is supported by the finite set $\mathrm{supp}(\mathbb{P}) \cup \{a\}$. It would be convenient to be able to make $(-)^{\#a}$ into a functor, but this is not possible on **FMPre**. To see this, let $\mathbb{P}$ be any FM-preorder and consider the unique map $f : \mathbb{P} \to \{a\}$ which is necessarily monotone and finitely-supported, and hence an arrow of **FMPre**: however if $\mathbb{P}$ contains an element without a in its support then $\mathbb{P}^{\#a} \neq \varnothing$ so there are no possible candidates for the arrow $f^{\#a} : \mathbb{P}^{\#a} \to \{a\}^{\#a} = \varnothing$.

In summary **NPre** is too small to define $(-)^{\#a}$ on objects but **FMPre** is too large to define $(-)^{\#a}$ on arrows. Fortunately there is a middle ground:

3.2.1.1 Definition. *For any finite set of names s let $\mathbf{FMPre}_s$ be the subcategory of $\mathbf{FMPre}$ consisting of only those objects and arrows which are supported by s.*

The categories $\mathbf{FMPre}_s$ are appropriate for defining a functor $(-)^{\#a}$ as follows.

3.2.1.2 Definition. *If $\mathbb{P}$ is a FM-preorder then define the FM-preorder*

$$\mathbb{P}^{\#a} =_{\mathrm{def}} \{p \in \mathbb{P} \mid a \# p\} \tag{3.2.1.3}$$

ordered by the restriction of the order on $\mathbb{P}$, and if $f : \mathbb{P} \to \mathbb{Q}$ is a finitely-supported monotone function then for all $p \in \mathbb{P}^{\#a}$ define

$$f^{\#a}(p) =_{\mathrm{def}} f(p). \tag{3.2.1.4}$$

3.2.1.5 Lemma. *If $a \notin s$ then the operation $(-)^{\#a}$ defined in 3.2.1.2 is a functor $\mathbf{FMPre}_s \to \mathbf{FMPre}_{s\dot\cup\{a\}}$.*

Proof. It is only unclear that if $p \in \mathbb{P}^{\#a}$ then $f^{\#a}(p) \in \mathbb{Q}^{\#a}$. To see this, note that if $p \in \mathbb{P}^{\#a}$ then $a \# p$, and if f is an arrow of $\mathbf{FMPre}_s$ and $a \notin s$ then $a \# f$ and hence $a \# f(p)$ so that $f(p) \in \mathbb{Q}^{\#a}$ as required. It is now straightforward to see that $(-)^{\#a}$ sends objects and arrows of $\mathbf{FMPre}_s$ to those of $\mathbf{FMPre}_{s\dot\cup\{a\}}$ and that its action on arrows is functorial. □

The discussion in 6.1.2.3 demonstrates that this functor arises naturally from the tensor operation $(-) \otimes \mathbb{A}$ in $\mathbf{NPre}$. Furthermore, its right adjoint δ gives rise to a right adjoint to $(-)^{\#a}$ which is given a concrete description here.

3.2.1.6 Definition. *If $\mathbb{P}$ is a FM-preorder then define the α-equivalence relation $\sim_\alpha$ on $\mathbb{P} \times \mathbb{A}$ as in 3.1.2.3 and define the FM-preorder*

$$\delta_a\mathbb{P} =_{\mathrm{def}} \{p' \in (\mathbb{P} \times \mathbb{A})/{\sim_\alpha} \mid \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in (ab) \cdot \mathbb{P}\}, \tag{3.2.1.7}$$

where if p_1' and p_2' are elements of $\delta_a\mathbb{P}$ then

$$p_1' \leq_{\delta_a\mathbb{P}} p_2' \qquad \Leftrightarrow_{\mathrm{def}} \qquad \mathbf{fresh}\, b\, \mathbf{in}\, p_1'@b \leq_{(ab)\cdot\mathbb{P}} p_2'@b. \tag{3.2.1.8}$$

If $f : \mathbb{P} \to \mathbb{Q}$ is a finitely-supported monotone function then for all $p' \in \delta_a\mathbb{P}$ define

$$\delta_a f(p') =_{\mathrm{def}} \mathbf{fresh}\, b\, \mathbf{in}\, [b].\big((ab) \cdot f\big)(p'@b). \tag{3.2.1.9}$$

Intuitively, δ_a captures the idea of 'hiding' the name a within a FM preorder that might have had a in its support, which is closely related to the idea of binding a in a syntactic object. To see this, notice that $(\mathbb{A} \times \mathbb{A})/\sim_\alpha = \delta\mathbb{A} \cong \mathbb{A} \uplus \{*\}$ where the new element $*$ is the α-equivalence class containing $\langle a, a\rangle$ for all $a \in \mathbb{A}$. If

$s \subseteq_{\text{fin}} \mathbb{A} \setminus \{a\}$ then under this bijection $\delta_a s \cong s$ and $\delta_a(s \mathbin{\dot\cup} \{a\}) \cong s \mathbin{\dot\cup} \{*\}$: in the latter case, a is hidden by replacing it with the new element $*$. Furthermore $\delta_a(\mathbb{A} \setminus s) \cong \mathbb{A} \setminus s \mathbin{\dot\cup} \{*\}$ and $\delta_a(\mathbb{A} \setminus (s \mathbin{\dot\cup} \{a\})) \cong \mathbb{A} \setminus s$: again, a is hidden by using the new element $*$ in its place. Note carefully that δ_a does not simply delete a: if $x \subseteq_{\text{fs}} \mathbb{A}$ then $a \in \delta_a x$ if and only if x is cofinite.

3.2.1.10 Lemma. *If $f : \mathbb{P} \to \mathbb{Q}$ is a finitely-supported monotone function, $p \in \mathbb{P}$ and $f \# b \in \mathbb{A} \setminus \{a\}$ then*

$$\delta_a f([b].p) = [b].(((ab) \cdot f)p).$$

Proof.

$$\begin{aligned} \delta_a f([b].p) &= \textbf{fresh}\, c\, \textbf{in}\, [c].(((ac) \cdot f)((bc) \cdot p)) \qquad (3.2.1.11) \\ &= \textbf{fresh}\, c\, \textbf{in}\, [c].(ac) \cdot (f((ac) \cdot (bc) \cdot p)) \\ &= \textbf{fresh}\, c\, \textbf{in}\, [c].(ac) \cdot (f((bc) \cdot (ab) \cdot p)) \\ &= \textbf{fresh}\, c\, \textbf{in}\, [c].(ac) \cdot (bc) \cdot (f((ab) \cdot p)) \\ &= \textbf{fresh}\, c\, \textbf{in}\, [c].(bc) \cdot (ab) \cdot (f((ab) \cdot p)) \\ &= \textbf{fresh}\, c\, \textbf{in}\, (bc) \cdot ([b].(ab) \cdot (f((ab) \cdot p))) \\ &= \textbf{fresh}\, c\, \textbf{in}\, (bc) \cdot ([b].((ab) \cdot f)p) \\ &= \textbf{fresh}\, c\, \textbf{in}\, ([b].((ab) \cdot f)p) \\ &= [b].(((ab) \cdot f)p) \end{aligned}$$

where the penultimate step follows since c is fresh and $b \# [b].((ab) \cdot f)p$. □

3.2.1.12 Lemma. *If $a \notin s$ then the operation δ_a defined in 3.2.1.6 is a functor $\mathbf{FMPre}_{s \dot\cup \{a\}} \to \mathbf{FMPre}_s$.*

Proof. Let $\mathbb{P}$ be an object of $\mathbf{FMPre}_{s \dot\cup \{a\}}$. Suppose that σ is a permutation that fixes s, let $p' \in \delta_a \mathbb{P}$ and let b be a fresh name, then $p'@b \in (ab) \cdot \mathbb{P}$ and hence $(\sigma \cdot p')@b = \sigma \cdot (p'@b) \in \sigma \cdot (ab) \cdot \mathbb{P}$. However, $\text{supp}((ab) \cdot \mathbb{P}) \subseteq s \mathbin{\dot\cup} \{b\}$ and σ fixes s and b was fresh so that σ fixes b too, and hence $\sigma \cdot (ab) \cdot \mathbb{P} = (ab) \cdot \mathbb{P}$. Therefore $\sigma \cdot p' \in \delta_a \mathbb{P}$ and hence $\sigma \cdot \delta_a \mathbb{P} \subseteq \delta_a \mathbb{P}$. By the same argument $\sigma^{-1} \cdot \delta_a \mathbb{P} \subseteq \delta_a \mathbb{P}$ and hence $\sigma \cdot \delta_a \mathbb{P} = \delta_a \mathbb{P}$ by equivariance. Therefore s supports $\delta_a \mathbb{P}$ so that the action of δ_a on objects is well-defined.

Now let $f : \mathbb{P} \to \mathbb{Q}$ be an arrow of $\mathbf{FMPre}_{s \dot\cup \{a\}}$, let $p' \in \delta_a \mathbb{P}$ and let b be a fresh name. Therefore $p'@b \in (ab) \cdot \mathbb{P}$ so that

$$\big((ab) \cdot f\big)(p'@b) = \big([b].\big((ab) \cdot f\big)(p'@b)\big)@b \in (ab) \cdot \mathbb{Q} \qquad (3.2.1.13)$$

and hence $[b].\big((ab) \cdot f\big)(p'@b) \in \delta_a \mathbb{Q}$ so that $\delta_a f$ is well-defined.

Let $p'_1 \leq_{\delta_a \mathbb{P}} p'_2$ and let b be fresh. Therefore $p'_1@b \leq_{(ab)\cdot\mathbb{P}} p'_2@b$ so that by monotonicity $((ab) \cdot f)(p'_1@b) \leq_{(ab)\cdot\mathbb{Q}} ((ab) \cdot f)(p'_2@b)$ and hence it is the case that $\delta_a f(p'_1) \leq_{\delta_a \mathbb{Q}} \delta_a f(p'_2)$ which demonstrates that $\delta_a f$ is monotonic.

Let σ be a permutation that fixes s and suppose that b is fresh. The support of $(ab) \cdot f$ is contained in $s \mathbin{\dot\cup} \{b\}$ so that $\sigma \cdot (ab) \cdot f = (ab) \cdot f$. Therefore

$$\begin{aligned} \sigma \cdot (\delta_a f(p')) &= \sigma \cdot [b].\big((ab) \cdot f\big)(p'@b) \\ &= [\sigma \cdot b].\big(\sigma \cdot (ab) \cdot f\big)(\sigma \cdot (p'@b)) \\ &= [b].\big((ab) \cdot f\big)((\sigma \cdot p')@b) \\ &= \delta_a f(\sigma \cdot p') \end{aligned} \tag{3.2.1.14}$$

so that s supports $\delta_a f$ so that the action of δ_a on arrows is also well-defined. It is not hard to see that the action of δ_a is functorial. □

3.2.1.15 Lemma. *If $a \notin s$ then there is an adjunction*

$$(-)^{\#a} \dashv \delta_a : \mathbf{FMPre}_s \leftrightarrows \mathbf{FMPre}_{s \dot\cup \{a\}}.$$

Proof. For each object $\mathbb{P}$ of $\mathbf{FMPre}_s$ define $\xi_{\mathbb{P}} : \mathbb{P} \to \delta_a(\mathbb{P}^{\#a})$ as an arrow of $\mathbf{FMPre}_s$ by

$$\xi_{\mathbb{P}}(p) =_{\text{def}} \mathbf{fresh}\, b\, \mathbf{in}\, [b].p. \tag{3.2.1.16}$$

Let $p \in \mathbb{P}$ and let b be a fresh name. Since s supports $\mathbb{P}$, $(ab) \cdot \mathbb{P} = \mathbb{P}$ so that $(ab) \cdot p \in \mathbb{P}$. Furthermore $b \# p$ so that $a \# (ab) \cdot p$ and hence $(ab) \cdot p \in \mathbb{P}^{\#a}$, so that $([b].p)@b = p \in (ab) \cdot (\mathbb{P}^{\#a})$. Therefore $\xi_{\mathbb{P}}(p) = [b].p \in \delta_a(\mathbb{P}^{\#a})$ so that $\xi_{\mathbb{P}}$ is well-defined. It is not hard to see that $\xi_{\mathbb{P}}$ is supported by s and that it is monotone.

For each object $\mathbb{Q}$ of $\mathbf{FMPre}_{s \dot\cup \{a\}}$ define $\zeta_{\mathbb{Q}} : (\delta_a \mathbb{Q})^{\#a} \to \mathbb{Q}$ as an arrow of $\mathbf{FMPre}_{s \dot\cup \{a\}}$ by

$$\zeta_{\mathbb{Q}}(q') =_{\text{def}} q'@a. \tag{3.2.1.17}$$

Let $q' \in (\delta_a \mathbb{Q})^{\#a}$ so that $a \# q'$. Also let b be fresh, then $q'@b \in (ab) \cdot \mathbb{Q}$ and hence $(ab) \cdot (q'@b) = q'@a \in \mathbb{Q}$ so that $\zeta_{\mathbb{Q}}$ is well-defined. It is not hard to see that $\zeta_{\mathbb{Q}}$ is supported by $s \mathbin{\dot\cup} \{a\}$ and that it is monotone.

Let $f : \mathbb{P} \to \mathbb{P}'$ be an arrow of $\mathbf{FMPre}_s$, let $p \in \mathbb{P}$ and let b be a fresh name. Then

$$\begin{aligned} (\delta_a(f^{\#a}) \circ \xi_{\mathbb{P}})(p) &= \delta_a(f^{\#a})([b].p) \\ &= [b].\big((ab) \cdot f^{\#a}\big)(([b].p)@b) \\ &= [b].(ab) \cdot \big(f((ab) \cdot p)\big) \\ &= [b].f(p) \qquad \text{as } a \# f \text{ and } b \# f \\ &= (\xi_{\mathbb{P}'} \circ f)(p) \end{aligned} \tag{3.2.1.18}$$

so that $\xi_{\mathbb{P}}$ is natural in $\mathbb{P}$. Let $g : \mathbb{Q} \to \mathbb{Q}'$ be an arrow of $\mathbf{FMPre}_{s \dot\cup \{a\}}$, let

$q' \in (\delta_a\mathbb{Q})^{\#a}$ and let b be a fresh name. Then

$$\begin{aligned}(\zeta'_{\mathbb{Q}} \circ (\delta_a g)^{\#a})(q') &= \big([b].\big((ab)\cdot g\big)(q'@b)\big)@a \\ &= (ab)\cdot\big((ab)\cdot g\big)(q'@b) \\ &= g\big((ab)\cdot(q'@b)\big) \\ &= g\big(q'@a)\big) \qquad \text{as } a \# q' \\ &= (g\circ\zeta_{\mathbb{Q}})(q')\end{aligned} \tag{3.2.1.19}$$

so that $\zeta_{\mathbb{Q}}$ is natural in $\mathbb{Q}$.

It remains to show that ξ and ζ satisfy the triangular identities. Let $p \in \mathbb{P}^{\#a}$ and let b be fresh, then $(\zeta_{\mathbb{P}^{\#a}} \circ \xi_{\mathbb{P}}^{\#a})(p) = ([b].p)@a = (ab)\cdot p = p$ as $a \# p$ so that $\zeta_{(-)^{\#a}} \circ \xi^{\#a} = \mathbf{1}_{(-)^{\#a}}$.

Let $q' \in \delta_a\mathbb{Q}$ and let b be fresh, then

$$\begin{aligned}(\delta_a\zeta_{\mathbb{Q}} \circ \xi_{\delta_a\mathbb{Q}})(q') &= \delta_a\zeta_{\mathbb{Q}}([b].q') \\ &= [b].\big((ab)\cdot\zeta_{\mathbb{Q}})(([b].q')@b) \\ &= [b].(ab)\cdot\zeta_{\mathbb{Q}}((ab)\cdot q') \\ &= [b].(ab)\cdot\big(((ab)\cdot q')@a\big) \\ &= [b].(q'@b) = q'\end{aligned} \tag{3.2.1.20}$$

so that $\delta_a\zeta \circ \xi_{\delta_a} = \mathbf{1}_{\delta_a}$ as required. □

Finally, it is worth giving a name, $\tau^a_{\mathbb{P}}$, to the obvious natural embedding of $\mathbb{P}^{\#a}$ into $\mathbb{P}$ as follows.

3.2.1.21 Definition. *If $a \notin s$ and $J : \mathbf{FMPre}_s \hookrightarrow \mathbf{FMPre}_{s\dot\cup\{a\}}$ is the inclusion of categories then define the natural transformation*

$$\tau^a : (-)^{\#a} \Rightarrow J : \mathbf{FMPre}_s \rightrightarrows \mathbf{FMPre}_{s\dot\cup\{a\}}$$

by $\tau^a_{\mathbb{P}}(p) =_{\mathrm{def}} p$ for all $p \in \mathbb{P}^{\#a}$.

3.3 Nominal Nondeterministic Domains

Recall from section 2.1.5 that the development of the classical domain theory that underpins the language HOPLA is based on a semantics of paths, where intuitively a process denotes the collection of computation paths that it may perform. A similar intuition applies when working within FM set theory, with the added constraint that each process can only access finitely many names, so the denotation of a process must be finitely supported. Apart from this additional constraint the development of categories of FM-linear maps follows closely that of the development of the classical category **Lin** of linear maps. The

FM-linear categories are similar in structure to **Lin**, but they also have enough structure to capture the idea of names and binding, as demonstrated in 3.3.5.9 below.

3.3.1 Free Join-Completions of Path Orders

The collection of path sets over a nominal path order $\mathbb{P}$ may be ordered by inclusion to form the nominal partial order $\widehat{\mathbb{P}}$, which can be interpreted as a domain of the meanings of processes of type $\mathbb{P}$. More generally, if $\mathbb{P}$ is any FM-preorder then define

$$\widehat{\mathbb{P}} =_{\mathrm{def}} \{x_{\downarrow} \mid x \subseteq_{\mathrm{fs}} \mathbb{P}\}, \tag{3.3.1.1}$$

ordered by inclusion. (For the sake of definiteness, the set comprehension involved in the definition above is external to the universe of nominal sets.) Such a poset is an FM-complete join-semilattice in the sense that every finitely-supported subset of $\widehat{\mathbb{P}}$ has a join in $\widehat{\mathbb{P}}$ given by its union.

Alternatively, $\widehat{\mathbb{P}}$ may be viewed as the monotone function space $\mathbb{P}^{\mathrm{op}} \to_{\mathrm{fs}} \mathbf{2}$ where $\mathbf{2}$ is the nontrivial poset on $\{\top, \bot\}$. Elements $x \in \widehat{\mathbb{P}}$ correspond to their characteristic functions χ_x such that

$$\chi_x(p) = \begin{cases} \top & \text{if } p \in x \\ \bot & \text{otherwise.} \end{cases} \tag{3.3.1.2}$$

The order $\widehat{\mathbb{P}}$ contains elements of the form $\{p\}_{\downarrow}$ for each $p \in \mathbb{P}$. These elements are (completely) prime: if $X \subseteq \widehat{\mathbb{P}}$ and $\{p\}_{\downarrow} \subseteq \bigcup X$ then $p \in \bigcup X$ so there exists $x \in X$ such that $p \in x$ and hence $\{p\}_{\downarrow} \subseteq x$. Conversely, every prime element is of this form: certainly $x \subseteq \bigcup\{\{p\}_{\downarrow} \mid p \in x\}$ so if x is prime then there exists $p \in x$ such that $x \subseteq \{p\}_{\downarrow}$ and hence $x = \{p\}_{\downarrow}$ as required.

Furthermore, if $x \in \widehat{\mathbb{P}}$ then

$$x = \bigcup\{\{p\}_{\downarrow} \mid p \in x\} \tag{3.3.1.3}$$

so that $\widehat{\mathbb{P}}$ is prime algebraic.

Finally, $\widehat{\mathbb{P}}$ can be characterised abstractly as the free finitely-supported join completion of $\mathbb{P}$. In other words, $\widehat{\mathbb{P}}$ has all finitely-supported joins and if C is a FM-poset that also has all finitely supported joins and $f : \mathbb{P} \to C$ is a monotone finitely-supported function then there is a unique finitely-supported $f^{\dagger} : \widehat{\mathbb{P}} \to C$ that preserves all finitely supported joins and such that the following diagram commutes. Note that the $\mathbb{P}$th component of the natural transformation $\{\cdot\}_\downarrow : \mathbf{1} \to \widehat{(-)}$ is written $\{\cdot\}_{\mathbb{P}}$ and not $\{\cdot\}_{\downarrow\mathbb{P}}$.

$$\begin{tikzcd}
\mathbb{P} \arrow[r, "\{\cdot\}_{\mathbb{P}}"] \arrow[dr, "f"'] & \widehat{\mathbb{P}} \arrow[d, "f^\dagger"] \\
& C
\end{tikzcd} \tag{3.3.1.4}$$

The function $f^\dagger$ is given by

$$f^\dagger x =_{\text{def}} \bigvee\{fp \mid p \in x\}. \tag{3.3.1.5}$$

The uniqueness of $f^\dagger$ follows from the algebraicity of $\widehat{\mathbb{P}}$, and it is not hard to show that it preserves finitely-supported joins and that it is itself finitely-supported. In particular, if C is of the form $\widehat{\mathbb{Q}}$ then this shows that finitely-supported monotone maps $\mathbb{P} \to \widehat{\mathbb{Q}}$ are in bijective correspondence with finitely-supported maps $\widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ that preserve finitely-supported joins. For brevity, call such maps 'FM-linear' or simply 'linear'.

Notice that if f and g are linear maps $\widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ and $f \circ \{\cdot\}_{\mathbb{P}} = g \circ \{\cdot\}_{\mathbb{P}}$ then by 3.3.1.4

$$f = (f \circ \{\cdot\}_{\mathbb{P}})^\dagger = (g \circ \{\cdot\}_{\mathbb{P}})^\dagger = g. \tag{3.3.1.6}$$

3.3.2 Categories of FM-Linear Maps

Let $\mathbf{FMLin}_s$ be the category whose objects are FM-preorders $\mathbb{P}$, $\mathbb{Q}$, ... supported by s and whose arrows $\mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ are FM-linear maps $\widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ supported by s. The discussion at the end of the previous section shows that there is a bijection

$$\mathbf{FMPre}_s(\mathbb{P}, \widehat{\mathbb{Q}}) \cong \mathbf{FMLin}_s(\mathbb{P}, \mathbb{Q}). \tag{3.3.2.1}$$

FM-linear maps are necessarily monotone, since if $p \leq p'$ then $p \vee p' = p'$ so that if f preserves joins then $fp \vee fp' = f(p \vee p') = fp'$ and hence $fp \leq fp'$. Therefore there is a functor $J : \mathbf{FMLin}_s \to \mathbf{FMPre}_s$, where $J\mathbb{P} =_{\text{def}} \widehat{\mathbb{P}}$ and if $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_s$ then $Jf =_{\text{def}} f : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ in $\mathbf{FMPre}_s$. Furthermore the freeness property 3.3.1.4 says that $(\mathbb{P}, \{\cdot\}_{\mathbb{P}})$ is initial in $(\mathbb{P} \downarrow J)$ which implies that 3.3.2.1 is an adjunction with J the right adjoint to the functor which acts as the identity on objects and takes an arrow $f : \mathbb{P} \to \mathbb{Q}$ of $\mathbf{FMPre}_s$ to $\widehat{f} =_{\text{def}} (\{\cdot\}_{\mathbb{Q}} \circ f)^\dagger : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ in $\mathbf{FMLin}_s$. Concretely this means that if $x \in \widehat{\mathbb{P}}$ then

$$\widehat{f}x = \{q \in \mathbb{Q} \mid \exists p \in x.q \leq_{\mathbb{Q}} fp\} = \{fp \mid p \in x\}_\downarrow. \tag{3.3.2.2}$$

The unit of this adjunction is $\{\cdot\}_\downarrow$ and the counit is given by $\cup =_{\mathrm{def}} \mathbf{1}^\dagger_{\widehat{(-)}}$ where if $X \in \widehat{\widehat{\mathbb{P}}}$ then

$$\cup_{\mathbb{P}} X = \mathbf{1}^\dagger_{\widehat{\mathbb{P}}} X = \bigcup X. \tag{3.3.2.3}$$

3.3.3 A relationship between $(-)^{\#a}$ and $\widehat{(-)}$

For the entirety of this section (3.3.3) let $s \subseteq_{\mathrm{fin}} \mathbb{A}$ and $a \in \mathbb{A} \setminus s$ be fixed. In order to construct a functor $(-)^{\#a+} : \mathbf{FMLin}_s \to \mathbf{FMLin}_{s\dot\cup\{a\}}$ that plays a similar role to $(-)^{\#a} : \mathbf{FMPre}_s \to \mathbf{FMPre}_{s\dot\cup\{a\}}$ as defined in 3.2.1.2 it is helpful to be able to commute the functors $(-)^{\#a}$ and $\widehat{(-)}$ in order to construct a composition such as

$$\begin{tikzcd} \widehat{\mathbb{P}\#a} \arrow[r] & \widehat{\mathbb{P}}\#a \arrow[r, "f^{\#a}"] & \widehat{\mathbb{Q}}\#a \arrow[r] & \widehat{\mathbb{Q}\#a} \end{tikzcd} \tag{3.3.3.1}$$

where $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$. The abstract results of chapter 6 demonstrate that this composition makes up a suitably canonical analogue of the functor $(-)^{\#a}$ on the linear categories, but for the purposes of this section it suffices to work more concretely. To that end, for each object $\mathbb{P}$ of $\mathbf{FMPre}_s$ define $\phi^{(a)}_{\mathbb{P}} : \widehat{\mathbb{P}}\#a \to \widehat{\mathbb{P}\#a}$ by

$$\phi^{(a)}_{\mathbb{P}}(x) =_{\mathrm{def}} \{p \in x \mid a \# p\} \tag{3.3.3.2}$$

and define also ${\phi^{(a)}_{\mathbb{P}}}^{-1} : \widehat{\mathbb{P}\#a} \to \widehat{\mathbb{P}}\#a$ by

$${\phi^{(a)}_{\mathbb{P}}}^{-1}(x) =_{\mathrm{def}} x \cup \bigcup_{b\#x,\mathbb{P}} (ab) \cdot x. \tag{3.3.3.3}$$

Often the parameter a is given by the context, and it is then simpler to denote $\phi^{(a)}$ simply as ϕ, and similarly for ϕ^{-1}. It is convenient to characterise the action of $\phi^{-1}_{\mathbb{P}}$ as follows.

3.3.3.4 Lemma. *If $\mathbb{P}$ is an object of $\mathbf{FMLin}_s$ and $x \in \widehat{\mathbb{P}\#a}$ then*

$$p \in \phi^{-1}_{\mathbb{P}}(x) \quad \Leftrightarrow \quad \mathbf{fresh}\, c\, \mathbf{in}\, p \in (ac) \cdot x.$$

Proof. Suppose that $p \in \phi^{-1}_{\mathbb{P}}(x)$, then either $p \in x$ or there exists b such that $b \# x$ and $p \in (ab) \cdot x$. If $p \in x$ then $a \# p$. Let c be fresh, then it is the case that $p = (ac) \cdot p \in (ac) \cdot x$ as required. On the other hand, let b be such that $b \# x$ and $p \in (ab) \cdot x$, then $b \# p$. If $a = b$ then this reduces to the previous case, so suppose that $a \neq b$. Let c be fresh, then

$$p = (bc) \cdot p \in (bc) \cdot (ab) \cdot x = (ac) \cdot (bc) \cdot x = (ac) \cdot x$$

as required. Conversely, suppose that c is fresh and $p \in (ac) \cdot x$, then $c \# x, \mathbb{P}$ so that $p \in \phi^{-1}_{\mathbb{P}}(x)$ as required. □

The notation is no accident: ϕ and ϕ^{-1} are natural and mutual inverses.

3.3.3.5 Lemma. *If $a \notin s$ and $\mathbb{P}$ is an object of $\mathbf{FMPre}_s$ then*

$$\phi_{\mathbb{P}} : \widehat{\mathbb{P}}^{\#a} \cong \widehat{\mathbb{P}^{\#a}} : \phi_{\mathbb{P}}^{-1}$$

is an isomorphism in $\mathbf{FMPre}_{s \dot\cup \{a\}}$ which is natural in $\mathbb{P}$. Put differently, the map ϕ is a natural isomorphism of the functors

$$\widehat{(-)}^{\#a}, \widehat{(-)^{\#a}} : \mathbf{FMPre}_s \rightrightarrows \mathbf{FMPre}_{s \dot\cup \{a\}}.$$

Proof. Firstly, it is clear that $\phi_{\mathbb{P}}$ is an arrow of $\mathbf{FMPre}_{s \dot\cup \{a\}}$. To see that $\phi_{\mathbb{P}}$ is natural in $\mathbb{P}$, let $f : \mathbb{P} \to \mathbb{Q}$ be an arrow of $\mathbf{FMPre}_s$ and let $x \in \widehat{\mathbb{P}}^{\#a}$. Let $q \in \big(\widehat{f^{\#a}} \circ \phi_{\mathbb{P}}\big)(x)$, then there is some $p \in x$ such that $a \# p$ and $q \leq fp$, and also $a \# q$. However if $a \# p$ then $a \# fp$ since $a \# f$ by assumption, so that $q \in \big(\phi_{\mathbb{Q}} \circ \widehat{f}^{\#a}\big)(x)$. Conversely let $q \in \big(\phi_{\mathbb{Q}} \circ \widehat{f}^{\#a}\big)(x)$, then $a \# q$ and $q \leq fp$ for some $p \in x$. Let b be a fresh name, then $q = (ab) \cdot q \leq (ab) \cdot (fp) = f((ab) \cdot p)$, and $b \# p$ so that $a \# (ab) \cdot p$. Therefore $q \leq f((ab) \cdot p) \in \big(\widehat{f^{\#a}} \circ \phi_{\mathbb{P}}\big)(x)$ and hence $\big(\widehat{f^{\#a}} \circ \phi_{\mathbb{P}}\big) = \big(\phi_{\mathbb{Q}} \circ \widehat{f}^{\#a}\big)(x)$ so that $\phi_{\mathbb{P}}$ is natural in $\mathbb{P}$.

Now it is necessary to show that $\phi_{\mathbb{P}}^{-1}$ is well-defined, so let $x \in \widehat{\mathbb{P}^{\#a}}$. Let $p \in \phi_{\mathbb{P}}^{-1}(x)$ and let b be a fresh name, then by lemma 3.3.3.4 it follows that $p \in (ab) \cdot x \subseteq (ab) \cdot \mathbb{P} = \mathbb{P}$ so that $\phi_{\mathbb{P}}^{-1}(x) \subseteq \mathbb{P}$. Now let $p' \leq_{\mathbb{P}} p$, and let b' be fresh, then $a \# (ab') \cdot p'$ so that $(ab') \cdot p' \leq_{\mathbb{P}^{\#a}} (ab') \cdot p \in x \in \widehat{\mathbb{P}^{\#a}}$ so that $(ab') \cdot p' \in x$ and hence $p' \in \phi_{\mathbb{P}}^{-1}(x)$. Therefore $\phi_{\mathbb{P}}^{-1}(x) \in \widehat{\mathbb{P}}$. It remains to show that $a \# \phi_{\mathbb{P}}^{-1}(x)$, so let b be a fresh name and show that $\phi_{\mathbb{P}}^{-1}(x) = (ab) \cdot \phi_{\mathbb{P}}^{-1}(x)$ as follows. Let $p \in \phi_{\mathbb{P}}^{-1}(x)$ and let c be a fresh name, then $p \in (ac) \cdot x$ and hence $(ab) \cdot p \in (ab) \cdot (ac) \cdot x = (ac) \cdot (bc) \cdot x = (ac) \cdot x$ so that $(ab) \cdot p \in \phi_{\mathbb{P}}^{-1}(x)$. Therefore $\phi_{\mathbb{P}}^{-1}(x) \subseteq (ab) \cdot \phi_{\mathbb{P}}^{-1}(x)$ so that by equivariance $\phi_{\mathbb{P}}^{-1}(x) = (ab) \cdot \phi_{\mathbb{P}}^{-1}(x)$ and hence $\phi_{\mathbb{P}}^{-1}(x) \in \widehat{\mathbb{P}}^{\#a}$. Now to see that $\phi_{\mathbb{P}}^{-1}$ is supported by $s \dot\cup \{a\}$ let $x \in \widehat{\mathbb{P}^{\#a}}$ and let σ be a permutation that fixes $s \dot\cup \{a\}$. Let $p \in \sigma \cdot (\phi_{\mathbb{P}}^{-1}(x))$ and let b be a fresh name, then $p \in \sigma \cdot (ab) \cdot x = (ab) \cdot \sigma \cdot x$ and hence $p \in \phi_{\mathbb{P}}^{-1}(\sigma \cdot x)$. Conversely let $p \in \phi_{\mathbb{P}}^{-1}(\sigma \cdot x)$ and let b be a fresh name, then $p \in (ab) \cdot \sigma \cdot x = \sigma \cdot (ab) \cdot x$ and hence $p \in \sigma \cdot \phi_{\mathbb{P}}^{-1}(x)$ as required. Therefore $\sigma \cdot \phi_{\mathbb{P}}^{-1}(x) = \phi_{\mathbb{P}}^{-1}(\sigma \cdot x)$. It is clear that $\phi_{\mathbb{P}}^{-1}$ is monotone, so this has shown that $\phi_{\mathbb{P}}^{-1}$ is indeed an arrow of $\mathbf{FMPre}_{s \dot\cup \{a\}}$.

To see that $\phi_{\mathbb{P}}^{-1} \circ \phi_{\mathbb{P}} = \mathbf{1}_{\widehat{\mathbb{P}}^{\#a}}$, let $x \in \widehat{\mathbb{P}}^{\#a}$ and show that $(\phi_{\mathbb{P}}^{-1} \circ \phi_{\mathbb{P}})(x) = x$ as follows. Let $p \in x$ and let b be a fresh name. By assumption, $a \# x$ so that $(ab) \cdot p \in x$. Also $b \# p$ so that $a \# (ab) \cdot p$ and hence $(ab) \cdot p \in \phi_{\mathbb{P}}(x)$. Therefore $p \in (ab) \cdot \phi_{\mathbb{P}}(x)$ and b is fresh so that $p \in (\phi_{\mathbb{P}}^{-1} \circ \phi_{\mathbb{P}})(x)$. Conversely let $p \in (\phi_{\mathbb{P}}^{-1} \circ \phi_{\mathbb{P}})(x)$ and let b be a fresh name, then $p \in (ab) \cdot \phi_{\mathbb{P}}(x)$ so that $p \in (ab) \cdot x = x$. Therefore $\phi_{\mathbb{P}}^{-1} \circ \phi_{\mathbb{P}} = \mathbf{1}_{\widehat{\mathbb{P}}^{\#a}}$.

To see that $\phi_{\mathbb{P}} \circ \phi_{\mathbb{P}}^{-1} = \mathbf{1}_{\widehat{\mathbb{P}\#a}}$, let $x \in \widehat{\mathbb{P}\#a}$ and show that $(\phi_{\mathbb{P}} \circ \phi_{\mathbb{P}}^{-1})(x) = x$ as follows. Let $p \in x$, then $p \in \phi_{\mathbb{P}}^{-1}(x)$ and furthermore $a \# p$ so it follows that $p \in (\phi_{\mathbb{P}} \circ \phi_{\mathbb{P}}^{-1})(x)$. Conversely, let $p \in (\phi_{\mathbb{P}} \circ \phi_{\mathbb{P}}^{-1})(x)$, then $a \# p$ and $p \in \phi_{\mathbb{P}}^{-1}(x)$ so that for a fresh name b, $p \in (ab) \cdot x$ and hence $p = (ab) \cdot p \in x$. Therefore $\phi_{\mathbb{P}} \circ \phi_{\mathbb{P}}^{-1} = \mathbf{1}_{\widehat{\mathbb{P}\#a}}$. It follows immediately that $\phi_{\mathbb{P}}^{-1}$ is natural in $\mathbb{P}$, and hence that ϕ is a natural isomorphism with inverse given by ϕ^{-1} as required. □

The transformation ϕ also interacts well with the unit $\{\cdot\}_{\downarrow}$ of the $\widehat{(-)}$ monad as follows.

3.3.3.6 Lemma. *The following diagram commutes.*

$$\begin{tikzcd}
(-)^{\#a} \arrow[r, "\{\cdot\}_{\downarrow}^{\#a}"] \arrow[dr, "\{\cdot\}_{(-)\#a}"'] & \widehat{(-)}^{\#a} \arrow[d, "\phi"] \\
& \widehat{(-)^{\#a}}
\end{tikzcd}$$

Proof. Let $\mathbb{P}$ be an object of $\mathbf{FMPre}_s$ and let $p \in \mathbb{P}^{\#a}$, then

$$\begin{aligned}
\big(\{\cdot\}_{\mathbb{P}^{\#a}}\big)p &= \{p' \in \mathbb{P}^{\#a} \mid p' \leq_{\mathbb{P}^{\#a}} p\} \qquad (3.3.3.7)\\
&= \{p' \in \mathbb{P} \mid a \# p' \wedge p' \leq_{\mathbb{P}} p\}\\
&= \phi_{\mathbb{P}}\{p' \in \mathbb{P} \mid p' \leq_{\mathbb{P}} p\}\\
&= \big(\phi_{\mathbb{P}} \circ \{\cdot\}_{\mathbb{P}}^{\#a}\big)p.
\end{aligned}$$

□

3.3.4 A relationship between δ_a and $\widehat{(-)}$

For the entirety of this section (3.3.4) let $s \subseteq_{\text{fin}} \mathbb{A}$ and $a \in \mathbb{A} \setminus s$ be fixed. In order to construct a functor $\delta_a^{+} : \mathbf{FMLin}_{s\dot\cup\{a\}} \to \mathbf{FMLin}_s$ that plays a similar role to $\delta_a : \mathbf{FMPre}_{s\dot\cup\{a\}} \to \mathbf{FMPre}_s$ it is helpful to be able to commute the functors δ_a and $\widehat{(-)}$ in order to construct a composition such as

$$\widehat{\delta_a\mathbb{P}} \longrightarrow \delta_a\widehat{\mathbb{P}} \xrightarrow{\delta_a f} \delta_a\widehat{\mathbb{Q}} \longrightarrow \widehat{\delta_a\mathbb{Q}} \qquad (3.3.4.1)$$

where $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$. The abstract results of chapter 6 demonstrate that this composition makes up a suitably canonical analogue of the functor δ_a on the linear categories, but for the purposes of this section it suffices to work more concretely. To that end, for each object $\mathbb{P}$ of $\mathbf{FMPre}_{s\dot\cup\{a\}}$ define the map $\theta_{\mathbb{P}}^{(a)} : \delta_a\widehat{\mathbb{P}} \to \widehat{\delta_a\mathbb{P}}$ by

$$\theta_{\mathbb{P}}^{(a)}(x') =_{\text{def}} \{p' \mid \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in x'@b\} \qquad (3.3.4.2)$$

and define ${\theta^{(a)}_{\mathbb{P}}}^{-1} : \widehat{\delta_a\mathbb{P}} \to \delta_a\widehat{\mathbb{P}}$ by

$$ {\theta^{(a)}_{\mathbb{P}}}^{-1}(x) =_{\mathrm{def}} \mathbf{fresh}\, b\, \mathbf{in}\, [b].\{p \mid [b].p \in x\}. \qquad (3.3.4.3)$$

As was the case for ϕ in 3.3.3, the notation is no accident as θ and θ^{-1} are natural and mutual inverses. It is also usual to drop the parameter a where it is otherwise clear from the context and hence to write $\theta^{(a)}$ simply as θ. A similar convention applies to ${\theta^{(a)}}^{-1}$ too.

It is perhaps interesting to note that the existence of this isomorphism seemed implausible when first it was realised that such a relationship was necessary for the development of this thesis. In personal correspondence Pitts pointed out that the isomorphism $\delta(X \to Y) \cong \delta X \to \delta Y$ in nominal sets[9] might provide useful insights, since letting $Y = \mathbf{2}$ draws attention to the isomorphism $\mathcal{P}\delta X \cong \delta\mathcal{P}X$ where $\mathcal{P}$, the powerset functor, corresponds closely to $\widehat{(-)}$. This observation led directly to the definitions above, and notice that if $\mathbb{P}$ is a nominal set then $\delta_a\mathbb{P} = \delta\mathbb{P}$ and if $\mathbb{P}$ has the discrete ordering then $\widehat{\mathbb{P}} = \mathcal{P}\mathbb{P}$ so that $\mathcal{P}\delta\mathbb{P} \cong \delta\mathcal{P}\mathbb{P}$ is a special case of the isomorphism given here.

3.3.4.4 Lemma. *If $a \notin s$ and $\mathbb{P}$ is an object of $\mathbf{FMPre}_{s\dot\cup\{a\}}$ then*

$$\theta_{\mathbb{P}} : \widehat{\delta_a\mathbb{P}} \cong \delta_a\widehat{\mathbb{P}} : \theta_{\mathbb{P}}^{-1}$$

is an isomorphism in $\mathbf{FMPre}_s$ which is natural in $\mathbb{P}$. Put differently, the map θ is a natural isomorphism of the functors

$$\delta_a\widehat{(-)}, \widehat{\delta_a -} : \mathbf{FMPre}_{s\dot\cup\{a\}} \rightrightarrows \mathbf{FMPre}_s.$$

Proof. It is necessary to show that $\theta_{\mathbb{P}}$ is well-defined, so let $x' \in \delta_a\widehat{\mathbb{P}}$ and show that $\theta_{\mathbb{P}}(x') \in \widehat{\delta_a\mathbb{P}}$ as follows. Let $p' \in \theta_{\mathbb{P}}(x')$ and let b be a fresh name, then $p'@b \in x'@b \in (ab)\cdot\widehat{\mathbb{P}}$ so that $p'@b \in (ab)\cdot\mathbb{P}$ and hence $p' \in \delta_a\mathbb{P}$. Let $p'' \leq_{\delta_a\mathbb{P}} p'$ and let b' be a fresh name, then $p''@b' \leq_{(ab')\cdot\mathbb{P}} p'@b' \in x'@b' \in (ab')\cdot\widehat{\mathbb{P}}$ so that $p''@b' \in x'@b'$ and hence $p'' \in x'$. Therefore $\theta_{\mathbb{P}}(x) \in \widehat{\delta_a\mathbb{P}}$ as required.

To see that $\theta_{\mathbb{P}}$ is supported by s let σ be a permutation that fixes s, let $x' \in \delta_a\widehat{\mathbb{P}}$ and show that $\sigma\cdot\theta_{\mathbb{P}}(x') = \theta_{\mathbb{P}}(\sigma\cdot x')$ as follows. Let $p' \in \sigma\cdot\theta_{\mathbb{P}}(x')$ and let b be a fresh name, then $(\sigma\cdot p')@b \in x'@b$ and hence $p'@b \in (\sigma\cdot x')@b$ as $\sigma\cdot b = b$ so that $p' \in \theta_{\mathbb{P}}(\sigma\cdot x')$. Conversely, let $p' \in \theta_{\mathbb{P}}(\sigma\cdot x')$ and let b be a fresh name, then $p'@b \in (\sigma\cdot x')@b$ and hence $(\sigma\cdot p')@b \in x'@b$ as $\sigma\cdot b = b$ so that $p' \in \sigma\cdot\theta_{\mathbb{P}}(x')$ as required. It is clear that $\theta_{\mathbb{P}}$ is monotone, so this has shown that $\theta_{\mathbb{P}}$ is an arrow of $\mathbf{FMPre}_s$.

To see that $\theta_{\mathbb{P}}$ is natural in $\mathbb{P}$, let $f : \mathbb{P} \to \mathbb{Q}$ be an arrow of $\mathbf{FMPre}_{s\dot\cup\{a\}}$, let $x' \in \delta_a\widehat{\mathbb{P}}$ and show that $(\theta_{\mathbb{Q}} \circ \delta_a\widehat{f})(x') = (\widehat{\delta_a f} \circ \theta_{\mathbb{P}})(x')$ as follows. Let

$q' \in (\theta_{\mathbb{Q}} \circ \delta_a \widehat{f})(x')$ and let b be a fresh name, then

$$q'@b \in (\delta_a \widehat{f}(x'))@b = \big((ab) \cdot \widehat{f}\big)(x'@b)$$

so that there exists $p \in x'@b$ such that $q'@b \leq_{\mathbb{Q}} ((ab) \cdot f)(p)$. Therefore

$$q' \leq_{\delta_a \mathbb{Q}} [b].((ab) \cdot f)(p) = [b].\big((ab) \cdot f\big)\big(([b].p)@b\big) = \delta_a f([b].p)$$

and $[b].p \in \theta_{\mathbb{P}}(x')$ so that $q' \in (\widehat{\delta_a f} \circ \theta_{\mathbb{P}})(x')$. Conversely, let $q' \in (\widehat{\delta_a f} \circ \theta_{\mathbb{P}})(x')$ so that there is $p' \in \theta_{\mathbb{P}}(x')$ such that $q' \leq_{\delta_a \mathbb{Q}} \delta_a f(p')$. Let b be a fresh name, then $p'@b \in x'@b$ and $q'@b \leq_{\mathbb{Q}} \big(\delta_a f(p')\big)@b = \big((ab) \cdot f\big)(p'@b)$. Therefore $q'@b \in \big((ab) \cdot \widehat{f}\big)(x'@b) = \big(\delta_a \widehat{f}(x')\big)@b$ and hence $q' \in (\theta_{\mathbb{Q}} \circ \delta_a \widehat{f})(x')$ as required.

It is necessary to show that $\theta_{\mathbb{P}}^{-1}$ is well-defined, so let $x \in \widehat{\delta_a \mathbb{P}}$ and show that $\theta_{\mathbb{P}}^{-1}(x) \in \delta_a \widehat{\mathbb{P}}$ as follows. Let b be a fresh name, and let $p \in \big(\theta_{\mathbb{P}}^{-1}(x)\big)@b$, then $[b].p \in x \subseteq \delta_a \mathbb{P}$ so that $[b].p \in \delta_a \mathbb{P}$ and hence $p \in (ab) \cdot \mathbb{P}$. Let $p' \leq_{\mathbb{P}} p$, then $[b].p' \leq_{\delta_a \mathbb{P}} [b].p \in x \in \widehat{\delta_a \mathbb{P}}$ so that $[b].p' \in x$ and hence $p' \in \big(\theta_{\mathbb{P}}^{-1}(x)\big)@b$. Therefore $\big(\theta_{\mathbb{P}}^{-1}(x)\big)@b \in (ab) \cdot \widehat{\mathbb{P}}$ so that $\theta_{\mathbb{P}}^{-1}(x) \in \delta_a \widehat{\mathbb{P}}$ as required.

To see that $\theta_{\mathbb{P}}^{-1}$ is supported by s, let σ be a permutation that fixes s, let $x \in \widehat{\delta_a \mathbb{P}}$, let b be a fresh name and show that $\sigma \cdot \theta_{\mathbb{P}}^{-1}(x) = \theta_{\mathbb{P}}^{-1}(\sigma \cdot x)$ as follows. Let $p \in \big(\sigma \cdot \theta_{\mathbb{P}}^{-1}(x)\big)@b$ then $[b].(\sigma^{-1} \cdot p) \in x$ and hence $[b].p \in \sigma \cdot x$ since $\sigma \cdot b = b$. Therefore $p \in \big(\theta_{\mathbb{P}}^{-1}(\sigma \cdot x)\big)@b$. Conversely let $p \in \big(\theta_{\mathbb{P}}^{-1}(\sigma \cdot x)\big)@b$ then $[b].p \in \sigma \cdot x$ and hence $[b].(\sigma^{-1} \cdot p) \in x$ since $\sigma \cdot b = b$. Therefore $p \in \big(\sigma \cdot \theta_{\mathbb{P}}^{-1}(x)\big)@b$ so that $\big(\sigma \cdot \theta_{\mathbb{P}}^{-1}(x)\big)@b = \big(\theta_{\mathbb{P}}^{-1}(\sigma \cdot x)\big)@b$ and hence $\sigma \cdot \theta_{\mathbb{P}}^{-1}(x) = \theta_{\mathbb{P}}^{-1}(\sigma \cdot x)$ as required. It is not hard to see that $\theta_{\mathbb{P}}^{-1}$ is monotone, so this has shown that $\theta_{\mathbb{P}}^{-1}$ is an arrow of $\mathbf{FMPre}_s$.

To see that $\theta_{\mathbb{P}}^{-1} \circ \theta_{\mathbb{P}} = \mathbf{1}_{\delta_a \widehat{\mathbb{P}}}$, let $x' \in \delta_a \widehat{\mathbb{P}}$ and show that $(\theta_{\mathbb{P}}^{-1} \circ \theta_{\mathbb{P}})(x') = x'$ as follows. Let b be a fresh name, then it suffices to show $\big((\theta_{\mathbb{P}}^{-1} \circ \theta_{\mathbb{P}})(x')\big)@b = x'@b$. Let $p \in x'@b$, then $([b].p)@b \in x'@b$ and hence $[b].p \in \theta_{\mathbb{P}}(x')$ so that finally $p \in \big(\theta_{\mathbb{P}}^{-1}(\theta_{\mathbb{P}}(x'))\big)@b$. Conversely, let $p \in \big(\theta_{\mathbb{P}}^{-1}(\theta_{\mathbb{P}}(x'))\big)@b$, then $[b].p \in \theta_{\mathbb{P}}(x')$ so that if c is a fresh name then $(bc) \cdot p \in x'@c$ and as b and c were fresh this means that $p \in ((bc) \cdot x')@b = x'@b$ as required.

To see that $\theta_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{-1} = \mathbf{1}_{\widehat{\delta_a \mathbb{P}}}$, let $x \in \widehat{\delta_a \mathbb{P}}$ and show that $(\theta_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{-1})(x) = x$ as follows. Let $p' \in x$ and let b be a fresh name, then $p'@b \in \big(\theta_{\mathbb{P}}^{-1}(x)\big)@b$ so that $p' \in (\theta_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{-1})(x)$. Conversely, let $p' \in (\theta_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{-1})(x)$ and let b be a fresh name, then $p'@b \in \big(\theta_{\mathbb{P}}^{-1}(x)\big)@b$ so that $p' = [b].(p'@b) \in x$ as required. It follows immediately that $\theta_{\mathbb{P}}^{-1}$ is natural in $\mathbb{P}$, and hence θ is a natural isomorphism with inverse given by θ^{-1} as required. □

The transformation θ also interacts well with the unit $\{\cdot\}_{\downarrow}$ of the $\widehat{(-)}$ monad as follows.

3.3.4.5 Lemma. *The following diagram commutes.*

$$\begin{tikzcd}
\delta_a\mathbb{P} \arrow[r, "\delta_a\{\cdot\}_{\mathbb{P}}"] \arrow[dr, "\{\cdot\}_{\delta_a\mathbb{P}}"'] & \delta_a\widehat{\mathbb{P}} \arrow[d, "\theta_{\mathbb{P}}"] \\
 & \widehat{\delta_a\mathbb{P}}
\end{tikzcd}$$

Proof. Let $\mathbb{P}$ be an object of $\mathbf{FMPre}_{s\dot{\cup}\{a\}}$ and let $p \in \delta_a\mathbb{P}$, then

$$\begin{aligned}
\big(\{\cdot\}_{\delta_a\mathbb{P}}\big)p &= \{p' \in \delta_a\mathbb{P} \mid p' \leq_{\delta_a\mathbb{P}} p\} \\
&= \{p' \in \delta_a\mathbb{P} \mid \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \leq_{(ab)\cdot\mathbb{P}} p@b\} \\
&= \{p' \in \delta_a\mathbb{P} \mid \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in \{p@b\}_{\downarrow}\} \\
&= \big\{p' \in \delta_a\mathbb{P} \mid \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in ([b].\{p@b\}_{\downarrow})@b\big\} \\
&= \theta_{\mathbb{P}}\big(\mathbf{fresh}\, b\, \mathbf{in}\, [b].\{p@b\}_{\downarrow})\big) \\
&= \big(\theta_{\mathbb{P}} \circ \delta_a\{\cdot\}_{\mathbb{P}}\big)p.
\end{aligned} \tag{3.3.4.6}$$

□

3.3.5 The Structure of $\mathbf{FMLin}_s$

The use of the word 'linear' in this context stems from the observation that each $\mathbf{FMLin}_s$ has enough structure to be understood as a categorical model of multiplicative-exponential linear logic[3]. It has other features that are important for the development of a rich domain theory too, and this section explores some of their details. This exploration closely follows the pattern of Winskel and Nygaard's exploration of **Lin** in the development of HOPLA[20].

3.3.5.1 Hom-sets. Recall from 2.2.3.11 that $\mathbb{P} \rightarrow_{\mathrm{fs}} \mathbb{Q}$ means all finitely-supported functions from $\mathbb{P}$ to $\mathbb{Q}$, and write $\mathbb{P} \rightarrow_s \mathbb{Q}$ for just those functions supported by s. The chain of isomorphisms

$$\begin{aligned}
\mathbf{FMLin}_s(\mathbb{P},\mathbb{Q}) &\cong \mathbb{P} \rightarrow_s \widehat{\mathbb{Q}} && \text{by 3.3.2.1} \\
&\cong \mathbb{P} \rightarrow_s (\mathbb{Q}^{\mathrm{op}} \rightarrow_{\mathrm{fs}} \mathbf{2}) && \text{by 3.3.1.2} \\
&\cong (\mathbb{P} \times \mathbb{Q}^{\mathrm{op}}) \rightarrow_s \mathbf{2} && \text{by currying} \\
&= (\mathbb{P}^{\mathrm{op}} \times \mathbb{Q})^{\mathrm{op}} \rightarrow_s \mathbf{2} \\
&\cong \{x \in \widehat{\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}} \mid \mathrm{supp}(x) \subseteq s\}
\end{aligned} \tag{3.3.5.2}$$

characterises hom-sets in $\mathbf{FMLin}_s$. Under this correspondence the ordering (given by inclusion) on $\widehat{\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}}$ gives rise to an ordering (given by pointwise inclusion) on $\mathbf{FMLin}_s(\mathbb{P},\mathbb{Q})$, and joins (given by union) in $\widehat{\mathbb{P}^{\mathrm{op}} \times \mathbb{Q}}$ give rise to joins (given by pointwise union) in $\mathbf{FMLin}_s(\mathbb{P},\mathbb{Q})$. Furthermore composition preserves joins in both its arguments, and in particular it is monotone. The order structure on the hom-sets gives rise to a commutative monoid structure, with multiplication given by binary pointwise union and unit given by the empty map $\varnothing$. Moreover composition is a monoid homomorphism which makes it possible to view each $\mathbf{FMLin}_s$ as being enriched over the category of commutative monoids and monoid homomorphisms.

3.3.5.3 Coproducts. Since left adjoints preserve coproducts, the disjoint union of the orders $\mathbb{P}_1$ and $\mathbb{P}_2$ forms the binary coproduct $\mathbb{P}_1 + \mathbb{P}_2$ in $\mathbf{FMLin}_s$. The ith injection $\mathbf{in}_i : \mathbb{P}_i \underset{\mathbf{L}}{\rightarrow} \mathbb{P}_1 + \mathbb{P}_2$ is given by the action of $\widehat{(-)}$ on the ith injection of the underlying preorders: concretely, $\mathbf{in}_1(x) = x \uplus \varnothing$ for example. The empty order $\mathbb{O}$ is the empty coproduct. More general coproducts are defined similarly.

3.3.5.4 Products. Since $\mathbf{FMLin}_s$ is enriched over **CMon** and it has binary coproducts it follows that the coproduct of the orders $\mathbb{P}_1$ and $\mathbb{P}_2$ is also a binary product $\mathbb{P}_1 \,\&\, \mathbb{P}_2$. The projections are defined by $\mathbf{out}_1 =_{\mathrm{def}} [\mathbf{1}_{\mathbb{P}_1}, \varnothing]$ and $\mathbf{out}_2 =_{\mathrm{def}} [\varnothing, \mathbf{1}_{\mathbb{P}_2}]$, and concretely for each i the map $\mathbf{out}_i : \mathbb{P}_1 \,\&\, \mathbb{P}_2 \underset{\mathbf{L}}{\rightarrow} \mathbb{P}_i$ is given by $\mathbf{out}_i x =_{\mathrm{def}} \{p \in \mathbb{P}_i \mid \mathbf{in}_i p \in x\}$. Furthermore, the commutative monoid structure means that injections and projections satisfy

$$\begin{gathered}\mathbf{out}_i \circ \mathbf{in}_i = \mathbf{1}_{\mathbb{P}_i} \quad \mathbf{out}_i \circ \mathbf{in}_j = \varnothing \ (i \neq j) \\ ((\mathbf{in}_1 \circ \mathbf{out}_1) \cup (\mathbf{in}_2 \circ \mathbf{out}_2)) = \mathbf{1}_{\mathbb{P}_1 + \mathbb{P}_2}\end{gathered} \tag{3.3.5.5}$$

and in short the object $\mathbb{P}_1 + \mathbb{P}_2$ is the biproduct of $\mathbb{P}_1$ and $\mathbb{P}_2$ with respect to this monoid. More general biproducts may be defined in $\mathbf{FMLin}_s$ in the same way.

3.3.5.6 Generalised Biproducts. In fact, if $(\mathbb{P}_\ell)_{\ell \in L}$ is any collection of FM-preorders where the mapping $\ell \mapsto \mathbb{P}_\ell$ is supported by s then it makes sense to define the object $\bigoplus_{\ell \in L} \mathbb{P}_\ell$ as the disjoint union of the $\mathbb{P}_\ell$. It is not hard to see that $\bigoplus_{\ell \in L} \mathbb{P}_\ell$ is supported by s and therefore an object of $\mathbf{FMLin}_s$. Furthermore, for each $\ell_0 \in L$ the ℓ_0th component $\mathbb{P}_{\ell_0}$ is an object of $\mathbf{FMLin}_{s \cup \mathrm{supp}(\ell_0)}$, and if $J : \mathbf{FMLin}_s \hookrightarrow \mathbf{FMLin}_{s \cup \mathrm{supp}(\ell_0)}$ is an inclusion of categories then the ℓ_0th injection is an arrow $\mathbf{in}_{\ell_0} : \mathbb{P}_{\ell_0} \underset{\mathbf{L}}{\rightarrow} J \bigoplus_{\ell \in L} \mathbb{P}_\ell$ of $\mathbf{FMLin}_{s \cup \mathrm{supp}(\ell_0)}$. Moreover if $\mathbb{Q}$ is an object of $\mathbf{FMLin}_s$ and $(f_\ell : \mathbb{P}_\ell \underset{\mathbf{L}}{\rightarrow} \mathbb{Q})_{\ell \in L}$ is a collection of arrows (in the appropriate categories) such that the mapping $\ell \mapsto f_\ell$ is supported by $s' \supseteq s$ then there is a unique $[f_\ell]_{\ell \in L} : \bigoplus_{\ell \in L} \mathbb{P}_\ell \underset{\mathbf{L}}{\rightarrow} \mathbb{Q}$ in $\mathbf{FMLin}_{s'}$ such that $[f_\ell]_{\ell \in L} \circ \mathbf{in}_{\ell_0} = f_{\ell_0}$ in $\mathbf{FMLin}_{s' \cup \mathrm{supp}(\ell_0)}$ for each ℓ_0. In short, $\bigoplus_{\ell \in L} \mathbb{P}_\ell$ behaves much like a coproduct, except that its injections do not necessarily all inhabit the same categories.

This generalised coproduct $\bigoplus_{\ell\in L}\mathbb{P}_\ell$ also has projections $\mathbf{out}_{\ell_0} : \bigoplus_{\ell\in L}\mathbb{P}_\ell \xrightarrow[\mathbf{L}]{} \mathbb{P}_{\ell_0}$ that satisfy conditions much like those of a product, except that again they also do not all lie in the same category. Also, the projections and injections interact as for a biproduct: $\mathbf{out}_\ell \circ \mathbf{in}_\ell = \mathbf{1}_{\mathbb{P}_i}$ in $\mathbf{FMLin}_{s'}$ where $s' \supseteq s \cup \mathrm{supp}(\ell)$ and $\mathbf{out}_\ell \circ \mathbf{in}_{\ell'} = \varnothing$ in $\mathbf{FMLin}_{s'}$ where $\ell \neq \ell'$ and $s' \supseteq s \cup \mathrm{supp}(\ell) \cup \mathrm{supp}(\ell')$. Finally $\bigcup_{\ell\in L}(\mathbf{in}_\ell \circ \mathbf{out}_\ell) = \mathbf{1}_{\bigoplus_{\ell\in L}\mathbb{P}_\ell}$ where the union is a join taken in the complete partial order

$$\widehat{\left(\bigoplus_{\ell\in L}\mathbb{P}_\ell\right)^{\mathrm{op}} \times \bigoplus_{\ell\in L}\mathbb{P}_\ell}$$

which contains all the function spaces $\mathbf{FMLin}_{s'}(\bigoplus_{\ell\in L}\mathbb{P}_\ell, \bigoplus_{\ell\in L}\mathbb{P}_\ell)$ by 3.3.5.2. In short, $\bigoplus_{\ell\in L}\mathbb{P}_\ell$ behaves much like a biproduct, except that its injections and projections are spread over many different categories.

3.3.5.7 Tensor. A tensor product on $\mathbf{FMLin}_s$ can be defined as the product $\mathbb{P}_1 \times \mathbb{P}_2$ of the underlying preorders. It is straightforward to see that this product is associative, and the (discrete) nominal preorder on the one-element set is a unit for this operation on both sides, so that this is a monoidal structure on $\mathbf{FMLin}_s$. Via 3.3.5.2,

$$\begin{aligned}\mathbf{FMLin}_s(\mathbb{P}\times\mathbb{Q},\mathbb{R}) &\cong \{x \in \widehat{(\mathbb{P}\times\mathbb{Q})^{\mathrm{op}}\times\mathbb{R}} \mid \mathrm{supp}(x)\subseteq s\} \\ &\cong \{x \in \widehat{\mathbb{P}^{\mathrm{op}}\times\mathbb{Q}^{\mathrm{op}}\times\mathbb{R}} \mid \mathrm{supp}(x)\subseteq s\} \\ &\cong \mathbf{FMLin}_s(\mathbb{P},\mathbb{Q}^{\mathrm{op}}\times\mathbb{R})\end{aligned} \qquad (3.3.5.8)$$

so that $\mathbf{FMLin}_s$ is closed with respect to the × tensor, and the internal function space is given by $\mathbb{Q} \multimap \mathbb{R} =_{\mathrm{def}} \mathbb{Q}^{\mathrm{op}} \times \mathbb{R}$.

3.3.5.9 Name Binding. If $a \in \mathbb{A} \setminus s$ then there is an adjunction on the FM-linear categories that is analogous to (and built from) the adjunction $(-)^{\#a} \dashv \delta_a$ described in lemma 3.2.1.15. Abstractly this adjunction arises from the general argument of section 6.3, but this section describes it in concrete terms. This adjunction is the key structure in the FM-linear categories that makes them a suitable setting for a domain theory that is sensitive to names and binding. More precisely, for any s and $a \notin s$ there is an adjunction

$$(-)^{\#a+} \dashv \delta_a^+ : \mathbf{FMLin}_s \leftrightarrows \mathbf{FMLin}_{s\dot\cup\{a\}} \qquad (3.3.5.10)$$

as follows. If $\mathbb{P}$ is an object then $\mathbb{P}^{\#a+} =_{\mathrm{def}} \mathbb{P}^{\#a}$ and $\delta_a^+\mathbb{P} =_{\mathrm{def}} \delta_a\mathbb{P}$. The unit and counit are given by $\widehat{\xi}$ and $\widehat{\zeta}$. If $f : \mathbb{P} \xrightarrow[\mathbf{L}]{} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_s$ then $f^{\#a+} : \mathbb{P}^{\#a} \xrightarrow[\mathbf{L}]{} \mathbb{Q}^{\#a}$ is the arrow of $\mathbf{FMLin}_{s\dot\cup\{a\}}$ defined concretely as

$$f^{\#a+}x =_{\mathrm{def}} \{q \in f(x_\downarrow) \mid a \# q\}. \qquad (3.3.5.11)$$

Also if $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_{s\dot{\cup}\{a\}}$ then $\delta_a^+ f : \delta_a\mathbb{P} \underset{\mathbf{L}}{\to} \delta_a\mathbb{Q}$ is the arrow of $\mathbf{FMLin}_s$ defined concretely as

$$\delta_a^+ fx =_{\text{def}} \{q \in \delta_a\mathbb{Q} \mid \exists p \in x.\ \mathbf{fresh}\, b\, \mathbf{in}\, q@b \in ((ab)\cdot f)(\{p@b\}_\downarrow)\}. \quad (3.3.5.12)$$

For the purposes of calculation these definitions are a little unwieldy and it is useful to have a slightly more abstract description of $(-)^{\#a+}$ and δ_a^+ in terms of $(-)^{\#a}$, δ_a, ϕ and θ as shown by the following results.

3.3.5.13 Lemma. *If $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_s$ then*

$$f^{\#a+} = \phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi_{\mathbb{P}}^{-1}.$$

Proof. Let $x \in \widehat{\mathbb{P}\#a}$. Suppose that $q \in f^{\#a+}x$ so that $q \in f(x_\downarrow)$ and $q \# a$. By definition $x \subseteq \phi_{\mathbb{P}}^{-1}x$ and $\phi_{\mathbb{P}}^{-1}x \in \widehat{\mathbb{P}}$ so that $x_\downarrow \subseteq \phi_{\mathbb{P}}^{-1}x$ and by the monotonicity of f it follows that $q \in f(\phi_{\mathbb{P}}^{-1}x)$ and hence that $q \in \big(\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi_{\mathbb{P}}^{-1}\big)x$. Conversely, suppose that $q \in \big(\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi_{\mathbb{P}}^{-1}\big)x$ then certainly $q \# a$ by definition of $\phi_{\mathbb{Q}}$. Also by the linearity of f there exists $p \in \phi_{\mathbb{P}}^{-1}x$ such that $q \in f\{p\}_\downarrow$ and by 3.3.3.4 if c is a fresh name then $(ac)\cdot p \in x$. As $a \# q$ and $a \# f$ it follows that $q = (ac)\cdot q \in (ac)\cdot f\{p\}_\downarrow = f\{(ac)\cdot p\}_\downarrow \subseteq f(x_\downarrow)$ and hence $q \in f^{\#a+}x$ as required. □

3.3.5.14 Lemma. *If $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_{s\dot{\cup}\{a\}}$ then*

$$\delta_a^+ f = \theta_{\mathbb{Q}} \circ \delta_a f \circ \theta_{\mathbb{P}}^{-1}.$$

Proof. Let $x \in \widehat{\delta_a\mathbb{P}}$, then

$$\begin{array}{rcl}
\big(\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta_{\mathbb{P}}^{-1}\big)(x) & = & \big(\theta_{\mathbb{Q}} \circ \delta_a f\big)\big(\mathbf{fresh}\, b\, \mathbf{in}\, [b].\{p \mid [b].p \in x\}\big) \\
& = & \theta_{\mathbb{Q}}\big(\mathbf{fresh}\, b\, \mathbf{in}\, [b].((ab)\cdot f)\{p \mid [b].p \in x\}\big) \\
& = & \{q' \in \delta_a\mathbb{Q} \mid \mathbf{fresh}\, b\, \mathbf{in} \\
& & \quad q'@b \in ((ab)\cdot f)\{p \mid [b].p \in x\}\} \\
\text{by linearity of } f & = & \{q' \in \delta_a\mathbb{Q} \mid \mathbf{fresh}\, b\, \mathbf{in} \\
& & \quad \exists p' \in x. b \# p' \wedge q'@b \in ((ab)\cdot f)\{p'@b\}_\downarrow\} \\
& = & \{q' \in \delta_a\mathbb{Q} \mid \exists p' \in x.\, \mathbf{fresh}\, b\, \mathbf{in} \\
& & \quad q'@b \in ((ab)\cdot f)\{p'@b\}_\downarrow\}, \\
& = & \delta_a^+ f(x)
\end{array} \quad (3.3.5.15)$$

as required. □

3.4 Continuity in FM Domain Theory

As in the development of HOPLA, linear maps are too restrictive to give a semantics for concurrent processes: it is desirable for a process to be able to perform actions spontaneously, without having received any input, but linear maps are join-preserving so that they preserve empty joins in particular and hence a linear process with no input can generate no output.

The solution to this issue is to consider maps which are continuous (with respect to some notion of approximation) rather than merely linear. In the development of HOPLA[20], continuity was chosen to mean ‘preserves directed joins’ but the discussion of section 3.4.1 demonstrates that this is not a suitable choice in $\mathbf{FMLin}_s$. Reconsidering directed sets as generalised sequences in section 3.4.2 suggests a more satisfactory notion of approximation. Section 3.4.3 demonstrates that it is simple to characterise the elements that are isolated with respect to this notion of approximation, and this gives rise to an appropriate exponential ! that captures the continuity in much the same way that the finite-join-completion exponential does in HOPLA, and section 3.4.4 demonstrates that this exponential can be characterised by a coKleisli construction.

Sections 3.4.5 and 3.4.6 demonstrate some technical results that show that the natural isomorphisms ϕ and θ defined in 3.3.3.2 and 3.3.4.2 interplay well with the exponential !, and section 3.4.7 demonstrates that the functors $(-)^{\#a+}$ and δ_a^+, as described in lemmas 3.3.5.13 and 3.3.5.14, preserve continuity as well as linearity. This ensures that these functors are suitable candidates for a binding adjunction on the continuous categories.

Finally, section 3.4.8 studies the structure of the induced categories of continuous maps and describes certain universal constructions that motivate the design of the language Nominal HOPLA in the following chapter.

3.4.1 Name-Binding is not Directed-Join Continuous

It is now necessary to speculate a little on what a nominal extension to the process calculus HOPLA may look like. This speculation is motivated by the development of new-HOPLA[38].

A key feature of a nominal extension to HOPLA would be terms of the form $\texttt{new}\,a.t$ whose intended meaning is to bind the name a in the outputs of t. In particular there will be a term $\texttt{new}\,a.\texttt{x}$ (where $\texttt{x}$ is a free variable) which receives a process as input in the variable $\texttt{x}$ and binds the name a in its output. The term $\texttt{x}$ (set in `teletype` typeface), is an *object-level* variable which is therefore treated with the usual informal methods regarding binding and α-conversion as discussed in chapter 4. In particular, the permutation action on $\mathtt{x}$ is discrete.

As the effect of $\mathtt{new}\,a\,\mathtt{.}t$ is to bind a, if b is a fresh name then it should be the case that $\mathtt{new}\,a\,\mathtt{.}t = \mathtt{new}\,b\,\mathtt{.}\big((ab)\cdot t\big)$, and in particular the name a should not be in the support of $\mathtt{new}\,a\,\mathtt{.}t$. The semantics of $\mathtt{new}\,a\,\mathtt{.}t$ are motivated by the semantics of $\mathtt{new}\,\alpha\,\mathtt{.}t$ from new-HOPLA:

$$\frac{\mathbb{P} : t \xrightarrow{p} t'}{\delta\mathbb{P} : \mathtt{new}\,\alpha\,\mathtt{.}t \xrightarrow{\mathtt{new}\,\alpha\,\mathtt{.}p} \mathtt{new}\,\alpha\,\mathtt{.}t'}$$

where α is a variable that ranges over names.

Now consider the term $\sum_{b\in\mathbb{A}} b\,\mathtt{:\,!nil}$ which outputs a name b nondeterministically chosen from $\mathbb{A}$. Recall that the type of a HOPLA term corresponds to the actions that it can do, so the type of this process will be (isomorphic to) $\mathbb{A}$ since its possible paths are (essentially) outputs of elements of $\mathbb{A}$. Furthermore, it is closed so its denotation will simply be a subset of $\mathbb{A}$, and since it can output any name it must be that

$$[\![\textstyle\sum_{b\in\mathbb{A}} b\,\mathtt{:\,!nil}]\!] = \mathbb{A}. \tag{3.4.1.1}$$

The term $\mathtt{new}\,a\,\mathtt{.}\big(\sum_{b\in\mathbb{A}} b\,\mathtt{:\,!nil}\big)$ should be able to output a bound name $[a].a$ if it is to satisfy the same operational semantics as new-HOPLA, and therefore

$$[a].a \in [\![\mathtt{new}\,a\,\mathtt{.}\big(\textstyle\sum_{b\in\mathbb{A}} b\,\mathtt{:\,!nil}\big)]\!]. \tag{3.4.1.2}$$

Recall that in HOPLA substitution is given by composition of denotations. In new-HOPLA, substitution under a $\mathtt{new}\,a\,\mathtt{.}(-)$ ensures that a is a fresh name before proceeding, but any a is fresh here since $\sum_{b\in\mathbb{A}} b\,\mathtt{:\,!nil}$ has empty support. Therefore

$$[\![\mathtt{new}\,a\,\mathtt{.}\big(\textstyle\sum_{b\in\mathbb{A}} b\,\mathtt{:\,!nil}\big)]\!] = [\![\mathtt{new}\,a\,\mathtt{.x}]\!] \circ [\![\textstyle\sum_{b\in\mathbb{A}} b\,\mathtt{:\,!nil}]\!] = [\![\mathtt{new}\,a\,\mathtt{.x}]\!](\mathbb{A}). \tag{3.4.1.3}$$

Now $\mathbb{A} = \bigcup_{s\subseteq_{\mathrm{fin}}\mathbb{A}} s$ is a directed join, so if $[\![\mathtt{new}\,a\,\mathtt{.x}]\!]$ is directed-join continuous then

$$[\![\mathtt{new}\,a\,\mathtt{.x}]\!](\mathbb{A}) = [\![\mathtt{new}\,a\,\mathtt{.x}]\!]\Bigg(\bigcup_{s\subseteq_{\mathrm{fin}}\mathbb{A}} s\Bigg) = \bigcup_{s\subseteq_{\mathrm{fin}}\mathbb{A}} [\![\mathtt{new}\,a\,\mathtt{.x}]\!]s, \tag{3.4.1.4}$$

so it follows that $[a].a \in \bigcup_{s\subseteq_{\mathrm{fin}}\mathbb{A}} [\![\mathtt{new}\,a\,\mathtt{.x}]\!]s$. Let $\{b_1,\ldots,b_n\}$ be such that

$$[a].a \in [\![\mathtt{new}\,a\,\mathtt{.x}]\!]\{b_1,\ldots,b_n\} = [\![\mathtt{new}\,a\,\mathtt{.x}]\!]\circ[\![b_1\,\mathtt{:\,!nil} + \ldots + b_n\,\mathtt{:\,!nil}]\!]. \tag{3.4.1.5}$$

Again, note that substitution under a $\mathtt{new}\,a\,\mathtt{.}(-)$ ensures that a is a fresh name in new-HOPLA. In particular a' may be chosen so that it is distinct from the

b_i and then $\mathtt{new}\,a\,\mathtt{.x} = \mathtt{new}\,a'\mathtt{.x}$ since the term $\mathtt{x}$ is unaffected by permutations of names and therefore has empty support. Therefore

$$[\![\mathtt{new}\,a\,\mathtt{.x}]\!] \circ [\![b_1\,\mathtt{:!nil} + \ldots + b_n\,\mathtt{:!nil}]\!] = \{[a'].b_1, \ldots, [a'].b_n\}. \qquad (3.4.1.6)$$

But this is a contradiction: $[a].a = [a'].a' \neq [a'].b_i$ for any i as a' was chosen to be fresh.

If it is unpalatable to take a sum over the set $\mathbb{A}$ of all names, notice that the argument above also applies to a term of the form $\sum_{b \in B} b\,\mathtt{:!nil}$ for any cofinite $B \subseteq_{\mathrm{fs}} \mathbb{A}$, since it rests only on the fact that B is infinite but finitely supported. From all of the speculative assumptions made in this example, the least convincing is that $[\![\mathtt{new}\,a\,\mathtt{.x}]\!]$ must preserve directed joins, so it is sensible to seek an alternative notion of continuity that includes maps such as $[\![\mathtt{new}\,a\,\mathtt{.x}]\!]$.

3.4.2 FM-Continuity

Continuity in domain theory arises from considering the process of physically realising some computation by performing a sequence of computational steps. Taking 'continuous' to mean 'preserves joins of increasing ω-sequences', continuous functions are those that are physically feasible[23, Thesis 5]. Classically it makes little difference to the elementary domain theory whether one uses increasing sequences or directed sets. After all, a poset has limits of all (ordinal-indexed) increasing sequences iff it has limits of all directed sets: increasing sequences give rise to directed sets, and conversely given a well-ordered directed set the well-ordering gives rise to an increasing sequence. The Axiom of Choice ensures that all directed sets have well-orderings. However, the Axiom of Choice does not hold in the theory of FM sets, so the equivalence between the use of increasing sequences and the use of directed sets breaks down here. It is therefore worthwhile to study increasing sequences in FM set theory in order to define a suitable notion of approximation in this setting. Firstly, notice that the supports of elements of total orders are very constrained as the following lemma shows.

3.4.2.1 Lemma. *Let $\mathbb{P}$ be a totally ordered FM-poset. Then each $p \in \mathbb{P}$ has* $\mathrm{supp}(p) \subseteq \mathrm{supp}(\mathbb{P})$.

Proof. Let $\mathbb{A} \ni b, c \# \mathbb{P}$, then for all $p \in \mathbb{P}$ it follows that $(bc) \cdot p \in \mathbb{P}$ too, so that either $p \leq (bc) \cdot p$ or $(bc) \cdot p \leq p$. If $p \leq (bc) \cdot p$ then $(bc) \cdot p \leq p$ by equivariance, so that $p = (bc) \cdot p$, and if $(bc) \cdot p \leq p$ then $p = (bc) \cdot p$ by a similar argument. However if $\mathrm{supp}(p) \nsubseteq \mathrm{supp}(\mathbb{P})$ then letting $b \in \mathrm{supp}(p) \setminus \mathrm{supp}(\mathbb{P})$ and $c \notin \mathrm{supp}(p, \mathbb{P})$ means that $(bc) \cdot p \neq p$ by lemma 2.2.2.10 which is a contradiction, so that $\mathrm{supp}(p) \subseteq \mathrm{supp}(\mathbb{P})$ as required. □

3.4.2.2 Definition. *An* ***increasing FM-sequence*** *in a FM-poset D is an increasing sequence (i.e. a monotone function from some (external) ordinal α to D) that is additionally finitely-supported.*

Notice that the ordinal α is totally ordered, so if $d_0 \leq d_1 \leq \dots$ is an increasing FM-sequence in D then each d_i in the sequence is supported by the finite set $\mathrm{supp}(\alpha, d)$. In other words, $\mathrm{supp}(\alpha, d)$ is a *uniform* support for the sequence d since it supports the elements of d uniformly. This observation suggests the following definition.

3.4.2.3 Definition. *An FM set X is* ***uniformly supported*** *by s if every element $x \in X$ is supported by s. An FM set* ***has uniform support*** *if there exists a finite set s that uniformly supports it.*

The (external) Axiom of Choice gives a well-ordering of any set X, and if X is uniformly supported by s then this well-ordering must also be supported by s, so that X can be well-ordered within FM set theory. It follows that classical increasing sequences are to directed sets as increasing FM-sequences are to directed sets *that are also uniformly supported*. More precisely, an FM preorder has limits of all increasing FM-sequences iff it has limits of all uniformly supported directed sets: increasing FM-sequences are already uniformly supported directed sets, and conversely any uniformly supported directed set can be wellordered within FM set theory and this gives rise to a corresponding increasing FM-sequence.

Returning to the example of $[\![\mathtt{new}\, a\,\mathtt{.}\,\mathtt{x}]\!]$, let $X \subseteq \widehat{\mathbb{A}}$ be directed and uniformly supported by a finite set s. Therefore every $x \in X$ is either a subset of s or a superset of $\mathbb{A} \setminus s$, so X is finite. Since X is also directed it contains a maximal element x so that

$$[\![\mathtt{new}\, a\,\mathtt{.}\,\mathtt{x}]\!] \left(\bigcup X\right) = [\![\mathtt{new}\, a\,\mathtt{.}\,\mathtt{x}]\!] x = \bigcup_{x' \in X} [\![\mathtt{new}\, a\,\mathtt{.}\,\mathtt{x}]\!] x' \tag{3.4.2.4}$$

and hence $[\![\mathtt{new}\, a\,\mathtt{.}\,\mathtt{x}]\!]$ does preserve joins of directed sets that have uniform supports. It turns out that approximation by uniformly-supported directed limits is a satisfactory notion of approximation in nominal domain theory.

3.4.3 FM-Isolated Elements

Considering $\widehat{\mathbb{P}}$ as a domain of meanings for processes of type $\mathbb{P}$, it is now sensible to investigate the structure of the isolated elements of this domain with respect to uniformly-supported directed approximations, since the isolated elements correspond to those computations that can be realised in finitely many steps.

3.4.3.1 Definition. *An element $P \in \widehat{\mathbb{P}}$ is **FM-isolated** iff for all uniformly supported directed sets $X \subseteq \widehat{\mathbb{P}}$, if $P \subseteq \bigcup X$ then there exists $x \in X$ such that $P \subseteq x$.*

For the purposes of this discussion it is unambiguous to call FM-isolated elements simply 'isolated'.

For example, every element of $\widehat{\mathbb{A}}$ is isolated. To see this, let $x \in \widehat{\mathbb{A}}$ be such that $x \subseteq \bigcup X$ where $X \subseteq \widehat{\mathbb{A}}$ is directed and uniformly supported by s, then every element of X is either a subset of s or a superset of $\mathbb{A} \setminus s$. Therefore X is finite, and since it is also directed it contains a maximal element x' and hence $x \subseteq x'$.

The usual definition of isolation in the theory of FM sets would be with respect to approximation by finitely-supported directed sets. With this definition, isolated elements of $\widehat{\mathbb{A}}$ are precisely the finite subsets of $\mathbb{A}$. Intuitively, the cofinite subsets (with support s) of $\mathbb{A}$ can also be represented by a finite process since if $B \subseteq \mathbb{A}$ is supported by s then it is only necessary to check the membership status of the elements of s and a single $a \notin s$ to characterise B. Working towards a generalisation of this intuition,

3.4.3.2 Definition. *If $\mathbb{P}$ is a FM-preorder, F a finite subset of $\mathbb{P}$ and s a finite set of names that contains* $\mathrm{supp}(\mathbb{P})$ *then define*

$$\langle F \rangle_s =_{\mathrm{def}} \bigcup_{\sigma \# s} \sigma \cdot F. \tag{3.4.3.3}$$

Every $x \in \widehat{\mathbb{A}}$ is of this form: either x is finite and hence $x = \langle x \rangle_{\mathrm{supp}(x)}$ or else x is cofinite and hence $x = \langle \{a\} \rangle_{\mathrm{supp}(x)}$ for any $a \in x$. Lemma 3.4.3.7 shows that the isolated elements of $\widehat{\mathbb{P}}$ are precisely those elements generated by sets of this form. Corollary 3.4.3.10 also shows that an appropriate s may be chosen relatively freely, as long as it is sufficiently large. The set $\langle F \rangle_s$ is generated from F by closure under the action of certain permutations, and the origin of the notation $\langle F \rangle_s$ is by analogy with a presentation of a group in terms of its generators: $\langle a, b, \ldots \mid R(a, b, \ldots) \rangle$.

3.4.3.4 Lemma. *If* $\mathrm{supp}(\mathbb{P}) \subseteq s \subseteq s' \subseteq_{\mathrm{fin}} \mathbb{A}$ *and* $F', F \subseteq_{\mathrm{fin}} \mathbb{P}$ *are such that* $F' \subseteq F_{\downarrow}$ *then* $\langle F' \rangle_{s'} \subseteq \langle F \rangle_{s\downarrow}$.

Proof. Let $p' \in \langle F' \rangle_{s'}$, then there exists $\sigma' \# s'$ such that $\sigma' \cdot p' \in F' \subseteq F_{\downarrow}$ so that $\sigma \cdot p' \in F_{\downarrow}$ and hence there exists $p \in F$ such that $\sigma \cdot p' \leq_{\mathbb{P}} p$. Also $s \subseteq s'$ so that $\sigma \# s$ and hence $\sigma^{-1} \cdot p \in \langle F \rangle_s$ which implies that $p' \in \langle F \rangle_{s\downarrow}$ as required. □

3.4.3.5 Lemma. *If* $\mathrm{supp}(\mathbb{P}) \subseteq s \subseteq s' \subseteq_{\mathrm{fin}} \mathbb{A}$ *and* $F \subseteq \mathbb{P}$ *is finite then there exists a finite* $F' \subseteq \mathbb{P}$ *such that* $\langle F \rangle_s = \langle F' \rangle_{s'}$*.*

Proof. Let $\bar{s} = s' \cup \bigcup_{p \in F} \mathrm{supp}(p)$ and note that since F is finite it follows that $\bar{s}$ is finite and $\mathrm{supp}(F) \subseteq \bar{s}$. Let s'' be a finite set of fresh names of greater cardinality than $\bar{s}$ and let $\Sigma = \{\sigma \mid \mathrm{supp}(\sigma) \subseteq (\bar{s} \cup s'') \setminus s\}$. Let $F' = \bigcup_{\sigma \in \Sigma} \sigma \cdot F$ and note that Σ is finite so that F' is also finite. Show that $\langle F \rangle_s = \langle F' \rangle_{s'}$ as follows.

Let $p \in \langle F \rangle_s$, then by definition there exists $\sigma \# s$ such that $p \in \sigma \cdot F$. By lemma 2.2.1.6 there exists σ_1, σ_2, σ_3 such that $\sigma = \sigma_1 \sigma_2 \sigma_3$ where $\sigma_1 \# \bar{s}$ and $\sigma_3 \# \bar{s}$ and $\mathrm{supp}(\sigma_2) \subseteq (\bar{s} \cup s'') \setminus s$.

Firstly, $\mathrm{supp}(F) \subseteq \bar{s}$ so that $\sigma_3 \cdot F = F$ and hence $p \in \sigma_1 \cdot \sigma_2 \cdot F$. Also, since $\sigma_2 \in \Sigma$ it follows that $p \in \sigma_1 \cdot F'$. Finally since $\sigma_1 \# \bar{s} \supseteq s'$ it follows that $p \in \langle F' \rangle_{s'}$ as required.

Conversely, let $p \in \langle F' \rangle_{s'}$, then there exists $\sigma \# s'$ such that $p \in \sigma \cdot F'$. By definition of F', there exists $\sigma' \in \Sigma$ such that $p \in \sigma \cdot \sigma' \cdot F$ and therefore $\sigma' \# s$. Furthermore, $s \subseteq s'$ so that $\sigma\sigma' \# s$ and hence $p \in \langle F \rangle_s$ as required. □

3.4.3.6 Lemma. *The support of* $\langle F \rangle_s$ *is contained in* s*.*

Proof. It is sufficient to show that if $\sigma \# s$ then $\sigma \cdot \langle F \rangle_s = \langle F \rangle_s$. Let $p \in \langle F \rangle_s$, then there exists $\sigma' \# s$ such that $p \in \sigma' \cdot F$. Then $\sigma \cdot p \in \sigma \cdot \sigma' \cdot F$ and $\sigma\sigma' \# s$ so that $\sigma \cdot p \in \langle F \rangle_s$ and hence $\langle F \rangle_s \subseteq \sigma \cdot \langle F \rangle_s$. The converse is similar. □

3.4.3.7 Lemma. *An element* $P \in \widehat{\mathbb{P}}$ *is isolated iff it is of the form* $\langle F \rangle_{s\downarrow}$ *where* F *is a finite subset of* $\mathbb{P}$ *and* s *is a finite set of names that supports* $\mathbb{P}$*.*

Proof. Suppose that $\langle F \rangle_{s\downarrow} \subseteq \bigcup X$ where $X \subseteq \widehat{\mathbb{P}}$ is uniformly supported and directed. In the light of lemma 3.4.3.5 suppose without loss of generality that s is large enough to be a uniform support for X. Let $F = \{p_1, \ldots, p_n\}$, then for each $i \in \{1, \ldots, n\}$, $p_i \in \langle F \rangle_s \subseteq \bigcup X$ so there exists $x_i \in X$ such that $p_i \in x_i$. Since X is directed, it contains an upper bound $\bar{x}$ for the x_i. If $p \in \langle F \rangle_{s\downarrow}$ then $p \leq_{\mathbb{P}} \sigma \cdot p_i$ for some $i \in \{1, \ldots, n\}$ and some $\sigma \# s$. Therefore $p \in \sigma \cdot x_i \subseteq \sigma \cdot \bar{x}$, but s is a uniform support for X so that $\sigma \cdot \bar{x} = \bar{x}$ and hence $p \in \bar{x}$. Therefore $\langle F \rangle_{s\downarrow} \subseteq \bar{x}$ so that $\langle F \rangle_{s\downarrow}$ is isolated.

Conversely, let $P \in \widehat{\mathbb{P}}$ be isolated, let $s = \mathrm{supp}(P) \cup \mathrm{supp}(\mathbb{P})$ and define $X \subseteq \widehat{\mathbb{P}}$ by $X = \{\langle F\rangle_{s\downarrow} \mid F \subseteq_{\mathrm{fin}} P\}$. From lemma 3.4.3.6 it follows that X is uniformly supported by s, and it is clear that if $F_1 \subseteq_{\mathrm{fin}} P$ and $F_2 \subseteq_{\mathrm{fin}} P$ then $F_1 \cup F_2 \subseteq_{\mathrm{fin}} P$ and $\langle F_1 \cup F_2\rangle_{s\downarrow}$ is an upper bound for $\langle F_1\rangle_{s\downarrow}$ and $\langle F_2\rangle_{s\downarrow}$ so that X is directed. Therefore by the isolation of P there exists $F \subseteq_{\mathrm{fin}} P$ such that $P \subseteq \langle F\rangle_{s\downarrow}$. Furthermore, $\langle F\rangle_{s\downarrow} \subseteq P$ as follows. Let $p \in \langle F\rangle_{s\downarrow}$, then there exists $\sigma \# s$ and $p' \in F$ such that $p \leq_{\mathbb{P}} \sigma \cdot p'$. Also, $F \subseteq P$ so that $p' \in P$, and $\sigma \# s$ so that $\sigma \cdot P = P$ and hence $p \leq_{\mathbb{P}} \sigma \cdot p' \in P \in \widehat{\mathbb{P}}$ so that $p \in P$. Therefore $P = \langle F\rangle_s$ as required. □

3.4.3.8 Lemma. *If σ is a permutation and $\sigma \# \mathbb{P}$ then*

$$\sigma \cdot \langle F\rangle_s = \langle \sigma \cdot F\rangle_{\sigma\cdot s}.$$

Proof. Let $p \in \sigma \cdot \langle F\rangle_s$ then it follows that there exists $\sigma' \# s$ such that $p \in \sigma \cdot \sigma' \cdot F = \sigma \cdot \sigma' \cdot \sigma^{-1} \cdot \sigma \cdot F$, and $\sigma\sigma'\sigma^{-1} \# \sigma \cdot s$ so that $p \in \langle \sigma \cdot F\rangle_{\sigma\cdot s}$. The converse case is similar. □

3.4.3.9 Lemma. *If $F \subseteq_{\mathrm{fin}} \mathbb{P}$ and* $\mathrm{supp}(\mathbb{P}) \subseteq s$ *then*

$$\langle F\rangle_s = \langle F\rangle_{\mathrm{supp}(\langle F\rangle_s, \mathbb{P})}.$$

Proof. Let $p \in \langle F\rangle_s$ then there exists $\sigma \# s$ such that $p \in \sigma \cdot F$. By lemma 3.4.3.6, $\mathrm{supp}(\langle F\rangle_s) \subseteq s$ so that $\sigma \# \langle F\rangle_s$; furthermore $\sigma \# \mathbb{P}$ as $\mathrm{supp}(\mathbb{P}) \subseteq s$ so that $p \in \langle F\rangle_{\mathrm{supp}(\langle F\rangle_s, \mathbb{P})}$. Conversely, let $p \in \langle F\rangle_{\mathrm{supp}(\langle F\rangle_s, \mathbb{P})}$ then there exists $\sigma \# \langle F\rangle_s, \mathbb{P}$ such that $p \in \sigma \cdot F = \iota \cdot \sigma \cdot F$ and $\iota \# \sigma \cdot s$ so it follows that $p \in \langle \sigma \cdot F\rangle_{\sigma\cdot s} = \sigma \cdot \langle F\rangle_s = \langle F\rangle_s$ as required. □

3.4.3.10 Corollary. *If $P \in \widehat{\mathbb{P}}$ is isolated and* $\mathrm{supp}(P, \mathbb{P}) \subseteq s$ *then there exists $F \subseteq_{\mathrm{fin}} \mathbb{P}$ such that $P = \langle F\rangle_{s\downarrow}$.*

Proof. If P is isolated then by 3.4.3.7 there exists finite $F' \subseteq \mathbb{P}$ and finite $s' \supseteq \mathrm{supp}(\mathbb{P})$ such that $P = \langle F'\rangle_{s'\downarrow}$. By 3.4.3.9 therefore $P = \langle F'\rangle_{\mathrm{supp}(P,\mathbb{P})\downarrow}$ and by 3.4.3.5 there exists $F \subseteq_{\mathrm{fin}} \mathbb{P}$ such that $P = \langle F\rangle_{s\downarrow}$ as required. □

3.4.4 Categories of FM-Continuous Maps

The discussion of 3.4.2 indicates that 'continuous' could be taken to mean 'preserves joins of uniformly-supported directed sets' or equivalently 'preserves joins of increasing sequences' which suggests the following definition.

3.4.4.1 Definition. *If $\mathbb{P}$, $\mathbb{Q}$ are preorders then say that a function $f : \widehat{\mathbb{P}} \to \widehat{\mathbb{Q}}$ is* ***FM-continuous*** *if it preserves all joins of uniformly-supported directed sets.*

Let $\mathbf{FMCts}_s$ be the category whose objects are FM-preorders $\mathbb{P}$, $\mathbb{Q}$, ... supported by s and whose arrows $\mathbb{P} \underset{\mathbf{C}}{\rightarrow} \mathbb{Q}$ are functions $\widehat{\mathbb{P}} \rightarrow \widehat{\mathbb{Q}}$ that are FM-continuous and supported by s. Note that in particular FM-linear maps are FM-continuous so that $\mathbf{FMLin}_s$ is a subcategory of $\mathbf{FMCts}_s$.

Following the development of HOPLA, it will be possible to characterise FM-continuous maps in terms of FM-linear maps whose domain is under an exponential ! which captures the appropriate notion of approximation.

3.4.4.2 Definition. *The preorder* $!\mathbb{P}$ *consists of the all elements of the form* $\langle F\rangle_s$ *where* $F \subseteq_{\mathrm{fin}} \mathbb{P}$ *and* s *supports* $\mathbb{P}$ *and the ordering is given by letting* $P \leq_{!\mathbb{P}} P'$ *whenever* $P \subseteq P'_{\downarrow}$.

Write $i_{\mathbb{P}}$ for the map $i_{\mathbb{P}} : !\mathbb{P} \rightarrow \widehat{\mathbb{P}}$ given by $i_{\mathbb{P}}P =_{\mathrm{def}} P_{\downarrow}$.

Each $\widehat{\mathbb{P}}$ is the free uniformly-supported-directed-join completion of $!\mathbb{P}$. In detail, this means that $\widehat{\mathbb{P}}$ has all uniformly-supported directed joins (indeed, it has all finitely-supported joins) and if C is a FM-poset that also has all uniformly-supported directed joins and $f : !\mathbb{P} \rightarrow C$ is a monotone finitely-supported function then there is a unique finitely-supported FM-continuous $f^{\ddagger} : \widehat{\mathbb{P}} \rightarrow C$ that such that the following diagram commutes.

$$\begin{tikzcd} !\mathbb{P} \arrow[r, "i_{\mathbb{P}}"] \arrow[dr, "f"'] & \widehat{\mathbb{P}} \arrow[d, "f^{\ddagger}"] \\ & C \end{tikzcd} \tag{3.4.4.3}$$

The function $f^{\ddagger}$ is given by

$$f^{\ddagger}x =_{\mathrm{def}} \bigvee\{fP_{\downarrow} \mid P \in !\mathbb{P}, P \subseteq x \text{ and } \mathrm{supp}(P) \subseteq \mathrm{supp}(x, \mathbb{P})\}. \tag{3.4.4.4}$$

Note that this is well-defined since the join is taken over a directed set that is uniformly supported by $\mathrm{supp}(f, x, \mathbb{P})$, so the join exists in C. It is also clear that $f^{\ddagger}$ is supported by the finite set $\mathrm{supp}(f, C, \mathbb{P})$.

3.4.4.5 Lemma. *The map* $f^{\ddagger}$ *is FM-continuous.*

Proof. It not clear that $f^{\ddagger}$ is even monotone. Let $x, x' \in \widehat{\mathbb{P}}$ be such that $x \subseteq x'$ and show that $f^{\ddagger}x \leq f^{\ddagger}x'$ as follows. By the monotonicity of f it is sufficient to show that for all $P \in !\mathbb{P}$ such that $P \subseteq x$ and $\mathrm{supp}(P) \subseteq \mathrm{supp}(x, \mathbb{P})$ there exists $P' \in !\mathbb{P}$ such that $P \subseteq P'_{\downarrow}$ and $P' \subseteq x'$ and $\mathrm{supp}(P') \subseteq \mathrm{supp}(x', \mathbb{P})$, so let $P \in !\mathbb{P}$ be such that $P \subseteq x$ and without loss of generality (by 3.4.3.10) write $P = \langle F\rangle_{\mathrm{supp}(x,x',\mathbb{P})}$. Define $P' = \langle F\rangle_{\mathrm{supp}(x',\mathbb{P})}$. It is certainly the case that $P \subseteq P'_{\downarrow}$ (by 3.4.3.4) and $\mathrm{supp}(P') \subseteq \mathrm{supp}(x', \mathbb{P})$ (by 3.4.3.6) so it remains to show that $P' \subseteq x'$. Let $p' \in P'$, then there exists $\sigma \mathrel{\#} x'$ such that $p' \in \sigma \cdot F$.

However, since $P = \langle F\rangle_{\mathrm{supp}(x,x',\mathbb{P})} \subseteq x \subseteq x'$, it must be that $p' \in \sigma \cdot x'$. Since $\sigma \# x'$ it follows that $p' \in x'$ as required.

Having now shown that $f^\ddagger$ is monotone, the proof of its continuity is standard. Let $X \subseteq \widehat{\mathbb{P}}$ be uniformly-supported and directed. For all $x \in X$ it is the case that $f^\ddagger x \leq f^\ddagger\big(\bigcup X\big)$ by monotonicity, so $f^\ddagger\big(\bigcup X\big)$ is an upper bound for $\{f^\ddagger x \mid x \in X\}$. It remains to show that it is the least such upper bound, i.e. for any U such that $f^\ddagger x \leq U$ for all $x \in X$ it is the case that $f^\ddagger\big(\bigcup X\big) \leq U$. For this, it is sufficient to show that U is an upper bound for

$$\{fP_\downarrow \mid P \in {!\mathbb{P}} \text{ and } P \subseteq \bigcup X \text{ and } \mathrm{supp}(P) \subseteq \mathrm{supp}(\bigcup X, \mathbb{P})\} \tag{3.4.4.6}$$

since $f^\ddagger\big(\bigcup X\big)$ is the least such. Therefore, suppose that $P \in {!\mathbb{P}}$ is such that $P \subseteq \bigcup X$ and $\mathrm{supp}(P) \subseteq \mathrm{supp}(\bigcup X, \mathbb{P})$. Then $P_\downarrow$ is isolated, and X is uniformly-supported and directed, so there exists $x \in X$ such that $P_\downarrow \subseteq x$. It is clear that $f^\ddagger P_\downarrow = fP_\downarrow$, so by monotonicity $fP_\downarrow \leq f^\ddagger x$ and $f^\ddagger x \leq U$ so that $fP_\downarrow \leq U$ as required. □

It is also the case that $\widehat{\mathbb{P}}$ is algebraic with respect to approximation by uniformly-supported directed sets, as the following lemma shows.

3.4.4.7 Lemma (Algebraicity of $\widehat{\mathbb{P}}$). *If $x \in \widehat{\mathbb{P}}$ then*

$$x = \bigcup\{P_\downarrow \mid P \in {!\mathbb{P}}, P \subseteq x \text{ and } \mathrm{supp}(P) \subseteq \mathrm{supp}(x, \mathbb{P})\}$$

Proof. Certainly $x \supseteq \bigcup\{P_\downarrow \mid P \in {!\mathbb{P}}, P \subseteq x \text{ and } \mathrm{supp}(P) \subseteq \mathrm{supp}(x, \mathbb{P})\}$. To see the converse let $p \in x$ and let $P = \langle\{p\}\rangle_{\mathrm{supp}(x,\mathbb{P})}$. It is clear that $p \in P$. From 3.4.3.6 it follows that $\mathrm{supp}(P) \subseteq \mathrm{supp}(x, \mathbb{P})$ and from 3.4.3.7 it follows that $P \in {!\mathbb{P}}$. Let $p' \in P_\downarrow$ then by definition there exists $\sigma \# \mathrm{supp}(x, \mathbb{P})$ such that $p' \leq_{\mathbb{P}} \sigma \cdot p$, but $\sigma \cdot p \in \sigma \cdot x = x$ and hence $p' \in x$. Therefore $P_\downarrow \subseteq x$ so as required it follows that

$$p \in \bigcup\{P_\downarrow \mid P \in {!\mathbb{P}}, P \subseteq x \text{ and } \mathrm{supp}(P) \subseteq \mathrm{supp}(x, \mathbb{P})\}.$$ □

As a consequence,

3.4.4.8 Corollary.

$$\widehat{\mathbb{P}} \cong \mathrm{Idl}({!\mathbb{P}})$$

Proof. The quotient of $!\mathbb{P}$ by the preorder equivalence is isomorphic to $\widehat{\mathbb{P}}^\circ$ by lemma 3.4.3.10, and the result follows by the same argument as in lemma 2.1.2.3. □

Lemma 3.4.4.7, together with the argument of lemma 2.1.2.4, also ensures the uniqueness of $f^\ddagger$, and in particular if f and g are FM-continuous maps $\mathbb{P} \underset{\mathbf{C}}{\rightarrow} \mathbb{Q}$ and $f \circ i_\mathbb{P} = g \circ i_\mathbb{P}$ then

$$f = (f \circ i_\mathbb{P})^\ddagger = (g \circ i_\mathbb{P})^\ddagger = g. \tag{3.4.4.9}$$

Notice that there is a natural FM-continuous $\eta_\mathbb{P} : \mathbb{P} \underset{\mathbf{C}}{\rightarrow} !\mathbb{P}$ given by $\eta_\mathbb{P} =_{\mathrm{def}} \{\cdot\}^\ddagger_{!\mathbb{P}}$, or concretely $\eta_\mathbb{P} X = \{P \in !\mathbb{P} \mid P \subseteq X\}$. In particular,

$$\eta_\mathbb{P} \circ i_\mathbb{P} = \{\cdot\}_{!\mathbb{P}}. \tag{3.4.4.10}$$

Furthermore for every FM-continuous $f : \mathbb{P} \underset{\mathbf{C}}{\rightarrow} \mathbb{Q}$ it is the case that the FM-linear function $(f \circ i_\mathbb{P})^\dagger : !\mathbb{P} \underset{\mathbf{L}}{\rightarrow} \mathbb{Q}$ satisfies

$$(f \circ i_\mathbb{P})^\dagger \circ \eta_\mathbb{P} = ((f \circ i_\mathbb{P})^\dagger \circ \eta_\mathbb{P} \circ i_\mathbb{P})^\ddagger = ((f \circ i_\mathbb{P})^\dagger \circ \{\cdot\}_{!\mathbb{P}})^\ddagger = (f \circ i_\mathbb{P})^\ddagger = f \tag{3.4.4.11}$$

by 3.3.1.4 and 3.4.4.3. Also, if $g : !\mathbb{P} \underset{\mathbf{L}}{\rightarrow} \widehat{\mathbb{Q}}$ is a FM-linear function such that $g \circ \eta_\mathbb{P} = f$ then

$$(f \circ i_\mathbb{P})^\dagger = (g \circ \eta_\mathbb{P} \circ i_\mathbb{P})^\dagger = (g \circ \{\cdot\}_{!\mathbb{P}})^\dagger = g. \tag{3.4.4.12}$$

Therefore $(f \circ i_\mathbb{P})^\dagger$ is the unique FM-linear function such that the following diagram commutes.

$$\begin{tikzcd}
\widehat{\mathbb{P}} \arrow[r, "\eta_\mathbb{P}"] \arrow[dr, "f"'] & \widehat{!\mathbb{P}} \arrow[d, "(f \circ i_\mathbb{P})^\dagger"] \\
& \widehat{\mathbb{Q}}
\end{tikzcd} \tag{3.4.4.13}$$

In other words if $J : \mathbf{FMLin}_s \hookrightarrow \mathbf{FMCts}_s$ then $\langle \eta_\mathbb{P}, !\mathbb{P} \rangle$ is an initial object of $(\mathbb{P} \downarrow J)$. If

$$!f =_{\mathrm{def}} (\eta_\mathbb{Q} \circ f \circ i_\mathbb{P})^\dagger : !\mathbb{P} \underset{\mathbf{C}}{\rightarrow} !\mathbb{Q}, \tag{3.4.4.14}$$

for any arrow $f : \mathbb{P} \underset{\mathbf{C}}{\rightarrow} \mathbb{Q}$ of $\mathbf{FMCts}_s$ then this means that there is an adjunction

$$\mathbf{FMLin}_s(!\mathbb{P}, \mathbb{Q}) \cong \mathbf{FMCts}_s(\mathbb{P}, \mathbb{Q}). \tag{3.4.4.15}$$

with ! as the left adjoint. Concretely, 3.4.4.14 means that if $X \in \widehat{!\mathbb{P}}$ then $!f(X) = \{Q \in !\mathbb{Q} \mid \exists P \in X.\ Q \subseteq fP_\downarrow\}$. The unit of the adjunction is η as defined above and the counit ϵ is defined as $\epsilon_\mathbb{P} =_{\mathrm{def}} i^\dagger_\mathbb{P}$. Therefore $\epsilon_\mathbb{P}$ is the unique FM-linear map such that

$$\epsilon_\mathbb{P} \circ \{\cdot\}_{!\mathbb{P}} = i_\mathbb{P}. \tag{3.4.4.16}$$

3.4.5 A relationship between $(-)^{\#a}$ and !

This section shows that there is an isomorphism — indeed, a bijection —

$$(!\mathbb{P})^{\#a} \cong !(\mathbb{P}^{\#a}) \tag{3.4.5.1}$$

where $a \in \mathbb{A} \setminus s$ and $\mathbb{P}$ is an object of $\mathbf{FMPre}_s$. This bijection can be seen as arising from the action of $\phi_{\mathbb{P}}$ and its inverse, which have particularly simple characterisations when attention is restricted to just the isolated elements. This isomorphism is a key ingredient in an adjunction on the continuous categories that is analogous to the 'binding' adjunction $(-)^{\#a+} \dashv \delta_a^+$ on the linear categories, as demonstrated in section 6.4. It is also used in giving a denotational semantics to pattern matching in the process calculus Nominal HOPLA as demonstrated in 5.2.2.7

3.4.5.2 Lemma. *If $F \subseteq_{\text{fin}} \mathbb{P}^{\#a}$ and $a \notin s' \supseteq s$ where $\mathbb{P}$ is an object of $\mathbf{FMPre}_s$ then*

$$\phi_{\mathbb{P}}\langle F\rangle_{s'\downarrow} = \langle F\rangle_{s'\cup\{a\}\downarrow}.$$

Moreover any element of $(!\mathbb{P})^{\#a}$ can be written as $\langle F\rangle_{s'}$ where $F \subseteq_{\text{fin}} \mathbb{P}^{\#a}$ and $a \notin s' \supseteq s$ and in this situation it is also the case that $\langle F\rangle_{s'\cup\{a\}} \in !(\mathbb{P}^{\#a})$.

Proof. Let $p' \in \phi_{\mathbb{P}}\langle F\rangle_{s'\downarrow}$, then $a \# p'$ and there exists $p \in F$ and $\sigma \# s'$ such that $p' \leq_{\mathbb{P}} \sigma \cdot p$. Let b be a fresh name, then $p = (ab)\cdot p$ and $p' = (ab)\cdot p'$ so that $p' \leq_{\mathbb{P}} (ab)\cdot\sigma\cdot(ab)\cdot p$. Since $b \# \sigma$ it follows by the equivariance of $\#$ that $s' \dot\cup \{a\} = (ab)\cdot(s' \dot\cup \{b\}) \# (ab)\cdot\sigma = (ab)\sigma(ab)$ so that $p' \in \langle F\rangle_{s'\dot\cup\{a\}\downarrow}$. Conversely, let $p' \in \langle F\rangle_{s'\dot\cup\{a\}\downarrow}$, then there exists $p \in F$ and $\sigma \# s' \dot\cup \{a\}$ such that $p' \leq_{\mathbb{P}^{\#a}} \sigma\cdot p$. Therefore $\sigma \# s'$ and $p' \leq_{\mathbb{P}} \sigma\cdot p$ and $a \# p$ so that $p' \in \phi_{\mathbb{P}}\langle F\rangle_{s'\downarrow}$ as required.

If $P \in (!\mathbb{P})^{\#a}$ then $P \# a$ so that $a \notin \text{supp}(P, \mathbb{P}, s)$ and hence by 3.4.3.10 there exists $F \subseteq_{\text{fin}} \mathbb{P}$ such that $P = \langle F\rangle_{\text{supp}(P,\mathbb{P},s)}$. If b is a fresh name then $(ab)\cdot\text{supp}(P,\mathbb{P},s) = \text{supp}(P,\mathbb{P},s)$ and hence $P = (ab)\cdot P = \langle (ab)\cdot F\rangle_{\text{supp}(P,\mathbb{P},s)}$. Also $a \# (ab)\cdot F$ and F is finite so that $a \# p$ for all $p \in (ab)\cdot F$ and hence $(ab)\cdot F \subseteq \mathbb{P}^{\#a}$ as required. Clearly $\langle F\rangle_{s'\dot\cup\{a\}} \in !(\mathbb{P}^{\#a})$ by 3.4.3.7. □

3.4.5.3 Lemma. *If $F \subseteq_{\text{fin}} \mathbb{P}^{\#a}$ and $a \in s' \supseteq s \not\ni a$ where $\mathbb{P}$ is an object of $\mathbf{FMPre}_s$ then*

$$\phi_{\mathbb{P}}^{-1}\langle F\rangle_{s'\downarrow} = \langle F\rangle_{s'\setminus\{a\}\downarrow}.$$

Moreover any element of $!(\mathbb{P}^{\#a})$ can be written as $\langle F\rangle_{s'}$ where $F \subseteq_{\text{fin}} \mathbb{P}^{\#a}$ and $a \in s' \supseteq s$ and in this situation it is also the case that $\langle F\rangle_{s'\setminus\{a\}} \in (!\mathbb{P})^{\#a}$.

Proof. Recall from 3.3.3.4 that

$$p' \in \phi_{\mathbb{P}}^{-1}\langle F\rangle_{s'\downarrow} \quad\Leftrightarrow\quad \mathbf{fresh}\, b\, \mathbf{in}\, p' \in (ab)\cdot\langle F\rangle_{s'\downarrow}. \tag{3.4.5.4}$$

Let $p' \in \phi_{\mathbb{P}}^{-1}\langle F\rangle_{s'\downarrow}$ and let b be a fresh name, then $p' \in (ab)\cdot\langle F\rangle_{s'\downarrow}$ so that by 3.4.3.8 there exists $p \in (ab)\cdot F$ and $\sigma \# (ab)\cdot s'$ such that $p' \leq_{\mathbb{P}\#a} \sigma\cdot p$. Therefore $p' \leq_{\mathbb{P}} \sigma\cdot(ab)\cdot(ab)\cdot p$ and $(ab)\cdot p \in F$. Furthermore if $a' \in s'\setminus\{a\}$ then $\sigma\cdot(ab)\cdot a' = a'$ as b is fresh and $\sigma \# (ab)\cdot s' \supseteq s'\setminus\{a\}$ so that $p' \in \langle F\rangle_{s'\setminus\{a\}\downarrow}$. Conversely, suppose that $p' \in \langle F\rangle_{s'\setminus\{a\}\downarrow}$, then there exists $p \in F$ and $\sigma \# s'\setminus\{a\}$ such that $p' \leq_{\mathbb{P}} \sigma\cdot p$. Let b be a fresh name. Since $p \in F$, $a \# p$ and hence $(ab)\cdot p = p$. Therefore $(ab)\cdot p' \leq (ab)\cdot\sigma\cdot(ab)\cdot p$ and $(ab)\sigma(ab) \# s'$ since $\sigma \# (s'\setminus\{a\}) \dot\cup \{b\} = (ab)\cdot s'$. Hence $(ab)\cdot p' \in \langle F\rangle_{s'\downarrow}$ so that $p' \in \phi_{\mathbb{P}}^{-1}\langle F\rangle_{s'\downarrow}$ as required.

Certainly $a \# \langle F\rangle_{s'\setminus a}$ by 3.4.3.6 so that $\langle F\rangle_{s'\setminus a} \in (!\mathbb{P})^{\#a}$ by 3.4.3.7. Finally if $P \in !(\mathbb{P}^{\#a})$ then by 3.4.3.10 there exists $F \subseteq \mathbb{P}^{\#a}$ such that $P = \langle F\rangle_{\mathrm{supp}(a,P,\mathbb{P},s)}$ as required. □

It therefore makes sense to define $\phi^! : (!-)^{\#a} \to !((-)^{\#a})$ and its inverse by letting

$$\phi^!_{\mathbb{P}}\langle F\rangle_{s'} =_{\mathrm{def}} \langle F\rangle_{s'\dot\cup\{a\}} \text{ where } a \notin s' \supseteq s \tag{3.4.5.5}$$

and

$$\phi^{!\ -1}_{\mathbb{P}}\langle F\rangle_{s'} =_{\mathrm{def}} \langle F\rangle_{s'\setminus\{a\}} \text{ where } a \in s' \supseteq s. \tag{3.4.5.6}$$

This latter map can importantly be characterised as follows.

3.4.5.7 Lemma. *The following diagram commutes.*

$$\begin{tikzcd}
!(\mathbb{P}^{\#a}) \arrow[r, "\phi^{!\ -1}_{\mathbb{P}}"] \arrow[d, "i_{\mathbb{P}\#a}"'] & (!\mathbb{P})^{\#a} \arrow[d, "i^{\#a}_{\mathbb{P}}"] \\
\widehat{\mathbb{P}\#a} \arrow[r, "\phi^{-1}_{\mathbb{P}}"] & \widehat{\mathbb{P}}^{\#a}
\end{tikzcd}$$

Proof. By 3.4.5.3. □

3.4.6 A relationship between δ_a and !

Similarly to section 3.4.5 this section shows that the isomorphism θ suggests a bijection

$$\delta_a!\mathbb{P} \cong !\delta_a\mathbb{P} \tag{3.4.6.1}$$

where $a \in \mathbb{A}\setminus s$ and $\mathbb{P}$ is an object of $\mathbf{FMPre}_{s\dot\cup\{a\}}$. As was the case with the relationship between $(-)^{\#a}$ and ! the path to this result is to observe that $\theta_{\mathbb{P}}$ and its inverse have particularly simple characterisations when attention is restricted to just the isolated elements, as shown below. This isomorphism, like the isomorphism $\phi^!$ defined above, is a key ingredient in an adjunction on the continuous categories that is analogous to the 'binding' adjunction $(-)^{\#a+} \dashv \delta_a^+$ on the linear categories, as demonstrated in section 6.4. Its development is sufficiently similar to that of $\phi^!$ that it seems sensible to present it here, although it is not revisited until section 6.4.

3.4.6.2 Lemma. *If b is a fresh name, $F \subseteq_{\text{fin}} (ab) \cdot \mathbb{P}$ and $b \in s' \supseteq s \not\ni a$ where $\mathbb{P}$ is an object of $\mathbf{FMPre}_{s \dot\cup \{a\}}$ then*

$$\theta_{\mathbb{P}}\big([b].\langle F \rangle_{s'\downarrow}\big) = \langle \{[b].p \mid p \in F\} \rangle_{s' \setminus \{b\} \downarrow}.$$

Moreover any element of $\delta_a !\mathbb{P}$ can be written in the form $[b].\langle F \rangle_{s'}$ where b is a fresh name, $F \subseteq_{\text{fin}} (ab) \cdot \mathbb{P}$ and $b \in s' \supseteq s$ and in this situation it is also the case that $\langle \{[b].p \mid p \in F\} \rangle_{s' \setminus \{b\}} \in !\delta_a \mathbb{P}$.

Proof. Let $p' \in \theta_{\mathbb{P}}\big([b].\langle F \rangle_{s'\downarrow}\big)$ and let c be a fresh name, then $p'@c \in (bc) \cdot \langle F \rangle_{s'\downarrow}$ so that there exists $p \in F$ and $\sigma \# s'$ so that $(bc) \cdot (p'@c) \leq_{(ab)\cdot\mathbb{P}} \sigma \cdot p$. Therefore $\big((bc) \cdot p'\big)@b \leq_{(ab)\cdot\mathbb{P}} \sigma \cdot p = \big([b].(\sigma \cdot p)\big)@b$ so that $(bc) \cdot p' \leq_{\delta_a \mathbb{P}} [b].(\sigma \cdot p)$. However $\sigma \# s' \ni b$ so $[b].(\sigma \cdot p) = \sigma \cdot ([b].p)$ and hence $p' \leq_{\delta_a \mathbb{P}} (bc) \cdot \sigma \cdot ([b].p)$ so it follows that $p' \in \langle \{[b].p \mid p \in F\} \rangle_{s' \setminus \{b\} \downarrow}$ as was required. Conversely, let $p' \in \langle \{[b].p \mid p \in F\} \rangle_{s' \setminus \{b\} \downarrow}$, then there exists $p \in F$ and $\sigma \# s' \setminus \{b\}$ such that $p' \leq_{\delta_a \mathbb{P}} \sigma \cdot ([b].p)$. Let c be a fresh name, then

$$p'@c \leq_{(ac)\cdot\mathbb{P}} \sigma \cdot (bc) \cdot p = (bc) \cdot (bc) \cdot \sigma \cdot (bc) \cdot p. \tag{3.4.6.3}$$

Furthermore, $(bc)\sigma(bc) \# s'$ so that $p'@c \in (bc) \cdot \langle F \rangle_{s'\downarrow}$ and hence as required $p' \in \theta_{\mathbb{P}}\big([b].\langle F \rangle_{s'\downarrow}\big)$.

If $P' \in \delta_a !\mathbb{P}$ and b is a fresh name then it follows that $P'@b \in (ab) \cdot !\mathbb{P}$ so that if $a \in s' \supseteq s \cup \text{supp}(P')$ then there exists $F \subseteq_{\text{fin}} \mathbb{P}$ such that $(ab) \cdot (P'@b) = \langle F \rangle_{s'}$ and hence $P' = [b].\big(\langle (ab) \cdot F \rangle_{(ab)\cdot s'}\big)$ and $b \in (ab) \cdot s'$ as required.

Finally, if $F' \subseteq (ab) \cdot \mathbb{P}$ it follows that $\{[b].p \mid p \in F'\} \subseteq \delta_a \mathbb{P}$ so that by 3.4.3.7

$$\langle \{[b].p \mid p \in F'\} \rangle_{s' \setminus \{b\}} \in !\delta_a \mathbb{P}. \tag{3.4.6.4}$$

□

3.4.6.5 Lemma. *If b is a fresh name, $F \subseteq_{\text{fin}} (ab) \cdot \mathbb{P}$ and $b \notin s' \supseteq s \not\ni a$ where $\mathbb{P}$ is an object of $\mathbf{FMPre}_{s \dot\cup \{a\}}$ then*

$$\theta_{\mathbb{P}}^{-1} \langle \{[b].p \mid p \in F\} \rangle_{s'\downarrow} = [b].\langle F \rangle_{s' \dot\cup \{b\} \downarrow}.$$

Moreover any element of $!\delta_a \mathbb{P}$ can be written in the form $\langle \{[b].p \mid p \in F\} \rangle_{s'}$ where b is a fresh name, $F \subseteq_{\text{fin}} (ab) \cdot \mathbb{P}$ and $b \notin s' \supseteq s$ and in this situation it is also the case that $[b].\langle F \rangle_{s' \dot\cup \{b\}} \in \delta_a !\mathbb{P}$.

Proof. Let c be a fresh name, then it is sufficient so show that

$$\big(\theta_{\mathbb{P}}^{-1}\langle\{[b].p \mid p \in F\}\rangle_{s'\downarrow}\big)@c = ([b].\langle F\rangle_{s'\dot\cup\{b\}\downarrow})@c. \tag{3.4.6.6}$$

Let $p' \in \big(\theta_{\mathbb{P}}^{-1}\langle\{[b].p \mid p \in F\}\rangle_{s'\downarrow}\big)@c$, then $[c].p' \in \langle\{[b].p \mid p \in F\}\rangle_{s'\downarrow}$ so that there exists $p \in F$ and $\sigma \# s'$ such that $[c].p' \leq_{\delta_a\mathbb{P}} \sigma \cdot [b].p$. Let d be a fresh name, then it follows that $(cd) \cdot p' \leq_{(ad)\cdot\mathbb{P}} (\sigma \cdot [b].p)@d = \sigma \cdot (bd) \cdot p$ so it is the case that $p' \leq_{(ac)\cdot\mathbb{P}} (cd) \cdot \sigma \cdot (bd) \cdot p$. Since $c \# F$ and F is finite $c \# p$ so that $p = (cd) \cdot p$; furthermore $(cd) = (bc)(cd)(bd)$, so that

$$(cd) \cdot \sigma \cdot (bd) \cdot p = (bc) \cdot (cd) \cdot (bd) \cdot \sigma \cdot (bd) \cdot (cd) \cdot p. \tag{3.4.6.7}$$

Also it is the case that $s' \cup \{d\} \# \sigma$ so that $s' \cup \{b\} \# (cd)(bd)\sigma(bd)(cd)$, from which it follows that $p' \in (bc) \cdot \langle F\rangle_{s'\dot\cup\{b\}\downarrow} = ([b].\langle F\rangle_{s'\dot\cup\{b\}\downarrow})@c$. Conversely, let $p' \in ([b].\langle F\rangle_{s'\dot\cup\{b\}\downarrow})@c = (bc) \cdot \langle F\rangle_{s'\dot\cup\{b\}\downarrow}$, then there exists $p \in F$ and $\sigma \# s' \dot\cup \{b\}$ such that $p' \leq_{(ac)\cdot\mathbb{P}} (bc) \cdot \sigma \cdot p$. Therefore

$$[c].p' \leq_{\delta_a\mathbb{P}} [c].\big((bc) \cdot \sigma \cdot p\big) = (bc) \cdot \sigma \cdot [b].p \tag{3.4.6.8}$$

and $(bc)\sigma \# s'$ so that $[c].p' \in \langle\{[b].p \mid p \in F\}\rangle_{s'\downarrow}$ and hence as required it is the case that $p' \in \big(\theta_{\mathbb{P}}^{-1}\langle\{[b].p \mid p \in F\}\rangle_{s'\downarrow}\big)@c$.

If $P \in !\delta_a\mathbb{P}$ then by 3.4.3.10 there exists $F' \subseteq \delta_a\mathbb{P}$ such that $P = \langle F'\rangle_{\mathrm{supp}(P,s,\mathbb{P})}$. Let b be a fresh name, then as $b \# F'$ and F' is finite it follows that $b \# p'$ for all $p' \in F'$, so that $F' = \{[b].(p'@b) \mid p' \in F'\}$ and $\{p'@b \mid p' \in F'\} \subseteq_{\mathrm{fin}} (ab) \cdot \mathbb{P}$ as required. Finally since $s' \supseteq s$ it follows that $(ab) \cdot (s' \dot\cup \{b\}) \supseteq (ab) \cdot s = s$. Therefore it is the case that $(ab) \cdot \langle F\rangle_{s'\dot\cup\{b\}} = \langle (ab) \cdot F\rangle_{(ab)\cdot(s'\dot\cup\{b\})} \in !\mathbb{P}$ so that as required $[b].\langle F\rangle_{s'\dot\cup\{b\}} \in \delta_a!\mathbb{P}$. □

It therefore makes sense to define $\theta^! : \delta_a! \to !\delta_a$ and its inverse by letting

$$\theta_{\mathbb{P}}^!\big([b].\langle F\rangle_{s'}\big) =_{\mathrm{def}} \langle\{[b].p \mid p \in F\}\rangle_{s'\setminus\{b\}} \text{ where } b \in s' \supseteq s \tag{3.4.6.9}$$

and

$$\theta_{\mathbb{P}}^{!\,-1}\langle\{[b].p \mid p \in F\}\rangle_{s'} =_{\mathrm{def}} [b].\langle F\rangle_{s'\dot\cup\{b\}} \text{ where } b \notin s' \supseteq s. \tag{3.4.6.10}$$

This latter map can importantly be characterised as follows.

3.4.6.11 Lemma. *The following diagram commutes.*

$$\begin{tikzcd}
!\delta_a\mathbb{P} \arrow[r, "\theta_{\mathbb{P}}^{!\,-1}"] \arrow[d, "i_{\delta_a\mathbb{P}}"'] & \delta_a!\mathbb{P} \arrow[d, "\delta_a i_{\mathbb{P}}"] \\
\widehat{\delta_a\mathbb{P}} \arrow[r, "\theta_{\mathbb{P}}^{-1}"] & \delta_a\widehat{\mathbb{P}}
\end{tikzcd}$$

Proof. By 3.4.6.5. □

3.4.7 Binding and Continuity

This section demonstrates that the actions of $(-)^{\#a+}$ and δ^{+}_{a}, as described in lemmas 3.3.5.13 and 3.3.5.14, preserve continuity as well as linearity. This ensures that these functors are suitable candidates for a binding adjunction on the continuous categories, as discussed in 3.4.8.26.

3.4.7.1 Lemma. *If $a \notin s$ and $f : \mathbb{P} \underset{\mathbf{C}}{\rightarrow} \mathbb{Q}$ is an arrow of $\mathbf{FMCts}_s$ then the composition*

$$\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}} : \widehat{\mathbb{P}\#a} \rightarrow \widehat{\mathbb{Q}\#a}$$

is also FM-continuous.

Proof. Let $X \subseteq \widehat{\mathbb{P}\#a}$ be directed and uniformly supported by s'. Without loss of generality assume that $a \in s'$ and $s \subseteq s'$. It is required to show that

$$\bigcup\{(\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}})(x) \mid x \in X\} = (\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}}) \left(\bigcup X\right).$$

Let $p \in \bigcup\{(\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}})(x) \mid x \in X\}$, then there exists $x \in X$ such that $p \in (\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}})(x)$. It is also the case that $x \subseteq \bigcup X$ so that by monotonicity $p \in (\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}}) (\bigcup X)$. Conversely let $p \in (\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}}) (\bigcup X)$ then $p \# a$ and $p \in f\left(\phi^{-1}_{\mathbb{P}} (\bigcup X)\right)$. By 3.4.4.7 it is the case that

$$\begin{aligned}\phi^{-1}_{\mathbb{P}} (\textstyle\bigcup X) \quad = \quad & \textstyle\bigcup\{P_{\downarrow} \mid P \in !\mathbb{P} \text{ and } P \subseteq \phi^{-1}_{\mathbb{P}} (\bigcup X) \\ & \text{and } \mathrm{supp}(P) \subseteq \mathrm{supp}(\phi^{-1}_{\mathbb{P}} (\textstyle\bigcup X))\}.\end{aligned} \tag{3.4.7.2}$$

Since f is FM-continuous there exists $P \in !\mathbb{P}$ such that $P \subseteq \phi^{-1}_{\mathbb{P}} (\bigcup X)$ and $\mathrm{supp}(P) \subseteq \mathrm{supp}(\phi^{-1}_{\mathbb{P}} (\bigcup X))$ and $p \in fP_{\downarrow}$. Notice that $\mathrm{supp}(\phi^{-1}_{\mathbb{P}} (\bigcup X)) \subseteq s'$ so it follows that by 3.4.3.10 there exists a finite set $\{p_1, \ldots, p_n\} \subseteq \phi^{-1}_{\mathbb{P}} (\bigcup X)$ such that $p \in f(\langle\{p_1, \ldots, p_n\}\rangle_{s'\downarrow})$. Let b be a fresh name, then from 3.3.3.4 it follows that $(ab) \cdot p_i \in \bigcup X$ for all i so that there exist $x_1, \ldots, x_n \in X$ such that $(ab) \cdot p_i \in x_i$. However, X is directed so there exists $x \in X$ such that $x_i \subseteq x$ for all i, and X is uniformly supported by s' so that $\mathrm{supp}(x) \subseteq s'$ and hence $\langle\{p_1, \ldots, p_n\}\rangle_{s'} \subseteq (ab) \cdot x \subseteq \phi^{-1}_{\mathbb{P}}(x)$. Therefore $p \in (\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}})(x)$ as required. □

It is useful to characterise the interaction between $\phi^{!}$ and the unit η of the ! comonad with the following technical lemma.

3.4.7.3 Lemma. *If $\mathbb{P}$ is an object of $\mathbf{FMPre}_s$ then*

$$\widehat{\phi^{!\,-1}_{\mathbb{P}}} \circ \eta_{\mathbb{P}\#a} = \phi_{!\mathbb{P}} \circ \eta^{\#a}_{\mathbb{P}} \circ \phi^{-1}_{\mathbb{P}}$$

Proof.

$$\begin{array}{cll}
\multicolumn{2}{l}{\widehat{\phi^{!}_{\mathbb{P}}{}^{-1}} \circ \eta_{\mathbb{P}\#a} \circ i_{\mathbb{P}\#a}} & \\
= & \widehat{\phi^{!}_{\mathbb{P}}{}^{-1}} \circ \{\cdot\}_{!(\mathbb{P}\#a)} & \text{by 3.4.4.10} \\
= & \{\cdot\}_{(!\mathbb{P})\#a} \circ \phi^{!}_{\mathbb{P}}{}^{-1} & \text{by naturality of } \{\cdot\}_{\downarrow} \\
= & \phi_{!\mathbb{P}} \circ \{\cdot\}^{\#a}_{!\mathbb{P}} \circ \phi^{!}_{\mathbb{P}}{}^{-1} & \text{by 3.3.3.6} \\
= & \phi_{!\mathbb{P}} \circ \eta^{\#a}_{\mathbb{P}} \circ i^{\#a}_{\mathbb{P}} \circ \phi^{!}_{\mathbb{P}}{}^{-1} & \text{by 3.4.4.10} \\
= & \phi_{!\mathbb{P}} \circ \eta^{\#a}_{\mathbb{P}} \circ \phi^{-1}_{\mathbb{P}} \circ i_{\mathbb{P}\#a} & \text{by 3.4.5.7.}
\end{array}$$

But using 3.4.7.1 the maps $\widehat{\phi^{!}_{\mathbb{P}}{}^{-1}} \circ \eta_{\mathbb{P}\#a}$ and $\phi_{!\mathbb{P}} \circ \eta^{\#a}_{\mathbb{P}} \circ \phi^{-1}_{\mathbb{P}}$ are both continuous, so the result follows by 3.4.4.9. □

3.4.7.4 Lemma. *If $a \notin s$ and $f : \mathbb{P} \underset{\mathbf{C}}{\rightarrow} \mathbb{Q}$ is an arrow of $\mathbf{FMCts}_{s\dot{\cup}\{a\}}$ then the composition*

$$\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}} : \widehat{\delta_a \mathbb{P}} \rightarrow \widehat{\delta_a \mathbb{Q}}$$

is also FM-continuous.

Proof. Let $X \subseteq \widehat{\delta_a \mathbb{P}}$ be directed and uniformly supported by s'. Without loss of generality assume that $a \in s'$ and $s \subseteq s'$. It is required to show that

$$\bigcup\{(\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}})(x) \mid x \in X\} = (\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}}) \left(\bigcup X\right).$$

Let $p \in \bigcup\{(\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}})(x) \mid x \in X\}$, then there exists $x \in X$ such that $p \in (\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}})(x)$. It is also the case that $x \subseteq \bigcup X$ so that by monotonicity $p \in (\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}}) (\bigcup X)$. Conversely let $p \in (\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}}) (\bigcup X)$ and let b be a fresh name. Therefore $p@b \in \left((\delta_a f \circ \theta^{-1}_{\mathbb{P}}) (\bigcup X)\right) @b = f\left(\left(\theta^{-1}_{\mathbb{P}} (\bigcup X)\right) @b\right)$. By 3.4.4.7 it is the case that

$$\begin{array}{rcl}
\left(\theta^{-1}_{\mathbb{P}} (\bigcup X)\right) @b & = & \bigcup\{P_{\downarrow} \mid P \in !\mathbb{P} \text{ and } \subseteq \left(\theta^{-1}_{\mathbb{P}} (\bigcup X)\right) @b \\
& & \qquad \text{and } \mathrm{supp}(P) \subseteq \mathrm{supp}(\left(\theta^{-1}_{\mathbb{P}} (\bigcup X)\right) @b)\}.
\end{array} \tag{3.4.7.5}$$

Since f is FM-continuous there exists $P \in !\mathbb{P}$ such that $P \subseteq \left(\theta^{-1}_{\mathbb{P}} (\bigcup X)\right) @b$ and $\mathrm{supp}(P) \subseteq \mathrm{supp}(\left(\theta^{-1}_{\mathbb{P}} (\bigcup X)\right) @b)$ and $p@b \in fP$. Notice the fact that $\mathrm{supp}(\left(\theta^{-1}_{\mathbb{P}} (\bigcup X)\right) @b) \subseteq s' \cup \{b\}$ so that by 3.4.3.10 there exists a finite set $\{p_1, \ldots, p_n\} \subseteq \left(\theta^{-1}_{\mathbb{P}} (\bigcup X)\right) @b$ such that $p@b \in f(\langle\{p_1, \ldots, p_n\}\rangle_{s' \cup \{b\}})$. As $b \# \bigcup X$ it follows that $[b].p_i \in \bigcup X$ and hence there exists $x_i \in X$ such that $[b].p_i \in x_i$ for all i. However X is directed so there exists $x \in X$ such that $x_i \subseteq x$ for all i, and X is uniformly supported by s' so that $\mathrm{supp}(x) \subseteq s'$ and hence $\langle\{[b].p_1, \ldots, [b].p_n\}\rangle_{s'} \subseteq x$. Using 3.4.6.5 it follows that

$$\langle\{p_1, \ldots, p_n\}\rangle_{s' \dot{\cup} \{b\}} = \left(\theta^{-1}_{\mathbb{P}} \langle\{[b].p_1, \ldots, [b].p_n\}\rangle_{s'}\right) @b \subseteq \left(\theta^{-1}_{\mathbb{P}} x\right) @b$$

so that $p@b \in f\left(\left(\theta^{-1}_{\mathbb{P}}(x)\right) @b\right) = \left((\delta_a f \circ \theta^{-1}_{\mathbb{P}})(x)\right) @b$ and hence as required $p \in (\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}})(x)$. □

Finally, it is useful to characterise the interaction between $\theta^!$ and the unit η of the ! comonad as follows.

3.4.7.6 Lemma. *If $\mathbb{P}$ is an object of $\mathbf{FMPre}_{s\dot\cup\{a\}}$ then*

$$\widehat{\theta^{!}_{\mathbb{P}}}^{-1} \circ \eta_{\delta_a\mathbb{P}} = \theta_{!\mathbb{P}} \circ \delta_a\eta_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{-1}$$

Proof.

$$\begin{array}{lll} \multicolumn{3}{l}{\widehat{\theta^{!}_{\mathbb{P}}}^{-1} \circ \eta_{\delta_a\mathbb{P}} \circ i_{\delta_a\mathbb{P}}} \\ = & \widehat{\theta^{!}_{\mathbb{P}}}^{-1} \circ \{\cdot\}_{!\delta_a\mathbb{P}} & \text{by 3.4.4.10} \\ = & \{\cdot\}_{\delta_a!\mathbb{P}} \circ \theta^{!}_{\mathbb{P}}{}^{-1} & \text{by naturality of } \{\cdot\}_{\downarrow} \\ = & \theta_{!\mathbb{P}} \circ \delta_a\{\cdot\}_{!\mathbb{P}} \circ \theta^{!}_{\mathbb{P}}{}^{-1} & \text{by 3.3.4.5} \\ = & \theta_{!\mathbb{P}} \circ \delta_a\eta_{\mathbb{P}} \circ \delta_a i_{\mathbb{P}} \circ \theta^{!}_{\mathbb{P}}{}^{-1} & \text{by 3.4.4.10} \\ = & \theta_{!\mathbb{P}} \circ \delta_a\eta_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{-1} \circ i_{\delta_a\mathbb{P}} & \text{by 3.4.6.11.} \end{array}$$

But using 3.4.7.4 the maps $\widehat{\theta^{!}_{\mathbb{P}}}^{-1} \circ \eta_{\delta_a\mathbb{P}}$ and $\theta_{!\mathbb{P}} \circ \delta_a\eta_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{-1}$ are both continuous, so the result follows by 3.4.4.9. □

3.4.8 The Structure of $\mathbf{FMCts}_s$

This section studies the rich structure of the categories $(\mathbf{FMCts}_s)_{s\subseteq_{\text{fin}}\mathbb{A}}$ and in particular describes the universal constructions that motivate the design of the language Nominal HOPLA in chapter 4.

3.4.8.1 Hom-sets. The chain of isomorphisms

$$\begin{array}{rcll} \mathbf{FMCts}_s(\mathbb{P},\mathbb{Q}) & \cong & \mathbf{FMLin}_s(!\mathbb{P},\mathbb{Q}) & \text{by 3.4.4.15} \\ & \cong & \{x \in \widehat{!\mathbb{P}^{\text{op}} \times \mathbb{Q}} \mid \text{supp}(x) \subseteq s\} & \text{by 3.3.5.2} \end{array} \tag{3.4.8.2}$$

characterises hom-sets in $\mathbf{FMCts}_s$. As in $\mathbf{FMLin}_s$ each hom-set may be endowed with a partial order structure, given by $\subseteq$, and joins given by union. Furthermore composition preserves joins in both its arguments, and in particular it is monotone.

More generally, if $(f_i)_{i\in I}$ is a collection of continuous maps $\mathbb{P} \underset{\mathbf{C}}{\to} \mathbb{Q}$ where the mapping $i \mapsto f_i$ is supported by s then their union $\bigcup_{i\in I} f_i$ is an element of $\widehat{!\mathbb{P}^{\text{op}} \times \mathbb{Q}}$ and it is not hard to see that it is supported by s. By 3.4.8.2 it makes sense to denote the corresponding arrow of $\mathbf{FMCts}_s$ as $\sum_{i\in I} f_i : \mathbb{P} \underset{\mathbf{C}}{\to} \mathbb{Q}$. Concretely, this map is given by a pointwise union.

3.4.8.3 Products. Since right adjoints preserve products, $\mathbf{FMCts}_s$ has finite products given by the disjoint union of the underlying preorders as in $\mathbf{FMLin}_s$. The product of the objects $\mathbb{P}_1$ and $\mathbb{P}_2$ is written as $\mathbb{P}_1 \mathbin{\&} \mathbb{P}_2$, and its ith projection is $\mathbf{out}_i$. As in $\mathbf{FMLin}_s$, $\mathbf{out}_i x =_{\mathrm{def}} \{p \in \mathbb{P}_i \mid \mathbf{in}_i p \in x\}$.

3.4.8.4 Definition. *If $f_i : \mathbb{Q} \underset{\mathbf{C}}{\to} \mathbb{P}_i$ are arrows of $\mathbf{FMCts}_s$ for $i \in \{1,2\}$ then write $\langle f_1, f_2\rangle^{\&}$ for the unique arrow such that*

$$\mathbf{out}_i \circ \langle f_1, f_2\rangle^{\&} = f_i \text{ for each } i. \tag{3.4.8.5}$$

If $g_i : \mathbb{P}_i \underset{\mathbf{C}}{\to} \mathbb{Q}_i$ are arrows of $\mathbf{FMCts}_s$ for $i \in \{1,2\}$ then define

$$g_1 \mathbin{\&} g_2 =_{\mathrm{def}} \langle g_1 \circ \mathbf{out}_1, g_2 \circ \mathbf{out}_2\rangle^{\&} : \mathbb{P}_1 \mathbin{\&} \mathbb{P}_2 \underset{\mathbf{C}}{\to} \mathbb{Q}_1 \mathbin{\&} \mathbb{Q}_2. \tag{3.4.8.6}$$

It is not hard to see that the operation

$$\& : \mathbf{FMCts}_s(\mathbb{P}_1, \mathbb{Q}_1) \times \mathbf{FMCts}_s(\mathbb{P}_2, \mathbb{Q}_2) \to \mathbf{FMCts}_s(\mathbb{P}_1 \mathbin{\&} \mathbb{P}_2, \mathbb{Q}_1 \mathbin{\&} \mathbb{Q}_2)$$

preserves all joins in each argument.

If $\mathbb{P}_1$ and $\mathbb{P}_2$ are objects of $\mathbf{FMPre}_s$ then it follows that there exists an isomorphism $m_{\mathbb{P}_1,\mathbb{P}_2} : \widehat{\mathbb{P}_1} \times \widehat{\mathbb{P}_2} \cong \widehat{\mathbb{P}_1 \mathbin{\&} \mathbb{P}_2}$ where if $x_i \in \widehat{\mathbb{P}_i}$ for $i \in \{1,2\}$ then

$$m_{\mathbb{P}_1,\mathbb{P}_2}\langle x_1, x_2\rangle =_{\mathrm{def}} x_1 \uplus x_2. \tag{3.4.8.7}$$

It is not hard to see that this isomorphism is supported by s and monotone, and hence an arrow of $\mathbf{FMPre}_s$. Furthermore it is straightforward to show that it is natural in $\mathbb{P}_1$ and $\mathbb{P}_2$. The isomorphism m can be used to perform calculations with products more easily than dealing directly with elements of $\widehat{\mathbb{P}_1 \mathbin{\&} \mathbb{P}_2}$. In particular,

$$\mathbf{out}_i \circ m_{\mathbb{P}_1,\mathbb{P}_2} = \pi_i : \widehat{\mathbb{P}_1} \times \widehat{\mathbb{P}_2} \to \widehat{\mathbb{P}_i}. \tag{3.4.8.8}$$

so that if $f_i : \mathbb{Q} \underset{\mathbf{C}}{\to} \mathbb{P}_i$ are arrows of $\mathbf{FMCts}_s$ for $i \in \{1,2\}$ then

$$\langle f_1, f_2\rangle^{\&} = m_{\mathbb{P}_1,\mathbb{P}_2} \circ \langle f_1, f_2\rangle \tag{3.4.8.9}$$

and if $g_i : \mathbb{P}_i \underset{\mathbf{C}}{\to} \mathbb{Q}_i$ are arrows of $\mathbf{FMCts}_s$ for $i \in \{1,2\}$ then

$$(g_1 \mathbin{\&} g_2) \circ m_{\mathbb{P}_1,\mathbb{P}_2} = m_{\mathbb{Q}_1,\mathbb{Q}_2} \circ (g_1 \times g_2). \tag{3.4.8.10}$$

This helps to make the presentation of a denotational semantics in the FM-continuous categories look a little more familiar.

Also, since each $\mathbf{FMCts}_s$ is cartesian it follows that it supports all the usual constructions in cartesian categories. A few of these constructions are described here to fix their notation. For any objects $\mathbb{P}_1$ and $\mathbb{P}_2$ there is a twist map

$$\varsigma_{\mathbb{P}_1,\mathbb{P}_2} =_{\mathrm{def}} \langle \mathbf{out}_2, \mathbf{out}_1\rangle^{\&} : \mathbb{P}_1 \mathbin{\&} \mathbb{P}_2 \underset{\mathbf{C}}{\to} \mathbb{P}_2 \mathbin{\&} \mathbb{P}_1 \tag{3.4.8.11}$$

In particular ς is natural in $\mathbb{P}_1$ and $\mathbb{P}_2$ in the sense that if $f_i : \mathbb{P}_i \underset{\mathbf{C}}{\to} \mathbb{Q}_i$ for $i \in \{1, 2\}$ then

$$(f_2 \mathbin{\&} f_1) \circ \varsigma_{\mathbb{P}_1,\mathbb{P}_2} = \varsigma_{\mathbb{Q}_1,\mathbb{Q}_2} \circ (f_1 \mathbin{\&} f_2). \tag{3.4.8.12}$$

Finally for any objects $\mathbb{P}$ of $\mathbf{FMCts}_s$ there is a diagonal map

$$\Delta_{\mathbb{P}} =_{\mathrm{def}} \langle \mathbf{1}_{\mathbb{P}}, \mathbf{1}_{\mathbb{P}} \rangle^{\&} : \mathbb{P} \underset{\mathbf{C}}{\to} \mathbb{P} \mathbin{\&} \mathbb{P}. \tag{3.4.8.13}$$

3.4.8.14 Coproducts. In $\mathbf{FMLin}_s$ the finite products were also coproducts and there were relationships between the injections and projections that made them into biproducts. This structure carries across to $\mathbf{FMCts}_s$, except that there may not be a mediating arrow in the coproduct diagram if its legs are continuous but not linear. In detail, the injections into the product $\mathbb{P}_1 \mathbin{\&} \mathbb{P}_2$ are given as in $\mathbf{FMLin}_s$ and the projections and injections satisfy

$$\begin{gathered} \mathbf{out}_i \circ \mathbf{in}_i = \mathbf{1}_{\mathbb{P}_i} \quad \mathbf{out}_i \circ \mathbf{in}_j = \varnothing \ (i \neq j) \\ ((\mathbf{in}_1 \circ \mathbf{out}_1) \cup (\mathbf{in}_2 \circ \mathbf{out}_2)) = \mathbf{1}_{\mathbb{P}_1 + \mathbb{P}_2}. \end{gathered} \tag{3.4.8.15}$$

3.4.8.16 Generalised Biproducts. Similarly to the situation in $\mathbf{FMLin}_s$, the object $\bigoplus_{\ell \in L} \mathbb{P}_\ell$ behaves as a 'generalised biproduct' in $\mathbf{FMCts}_s$ when the mapping $\ell \mapsto \mathbb{P}_\ell$ is supported by s. In detail, for each $\ell_0 \in L$ the ℓ_0th component $\mathbb{P}_{\ell_0}$ is an object of $\mathbf{FMLin}_{s \cup \mathrm{supp}(\ell_0)}$, and the ℓ_0th injection and projection are arrows $\mathbf{in}_{\ell_0}$ and $\mathbf{out}_{\ell_0}$ of $\mathbf{FMLin}_{s \cup \mathrm{supp}(\ell_0)}$. Also, the projections and injections interact as for a biproduct: $\mathbf{out}_\ell \circ \mathbf{in}_\ell = \mathbf{1}_{\mathbb{P}_i}$ in $\mathbf{FMLin}_{s'}$ where $s' \supseteq s \cup \mathrm{supp}(\ell)$ and $\mathbf{out}_\ell \circ \mathbf{in}_{\ell'} = \varnothing$ in $\mathbf{FMLin}_{s'}$ where $\ell \neq \ell'$ and $s' \supseteq s \cup \mathrm{supp}(\ell) \cup \mathrm{supp}(\ell')$. Finally $\bigcup_{\ell \in L} (\mathbf{in}_\ell \circ \mathbf{out}_\ell) = \mathbf{1}_{\bigoplus_{\ell \in L} \mathbb{P}_\ell}$ where the union is a join taken in the complete partial order

$$\widehat{\left(!\left(\bigoplus_{\ell \in L} \mathbb{P}_\ell\right)\right)^{\mathrm{op}} \times \bigoplus_{\ell \in L} \mathbb{P}_\ell}$$

which contains all the function spaces $\mathbf{FMCts}_{s'}(\bigoplus_{\ell \in L} \mathbb{P}_\ell, \bigoplus_{\ell \in L} \mathbb{P}_\ell)$ by 3.4.8.2.

3.4.8.17 Exponentials. If $\mathbb{P}$ and $\mathbb{Q}$ are objects of $\mathbf{FMCts}_s$ then there is an isomorphism $m^!_{\mathbb{P},\mathbb{Q}} : !\mathbb{P} \times !\mathbb{Q} \cong !(\mathbb{P} \mathbin{\&} \mathbb{Q})$ which maps a pair $\langle P, Q \rangle$ to the union $P \uplus Q$. To see that this is well-defined, pick $s' \supseteq \mathrm{supp}(P, Q, \mathbb{P}, \mathbb{Q})$ and write $P = \langle F \rangle_{s'}$ and $Q = \langle G \rangle_{s'}$ by 3.4.3.10, then it is straightforward to see that $P \uplus Q = \langle F \uplus G \rangle_{s'} \in !(\mathbb{P} \mathbin{\&} \mathbb{Q})$. Moreover, any element of $!(\mathbb{P} \mathbin{\&} \mathbb{Q})$ can be written as $\langle F \uplus G \rangle_{s'}$ and in this case $\langle F \rangle_{s'} \in !\mathbb{P}$ and $\langle G \rangle_{s'} \in !\mathbb{Q}$, so that $m^!_{\mathbb{P},\mathbb{Q}}$ is an isomorphism. It is straightforward to see that $m^!_{\mathbb{P},\mathbb{Q}}$ is supported by s and natural in $\mathbb{P}$ and $\mathbb{Q}$, and finally it is clearly linear in both its arguments.

Recall from 3.3.5.7 that $\mathbf{FMLin}_s$ is monoidal closed, so that each $(-) \times \mathbb{Q}$ has a right adjoint $\mathbb{Q} \multimap (-)$. Together with the natural isomorphism $m^!$ and the adjunction 3.4.4.15 this gives rise to the following chain of isomorphisms (naturally in $\mathbb{P}$ and $\mathbb{R}$).

$$\begin{aligned} \mathbf{FMCts}_s(\mathbb{P} \,\&\, \mathbb{Q}, \mathbb{R}) &\cong \mathbf{FMLin}_s(!(\mathbb{P} \,\&\, \mathbb{Q}), \mathbb{R}) \\ &\cong \mathbf{FMLin}_s(!\mathbb{P} \times !\mathbb{Q}, \mathbb{R}) \\ &\cong \mathbf{FMLin}_s(!\mathbb{P}, !\mathbb{Q} \multimap \mathbb{R}) \\ &\cong \mathbf{FMCts}_s(\mathbb{P}, !\mathbb{Q} \multimap \mathbb{R}) \end{aligned} \tag{3.4.8.18}$$

so that $\mathbb{Q} \to (-) =_{\mathrm{def}} !\mathbb{Q} \multimap (-)$ gives a right adjoint to the cartesian product $(-) \,\&\, \mathbb{Q}$ in $\mathbf{FMCts}_s$, so that $\mathbf{FMCts}_s$ is cartesian closed.

In detail, the exponential counit is $\mathbf{apply} : (!\mathbb{Q} \multimap (-)) \,\&\, \mathbb{Q} \underset{\mathbf{C}}{\to} \mathbf{1}$ where if $f \in \widehat{!\mathbb{Q} \multimap \mathbb{P}}$ and $y \in \widehat{\mathbb{Q}}$ then

$$\mathbf{apply}_{\mathbb{P}}(f \uplus y) = \{p \in \mathbb{P} \mid \exists Q \in !\mathbb{Q}.\ Q \subseteq y \wedge \langle Q, p \rangle \in f\}. \tag{3.4.8.19}$$

The exponential transpose of a continuous arrow $f : \mathbb{P} \,\&\, \mathbb{Q} \underset{\mathbf{C}}{\to} \mathbb{R}$ is written $\mathbf{abstract}(f) : \mathbb{P} \underset{\mathbf{C}}{\to} (!\mathbb{Q} \multimap \mathbb{R})$ where if $x \in \widehat{\mathbb{P}}$ then

$$\mathbf{abstract}(f)(x) = \{\langle Q, r \rangle \in !\mathbb{Q} \multimap \mathbb{R} \mid r \in f(x \uplus Q)\}. \tag{3.4.8.20}$$

3.4.8.21 Strong Monad. There is a map $n_{\mathbb{P},\mathbb{Q}} : \widehat{\mathbb{P}} \times \widehat{\mathbb{Q}} \to \widehat{\mathbb{P} \times \mathbb{Q}}$ defined by

$$n_{\mathbb{P},\mathbb{Q}}\langle x, y \rangle =_{\mathrm{def}} \{\langle p, q \rangle \in \mathbb{P} \times \mathbb{Q} \mid p \in x \wedge q \in y\}. \tag{3.4.8.22}$$

It is straightforward to show that n is natural in $\mathbb{P}$ and $\mathbb{Q}$. The composition

$$\widehat{\mathbb{P} \,\&\, !\mathbb{Q}} \xrightarrow{\eta_{\mathbb{P}} \,\&\, \mathbf{1}_{!\mathbb{Q}}} \widehat{!\mathbb{P} \,\&\, !\mathbb{Q}} \xrightarrow{m^{-1}_{!\mathbb{P},!\mathbb{Q}}} !\widehat{\mathbb{P}} \times \widehat{!\mathbb{Q}} \xrightarrow{n_{!\mathbb{P},!\mathbb{Q}}} \widehat{!\mathbb{P} \times !\mathbb{Q}} \xrightarrow{\widehat{m^!_{\mathbb{P},\mathbb{Q}}}} \widehat{!(\mathbb{P} \,\&\, \mathbb{Q})} \tag{3.4.8.23}$$

defines a natural strength map $S_{\mathbb{P},\mathbb{Q}}$ for the monad $(!, \eta, \epsilon_!)$. Concretely, if $x \in \widehat{\mathbb{P}}$ and $Y \in \widehat{!\mathbb{Q}}$ then

$$S_{\mathbb{P},\mathbb{Q}}(x \uplus Y) = \{P \uplus Q \mid P \subseteq x, P \in !\mathbb{P} \text{ and } Q \in Y\}. \tag{3.4.8.24}$$

It follows that if $x \in \widehat{\mathbb{P}}$ and $y \in \widehat{\mathbb{Q}}$ then

$$\begin{aligned} &S_{\mathbb{P},\mathbb{Q}} \circ (\mathbf{1}_{\mathbb{P}} \,\&\, \eta_{\mathbb{Q}})(x \uplus y) && (3.4.8.25) \\ &= S_{\mathbb{P},\mathbb{Q}} \circ (x \uplus \{Q \in !\mathbb{Q} \mid Q \subseteq y\}) \\ &= \{P \uplus Q \mid P \in !\mathbb{P} \wedge Q \in !\mathbb{Q} \wedge P \subseteq x \wedge Q \subseteq y\}) \\ &= \eta_{\mathbb{P} \,\&\, \mathbb{Q}}(x \uplus y). \end{aligned}$$

3.4.8.26 Name-Binding. There is an adjunction on the FM-continuous categories that is analogous to the adjunctions $(-)^{\#a} \dashv \delta_a$ and $(-)^{\#a+} \dashv \delta_a^+$ described in 3.2.1.15 and 3.3.5.9. This adjunction is the key structure in the FM-continuous categories that makes them a suitable setting for a domain theory that is sensitive to names and binding. More precisely, for any $s \subseteq_{\text{fin}} \mathbb{A}$ and $a \notin s$ there is an adjunction

$$(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s \dot\cup \{a\}} \tag{3.4.8.27}$$

as follows. If $\mathbb{P}$ is an object then $\mathbb{P}^{\#a++} =_{\text{def}} \mathbb{P}^{\#a}$ and $\delta_a^{++}\mathbb{P} =_{\text{def}} \delta_a\mathbb{P}$. The unit and counit are given by $\widehat{\xi}$ and $\widehat{\zeta}$. If $f : \mathbb{P} \underset{\mathbf{C}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMCts}_s$ then $f^{\#a++} : \mathbb{P}^{\#a} \underset{\mathbf{C}}{\to} \mathbb{Q}^{\#a}$ is defined by

$$f^{\#a++} =_{\text{def}} \phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi_{\mathbb{P}}^{-1}. \tag{3.4.8.28}$$

Also if $f : \mathbb{P} \underset{\mathbf{C}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMCts}_{s \dot\cup \{a\}}$ then $\delta_a^{++} f : \delta_a\mathbb{P} \underset{\mathbf{C}}{\to} \delta_a\mathbb{Q}$ is defined by

$$\delta_a^{++} f =_{\text{def}} \theta_{\mathbb{Q}} \circ \delta_a f \circ \theta_{\mathbb{P}}^{-1}. \tag{3.4.8.29}$$

Notice that the definitions above are the same as those for the adjunction $(-)^{\#a+} \dashv \delta_a^+$ described in 3.3.5.9. As shown in 3.4.7.1 and 3.4.7.4, the actions of $(-)^{\#a++}$ and δ_a^{++} do send continuous maps to continuous maps as required. That this genuinely is an adjunction follows from the abstract arguments of 6.4.2. Also, the action of $(-)^{\#a+}$ interacts well with the unit η of the ! comonad as follows.

3.4.8.30 Lemma. *The following diagram commutes.*

$$\begin{tikzcd}
(-)^{\#a} \arrow[r, "\eta^{\#a++}"] \arrow[dr, "\eta_{(-)^{\#a}}"'] & (!-)^{\#a} \arrow[d, "\widehat{\phi}^!"] \\
 & !((-)^{\#a})
\end{tikzcd}$$

Proof. Let $\mathbb{P}$ be an object of $\mathbf{FMCts}_s$, then

$$\begin{aligned}
\widehat{\phi^!_{\mathbb{P}}} \circ \eta^{\#a++}_{\mathbb{P}} &= \widehat{\phi^!_{\mathbb{P}}} \circ \phi_{!\mathbb{P}} \circ \eta^{\#a}_{\mathbb{P}} \circ \phi^{-1}_{\mathbb{P}} && \text{by 3.4.8.28} \\
&= \widehat{\phi^!_{\mathbb{P}}} \circ \widehat{\phi^!_{\mathbb{P}}}^{-1} \circ \eta_{\mathbb{P}^{\#a}} && \text{by 3.4.7.3} \\
&= \eta_{\mathbb{P}^{\#a}} && \text{as required.}
\end{aligned} \tag{3.4.8.31}$$

□

3.4.8.32 Freshness Assumptions. The machinery of freshness (such as the functor $(-)^{\#a}$ and the isomorphism $\phi : \widehat{(-)}^{\#a} \to \widehat{(-)^{\#a}}$) can be extended in order to model freshness with respect to a list $\vec{s'}$ of names. This can be used to capture the idea of 'freshness assumptions' in a type system: a variable of type $\mathbb{P}^{\#\vec{s'}}$ insists that it receives input that is fresh for s', and a term of type $\mathbb{P}^{\#\vec{s'}}$ avoids the names in s' in its evaluation. More precisely, if $s \cap s' = \varnothing$ and s' is the underlying set of a list $\vec{s'}$ of distinct names then there is a functor $(-)^{\#\vec{s'}} : \mathbf{FMPre}_s \to \mathbf{FMPre}_{s \dot{\cup} \vec{s'}}$ defined as

$$(-)^{\#[]} = \mathbf{1}_{\mathbf{FMPre}_s} \quad \text{and} \quad (-)^{\#a::\vec{s'}} = (-)^{\#a} \circ (-)^{\#\vec{s'}}. \tag{3.4.8.33}$$

Concretely this means that $\mathbb{P}^{\#\vec{s'}} = \{p \in \mathbb{P} \mid p \# s'\}$ where the order on $\mathbb{P}^{\#\vec{s'}}$ is given by the restriction of the order on $\mathbb{P}$, and the action of $(-)^{\#\vec{s'}}$ on arrows is also given by restriction. Therefore the order of the names in $\vec{s'}$ is unimportant, so the functor $(-)^{\#\vec{s'}}$ may be written simply $(-)^{\#s'}$.

It follows that the isomorphism $\phi^{(a)}$ of 3.3.3.2 can be used to construct an isomorphism $\phi^{(\vec{s'})} : \widehat{(-)}^{\#s'} \to \widehat{(-)^{\#s'}}$ by setting

$$\phi^{([])} = \mathbf{1}_{\widehat{(-)}} \quad \text{and} \quad \phi^{(a::\vec{s'})} = \phi^{(a)}_{(-)^{\#s'}} \circ \phi^{(s')^{\#a}}. \tag{3.4.8.34}$$

Concretely this means that if $x \in \widehat{\mathbb{P}}^{\#s'}$ then $\phi^{(\vec{s'})}_{\mathbb{P}} x = \{p \in x \mid p \# s'\}$. Therefore the order of the names in $\vec{s'}$ is again unimportant, so the transformation $\phi^{(\vec{s'})}$ may be written simply $\phi^{(s')}$. The inverse ${\phi^{(s')}}^{-1}$ is defined similarly, and by repeatedly applying 3.3.3.4 the inverse can be characterised concretely by letting $s' = \{a_1, \ldots, a_n\}$, then $p \in {\phi^{(s')}}^{-1}(x)$ if and only if there are n fresh and distinct names $c_1, \ldots, c_n$ such that $p \in (a_1 c_1) \ldots (a_n c_n) \cdot x$. It is clear that if s, s_1 and s_2 are mutually disjoint then

$$\phi^{(s_1 \dot{\cup} s_2)} = \phi^{(s_1)}_{(-)^{\#s_2}} \circ \phi^{(s_2)^{\#s_1}}. \tag{3.4.8.35}$$

A similar argument applies to the similarly-defined map

$$\phi^{!(s')} : (!-)^{\#s'} \to !((-)^{\#s'}) \tag{3.4.8.36}$$

and in particular if $\mathbb{P}$ is an object of $\mathbf{FMCts}_s$ then

$$\eta_{\mathbb{P}^{\#s'}} = \widehat{\phi^{!(s')}_{\mathbb{P}}} \circ \eta^{\#s'++}_{\mathbb{P}} \tag{3.4.8.37}$$

by repeatedly applying lemma 3.4.8.30.

Furthermore, the map $\tau^a : (-)^{\#a} \to J$ of 3.2.1.21 extends in the obvious fashion to a natural transformation $\tau^{(s')} : (-)^{\#s'} \to J$ where J is the inclusion functor. Concretely if $p \in \mathbb{P}^{\#s'}$ then $\tau^{(s')}_{\mathbb{P}}(p) =_{\text{def}} p$. The maps $\tau^{(s')}$ and $\phi^{(s')}$ can be used to construct the map

$$\tau^{(s')++} =_{\text{def}} \tau^{(s')}_{\widehat{(-)}} \circ {\phi^{(s')}}^{-1} : \widehat{(-)^{\#s'}} \to \widehat{(-)}. \tag{3.4.8.38}$$

3.4.8.39 Lemma. *The map $\tau^{(s')++}$ is a natural transformation of the type $(-)^{\# s'++} \to J : \mathbf{FMCts}_s \rightrightarrows \mathbf{FMCts}_{s \dot{\cup} s'}$ where J is the inclusion of categories.*

Proof. The naturality of $\tau^{(s')++}$ follows directly from that of $\tau^{(s')}$. It remains to show that each $\tau_{\mathbb{P}}^{(s')++}$ is FM-continuous so let $X \subseteq \widehat{\mathbb{P}\# s'}$ be directed and uniformly supported by s'', then it is necessary to show that

$$\tau_{\mathbb{P}}^{(s')++}\big(\bigcup X\big) = \bigcup_{x \in X} \tau_{\mathbb{P}}^{(s')++} x. \tag{3.4.8.40}$$

Let $s' = \{a_1, \ldots, a_n\}$. Let $p \in \tau_{\mathbb{P}}^{(s')++}\big(\bigcup X\big)$, let c_1, …, c_n be fresh and let $\sigma = (a_1 c_1) \ldots (a_n c_n)$. Therefore $p \in \sigma \cdot \bigcup X$ so there exists $x \in X$ such that $p \in \sigma \cdot x$. However, the c_i were chosen to be fresh for s'' so that as required $p \in \tau_{\mathbb{P}}^{(s')++}(x) \subseteq \bigcup_{x \in X} \tau_{\mathbb{P}}^{(s')++} x$. Conversely let $p \in \bigcup_{x \in X} \tau_{\mathbb{P}}^{(s')++} x$, let $x \in X$ be such that $p \in \tau_{\mathbb{P}}^{(s')++}(x)$, let c_1, …, c_n be fresh and let $\sigma = (a_1 c_1) \ldots (a_n c_n)$, then $p \in \sigma \cdot x \subseteq \sigma \cdot \bigcup X$ so that $p \in \tau_{\mathbb{P}}^{(s')++}\big(\bigcup X\big)$ as required. □

3.5 Conclusion

As promised, this chapter has developed a domain theory for nondeterministic processes with names by following the development of the domain theory behind the language HOPLA within the theory of nominal sets. An important insight was to demonstrate that a sensible notion of approximation in nominal domain theory is that of approximation by directed sets that are also uniformly supported. This notion of approximation gives rise to the collection of categories $(\mathbf{FMCts}_s)_{s \subseteq_{\text{fin}} \mathbb{A}}$ that are very rich in structure, and this structure motivates a process calculus, Nominal HOPLA, which is described in the next chapter. Nominal HOPLA is designed so that the universal constructions in $(\mathbf{FMCts}_s)_{s \subseteq_{\text{fin}} \mathbb{A}}$ correspond very closely to its denotational semantics, as shown in chapter 5. Furthermore, although some of the development of the continuous categories within this chapter may appear to be a little ad-hoc, chapter 6 gives a more universal view of the situation which justifies abstractly the definitions and discussion above. If desired, it is acceptable to bypass chapters 4 and 5 as chapter 6 is independent of their contents.

Chapter 4

The Syntax and Operational Semantics of Nominal HOPLA

Nominal HOPLA is an expressive calculus for higher-order processes with non-determinism and name-binding which can now be used to illustrate the domain theory of the previous chapter. The development of Nominal HOPLA follows closely that of HOPLA (a Higher-Order Process LAnguage)[20] and is inspired by the language new-HOPLA[38].

Section 2.2.4 shows that the category **NSet** is equivalent to the Schanuel topos **Sch** of pullback-preserving presheaves over $\mathbb{I}^{\mathrm{op}}$. In fact, Johnstone[13, Examples A2.1.11(g)] comments that **Sch** is a category of sheaves over $\mathbb{I}^{\mathrm{op}}$. Others[8, 4] have made sense of binding operators in a more general functor category $[\mathbb{I}, \mathcal{C}]$ for some suitable $\mathcal{C}$. Note that if $\mathcal{C} = \mathbf{Set}$ then this draws attention to the category of presheaves over $\mathbb{I}^{\mathrm{op}}$ which is different from **NSet** in that in the presheaf setting there is no well-defined *minimal* support in general, so that supports must be treated more explicitly than when working in **NSet**. Roughly speaking, HOPLA, new-HOPLA and Nominal HOPLA are related to each other in the same way that respectively set theory, presheaves over $\mathbb{I}^{\mathrm{op}}$ and sheaves over $\mathbb{I}^{\mathrm{op}}$ are related: HOPLA has no names, new-HOPLA treats names explicitly in its syntax, whereas the treatment of names in Nominal HOPLA is much more implicit.

In order to present Nominal HOPLA it is necessary to give the language an abstract syntax, and this syntax includes some binding operators such as the usual function abstraction $\lambda\,\mathtt{x}\,.\,t$ which binds free occurrences of the variable $\mathtt{x}$ in the term t. One of the earliest applications of nominal sets[22] was to formalise common informal arguments about syntax with binding. For example, when performing a proof by induction over the syntax of a language with binding operators, bound variables are commonly assumed to be fresh, and this can be shown to be a valid induction principle by representing the syntax within nominal sets, where variables are represented by names.

However, the structure of interest here is not the syntax of Nominal HOPLA but its semantics, and the binding of variables in its syntax is a distraction. To avoid confusion the binding of variables is treated in the usual informal fashion: bound variables are always distinct from the other variables in scope, and substitution silently avoids capturing free variables. In particular, if $\mathtt{x}$ is a variable and σ is a permutation of the set of atoms then $\sigma \cdot \mathtt{x} = \mathtt{x}$.

The design of Nominal HOPLA is motivated by universal constructions in the categories $(\mathbf{FMCts}_s)_{s \subseteq_{\mathrm{fin}} \mathbb{A}}$, such as their cartesian closed structure (for higher-order processes) and the adjunction $(-)^{\#a{+}{+}} \dashv \delta_a^{++}$ (for names and name binding). This motivation will become clear in chapter 5 which shows the details of its denotational semantics. Before that, the syntax (section 4.1), type system (section 4.2) and operational semantics (section 4.4) are presented.

This development follows extremely closely that of HOPLA by Nygaard and Winskel[20], translated to a nominal setting. The main differences are the mention of supports in typing judgements (although lemma 4.2.2.9 demonstrates that these may eventually be dropped) and the two new term formers $\mathtt{new}\,a\,.\,t$ and $t\,\mathtt{[}a\mathtt{]}$ which are very similar to the $\mathtt{new}\,\alpha\,.\,t$ and $t\,\mathtt{[}\alpha\mathtt{]}$ of Winskel and Zappa Nardelli's new-HOPLA[38], and which arise directly from the adjunction $(-)^{\#a{+}{+}} \dashv \delta_a^{++}$ of 3.4.8.27.

4.1 Syntax

4.1.1 Preliminaries

Fix a set of (term) variables $\mathtt{x}, \mathtt{y}, \ldots$ and a set of type variables $P, \ldots$, each with a discrete permutation action. Also fix a set $\mathcal{L}$ of nominal label-sets, also with the discrete permutation action. Labels are written $\ell, \ell_0, \ldots \in L \in \mathcal{L}$. Note that each $L \in \mathcal{L}$ does not necessarily have the discrete permutation action, so that for some labels ℓ and some permutations σ it may be the case that $\sigma \cdot \ell \neq \ell$.

4.1.2 Syntax of Types

Types are given by the grammar

$$\mathbb{P}, \mathbb{Q} ::= P \mid !\mathbb{P} \mid \mathbb{Q} \rightarrow \mathbb{P} \mid \delta\mathbb{P} \mid \bigoplus_{\ell \in L} \mathbb{P}_\ell \mid \mu_j \vec{P}.\ \vec{\mathbb{P}}, \tag{4.1.2.1}$$

where P is a type variable, $\vec{P}$ is a list of type variables, and $\mu_j \vec{P}.\ \vec{\mathbb{P}}$ binds $\vec{P}$. Note that this grammar allows types to be defined by *mutual* recursion, where $\mu_j \vec{P}.\ \vec{\mathbb{P}}$ defines the jth component of the mutual definition, and its first unfolding (substituting $\mu_k \vec{P}.\ \vec{\mathbb{P}}$ in for the kth variable for each k) is written $\mathbb{P}_j[\mu \vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$. A closed type is a type with no free variables, and in the following, closed types are normally simply called 'types'. The permutation action on types is the discrete action: for all types $\mathbb{P}$ and permutations σ, $\sigma \cdot \mathbb{P} = \mathbb{P}$. Intuitively, a process of type $!\mathbb{P}$ may perform an anonymous action — written '`!`' and pronounced 'bang' — and resume as a process of type $\mathbb{P}$. A process of type $\mathbb{Q} \rightarrow \mathbb{P}$ awaits input of type $\mathbb{Q}$ and acts as a process of type $\mathbb{P}$ on its receipt. A process of type $\delta\mathbb{P}$ behaves as a process of type $\mathbb{P}$ with one of its names bound, which can be used to model the dynamic generation of names. A process of type $\bigoplus_{\ell \in L} \mathbb{P}_\ell$ behaves as any of its components $\mathbb{P}_{\ell_0}$ with the actions of the component having been tagged by the label ℓ_0. The recursively-defined type $\mu_j \vec{P}.\ \vec{\mathbb{P}}$ is isomorphic to the jth component of its unfolding $\mathbb{P}_j[\mu \vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$ and so processes of recursively-defined types behave similarly to processes typed by their unfoldings. For a more detailed intuition about this system of types, see the operational semantics of Nominal HOPLA as described in section 4.4. Note that this type system corresponds closely to that of HOPLA: the only difference is the new type former δ.

4.1.3 Syntax of Environments

Environments are given by the grammar

$$\Gamma ::= () \mid \Gamma, \mathtt{x} : \mathbb{P}^{\# s} \tag{4.1.3.1}$$

where $\mathtt{x}$ ranges over variables, $\mathbb{P}$ ranges over types and s ranges over finite sets of names, and the variables in Γ are distinct from $\mathtt{x}$. In new-HOPLA[38] the typing environments of HOPLA were augmented by a separate set of freshness distinctions, written d, consisting of pairs of names and variables that were asserted to be fresh for each other. Although the syntax of nominal HOPLA's environments is quite different from those of new-HOPLA, there is a close correspondence between the two approaches. The syntax of environments in nominal HOPLA was chosen as it is a much cleaner way to represent common operations such as adding a freshness constraint or replacing a variable in a substitution operation. This representation of freshness constraints is very similar to that which is used in Nominal Equational Logic[6], Nominal Unification[33] and Nominal Algebra[11].

The intended meaning of $\mathtt{x} : \mathbb{P}^{\# s}$ is that the variable $\mathtt{x}$ takes values of type $\mathbb{P}$ that are assumed to be fresh for s. The set of environments may be equipped with a permutation action which simply permutes the freshness assumptions:

$$\sigma \cdot () = () \qquad \text{and} \qquad \sigma \cdot (\Gamma, \mathtt{x} : \mathbb{P}^{\# s}) = (\sigma \cdot \Gamma), \mathtt{x} : \mathbb{P}^{\#(\sigma \cdot s)}, \tag{4.1.3.2}$$

so that the support of an environment is the union of its freshness constraints. This is a finite support, so the collection of environments forms a nominal set.

Define Γ, Λ by the obvious recursion

$$\Gamma, () = \Gamma \qquad \text{and} \qquad \Gamma, (\Lambda, \mathtt{x} : \mathbb{P}^{\# s}) = (\Gamma, \Lambda), \mathtt{x} : \mathbb{P}^{\# s}, \tag{4.1.3.3}$$

which only makes sense when the variables in Γ and Λ are distinct. It is sometimes useful to be able to simultaneously alter all the freshness assumptions in an environment: define $\Gamma^{\# s}$ by

$$()^{\# s} = () \qquad \text{and} \qquad \left(\Gamma, \mathtt{x} : \mathbb{P}^{\# s'}\right)^{\# s} = \Gamma^{\# s}, \mathtt{x} : \mathbb{P}^{\# s' \cup s}, \tag{4.1.3.4}$$

and define $\Gamma|_s$ by

$$()|_s = () \qquad \text{and} \qquad \left(\Gamma, \mathtt{x} : \mathbb{P}^{\# s'}\right)|_s = \left(\Gamma|_s\right), \mathtt{x} : \mathbb{P}^{\# s' \cap s}. \tag{4.1.3.5}$$

Finally, omit $()$ where it is unambiguous to do so. In particular, write $\mathtt{x} : \mathbb{P}^{\# s}$ for the single-variable environment $(), \mathtt{x} : \mathbb{P}^{\# s}$.

4.1.4 Syntax of Terms

Terms are given by the following grammar, where $\mathtt{x}$ ranges over variables, a ranges over names, s over finite sets of names, p over actions (see 4.1.5), ℓ over labels and $\mathbb{P}$ over types. To build an intuition for this definition, it is worth studying it in parallel with the typing rules in section 4.2 and the operational semantics as defined in section 4.4. The term formers $\mathtt{new}\,a\mathtt{.}t$ and $t\mathtt{[}a\mathtt{]}$ are very close in meaning to the terms $\mathtt{new}\,\alpha\mathtt{.}t$ and $t\mathtt{[}\alpha\mathtt{]}$ of new-HOPLA, and the remaining terms correspond closely to their counterparts in both HOPLA and new-HOPLA.

$$
\begin{array}{lllll}
t, u & ::= & \mathtt{x} \;\Big|\; \mathtt{rec}\,\mathtt{x}\mathtt{.}t & & \text{variables; recursion} \\
& & \Big|\; \mathtt{!}t \;\Big|\; \mathtt{[}u\,\mathtt{>}\,p\mathtt{(x:}\mathbb{P}\,\mathtt{\#}\,s\mathtt{)}\;\mathtt{=>}\;t\mathtt{]} & & \text{prefixing; matching} \\
& & \Big|\; \lambda\mathtt{x}\mathtt{.}t \;\Big|\; t\mathtt{(}u\mathtt{:}\mathbb{P}\mathtt{)} & & \text{abstraction and application} \\
& & \Big|\; \mathtt{new}\,a\mathtt{.}t \;\Big|\; t\mathtt{[}a\mathtt{]} & & \text{binding and concretion} \\
& & \Big|\; \ell\mathtt{:}t \;\Big|\; \pi_\ell t & & \text{labelling} \\
& & \Big|\; \sum_{i\in I} t_i & & \text{nondeterministic sum} \\
& & \Big|\; \mathtt{abs}\,t \;\Big|\; \mathtt{rep}\,t & & \text{recursive types}
\end{array}
\tag{4.1.4.1}
$$

The forms

$$\mathtt{rec}\,\mathtt{x}\mathtt{.}t \qquad \mathtt{[}u\,\mathtt{>}\,p\mathtt{(x:}\mathbb{P}\,\mathtt{\#}\,s\mathtt{)}\;\mathtt{=>}\;t\mathtt{]} \qquad \lambda\mathtt{x}\mathtt{.}t \tag{4.1.4.2}$$

all bind $\mathtt{x}$ in t, and the set of free variables of t is defined in the usual way.

The nondeterministic sum may be over an infinite set I, but there are constraints to ensure that it behaves properly: the mapping $i \mapsto t_i$ is a finitely supported function from a nominal set I to the set of terms, and is such that there exists a finite set X of variables such that for all i the free variables of t_i are contained in X. Write $\mathtt{nil}$ for the term $\sum_{i\in\varnothing} t_i$.

Although the binding of variables is treated informally, the binding of the name a in the form $\mathtt{new}\,a\mathtt{.}t$ must be treated more carefully. Strictly speaking, the form $\mathtt{new}\,a\mathtt{.}t$ is the equivalence class of pairs of names and terms that contains $\langle a, t\rangle$ under the usual α-equivalence relation

$$\langle a, t\rangle \sim_\alpha \langle a', t'\rangle \quad \Leftrightarrow \quad \mathbf{fresh}\,b\,\mathbf{in}\,(ab)\cdot t = (a'b)\cdot t'. \tag{4.1.4.3}$$

Therefore for any fresh name b, $\mathtt{new}\,a\mathtt{.}t = \mathtt{new}\,b\mathtt{.}(ab)\cdot t$, where the equality in this statement is literally equality, not simply equivalence-up-to-α.

4.1.5 Syntax of Actions

The operational semantics of Nominal HOPLA, as defined in section 4.4, is given in the style of a labelled transition system. The grammar of actions, labelling the transitions in the operational semantics, is given as follows. The symbol t ranges over closed terms, a ranges over names and ℓ over labels.

$$\begin{array}{lcll}
p & ::= & \mathtt{!} & \text{prefixing} \\
 & \mid & \ell\mathtt{:}p & \text{labelled actions} \\
 & \mid & t \mapsto p & \text{higher-order actions} \\
 & \mid & \mathtt{abs}\ p & \text{recursive type actions} \\
 & \mid & \mathtt{new}\,a.\,p & \text{new name actions}
\end{array} \tag{4.1.5.1}$$

The form $\mathtt{new}\,a.\,p$ binds the name a in the same way that a is bound in the term $\mathtt{new}\,a\,.t$.

4.1.6 Permutations on Terms and Actions

Naturally enough, actions and terms form nominal sets where the permutation action is given by a straightforward structural recursion as follows.

$$\begin{array}{rcl}
\sigma\cdot\mathtt{x} & = & \mathtt{x} \\
\sigma\cdot(\mathtt{!}t) & = & \mathtt{!}(\sigma\cdot t) \\
\sigma\cdot(\lambda\,\mathtt{x}.t) & = & \lambda\,\mathtt{x}.(\sigma\cdot t) \\
\sigma\cdot(\mathtt{new}\,a\,.t) & = & \mathtt{new}\,(\sigma a).(\sigma\cdot t) \\
\sigma\cdot(\ell\mathtt{:}t) & = & (\sigma\cdot\ell)\mathtt{:}(\sigma\cdot t) \\
\sigma\cdot(\mathtt{abs}\,t) & = & \mathtt{abs}\,(\sigma\cdot t)
\end{array}
\qquad
\begin{array}{rcl}
\sigma\cdot(\mathtt{rec}\,\mathtt{x}.t) & = & \mathtt{rec}\,\mathtt{x}.(\sigma\cdot t) \\
\sigma\cdot\left(\sum_{i\in I}t_i\right) & = & \sum_{i\in I}(\sigma\cdot t_{\sigma^{-1}\cdot i}) \\
\sigma\cdot(t(u\mathtt{:}\mathbb{P})) & = & (\sigma\cdot t)((\sigma\cdot u)\mathtt{:}\mathbb{P}) \\
\sigma\cdot(t[a]) & = & (\sigma\cdot t)[(\sigma a)] \\
\sigma\cdot(\pi_\ell t) & = & \pi_{(\sigma\cdot\ell)}(\sigma\cdot t) \\
\sigma\cdot(\mathtt{rep}\,t) & = & \mathtt{rep}\,(\sigma\cdot t)
\end{array}$$

$$\sigma\cdot([u > p(\mathtt{x}\mathtt{:}\mathbb{P}\ \#\ s)\ \mathtt{=>}\ t]) = [(\sigma\cdot u) > (\sigma\cdot p)(\mathtt{x}\mathtt{:}\mathbb{P}\ \#\ (\sigma\cdot s))\ \mathtt{=>}\ (\sigma\cdot t)]$$

$$\begin{array}{rcl}
\sigma\cdot\mathtt{!} & = & \mathtt{!} \\
\sigma\cdot(t\mapsto p) & = & (\sigma\cdot t)\mapsto(\sigma\cdot p) \\
\sigma\cdot(\mathtt{abs}\ p) & = & \mathtt{abs}\ (\sigma\cdot p)
\end{array}
\qquad
\begin{array}{rcl}
\sigma\cdot(\ell\mathtt{:}p) & = & (\sigma\cdot\ell)\mathtt{:}(\sigma\cdot p) \\
\sigma\cdot(\mathtt{new}\,a.\,p) & = & \mathtt{new}\,(\sigma\cdot a).\,(\sigma\cdot p)
\end{array} \tag{4.1.6.1}$$

4.1.7 Substitution

The substitution $t[v/\mathtt{y}]$ of a term v for the variable $\mathtt{y}$ in a term t is defined by recursion on t as follows.

$$\begin{array}{rcl}
\mathtt{y}[v/\mathtt{y}] = v & & \mathtt{x}[v/\mathtt{y}] = \mathtt{x} \qquad (\mathtt{x} \neq \mathtt{y}) \\
(\mathtt{rec}\,\mathtt{x}\,.\,t)[v/\mathtt{y}] = \mathtt{rec}\,\mathtt{x}\,.\,(t[v/\mathtt{y}]) & & (\mathtt{!}t)[v/\mathtt{y}] = \mathtt{!}(t[v/\mathtt{y}]) \\
\multicolumn{3}{c}{(\mathtt{[}u\,\mathtt{>}\,p\mathtt{(x:}\mathbb{P}\,\mathtt{\#}\,s\mathtt{)}\ \mathtt{=>}\ t\mathtt{]})[v/\mathtt{y}] = \mathtt{[}(u[v/\mathtt{y}])\,\mathtt{>}\,p\mathtt{(x:}\mathbb{P}\,\mathtt{\#}\,s\mathtt{)}\ \mathtt{=>}\ (t[v/\mathtt{y}])\mathtt{]}} \\
(\lambda\,\mathtt{x}\,.\,t)[v/\mathtt{y}] = \lambda\,\mathtt{x}\,.\,(t[v/\mathtt{y}]) & & (t\mathtt{(}u\mathtt{:}\mathbb{P}\mathtt{)})[v/\mathtt{y}] = (t[v/\mathtt{y}])\mathtt{(}u[v/\mathtt{y}]\mathtt{:}\mathbb{P}\mathtt{)} \\
(\mathtt{new}\,a\,.\,t)[v/\mathtt{y}] = \mathtt{new}\,a\,.\,(t[v/\mathtt{y}]) & & (t\mathtt{[}a\mathtt{]})[v/\mathtt{y}] = (t[v/\mathtt{y}])\mathtt{[}a\mathtt{]} \\
(\ell\mathtt{:}t)[v/\mathtt{y}] = \ell\mathtt{:}(t[v/\mathtt{y}]) & & (\pi_\ell t)[v/\mathtt{y}] = \pi_\ell(t[v/\mathtt{y}]) \\
\left(\sum_{i\in I} t_i\right)[v/\mathtt{y}] = \sum_{i\in I}(t_i[v/\mathtt{y}]) & &
\end{array} \tag{4.1.7.1}$$

As discussed above, substitution is capture-avoiding in both names and variables, in the sense that for substitution into a term of the forms

$$\mathtt{rec}\,\mathtt{x}\,.\,t \qquad \mathtt{[}u\,\mathtt{>}\,p\mathtt{(x:}\mathbb{P}\,\mathtt{\#}\,s\mathtt{)}\ \mathtt{=>}\ t\mathtt{]} \qquad \lambda\,\mathtt{x}\,.\,t \tag{4.1.7.2}$$

the variable $\mathtt{x}$ is assumed not to be free in v, and for substitution into a term of the form

$$\mathtt{new}\,a\,.\,t \tag{4.1.7.3}$$

the name a is chosen to be fresh for v.

4.1.7.4 Lemma (Equivariance of Substitution). *For any permutation σ, terms t and v and variable $\mathtt{y}$,*

$$\sigma \cdot (t[v/\mathtt{y}]) = (\sigma \cdot t)[(\sigma \cdot v)/\mathtt{y}]$$

Proof. By a straightforward induction over the structure of t. □

4.2 Typing Rules

Nominal HOPLA is a strongly typed language, and its terms and actions are typed as defined in this section. This definition is given by a structural recursion, and it is mutually recursive since the term t appears in the action $t \mapsto p$ and conversely the action p appears in the term $\mathtt{[}u\,\mathtt{>}\,p\mathtt{(x:}\mathbb{Q}'\,\mathtt{\#}\,s'\mathtt{)}\ \mathtt{=>}\ t\mathtt{]}$.

4.2.1 Typing Rules for Terms

Terms of Nominal HOPLA are typed with judgements of the form $\Gamma \vdash_s t : \mathbb{P}$, where Γ is an environment, s is a finite set of names, t is a term and $\mathbb{P}$ is a type.

The type $\mathbb{P}$ describes the actions that the term may perform. The environment Γ records types and freshness assumptions for the variables of t. The set s represents the 'current' set of names, and records a bound on the support of the typing judgement which helps to give a clean presentation of the denotational semantics. However, lemma 4.2.2.9 below demonstrates that it is not strictly necessary to include this information, since a suitable s may inferred from the rest of the typing judgement.

Simultaneously, actions are typed with judgements of the form $\vdash_s \mathbb{P} : p : \mathbb{P}'$ where s is a finite set of names and $\mathbb{P}$ and $\mathbb{P}'$ are types. Intuitively this means that p is an action that terms of type $\mathbb{P}$ may perform and the resumption is of type $\mathbb{P}'$. The typing rules for actions are defined in section 4.2.2.

4.2.1.1 Variable. A bare variable is typed by the environment in the obvious fashion.

$$\frac{-}{\mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_\varnothing \mathtt{x} : \mathbb{P}}$$

4.2.1.2 Weakening. The environment may be extended with extra variables.

$$\frac{\Gamma \vdash_s t : \mathbb{P}}{\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}}$$

4.2.1.3 Exchange. Two variables in the environment may be exchanged.

$$\frac{\Gamma, \mathtt{x2} : {\mathbb{Q}_2}^{\#s_2}, \mathtt{x1} : {\mathbb{Q}_1}^{\#s_1}, \Lambda \vdash_s t : \mathbb{P}}{\Gamma, \mathtt{x1} : {\mathbb{Q}_1}^{\#s_1}, \mathtt{x2} : {\mathbb{Q}_2}^{\#s_2}, \Lambda \vdash_s t : \mathbb{P}}$$

4.2.1.4 Contraction. It is possible to replace a pair of variables (with equal types) with a single variable.

$$\frac{\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}}{\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'} \vdash_s t[\mathtt{x1}/\mathtt{x2}] : \mathbb{P}}$$

4.2.1.5 Fresh-Weakening. It is possible to impose extra freshness assumptions on a variable.

$$\frac{\Gamma, \mathtt{x} : \mathbb{Q}^{\#s''} \vdash_s t : \mathbb{P}}{\Gamma, \mathtt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}} \; (s'' \subseteq s' \subseteq s)$$

4.2.1.6 Support-Weakening (Terms). It is possible to extend the 'current' set s of names.

$$\frac{\Gamma \vdash_{s'} t : \mathbb{P}}{\Gamma \vdash_s t : \mathbb{P}}\ (s' \subseteq s)$$

4.2.1.7 Prefix. The term constructor `!` takes a term t to a term $\texttt{!}t$ that intuitively may perform a primitive action `!` and resume as t. The possible action `!` is recorded in the type.

$$\frac{\Gamma \vdash_s t : \mathbb{P}}{\Gamma \vdash_s \texttt{!}t : !\mathbb{P}}$$

4.2.1.8 Match. A term of the form $\texttt{[}u\ \texttt{>}\ q\texttt{(x:}\mathbb{Q}'\ \texttt{\#}\ s'\texttt{)}\ \texttt{=>}\ t\texttt{]}$ intuitively matches the output of u against the action q and feeds the resumption of u into the variable `x` in t. If `x` has some freshness assumptions imposed on it then u and q must satisfy those assumptions.

$$\frac{\Gamma, \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_s t : \mathbb{P} \quad \Lambda \vdash_{s''} u : \mathbb{Q} \quad \vdash_{s''} \mathbb{Q} : q : \mathbb{Q}'}{\Gamma, \Lambda^{\#s'} \vdash_s \texttt{[}u \texttt{>} q\texttt{(x:}\mathbb{Q}'\ \texttt{\#}\ s'\texttt{)} \texttt{=>} t\texttt{]} : \mathbb{P}}\ (s'' \subseteq s \setminus s')$$

4.2.1.9 Recursion. A term of the form $\texttt{rec}\,\texttt{x.}t$ intuitively acts as its unfolding $t[\texttt{rec}\,\texttt{x.}t/\texttt{x}]$, so that `x` must be of the same type as t.

$$\frac{\Gamma, \texttt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}}{\Gamma \vdash_s \texttt{rec}\,\texttt{x.}t : \mathbb{P}}$$

4.2.1.10 Function Abstraction and Application. A term t of type $\mathbb{P}$ may be abstracted with respect to the free variable `x` of type $\mathbb{Q}$ to leave a term $\lambda\texttt{x.}t$ of type $\mathbb{Q} \to \mathbb{P}$ that can in turn be applied to a term of type $\mathbb{Q}$ in the usual fashion.

$$\frac{\Gamma, \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}}{\Gamma \vdash_s \lambda\texttt{x.}t : \mathbb{Q} \to \mathbb{P}} \qquad \frac{\Gamma \vdash_s t : \mathbb{Q} \to \mathbb{P} \quad \Lambda \vdash_s u : \mathbb{Q}}{\Gamma, \Lambda \vdash_s t\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)} : \mathbb{P}}$$

The inclusion of the type $\mathbb{Q}$ in terms of the form $t\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)}$ is simply so that lemma 4.4.1.4(i) holds: without it, the presence of terms that can be ambiguously typed — such as $\lambda\texttt{x.x}$ — makes the situation too complicated for the purposes of this discussion.

4.2.1.11 Labelling and Label Projection. The actions of a term t may be 'tagged' with a label ℓ_0 by forming the term $\ell_0\texttt{:}t$. The effect of the term former π_{ℓ_0} is that terms of the form $\pi_{\ell_0}t$ can perform only the actions of t that are tagged by the label ℓ_0. In both of these rules the support of ℓ_0 must be contained in s.

$$\frac{\Gamma \vdash_s t : \mathbb{P}_{\ell_0}}{\Gamma \vdash_s \ell_0 {:} t : \bigoplus_{\ell \in L} \mathbb{P}_\ell} \qquad \frac{\Gamma \vdash_s t : \bigoplus_{\ell \in L} \mathbb{P}_\ell}{\Gamma \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}}$$

4.2.1.12 Nondeterministic Sum. A term $\sum_{i \in I} t_i$ makes a nondeterministic choice amongst its components and behaves as the chosen component. The mapping $i \mapsto \Gamma \vdash_{s_i} t_i : \mathbb{P}$ must be supported by s.

$$\frac{\Gamma \vdash_{s_i} t_i : \mathbb{P} \qquad \text{each } i \in I}{\Gamma \vdash_s \sum_{i \in I} t_i : \mathbb{P}}$$

4.2.1.13 Recursive Type Folding and Unfolding. As the recursively-defined type $\mu_j \vec{P}.\, \vec{\mathbb{P}}$ is isomorphic (and not equal) to its unfolding $\mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]$ it is necessary to record any uses of the isomorphism $\mathbf{abs} = \mathbf{rep}^{-1}$ in the syntax of the term.

$$\frac{\Gamma \vdash_s t : \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]}{\Gamma \vdash_s \mathtt{abs}\, t : \mu_j \vec{P}.\, \vec{\mathbb{P}}} \qquad \frac{\Gamma \vdash_s t : \mu_j \vec{P}.\, \vec{\mathbb{P}}}{\Gamma \vdash_s \mathtt{rep}\, t : \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]}$$

4.2.1.14 Name Abstraction and Application. The only alteration to the syntax of terms over that of conventional HOPLA is the following pair of term formers. Intuitively the term $\mathtt{new}\, a.t$ can perform the same actions as t with the name a bound, whereas the term $t\mathtt{[}a\mathtt{]}$ takes the outputs of t, which contain a bound name since t is of type $\delta\mathbb{P}$, and instantiates that name as a.

$$\frac{\Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t : \mathbb{P}}{\Gamma \vdash_s \mathtt{new}\, a.t : \delta\mathbb{P}}\ (a \notin s) \qquad \frac{\Gamma \vdash_s t : \delta\mathbb{P}}{\Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\mathtt{[}a\mathtt{]} : \mathbb{P}}\ (a \notin s)$$

This concludes the definition of the type system for terms. The weakening, exchange, contraction, fresh-weakening and support-weakening rules are together called the *structural* rules. The syntax-directed nature of the non-structural typing rules means that it is sometimes possible to derive information about the type of subterms from the type of a term. The structural rules make this difficult to do in general, but for the purposes of this discussion it is enough to be able to derive type information about subterms of a closed term as shown in the following lemma.

4.2.1.15 Lemma. *If the conclusion of a non-structural typing rule for a closed term is derivable then so are its premises, in a sense made precise below.*

(i) If $\vdash_s \mathtt{!}t : !\mathbb{P}$ then $\vdash_s t : \mathbb{P}$.

(ii) If $\vdash_s \mathtt{rec}\,\mathtt{x}.t : \mathbb{P}$ then $\mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}$.

(iii) If $\vdash_s$ $\mathtt{[}u \mathtt{>} q\mathtt{(x:}\mathbb{Q}' \mathtt{\#}\, s'\mathtt{)} \mathtt{=>} t\mathtt{]}$: $\mathbb{P}$ then $s' \subseteq s$ and there exists $\mathbb{Q}$ and $s'' \subseteq s \setminus s'$ such that $\mathtt{x} : \mathbb{Q}'^{\#s'} \vdash_s t : \mathbb{P}$, $\vdash_{s''} u : \mathbb{Q}$ and $\vdash_{s''} \mathbb{Q} : q : \mathbb{Q}'$.

(iv) If $\vdash_s \lambda\,\mathtt{x}.t : \mathbb{Q} \to \mathbb{P}$ then $\mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$.

(v) If $\vdash_s t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)} : \mathbb{P}$ then $\vdash_s t : \mathbb{Q} \to \mathbb{P}$ and $\vdash_s u : \mathbb{Q}$.

(vi) If $\vdash_s \mathtt{new}\,a.t : \delta\mathbb{P}$ and $a \notin s$ then $\vdash_{s\dot\cup\{a\}} t : \mathbb{P}$.

(vii) If $\vdash_s t\mathtt{[}a\mathtt{]} : \mathbb{P}$ then $a \in s$ and $\vdash_{s\setminus\{a\}} t : \delta\mathbb{P}$.

(viii) If $\vdash_s \ell_0\mathtt{:}t : \bigoplus_{\ell\in L}\mathbb{P}_\ell$ then $\vdash_s t : \mathbb{P}_{\ell_0}$.

(ix) If $\vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}$ then $\vdash_s t : \bigoplus_{\ell\in L}\mathbb{P}_\ell$.

(x) If $\vdash_s \sum_{i\in I} t_i : \mathbb{P}$ then for each $i \in I$ there exists s_i such that $\vdash_{s_i} t_i : \mathbb{P}$, and such that the mapping $i \mapsto (s_i, t_i)$ is supported by s.

(xi) If $\vdash_s \mathtt{abs}\,t : \mu_j\vec{P}.\,\vec{\mathbb{P}}$ then $\vdash_s t : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]$.

(xii) If $\vdash_s \mathtt{rep}\,t : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]$ then $\vdash_s t : \mu_j\vec{P}.\,\vec{\mathbb{P}}$.

Proof. Each statement in this lemma is of the form “If C then ...” where C is some typing judgement. Consider the last steps of the derivation of C. By inspection, each C can arise from at most one non-structural rule, and each C has empty environment so C can arise from the support-weakening rule but no other structural rule concludes with an empty environment. Therefore the derivation of C must finish with the appropriate non-structural rule followed by some number of applications of the support-weakening rule. However, the support-weakening rule is transitive and reflexive, so the C may be derived by a sequence of rules finishing with the appropriate non-structural rule followed by exactly one application of the support-weakening rule, say one which extends the support from s_1 to $s_2 \supseteq s_1$.

The result follows immediately for cases (i), (ii), (iv), (v), (viii), (ix), (xi) and (xii) since the non-structural rule in each of these cases preserves the support, so if it is valid at support s_1 then it remains valid at s_2. For case (iii) the result follows similarly, since $s' \subseteq s_1$ so that if $s'' \subseteq s_1 \setminus s'$ then $s'' \subseteq s_2 \setminus s'$. Case (x) is immediate too, since this case does not depend on the support of C.

For case (vi) if a is fresh for s_2 then it is certainly fresh for s_1, so this case follows. Finally, for case (vii) by considering the structure of the derivation it must be that $a \in s_1$ so that $s_1 \setminus \{a\} \subseteq s_2 \setminus \{a\}$ and $\vdash_{s_1 \setminus \{a\}} t : \delta\mathbb{P}$, so that $\vdash_{s_2 \setminus \{a\}} t : \delta\mathbb{P}$ as required. □

4.2.2 Typing Rules for Actions

Actions are typed with judgements of the form $\vdash_s \mathbb{P} : p : \mathbb{P}'$ where s is a finite set of names and $\mathbb{P}$ and $\mathbb{P}'$ are types. Intuitively this means that p is an action that terms of type $\mathbb{P}$ may perform and the resumption is of type $\mathbb{P}'$. As with typing judgements for terms the set s helps with the presentation of the associated denotational semantics and lemma 4.2.2.9 shows that, if omitted, a suitable s may be inferred from the rest of the typing judgement.

4.2.2.1 Support-Weakening (Actions) It is possible to extend the 'current' set s of names.

$$\frac{\vdash_{s'} \mathbb{P} : p : \mathbb{P}'}{\vdash_s \mathbb{P} : p : \mathbb{P}'} \ (s' \subseteq s)$$

4.2.2.2 Prefix Action. The resumption of a process of type $!\mathbb{P}$ after performing a $!$ action is of type $\mathbb{P}$.

$$\frac{-}{\vdash_{\varnothing} !\mathbb{P} : ! : \mathbb{P}}$$

4.2.2.3 Higher-Order Action. A process t of type $\mathbb{Q} \to \mathbb{P}$ can perform the action $u \mapsto p$ if the application of t to u can perform the action p.

$$\frac{\vdash_s \mathbb{P} : p : \mathbb{P}' \qquad \vdash_s u : \mathbb{Q}}{\vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}'}$$

4.2.2.4 Labelled Action. Actions may be labelled as follows. As with the corresponding typing rules for terms, the support of ℓ_0 must be contained in s.

$$\frac{\vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}'}{\vdash_s \bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0 {:} p : \mathbb{P}'}$$

4.2.2.5 Recursive Type Action. As the recursively-defined type $\mu_j \vec{P}.\ \vec{\mathbb{P}}$ is isomorphic (and not equal) to its unfolding $\mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$ it is necessary to decorate actions in the unfolded type with the tag abs to record the use of the isomorphism **abs** = $\mathbf{rep}^{-1}$.

$$\frac{\vdash_s \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}'}{\vdash_s \mu_j\vec{P}.\,\vec{\mathbb{P}} : \texttt{abs}\ p : \mathbb{P}'}$$

4.2.2.6 New Name Action. An action p may have the name a 'bound' in it to form the action $\texttt{new}\,a.\,p$. Notice that the type of the resumption is $\delta\mathbb{P}'$: if the name a is bound in a term then it remains bound in its resumption.

$$\frac{\vdash_{s\dot\cup\{a\}} \mathbb{P} : p : \mathbb{P}'}{\vdash_s \delta\mathbb{P} : \texttt{new}\,a.\,p : \delta\mathbb{P}'}\ (a \notin s)$$

This concludes the definition of the type system for actions. As is the case for terms, it is possible to use the type system for actions 'backwards' in the following sense.

4.2.2.7 Lemma. *If the conclusion of a non-structural typing rule for an action is derivable then so are its premises, in a sense made precise below.*

(i) If $\vdash_s \mathbb{Q}\to\mathbb{P} : u \mapsto p : \mathbb{P}'$ *then* $\vdash_s u : \mathbb{Q}$ *and* $\vdash_s \mathbb{P} : p : \mathbb{P}'$.

(ii) If $\vdash_s \delta\mathbb{P} : \texttt{new}\,a.\,p : \delta\mathbb{P}'$ *and* $a \notin s$ *then* $\vdash_{s\dot\cup\{a\}} \mathbb{P} : p : \mathbb{P}'$.

(iii) If $\vdash_s \bigoplus_{\ell\in L}\mathbb{P}_\ell : \ell_0{:}p : \mathbb{P}'$ *then* $\vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}'$.

(iv) If $\vdash_s \mu_j\vec{P}.\,\vec{\mathbb{P}} : \texttt{abs}\ p : \mathbb{P}'$ *then* $\vdash_s \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}'$.

Proof. Each statement in this lemma is of the form "If C then ..." where C is some typing judgement. Consider the last steps of the derivation of C. By inspection, each C can arise from at most one non-structural rule, or from the support-weakening rule, so the derivation of C must finish with the appropriate non-structural rule followed by some number of applications of the support-weakening rule. However, the support-weakening rule is transitive and reflexive, so the C may be derived by a sequence of rules finishing with the appropriate non-structural rule followed by exactly one application of the support-weakening rule, say one which extends the support from s_1 to $s_2 \supseteq s_1$.

The result follows immediately for cases (i), (iii) and (iv) since the non-structural rule in each of these cases preserves the support, so if it is valid at support s_1 then it remains valid at s_2. For case (ii) if a is fresh for s_2 then it is certainly fresh for s_1, so this case follows. □

4.2.2.8 Lemma (Equivariance of Typing). *For all permutations σ,*

(a) If $\Gamma \vdash_s t : \mathbb{P}$ then $(\sigma \cdot \Gamma) \vdash_{(\sigma \cdot s)} (\sigma \cdot t) : \mathbb{P}$.

(b) If $\vdash_s \mathbb{P} : p : \mathbb{P}'$ then $\vdash_{(\sigma \cdot s)} \mathbb{P} : (\sigma \cdot p) : \mathbb{P}'$.

Proof. By a straightforward induction over the derivations of the judgements $\Gamma \vdash_s t : \mathbb{P}$ and $\vdash_s \mathbb{P} : p : \mathbb{P}'$. □

4.2.2.9 Lemma. *If the current set s of names is omitted in a typing judgement then a suitable s can be deduced from the syntax of the typing judgement:*

(a) If $\Gamma \vdash_s t : \mathbb{P}$ then $\Gamma|_{\mathrm{supp}(t)} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$.

(b) If $\vdash_s \mathbb{P} : p : \mathbb{P}'$ then $\vdash_{\mathrm{supp}(p)} \mathbb{P} : p : \mathbb{P}'$.

Proof. The proof is by mutual induction on the derivations of $\Gamma \vdash_s t : \mathbb{P}$ and $\vdash_s \mathbb{P} : p : \mathbb{P}'$. The proof is straightforward with the exception of the case for terms of the form $\sum_{i \in I} t_i$.

Variable Trivially, since $\mathrm{supp}(\mathtt{x}) = \varnothing$.

Weakening Suppose that $\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$ is derived from $\Gamma \vdash_s t : \mathbb{P}$, then by induction $\Gamma|_{\mathrm{supp}(t)} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$ and hence $\Gamma|_{\mathrm{supp}(t)}, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$ as required.

Exchange This case follows by a straightforward application of the induction hypothesis, similarly to the case of weakening above.

Contraction This case follows by a straightforward application of the induction hypothesis, similarly to the case of weakening above.

Fresh-Weakening Suppose that $\Gamma, \mathtt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}$ is derived from the judgement $\Gamma, \mathtt{x} : \mathbb{Q}^{\#s''} \vdash_s t : \mathbb{P}$ where $s'' \subseteq s' \subseteq s$, then by induction it follows that $\Gamma|_{\mathrm{supp}(t)}, \mathtt{x} : \mathbb{Q}^{\#s'' \cap \mathrm{supp}(t)} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$ and hence by fresh-weakening it follows that $\Gamma|_{\mathrm{supp}(t)}, \mathtt{x} : \mathbb{Q}^{\#s' \cap \mathrm{supp}(t)} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$ as required.

Support-Weakening (Terms) If $\Gamma \vdash_s t : \mathbb{P}$ is derived from $\Gamma \vdash_{s'} t : \mathbb{P}$ then by induction $\Gamma|_{\mathrm{supp}(t)} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$ as required.

Prefix Suppose that $\Gamma \vdash_s \mathtt{!}t : !\mathbb{P}$ is derived from $\Gamma \vdash_s t : \mathbb{P}$, then by induction $\Gamma|_{\mathrm{supp}(t)} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$, and $\mathrm{supp}(\mathtt{!}t) = \mathrm{supp}(t)$ so that $\Gamma|_{\mathrm{supp}(\mathtt{!}t)} \vdash_{\mathrm{supp}(\mathtt{!}t)} \mathtt{!}t : !\mathbb{P}$ as required.

Recursion Suppose that $\Gamma \vdash_s \mathtt{rec}\,\mathtt{x}.t : \mathbb{P}$ is derived from $\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}$, then by induction $\Gamma|_{\mathrm{supp}(t)}, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$, and $\mathrm{supp}(\mathtt{rec}\,\mathtt{x}.t) = \mathrm{supp}(t)$ so that $\Gamma|_{\mathrm{supp}(\mathtt{rec}\,\mathtt{x}.t)} \vdash_{\mathrm{supp}(\mathtt{rec}\,\mathtt{x}.t)} \mathtt{rec}\,\mathtt{x}.t : \mathbb{P}$ as required.

Match Suppose that $\Gamma, \Lambda^{\#s'} \vdash_s \mathtt{[}u \mathtt{>} q\mathtt{(x:}\mathbb{Q}' \mathtt{\#}\, s'\mathtt{)} \mathtt{=>} t\mathtt{]} : \mathbb{P}$ is derived from

$$\Gamma, \mathtt{x} : \mathbb{Q}'^{\#s'} \vdash_s t : \mathbb{P}, \quad \Lambda \vdash_{s''} u : \mathbb{Q} \quad \text{and} \quad \vdash_{s''} \mathbb{Q} : q : \mathbb{Q}'$$

where $s'' \subseteq s \setminus s'$, then by induction

$$\begin{aligned} &\Gamma|_{\mathrm{supp}(t)}, \mathtt{x} : \mathbb{Q}'^{\#s' \cap \mathrm{supp}(t)} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}, \\ &\Lambda|_{\mathrm{supp}(u)} \vdash_{\mathrm{supp}(u)} u : \mathbb{Q} \quad \text{and} \quad \vdash_{\mathrm{supp}(q)} \mathbb{Q} : q : \mathbb{Q}'. \end{aligned}$$

Notice that $\mathrm{supp}(\mathtt{[}u \mathtt{>} q\mathtt{(x:}\mathbb{Q}' \mathtt{\#}\, s'\mathtt{)} \mathtt{=>} t\mathtt{]}) = \mathrm{supp}(u, q, t, s')$. By applying fresh-weakening and support-weakening it follows that

$$\begin{aligned} &\Gamma|_{\mathrm{supp}(u,q,t,s')}, \mathtt{x} : \mathbb{Q}'^{\#s'} \vdash_{\mathrm{supp}(u,q,t,s')} t : \mathbb{P}, \\ &\Lambda|_{\mathrm{supp}(u)} \vdash_{\mathrm{supp}(u,q)} u : \mathbb{Q} \quad \text{and} \quad \vdash_{\mathrm{supp}(u,q)} \mathbb{Q} : q : \mathbb{Q}'. \end{aligned}$$

Also, $\mathrm{supp}(u, q) \subseteq s'' \cap \mathrm{supp}(u, q, t, s') \subseteq \mathrm{supp}(u, q, t, s') \setminus s'$ so that

$$\Gamma|_{\mathrm{supp}(u,q,t,s')}, \Lambda|_{\mathrm{supp}(u)}{}^{\#s'} \vdash_{\mathrm{supp}(u,q,t,s')} \mathtt{[}u \mathtt{>} q\mathtt{(x:}\mathbb{Q}' \mathtt{\#}\, s'\mathtt{)} \mathtt{=>} t\mathtt{]} : \mathbb{P}$$

and hence

$$\Gamma|_{\mathrm{supp}(u,q,t,s')}, \Lambda|_{\mathrm{supp}(u,q,t,s')}{}^{\#s'} \vdash_{\mathrm{supp}(u,q,t,s')} \mathtt{[}u \mathtt{>} q\mathtt{(x:}\mathbb{Q}' \mathtt{\#}\, s'\mathtt{)} \mathtt{=>} t\mathtt{]} : \mathbb{P}$$

by fresh-weakening as required.

Function Abstraction Suppose that $\Gamma \vdash_s \lambda\,\mathtt{x}.t : \mathbb{Q} \to \mathbb{P}$ is derived from the judgement $\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$, then by induction $\Gamma|_{\mathrm{supp}(t)}, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$, and also $\mathrm{supp}(\lambda\,\mathtt{x}.t) = \mathrm{supp}(t)$ so that $\Gamma|_{\mathrm{supp}(\lambda\,\mathtt{x}.t)} \vdash_{\mathrm{supp}(\lambda\,\mathtt{x}.t)} \lambda\,\mathtt{x}.t : \mathbb{Q} \to \mathbb{P}$ as required.

Function Application Suppose that $\Gamma, \Lambda \vdash_s t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)} : \mathbb{P}$ is derived from the judgements $\Gamma \vdash_s t : \mathbb{Q} \to \mathbb{P}$ and $\Lambda \vdash_s u : \mathbb{Q}$, then by induction followed by support-weakening and fresh-weakening $\Gamma|_{\mathrm{supp}(t,u)} \vdash_{\mathrm{supp}(t,u)} t : \mathbb{Q} \to \mathbb{P}$ and $\Lambda|_{\mathrm{supp}(t,u)} \vdash_{\mathrm{supp}(t,u)} u : \mathbb{Q}$, and $\mathrm{supp}(t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)}) = \mathrm{supp}(t, u)$ so it follows that $\Gamma|_{\mathrm{supp}(t,u)}, \Lambda|_{\mathrm{supp}(t,u)} \vdash_{\mathrm{supp}(t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)})} t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)} : \mathbb{P}$ as required.

Name Abstraction Suppose that $\Gamma \vdash_s \mathtt{new}\, a \mathtt{.} t : \delta\mathbb{P}$ is derived from the judgement $\Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t : \mathbb{P}$ where a is fresh, then by induction and support-weakening $\Gamma^{\#a}|_{\mathrm{supp}(t)} \vdash_{\mathrm{supp}(t) \cup \{a\}} t : \mathbb{P}$, and $\mathrm{supp}(\mathtt{new}\, a \mathtt{.} t) = \mathrm{supp}(t) \setminus \{a\}$ so that $\Gamma|_{\mathrm{supp}(t)\setminus\{a\}} \vdash_{\mathrm{supp}(t)\setminus\{a\}} \mathtt{new}\, a \mathtt{.} t : \mathbb{P}$ as required.

Name Application Suppose that $\Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\mathtt{[}a\mathtt{]} : \mathbb{P}$ is derived from the judgement $\Gamma \vdash_s t : \delta\mathbb{P}$ and $a \notin s$, then by induction $\Gamma|_{\mathrm{supp}(t)} \vdash_{\mathrm{supp}(t)} t : \delta\mathbb{P}$, and $\mathrm{supp}(t\mathtt{[}a\mathtt{]}) = \mathrm{supp}(t) \dot\cup \{a\}$, so that $\Gamma^{\#a}|_{\mathrm{supp}(t\mathtt{[}a\mathtt{]})} \vdash_{\mathrm{supp}(t\mathtt{[}a\mathtt{]})} t\mathtt{[}a\mathtt{]} : \mathbb{P}$ as required.

Labelling Suppose that $\Gamma \vdash_s \ell_0 {:} t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ is derived from $\Gamma \vdash_s t : \mathbb{P}_{\ell_0}$, then by induction followed by support-weakening and freshness-weakening it follows that $\Gamma|_{\mathrm{supp}(t,\ell_0)} \vdash_{\mathrm{supp}(t,\ell_0)} t : \mathbb{P}_{\ell_0}$ and $\mathrm{supp}(\ell_0 {:} t) = \mathrm{supp}(t, \ell_0)$ so that $\Gamma|_{\mathrm{supp}(t,\ell_0)} \vdash_{\mathrm{supp}(t,\ell_0)} \ell_0 {:} t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ as required.

Label Projection Suppose that $\Gamma \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}$ is derived from the judgement $\Gamma \vdash_s t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$, then by induction followed by support-weakening and freshness-weakening $\Gamma|_{\mathrm{supp}(t,\ell_0)} \vdash_{\mathrm{supp}(t,\ell_0)} t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$. Also it is the case that $\mathrm{supp}(\pi_{\ell_0} t) = \mathrm{supp}(t, \ell_0)$ so that $\Gamma|_{\mathrm{supp}(t,\ell_0)} \vdash_{\mathrm{supp}(t,\ell_0)} \pi_{\ell_0} t : \mathbb{P}_{\ell_0}$ as required.

Nondeterministic Sum Suppose that $\Gamma \vdash_s \sum_{i \in I} t_i : \mathbb{P}$ is derived from the judgements $\Gamma \vdash_{s_i} t_i : \mathbb{P}$ for all $i \in I$. Therefore by induction for each $i \in I$ it follows that $\Gamma|_{\mathrm{supp}(t_i)} \vdash_{\mathrm{supp}(t_i)} t_i : \mathbb{P}$. Let $i \in I$. Enumerate the names in $(\mathrm{supp}(i) \cap \mathrm{supp}(\Gamma)) \setminus \mathrm{supp}(\lambda i.t_i)$ as $a_1, \ldots, a_n$, let $b_1, \ldots, b_n$ be fresh and let $\pi = (a_1 b_1) \ldots (a_n b_n)$. As the a_j and b_j are fresh for $\lambda i.t_i$ it follows that $\pi \cdot t_i = t_{\pi \cdot i}$. Since typing is equivariant, $\pi^{-1} \cdot \left(\Gamma|_{\mathrm{supp}(t_{\pi \cdot i})}\right) \vdash_{\mathrm{supp}(t_i)} t_i : \mathbb{P}$. Suppose that the freshness assumption $\mathtt{x} \# a$ is in the environment $\pi^{-1} \cdot \left(\Gamma|_{\mathrm{supp}(t_{\pi \cdot i})}\right)$, then $\mathtt{x} \# (\pi \cdot a)$ is in $\Gamma|_{\mathrm{supp}(t_{\pi \cdot i})}$ and hence in Γ and furthermore $\pi \cdot a \in \mathrm{supp}(t_{\pi \cdot i})$ so that finally $a \in \mathrm{supp}(t_i) \subseteq \mathrm{supp}(i) \cup \mathrm{supp}(\lambda i.t_i)$. If $a \in \mathrm{supp}(\lambda i.t_i)$ then $\pi \cdot a = a$ and $\mathtt{x} \# a$ is in $\Gamma|_{\mathrm{supp}(\lambda i.t_i)}$.

On the other hand, suppose that $a \in \mathrm{supp}(i) \setminus \mathrm{supp}(\lambda i.t_i)$. Recall that $\mathtt{x} \# (\pi \cdot a)$ is a freshness assumption of Γ and hence that $\pi \cdot a \in \mathrm{supp}(\Gamma)$. It cannot be that $a = a_j$ for some j, because then $\pi \cdot a = b_j$ and the b_j were chosen fresh for Γ. Nor can it be that $a = b_j$ for some j, because $a \in \mathrm{supp}(i)$ and the b_j were chosen fresh for i. Finally, it cannot be that $\pi \cdot a = a$, because then $a \in \mathrm{supp}(\Gamma)$ and hence $a = a_j$ for some j, which has already been shown impossible. Therefore $a \notin \mathrm{supp}(i) \setminus \mathrm{supp}(\lambda i.t_i)$, so that $a \in \mathrm{supp}(\lambda i.t_i)$ and hence $\mathtt{x} \# a$ is in $\Gamma|_{\mathrm{supp}(\lambda i.t_i)}$.

Therefore, starting from $\pi^{-1} \cdot \left(\Gamma|_{\text{supp}(t_{\pi \cdot i})}\right) \vdash_{\text{supp}(t_i)} t_i : \mathbb{P}$ it is possible to apply fresh-weakening to conclude that $\Gamma|_{\text{supp}(\lambda i.t_i)} \vdash_{\text{supp}(t_i)} t_i : \mathbb{P}$ for each i, and hence $\Gamma|_{\text{supp}(\lambda i.t_i)} \vdash_{\text{supp}(\lambda i.t_i)} \sum_{i \in I} t_i : \mathbb{P}$.

Recursive Type Folding Suppose that $\Gamma \vdash_s \texttt{abs}\, t : \mu_j \vec{P}.\, \vec{\mathbb{P}}$ is derived from $\Gamma \vdash_s t : \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]$ then by induction $\Gamma|_{\text{supp}(t)} \vdash_{\text{supp}(t)} t : \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]$ and $\text{supp}(\texttt{abs}\, t) = \text{supp}(t)$ so that $\Gamma|_{\text{supp}(\texttt{abs}\, t)} \vdash_{\text{supp}(\texttt{abs}\, t)} \texttt{abs}\, t : \mu_j \vec{P}.\, \vec{\mathbb{P}}$ as required.

Recursive Type Unfolding Suppose that $\Gamma \vdash_s \texttt{rep}\, t : \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]$ is derived from $\Gamma \vdash_s t : \mu_j \vec{P}.\, \vec{\mathbb{P}}$ then by induction $\Gamma|_{\text{supp}(t)} \vdash_{\text{supp}(t)} t : \mu_j \vec{P}.\, \vec{\mathbb{P}}$ and $\text{supp}(\texttt{rep}\, t) = \text{supp}(t)$ so that $\Gamma|_{\text{supp}(\texttt{rep}\, t)} \vdash_{\text{supp}(\texttt{rep}\, t)} \texttt{rep}\, t : \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]$.

Prefix Action Trivially, since $\text{supp}(\texttt{!}) = \varnothing$.

Higher-Order Action Suppose that $\vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}'$ is derived from $\vdash_s \mathbb{P} : p : \mathbb{P}'$ and $\vdash_s u : \mathbb{Q}$, then by induction and support-weakening it follows that $\vdash_{\text{supp}(u,p)} \mathbb{P} : p : \mathbb{P}'$ and $\vdash_{\text{supp}(u,p)} u : \mathbb{Q}$. Also $\text{supp}(u \mapsto p) = \text{supp}(u, p)$ so that $\vdash_{\text{supp}(u,p)} \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}'$ as required.

Labelled Action Suppose that $\vdash_s \bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0 {:} p : \mathbb{P}'$ is derived from the judgement $\vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}'$, then by induction and support-weakening it follows that $\vdash_{\text{supp}(p,\ell_0)} \mathbb{P}_{\ell_0} : p : \mathbb{P}'$. Also $\text{supp}(\ell_0 {:} p) = \text{supp}(p, \ell_0)$ so that as required $\vdash_{\text{supp}(p,\ell_0)} \bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0 {:} p : \mathbb{P}'$.

New Name Action Suppose that $\vdash_s \delta\mathbb{P} : \texttt{new}\, a.\, p : \delta\mathbb{P}'$ is derived from the judgement $\vdash_{s \dot\cup \{a\}} \mathbb{P} : p : \mathbb{P}'$, then by induction and support-weakening it follows that $\vdash_{\text{supp}(p) \cup \{a\}} \mathbb{P} : p : \mathbb{P}'$. Also $\text{supp}(\texttt{new}\, a.\, p) = \text{supp}(p) \setminus \{a\}$ so that $\vdash_{\text{supp}(p) \setminus \{a\}} \delta\mathbb{P} : \texttt{new}\, a.\, p : \mathbb{P}'$ as required.

Recursive Type Action Suppose that $\vdash_s \mu_j \vec{P}.\, \vec{\mathbb{P}} : \texttt{abs}\ p : \mathbb{P}'$ is derived from $\vdash_s \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}'$, then by induction $\vdash_{\text{supp}(p)} \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}'$ and $\text{supp}(\texttt{abs}\ p) = \text{supp}(p)$ so that $\vdash_{\text{supp}(p)} \mu_j \vec{P}.\, \vec{\mathbb{P}} : \texttt{abs}\, p : \mathbb{P}'$ as required.

Support-Weakening (Actions) If $\vdash_s \mathbb{P} : p : \mathbb{P}'$ is derived from $\vdash_{s'} \mathbb{P} : p : \mathbb{P}'$ then $\vdash_{\text{supp}(p)} \mathbb{P} : p : \mathbb{P}'$ by induction as required.

□

4.2.2.10 Definition. *In the light of the previous lemma, write* $\Gamma \vdash t : \mathbb{P}$ *for* $\Gamma \vdash_{\mathrm{supp}(t)} t : \mathbb{P}$ *and* $\vdash \mathbb{P} : p : \mathbb{P}'$ *for* $\vdash_{\mathrm{supp}(p)} \mathbb{P} : p : \mathbb{P}'$.

4.3 The Substitution Lemma

The intuition behind the following lemma is that a variable $\mathtt{y}$ of type $\mathbb{R}$ in a term t may receive a term v as input, simply by substituting v for $\mathtt{y}$ in t. If $\mathtt{y}$ has freshness assumptions imposed on it then the given v must satisfy those assumptions.

4.3.0.1 Lemma (Syntactic Substitution Lemma). *Suppose that t and v satisfy $\Gamma, \mathtt{y} : \mathbb{R}^{\#r} \vdash_s t : \mathbb{P}$ and $\Delta \vdash_{s_1} v : \mathbb{P}$ where $s_1 \cap r = \varnothing$ and the variables in Γ are distinct from those in Δ. Then*

$$\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v/\mathtt{y}] : \mathbb{P}$$

Proof. The proof is by a very routine induction over the typing rules. It does not provide any especially deep insights and can be skipped by all but the most dedicated of readers. Because of the structural rules of contraction and exchange it is necessary to use a stronger induction hypothesis than that given by the statement of the lemma:

Suppose that $\Gamma' \vdash_s t : \mathbb{P}$, $\Delta \vdash_{s_1} v : \mathbb{R}$, and Γ' is a reordering of the environment $\Gamma, \mathtt{y1} : \mathbb{R}^{\#r_1}, \ldots, \mathtt{yn} : \mathbb{R}^{\#r_n}$. Let $r = r_1 \cup \ldots \cup r_n$. Suppose also that the variables in Γ are distinct from those in Δ and that $s_1 \cap r = \varnothing$. Then

$$\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P} \qquad (4.3.0.2)$$

where $t[v]$ is the term obtained by simultaneously substituting v for the variables $\mathtt{y1}, \ldots, \mathtt{yn}$ in t.

The proof now proceeds by induction on the derivation of $\Gamma' \vdash_s t : \mathbb{P}$. Throughout, variables beginning with $\mathtt{y}$ and $\mathtt{z}$ such as $\mathtt{y1}, \mathtt{ym}$ and $\mathtt{zn}$ represent those that are subject to substitution, whereas others such as $\mathtt{x}$ and $\mathtt{x1}$ stand for those that are not.

Variable There are two possibilities for the variable rule, depending on whether the variable in question is subject to substitution or not. If it is subject to substitution then $t = \mathtt{y1}$ and $s = \varnothing$ so by assumption $\Delta \vdash_{s_1} v : \mathbb{R}$ and $t[v/\mathtt{y1}] = v$, so that $\Delta \vdash_{\varnothing \cup s_1} t[v/\mathtt{y1}] : \mathbb{R}$ as required. On the other hand if the variable in question is not subject to substitution then $t = \mathtt{x}$ so by repeated use of weakening, support-weakening and fresh-weakening it is possible to derive $\mathtt{x} : \mathbb{P}^{\#\varnothing}, \Delta \vdash_{s_1} x : \mathbb{P}$ as required.

Weakening There are two possibilities for the weakening rule, depending on whether the newly-added variable is subject to substitution or not. Suppose that $\Gamma', \texttt{yn} : \mathbb{R}^{\#\varnothing} \vdash_s t : \mathbb{P}$ is derived from $\Gamma' \vdash_s t : \mathbb{P}$, then by induction $\Gamma, \Delta^{\# r_1 \cup \ldots \cup r_{n-1} \cup \varnothing} \vdash_{s \cup s_1} t[v] : \mathbb{P}$. Furthermore, $\texttt{yn}$ does not appear free in t so that the judgement above is as required.

On the other hand, suppose that $\Gamma', \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$ is derived from the judgement $\Gamma' \vdash_s t : \mathbb{P}$, then $\Gamma, \Delta^{\# r_1 \cup \ldots \cup r_n} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ by induction and hence by weakening $\Gamma, \Delta^{\# r_1 \cup \ldots \cup r_n}, \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ as required.

Exchange This case is trivial, by reordering the environments appropriately.

Contraction There are two possibilities for this case, depending on whether the contracted variable is subject to substitution or not. Suppose that the judgement $\Gamma', \texttt{yn} : \mathbb{R}^{\# r_n} \vdash_s t[\texttt{yn}/\texttt{ym}] : \mathbb{P}$ is derived from a judgement of the form $\Gamma', \texttt{yn} : \mathbb{R}^{\# r_n}, \texttt{ym} : \mathbb{R}^{\# r_n} \vdash_s t : \mathbb{P}$, then by induction $\Gamma, \Delta^{\# r} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ and $t[v] = t[\texttt{yn}/\texttt{ym}][v]$ so this judgement is as required.

On the other hand suppose that the judgement $\Gamma', \texttt{x1} : \mathbb{Q}^{\# s'} \vdash_s t[\texttt{x1}/\texttt{x2}] : \mathbb{P}$ is derived from the judgement $\Gamma', \texttt{x1} : \mathbb{Q}^{\# s'}, \texttt{x2} : \mathbb{Q}^{\# s'} \vdash_s t : \mathbb{P}$, then by induction $\Gamma, \texttt{x1} : \mathbb{Q}^{\# s'}, \texttt{x2} : \mathbb{Q}^{\# s'}, \Delta^{\# r} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ and hence by contraction and exchange $\Gamma, \texttt{x1} : \mathbb{Q}^{\# s'}, \Delta^{\# r} \vdash_{s \cup s_1} t[v][\texttt{x1}/\texttt{x2}] : \mathbb{P}$. Furthermore as the substitutions $[v]$ and $[\texttt{x1}/\texttt{x2}]$ are disjoint, $t[v][\texttt{x1}/\texttt{x2}] = t[\texttt{x1}/\texttt{x2}][v]$ so this judgement is as required.

Fresh-Weakening There are two possibilities for this case, corresponding to whether the variable with extra freshness assumptions is subject to substitution or not. If it is not subject to substitution then this case is a simple application of the inductive hypothesis. On the other hand, suppose that $r'_n \subseteq r_n \subseteq s$, that Γ' is a reordering of the environment $\Gamma, \texttt{y1} : \mathbb{R}^{\# r_1}, \ldots, \texttt{yn} : \mathbb{R}^{\# r_n}$ and also that Γ'' is a reordering of $\Gamma, \texttt{y1} : \mathbb{R}^{\# r_1}, \ldots, \texttt{yn} : \mathbb{R}^{\# r'_n}$. If $\Gamma' \vdash_s t : \mathbb{P}$ is derived from $\Gamma'' \vdash_s t : \mathbb{P}$ then $\Gamma, \Delta^{\# r_1 \cup \ldots \cup r_{n-1} \cup r'_n} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ by induction, hence $\Gamma, \Delta^{\# r_1 \cup \ldots \cup r_{n-1} \cup r_n} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ by repeated applications of fresh-weakening on the variables in Δ, as required.

Support-Weakening Suppose that $s' \subseteq s$ and that $\Gamma' \vdash_s t : \mathbb{P}$ is derived from $\Gamma' \vdash_{s'} t : \mathbb{P}$, then $\Gamma, \Delta^{\# r} \vdash_{s' \cup s_1} t[v] : \mathbb{P}$ by induction, and hence by support-weakening it follows that $\Gamma, \Delta^{\# r} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ as required.

Prefix Suppose that $\Gamma' \vdash_s \texttt{!}t : !\mathbb{P}$ is derived from $\Gamma' \vdash_s t : \mathbb{P}$, then by induction $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ and so it follows from the rule for prefixed terms that $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} \texttt{!}(t[v]) = (\texttt{!}t)[v] : !\mathbb{P}$ as required.

Recursion Suppose that $\Gamma' \vdash_s \texttt{rec}\,\texttt{x}.t : \mathbb{P}$ is derived from the judgement $\Gamma', \texttt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}$, then $\Gamma, \Delta^{\#r}, \texttt{x} : \mathbb{P}^{\#\varnothing} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ by induction and hence $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} \texttt{rec}\,\texttt{x}.(t[v]) = (\texttt{rec}\,\texttt{x}.t)[v] : \mathbb{P}$ as required.

Match Suppose that $\Gamma', \Lambda'^{\#s'} \vdash_s \texttt{[}u \texttt{>} q\texttt{(x:}\mathbb{Q}' \texttt{\#}\, s'\texttt{) =>}\, t\texttt{]} : \mathbb{P}$ is derived from $\Gamma', \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_s t : \mathbb{P}$, $\Lambda' \vdash_{s''} u : \mathbb{Q}$ and $\vdash_{s''} \mathbb{Q} : q : \mathbb{Q}'$ where $s'' \subseteq s \setminus s'$. Suppose that Γ' is a reordering of $\Gamma, \texttt{y1} : \mathbb{R}^{\#r_1}, \ldots, \texttt{yn} : \mathbb{R}^{\#r_n}$ and that Λ' is a reordering of $\Lambda, \texttt{z1} : \mathbb{R}^{\#r'_1}, \ldots, \texttt{zm} : \mathbb{R}^{\#r'_m}$. Let $r = r_1 \cup \ldots \cup r_n$ and $r' = r'_1 \cup \ldots \cup r'_m$.

There are now two possibilities, depending on whether $m = 0$ or not. If $m \neq 0$ then by induction $\Gamma, \Delta^{\#r}, \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ and $\Lambda, \Delta^{\#r'} \vdash_{s'' \cup s_1} u[v] : \mathbb{Q}$ and hence $\Lambda, \Delta'^{\#r'} \vdash_{s'' \cup s_1} u[v'] : \mathbb{Q}$ where Δ' and v' are Δ and v with all the variables renamed. Also by fresh-weakening $\vdash_{s'' \cup s_1} \mathbb{Q} : q : \mathbb{Q}'$. By hypothesis, $(r \cup s' \cup r') \cap s_1 = \varnothing$ so that $s' \cap s_1 = \varnothing$ and hence $s_1 \subseteq (s \cup s_1) \setminus s'$. Also $s'' \subseteq s \setminus s' \subseteq (s \cup s_1) \setminus s'$ so that finally $(s'' \cup s_1) \subseteq (s \cup s_1) \setminus s'$. Therefore

$$\Gamma, \Delta^{\#r}, \Lambda^{\#s'}, \Delta'^{\#r' \cup s'} \vdash_{s \cup s_1} \texttt{[}(u[v']) \texttt{>} q\texttt{(x:}\mathbb{Q}' \texttt{\#}\, s'\texttt{) =>}\, t[v]\texttt{]} : \mathbb{P}$$

so that by contraction, exchange and fresh-weakening,

$$\Gamma, \Lambda^{\#s'}, \Delta^{\#r \cup r' \cup s'} \vdash_{s \cup s_1} \texttt{[}u[v] \texttt{>} q\texttt{(x:}\mathbb{Q}' \texttt{\#}\, s'\texttt{) =>}\, t[v]\texttt{]} : \mathbb{P}$$

which is as required since

$$\texttt{[}u[v] \texttt{>} q\texttt{(x:}\mathbb{Q}' \texttt{\#}\, s'\texttt{) =>}\, t[v]\texttt{]} = \texttt{[}u \texttt{>} q\texttt{(x:}\mathbb{Q}' \texttt{\#}\, s'\texttt{) =>}\, t\texttt{]}\,[v].$$

If $m = 0$ then by induction $\Gamma, \Delta^{\#r}, \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ and furthermore $s'' \subseteq s \setminus s' \subseteq (s \cup s_1) \setminus s'$. Therefore

$$\Gamma, \Delta^{\#r}, \Lambda^{\#s'} \vdash_{s \cup s_1} \texttt{[}u \texttt{>} q\texttt{(x:}\mathbb{Q}' \texttt{\#}\, s'\texttt{) =>}\, t[v]\texttt{]} = \texttt{[}u \texttt{>} q\texttt{(x:}\mathbb{Q}' \texttt{\#}\, s'\texttt{) =>}\, t\texttt{]}\,[v] : \mathbb{P}$$

as required (up-to exchange).

Function Abstraction Suppose that $\Gamma' \vdash_s \lambda\texttt{x}.t : \mathbb{Q} \rightarrow \mathbb{P}$ is derived from the judgement $\Gamma', \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$, then $\Gamma, \Delta^{\#r}, \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_{s \cup s_1} t[v] : \mathbb{P}$ by induction and hence $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} \lambda\texttt{x}.(t[v]) = (\lambda\texttt{x}.t)[v] : \mathbb{Q} \rightarrow \mathbb{P}$ as required.

Function Application Suppose that $\Gamma', \Lambda' \vdash_s t\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)} : \mathbb{P}$ is derived from the judgement $\Gamma' \vdash_s t : \mathbb{Q} \to \mathbb{P}$ and $\Lambda' \vdash_s u : \mathbb{Q}$. Suppose that Γ' is a reordering of the environment $\Gamma, \texttt{y1} : \mathbb{R}^{\#r_1}, \ldots, \texttt{yn} : \mathbb{R}^{\#r_n}$ and that Λ' is a reordering of $\Lambda, \texttt{z1} : \mathbb{R}^{\#r'_1}, \ldots, \texttt{zm} : \mathbb{R}^{\#r'_m}$. Let $r = r_1 \cup \ldots \cup r_n$ and $r' = r'_1 \cup \ldots \cup r'_m$. By induction, it follows that $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{Q} \to \mathbb{P}$ and $\Lambda, \Delta^{\#r'} \vdash_{s \cup s_1} u[v] : \mathbb{Q}$ so that $\Gamma, \Delta^{\#r}, \Lambda, \Delta'^{\#r'} \vdash_{s \cup s_1} (t[v])\texttt{(}(u[v'])\texttt{:}\mathbb{Q}\texttt{)} : \mathbb{P}$ where Δ' and v' are Δ and v with freshly-named variables. Therefore by contraction and exchange and fresh-weakening, $\Gamma, \Lambda, \Delta^{\#r \cup r'} \vdash_{s \cup s_1} (t[v])\texttt{(}(u[v])\texttt{:}\mathbb{Q}\texttt{)} = (t\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)})[v] : \mathbb{P}$ as required.

Name Abstraction Suppose that $\Gamma' \vdash_s \texttt{new}\, a\texttt{.}t : \delta\mathbb{P}$ is derived from the judgement $\Gamma'^{\#a} \vdash_{s \dot\cup \{a\}} t : \mathbb{P}$ where a is a fresh name and Γ' is a reordering of $\Gamma, \texttt{y1} : \mathbb{R}^{\#r_1}, \ldots, \texttt{yn} : \mathbb{R}^{\#r_n}$. Notice that as a is fresh, therefore $a \notin r$ and $a \notin s_1$ so that $a \# v$. By induction $\Gamma^{\#a}, \Delta^{\#r \dot\cup \{a\}} \vdash_{s \cup s_1 \dot\cup \{a\}} t[v] : \mathbb{P}$ so that as required

$$\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} \texttt{new}\, a\texttt{.}(t[v]) = (\texttt{new}\, a\texttt{.}t)[v] : \delta\mathbb{P}.$$

Name Application Suppose that $a \notin s$ and that $\Gamma'^{\#a} \vdash_{s \dot\cup \{a\}} t\texttt{[}a\texttt{]} : \mathbb{P}$ is derived from $\Gamma' \vdash_s t : \delta\mathbb{P}$ where Γ' is a reordering of $\Gamma, \texttt{y1} : \mathbb{R}^{\#r_1}, \ldots, \texttt{yn} : \mathbb{R}^{\#r_n}$. By hypothesis, $s_1 \cap (r \dot\cup \{a\}) = \varnothing$ so that $a \notin s_1$ and hence $a \# v$. By induction $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \delta\mathbb{P}$ so that finally

$$\Gamma^{\#a}, \Delta^{\#r \dot\cup \{a\}} \vdash_{s \cup s_1 \dot\cup \{a\}} (t[v])\texttt{[}a\texttt{]} = (t\texttt{[}a\texttt{]})[v] : \mathbb{P}$$

as required.

Labelling Suppose that $\Gamma' \vdash_s \ell_0\texttt{:}t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ is derived from $\Gamma' \vdash_s t : \mathbb{P}_{\ell_0}$, then by induction $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P}_{\ell_0}$ and so as required it follows that $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} \ell_0\texttt{:}(t[v]) = (\ell_0\texttt{:}t)[v] : \bigoplus_{\ell \in L} \mathbb{P}_\ell$.

Label Projection Suppose that $\Gamma' \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}$ is derived from the judgement $\Gamma' \vdash_s t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$, then by induction $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ and so $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} \pi_{\ell_0}(t[v]) = (\pi_{\ell_0} t)[v] : \mathbb{P}_{\ell_0}$ as required.

Nondeterministic Sum Suppose that $\Gamma' \vdash_s \sum_{i \in I} t_i : \mathbb{P}$ is derived from the judgement $\Gamma' \vdash_{s_i} t_i : \mathbb{P}$ for all $i \in I$, then by induction $\Gamma, \Delta^{\#r} \vdash_{s_i \cup s_1} t_i[v] : \mathbb{P}$ for all $i \in I$. Note that the support of the mapping $i \mapsto s_i \cup s_1$ is at most $s \cup s_1$ and so $\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} \sum_{i \in I}(t[v]) = \big(\sum_{i \in I} t\big)[v] : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ as required.

Recursive Type Folding Suppose that $\Gamma' \vdash_s \mathtt{abs}\,t : \mu_j\vec{P}.\,\vec{\mathbb{P}}$ is derived from $\Gamma' \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]$, then by induction $\Gamma, \Delta^{\#r} \vdash_{s\cup s_1} t[v] : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]$ and so $\Gamma, \Delta^{\#r} \vdash_{s\cup s_1} \mathtt{abs}\,(t[v]) = (\mathtt{abs}\,t)[v] : \mu_j\vec{P}.\,\vec{\mathbb{P}}$ as required.

Recursive Type Unfolding Suppose that $\Gamma' \vdash_s \mathtt{rep}\,t : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]$ is derived from $\Gamma' \vdash_s t : \mu_j\vec{P}.\,\vec{\mathbb{P}}$, then by induction $\Gamma, \Delta^{\#r} \vdash_{s\cup s_1} t[v] : \mu_j\vec{P}.\,\vec{\mathbb{P}}$ and so $\Gamma, \Delta^{\#r} \vdash_{s\cup s_1} \mathtt{rep}\,(t[v]) = (\mathtt{rep}\,t)[v] : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]$ as required. □

4.4 Operational Semantics

Nominal HOPLA is given an operational semantics in the style of a labelled transition system. That a term t such that $\vdash t : \mathbb{P}$ may perform an action p such that $\vdash \mathbb{P} : p : \mathbb{P}'$ and resume as the term t' is written

$$\mathbb{P} : t \xrightarrow{p} t'. \tag{4.4.0.1}$$

The operational semantics of closed, well-typed terms are defined below. These rules define the semantics only of well-typed terms in the sense that each of these rules has an additional premise, omitted for clarity, that the term on the left of the transition in the conclusion has the correct type.

$$\frac{\mathbb{P} : t[\mathtt{rec}\,\mathtt{x}.t/\mathtt{x}] \xrightarrow{p} t'}{\mathbb{P} : \mathtt{rec}\,\mathtt{x}.t \xrightarrow{p} t'} \qquad \frac{-}{!\mathbb{P} : \mathtt{!}t \xrightarrow{!} t} \tag{4.4.0.2}$$

$$\frac{\mathbb{P} : t[u'/\mathtt{x}] \xrightarrow{p} t' \quad \mathbb{Q} : u \xrightarrow{q} u' \quad \vdash \mathbb{Q} : q : \mathbb{Q}'}{\mathbb{P} : \mathtt{[}u\ \mathtt{>}\ q\mathtt{(x:}\mathbb{Q}'\ \mathtt{\#}\ s'\mathtt{)}\ \mathtt{=>}\ t\mathtt{]} \xrightarrow{p} t'}$$

$$\frac{\mathbb{P} : t \xrightarrow{p} t'}{\delta\mathbb{P} : \mathtt{new}\,a.t \xrightarrow{\mathtt{new}\,a.\,p} \mathtt{new}\,a.t'} \qquad \frac{\delta\mathbb{P} : t \xrightarrow{\mathtt{new}\,a.\,p} \mathtt{new}\,a.t'}{\mathbb{P} : t\mathtt{[}a\mathtt{]} \xrightarrow{p} t'}$$

$$\frac{\mathbb{P} : t[u/\mathtt{x}] \xrightarrow{p} t'}{\mathbb{Q} \rightarrow \mathbb{P} : \lambda\mathtt{x}.t \xrightarrow{u \mapsto p} t'} \qquad \frac{\mathbb{Q} \rightarrow \mathbb{P} : t \xrightarrow{u \mapsto p} t'}{\mathbb{P} : t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)} \xrightarrow{p} t'}$$

$$\frac{\mathbb{P}_{\ell_0} : t \xrightarrow{p} t'}{\bigoplus_{\ell\in L}\mathbb{P}_\ell : \ell_0 \mathtt{:} t \xrightarrow{\ell_0 : p} t'} \qquad \frac{\bigoplus_{\ell\in L}\mathbb{P}_\ell : t \xrightarrow{\ell_0 : p} t'}{\mathbb{P}_{\ell_0} : \pi_{\ell_0} t \xrightarrow{p} t'}$$

$$\frac{\mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : t \xrightarrow{p} t'}{\mu_j\vec{P}.\,\vec{\mathbb{P}} : \mathtt{abs}\,t \xrightarrow{\mathsf{abs}\ p} t'} \qquad \frac{\mu_j\vec{P}.\,\vec{\mathbb{P}} : t \xrightarrow{\mathsf{abs}\ p} t'}{\mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : \mathtt{rep}\,t \xrightarrow{p} t'}$$

$$\frac{\mathbb{P} : t_{i_0} \xrightarrow{p} t'}{\mathbb{P} : \sum_{i\in I} t_i \xrightarrow{p} t'}$$

4.4.1 Properties of the Operational Semantics

The following few lemmas demonstrate that the operational semantics given above interacts well with the type system described in section 4.2.

4.4.1.1 Lemma. *If* $\mathbb{P} : t \xrightarrow{p} t'$ *then* $\vdash t : \mathbb{P}$*.*

Proof. By definition. □

4.4.1.2 Lemma. *If* $\mathbb{P} : t \xrightarrow{p} t'$ *then there exists a unique* $\mathbb{P}'$ *such that the judgement* $\vdash \mathbb{P} : p : \mathbb{P}'$ *holds.*

Proof. By induction over the derivation of $\mathbb{P} : t \xrightarrow{p} t'$. □

4.4.1.3 Lemma. *If* $\mathbb{P} : t \xrightarrow{p} t'$ *and* $\vdash \mathbb{P} : p : \mathbb{P}'$ *then* $\vdash t' : \mathbb{P}'$*.*

Proof. By induction on the derivation of $\mathbb{P} : t \xrightarrow{p} t'$ as follows.

Recursion The induction hypothesis says that $\vdash \mathbb{P} : p : \mathbb{P}'$ and furthermore that $\mathbb{P} : t[\mathtt{rec\,x}.t/\mathtt{x}] \xrightarrow{p} t'$ so that $\vdash t' : \mathbb{P}$ follows immediately by induction.

Prefix The induction hypothesis says that $\vdash \mathtt{!}t : !\mathbb{P}$ so that $\vdash t : \mathbb{P}$ by lemma 4.2.1.15(i), as required.

Match The induction hypothesis says that $\vdash \mathbb{P} : p : \mathbb{P}'$ and $\mathbb{P} : t[u'/\mathtt{x}] \xrightarrow{p} t'$, so that $\vdash t' : \mathbb{P}$ follows immediately by induction.

Name Abstraction The induction hypothesis says that $\vdash \delta\mathbb{P} : \mathtt{new}\, a.\, p : \mathbb{P}''$ and $\mathbb{P} : t \xrightarrow{p} t'$. Therefore $\mathbb{P}'' = \delta\mathbb{P}'$ for some $\mathbb{P}'$, and hence $\vdash \delta\mathbb{P} : \mathtt{new}\, a.\, p : \delta\mathbb{P}'$. From lemma 4.2.2.7(ii) therefore $\vdash \mathbb{P} : p : \mathbb{P}'$ so that by induction $\vdash t' : \mathbb{P}'$ and hence $\vdash \mathtt{new}\, a\,.\, t' : \delta\mathbb{P}'$ as required.

Name Application The induction hypothesis says that $\vdash \mathbb{P} : p : \mathbb{P}'$ and also $\delta\mathbb{P} : t \xrightarrow{\mathtt{new}\, a.\, p} \mathtt{new}\, a\,.\, t'$ so that $\vdash \delta\mathbb{P} : \mathtt{new}\, a.\, p : \delta\mathbb{P}'$ and hence by induction $\vdash \mathtt{new}\, a\,.\, t' : \delta\mathbb{P}'$. From lemma 4.2.1.15(vi) therefore $\vdash t' : \mathbb{P}'$ as required.

Function Abstraction The induction hypothesis says that $\vdash \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}'$ and $\mathbb{P} : t[u/\mathtt{x}] \xrightarrow{p} t'$. From lemma 4.2.2.7(i) therefore $\vdash \mathbb{P} : p : \mathbb{P}'$ and hence by induction $\vdash t' : \mathbb{P}'$ as required.

Function Application The induction hypothesis says that $\vdash \mathbb{P} : p : \mathbb{P}'$ and also $\mathbb{Q} \to \mathbb{P} : t \xrightarrow{u \mapsto p} t'$. By lemma 4.4.1.2 there exists a unique $\mathbb{P}''$ such that $\vdash \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}''$ and by lemma 4.2.2.7(i) $\vdash u : \mathbb{Q}$ and $\vdash \mathbb{P} : p : \mathbb{P}''$ so that $\mathbb{P}'' = \mathbb{P}'$. Therefore by induction $\vdash t' : \mathbb{P}'$ as required.

Labelling The induction hypothesis says that $\mathbb{P}_{\ell_0} : t \xrightarrow{p} t'$ and also that $\vdash \bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0{:}p : \mathbb{P}'$. From lemma 4.2.2.7(iii) therefore $\vdash \mathbb{P}_{\ell_0} : p : \mathbb{P}'$ so that $\vdash t' : \mathbb{P}'$ by induction as required.

Label Projection The induction hypothesis says that $\vdash \mathbb{P}_{\ell_0} : p : \mathbb{P}'$ and $\bigoplus_{\ell \in L} \mathbb{P}_\ell : t \xrightarrow{\ell_0 : p} t'$. From the typing rules $\vdash \bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0{:}p : t'$ so that $\vdash t' : \mathbb{P}'$ by induction as required.

Recursive Type Folding The induction hypothesis gives the statements that $\mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : t \xrightarrow{p} t'$ and $\vdash \mu_j \vec{P}.\, \vec{\mathbb{P}} : \mathtt{abs}\ p : \mathbb{P}'$. From lemma 4.2.2.7(iv) therefore $\vdash \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}'$, so that $\vdash t' : \mathbb{P}'$ by induction as required.

Recursive Type Unfolding The induction hypothesis gives the statements that $\vdash \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}'$ and $\mu_j \vec{P}.\, \vec{\mathbb{P}} : t \xrightarrow{\mathtt{abs}\ p} t'$. From the typing rules it follows that $\vdash \mu_j \vec{P}.\, \vec{\mathbb{P}} : \mathtt{abs}\ p : t'$ so that $\vdash t' : \mathbb{P}'$ by induction as required.

Nondeterministic Sum The induction hypothesis says that $\vdash \mathbb{P} : p : \mathbb{P}'$ and $\mathbb{P} : t_{i_0} \xrightarrow{p} t'$ so that $\vdash t' : \mathbb{P}'$ by induction as required. □

4.4.1.4 Lemma. *If the conclusion of certain derivation rules for the operational semantics is derivable then so are its premises, in a sense that is made precise below.*

(i) If $\mathbb{P} : t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)} \xrightarrow{p} t'$ *then* $\mathbb{Q} \to \mathbb{P} : t \xrightarrow{u \mapsto p} t'$.

(ii) If $\mathbb{P}_{\ell_0} : \pi_{\ell_0} t \xrightarrow{p} t'$ *then* $\bigoplus_{\ell \in L} \mathbb{P}_\ell : t \xrightarrow{\ell_0 : p} t'$.

(iii) If $\mathbb{P} : t\mathtt{[}a\mathtt{]} \xrightarrow{p} t'$ *then* $\delta\mathbb{P} : t \xrightarrow{\mathtt{new}\, a\,.\, p} \mathtt{new}\, a\,\mathtt{.}\, t'$.

(iv) If $\mathbb{P}_j[\mu_j \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : \mathtt{rep}\, t \xrightarrow{p} t'$ *then* $\mu_j \vec{P}.\, \vec{\mathbb{P}} : t \xrightarrow{\mathtt{abs}\ p} t'$.

Proof. The derivation rules are completely syntax-directed, so the result is immediate. □

Chapter 5

The Denotational Semantics of Nominal HOPLA

It was claimed that the nominal domain theory developed in chapter 3 was to motivate the design of the process calculus Nominal HOPLA as presented in the previous chapter. The present chapter provides the evidence to back up this claim. In particular, section 5.2 demonstrates that the denotational semantics of Nominal HOPLA arises directly from the various universal properties in the categories $(\mathbf{FMCts}_s)_{s \subseteq_{\mathrm{fin}} \mathbb{A}}$. That the denotational semantics is given by *universal* properties is important: it provides weight to the claim that the nominal domain theory captures the mathematical essence of the computational features that are supported. The design of the name-free process calculus HOPLA[20] was also guided by the principle of universal constructions, and Nominal HOPLA can be seen as a straightforward extension of HOPLA with terms of the form $\mathtt{new}\, a\,\mathtt{.}\, t$ and $t\,\mathtt{[}a\mathtt{]}$ which arise directly from the adjunction $(-)^{\#a{+}{+}} \dashv \delta_a^{++}$.

After the definition of the denotational semantics in sections 5.1 and 5.2, section 5.3 demonstrates that substitution in Nominal HOPLA effectively amounts to composition, and finally section 5.4 shows that the operational semantics defined above corresponds closely with the denotational semantics given here.

5.1 Types and Environments

A closed type denotes the collection of paths of the appropriate type, ordered by extension. Such *path orders*, even recursively-defined ones, can be constructed inductively out of syntactic tokens by a method inspired by the use of information systems to solve recursive domain equations[37] as demonstrated below. With no freshness assumptions an environment denotes the product of the path orders denoted by the types of its variables. The functors $\big((-)^{\#s++}\big)_{s\subseteq_{\mathrm{fin}}\mathbb{A}}$ are used to modify this product to incorporate freshness assumptions, as demonstrated in section 5.1.2.

5.1.1 Types as Path Orders

A closed type $\mathbb{P}$ denotes the collection of paths of type $\mathbb{P}$ ordered by extension. (It is conventional, albeit confusing, to omit the semantic brackets $[\![\cdot]\!]$ around the denotations of types in HOPLA and its derivatives.) It is therefore possible to construct a denotation for $\mathbb{P}$ in terms of a language of paths, given by the grammar

$$p ::= Q \mid Q \mapsto p \mid \ell{:}p \mid \texttt{abs}\ p \mid \texttt{new}\, a.\, p, \tag{5.1.1.1}$$

where ℓ is a label, a is a name, and Q (and later P) is a set of paths of the form $\langle\{p_1, \ldots, p_n\}\rangle_s$ which denotes an element of $!\mathbb{P}$ as defined in 3.4.4.2. A path of the form $Q \mapsto p$ denotes a pair $\langle Q, p\rangle$ in the continuous function space $\mathbb{Q} \to \mathbb{P} = !\mathbb{Q}^{\mathrm{op}} \times \mathbb{P}$. A path of the form $\texttt{new}\, a.\, p$ denotes an equivalence class $[a].p$ in the path order $\delta\mathbb{P}$, and in particular if b is a fresh name then $\texttt{new}\, a.\, p = \texttt{new}\, b.\, ((ab) \cdot p)$. A path of the form $\ell_0{:}p$ denotes a path in the ℓ_0th component of the biproduct $\bigoplus_{\ell\in L}\mathbb{P}_\ell$. Finally a path of the form $\texttt{abs}\ p$ denotes a path in the recursively-defined type $\mu_j\vec{P}.\ \vec{\mathbb{P}}$, where p is a path of the unfolded type $\mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$.

It is recognised that confusion might arise from using the letter p here to range over paths whereas in the previous chapter p was used for actions. There is an unfortunate clash between the use of a to range over actions in the standard treatment of HOPLA and the use of a to range over names in the standard treatment of nominal set theory. Using a for both of these purposes here is impossible, but there are no other appropriate letters, so because of the closeness of the relationship between paths and actions, p is used for both.

To capture the intuition described above, paths are typed by judgements of the form $p : \mathbb{P}$ according to the following rules.

$$\frac{p_1 : \mathbb{P} \quad \ldots \quad p_n : \mathbb{P}}{\langle\{p_1, \ldots, p_n\}\rangle_s : !\mathbb{P}} \qquad \frac{Q : !\mathbb{Q} \quad p : \mathbb{P}}{Q \mapsto p : \mathbb{Q} \to \mathbb{P}} \tag{5.1.1.2}$$

$$\frac{p : \mathbb{P}_{\ell_0}}{\ell_0{:}p : \bigoplus_{\ell\in L}\mathbb{P}_\ell}\ (\ell_0 \in L) \qquad \frac{p : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]}{\texttt{abs}\ p : \mu_j\vec{P}.\ \vec{\mathbb{P}}}$$

$$\frac{p : \mathbb{P}}{\texttt{new}\, a.\, p : \delta\mathbb{P}}$$

where the ordering $\leq_{\mathbb{P}}$ of paths of type $\mathbb{P}$ is given recursively as follows.

$$
\frac{P \preceq_{\mathbb{P}} P'}{P \leq_{!\mathbb{P}} P'} \qquad\qquad \frac{Q' \leq_{!\mathbb{Q}} Q \quad p \leq_{\mathbb{P}} p'}{Q \mapsto p \leq_{\mathbb{Q} \to \mathbb{P}} Q' \mapsto p'} \tag{5.1.1.3}
$$

$$
\frac{p \leq_{\mathbb{P}_{\ell_0}} p'}{\ell_0 : p \leq_{\bigoplus_{\ell \in L} \mathbb{P}_\ell} \ell_0 : p'} \ (\ell_0 \in L) \qquad\qquad \frac{p \leq_{\mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]} p'}{\texttt{abs}\ p \leq_{\mu_j \vec{P}.\, \vec{\mathbb{P}}} \texttt{abs}\ p'}
$$

$$
\frac{p \leq_{\mathbb{P}} p'}{\texttt{new}\, a.\, p \leq_{\delta\mathbb{P}} \texttt{new}\, a.\, p'}
$$

Here, $P \preceq_{\mathbb{P}} P'$ means that for all $p \in P$ there exists $p' \in P'$ such that $p \leq_{\mathbb{P}} p'$. Finally, the permutation action on paths is defined by the following.

$$
\begin{aligned}
\sigma \cdot P &=_{\text{def}} \{\sigma \cdot p \mid p \in P\} \\
\sigma \cdot (Q \mapsto p) &=_{\text{def}} (\sigma \cdot Q) \mapsto (\sigma \cdot p) \\
\sigma \cdot (\ell : p) &=_{\text{def}} (\sigma \cdot \ell) : (\sigma \cdot p) \\
\sigma \cdot (\texttt{abs}\ p) &=_{\text{def}} \texttt{abs}\ (\sigma \cdot p) \\
\sigma \cdot (\texttt{new}\, a.\, p) &=_{\text{def}} \texttt{new}\, (\sigma a).\, (\sigma \cdot p).
\end{aligned} \tag{5.1.1.4}
$$

This method of constructing recursive types syntactically is inspired by the similar process used in classical HOPLA[20] which in turn is inspired by the theory of information systems[37]. Here it is straightforward to show that these definitions construct path orders that are nominal preorders and hence objects of $\mathbf{FMPre}_{\varnothing}$. As in HOPLA, in a recursively-defined type $\mu_j \vec{P}.\, \vec{\mathbb{P}}$ each path is of the form $\texttt{abs}\ p$ which means there is an isomorphism

$$
\mathbf{rep} : \mu_j \vec{P}.\, \vec{\mathbb{P}} \cong \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : \mathbf{abs}, \tag{5.1.1.5}
$$

where $\mathbf{abs}(p) =_{\text{def}} \texttt{abs}\ p$ and $\mathbf{rep}(\texttt{abs}\ p) =_{\text{def}} p$.

It is straightforward to show that each type $\mathbb{P}$ is a FM-preorder with empty support and therefore an object of $\mathbf{FMPre}_{\varnothing}$. Importantly, this implies that for all names a it is the case that $\delta_a \mathbb{P} = \delta \mathbb{P}$.

5.1.2 Environments as Products

Environments Γ (with freshness constraints contained in s_0) denote objects of $\mathbf{FMCts}_{s_0}$ as follows. $\mathbb{O}$ is the empty preorder.

$$
[\![()]\!] = \mathbb{O} \qquad \text{and} \qquad [\![\Gamma, \mathtt{x} : \mathbb{P}^{\# s}]\!] = [\![\Gamma]\!] \,\&\, \mathbb{P}^{\# s} \tag{5.1.2.1}
$$

To help provide a suggestive notation for the elements of $[\![\Gamma]\!]$, environments Γ also denote objects of $\mathbf{FMPre}_{s_0}$ as follows.

$$
[\![()]\!]' = \widehat{\mathbb{O}} \qquad \text{and} \qquad [\![\Gamma, \mathtt{x} : \mathbb{P}^{\# s}]\!]' = [\![\Gamma]\!]' \times \widehat{\mathbb{P}}^{\# s} \tag{5.1.2.2}
$$

so that elements of $[\![\Gamma]\!]'$ are tuples of path-sets satisfying the appropriate freshness constraints. The transformations m and $\phi^{(s)}$ of 3.4.8.7 and 3.4.8.34 can be used to build a monotone map $\langle\cdot\rangle_\Gamma : [\![\Gamma]\!]' \to \widehat{[\![\Gamma]\!]}$ by recursion on Γ by setting $\langle\cdot\rangle_{()} : [\![()]\!]' \to \widehat{[\![()]\!]} \cong \{\varnothing\}$ to be the unique such function, and

$$\langle\cdot\rangle_{\Gamma,\mathtt{x}:\mathbb{P}^{\#s}} =_{\mathrm{def}} m_{[\![\Gamma]\!],\mathbb{P}^{\#s}} \circ \left(\langle\cdot\rangle_\Gamma \times \phi^{(s)}_{\mathbb{P}}\right). \tag{5.1.2.3}$$

Therefore

$$\langle\gamma, p\rangle_{\Gamma,\mathtt{x}:\mathbb{P}^{\#s}} = \langle\gamma\rangle_\Gamma \uplus \{x \in p \mid x \# s\}. \tag{5.1.2.4}$$

5.2 Terms and Actions

Typing judgements $\Gamma \vdash_s t : \mathbb{P}$ denote arrows

$$[\![\Gamma \vdash_s t : \mathbb{P}]\!] : [\![\Gamma]\!] \underset{\mathbf{C}}{\to} \mathbb{P} \tag{5.2.0.1}$$

in $\mathbf{FMCts}_s$. The denotation of a typing judgement is built by recursion on the derivation of the typing judgement, making use of the various universal constructions available in $\mathbf{FMCts}_s$ as described in section 3.4.8. Intuitively, the arrow $[\![\Gamma \vdash_s t : \mathbb{P}]\!]$ receives some input in its free variables, as typed by Γ, and returns the set of paths that the term t can perform with the given input.

Exponentials in $\mathbf{FMCts}_s$ give rise to a semantics for λ-abstraction (5.2.1.1) and function application (5.2.1.3) much as in the simply-typed λ-calculus[15].

The monad $(!, \eta, \epsilon_!)$ on $\mathbf{FMCts}_s$ gives rise to the semantics for the anonymous prefix action $\mathtt{!}$ (5.2.2.1) and for prefix-matching (5.2.2.7). Combined with biproducts $\bigoplus_{\ell\in L}\mathbb{P}_\ell$ this gives a semantics for labelled actions: the injection $\mathbf{in}_{\ell_0}$ into the biproduct corresponds to tagging an action with the label ℓ_0, whereas the projection $\mathbf{out}_{\ell_0}$ corresponds to matching against the label ℓ_0.

The isomorphism $\mathbf{rep} : \mu\vec{P}.\ \vec{\mathbb{P}} \cong \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}] : \mathbf{abs}$ of 5.1.1.5 gives a semantics for terms of recursively-defined types by folding (5.2.4.4) and unfolding (5.2.4.6) the definition. The hom-set $\mathbf{FMCts}_s([\![\Gamma]\!], \mathbb{P})$ is ω-complete which gives rise to a semantics for recursive processes (5.2.4.2) as the ω-limit of their finite unfoldings.

By 3.4.8.2 the hom-sets may be collected together to form

$$\bigcup_{\mathrm{supp}(\Gamma)\subseteq s\subseteq_{\mathrm{fin}}\mathbb{A}} \mathbf{FMCts}_s([\![\Gamma]\!], \mathbb{P}) \cong \widehat{![\![\Gamma]\!]^{\mathrm{op}} \times \mathbb{P}} \tag{5.2.0.2}$$

which has more general joins than merely those of ω-chains, and such joins give rise to a semantics for nondeterministic sums (5.2.5.1).

The adjunction $(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s\dot{\cup}\{a\}}$ of 3.4.8.27 gives a semantics of names and binding as demonstrated in 5.2.6.1 and 5.2.6.10.

Finally, the structural rules for variables, weakening, exchange and contraction make use of the cartesian structure of $\mathbf{FMCts}_s$ in the usual fashion, as shown in 5.2.7.3, 5.2.7.4, 5.2.7.6 and 5.2.7.8. The nominal structure gives rise to two extra structural rules. The first permits adding extra assumptions on the freshness of input with respect to certain names, which receives its semantics (5.2.7.10) from the map τ defined in 3.2.1.21. The second permits extending the the 'current' set of names from s to $s' \supseteq s$, and this rule receives its semantics (5.2.7.12) from the inclusion $\mathbf{FMCts}_s \hookrightarrow \mathbf{FMCts}_{s'}$.

Typing judgements $\vdash_s \mathbb{P} : p : \mathbb{P}'$ denote arrows

$$[\![\vdash_s \mathbb{P} : p : \mathbb{P}']\!] : \mathbb{P} \underset{\mathbf{C}}{\to} !\mathbb{P}' \tag{5.2.0.3}$$

in $\mathbf{FMCts}_s$ by recursion on the structure of p as shown below. Intuitively the arrow $[\![\vdash_s \mathbb{P} : p : \mathbb{P}']\!]$ matches its input against the action p and returns a collection of possible resumptions after performing p. If the type information is clear then $[\![\Gamma \vdash_s t : \mathbb{P}]\!]$ and $[\![\vdash_s \mathbb{P} : p : \mathbb{P}']\!]$ are abbreviated to $[\![t]\!]$ and $[\![p]\!]$ respectively.

5.2.1 Higher-Order Processes

The cartesian-closed structure of $\mathbf{FMPre}_s$, described in 3.4.8.17, gives rise to a semantics for higher-order processes as for the simply-typed λ-calculus. Firstly, abstraction of a variable of type $\mathbb{P}$ is simply given by transposition in the exponential adjunction $(-) \,\&\, \mathbb{P} \dashv \mathbb{P} \to (-)$ as follows.

5.2.1.1 Definition (Semantics of Function Abstraction). *If the judgement* $\Gamma \vdash_s \lambda\mathtt{x}.t : \mathbb{Q} \to \mathbb{P}$ *is derived from* $\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$ *then define* $[\![\Gamma \vdash_s \lambda\mathtt{x}.t : \mathbb{Q} \to \mathbb{P}]\!]$ *to be the exponential transpose of* $[\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]$ *in* $\mathbf{FMCts}_s$ *as defined in 3.4.8.20.*

The semantics of function abstraction can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.1.2 Lemma (Characterising Function Abstraction). *Suppose that* $\Gamma \vdash_s \lambda\mathtt{x}.t : \mathbb{Q} \to \mathbb{P}$ *is derived from* $\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$. *Let* $\gamma \in [\![\Gamma]\!]'$, $Q \in !\mathbb{Q}$ *and* $p \in \mathbb{P}$. *Then*

$$Q \mapsto p \in [\![\Gamma \vdash_s \lambda\mathtt{x}.t : \mathbb{Q} \to \mathbb{P}]\!]\langle\gamma\rangle_\Gamma \text{ iff } p \in [\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]\langle\gamma, Q_\downarrow\rangle_{\Gamma, \mathtt{x}:\mathbb{Q}^{\#\varnothing}}.$$

Proof. By 3.4.8.20. □

Application of a function to its argument is given by the counit of the exponential adjunction as follows.

5.2.1.3 Definition (Semantics of Function Application). *If the judgement $\Gamma, \Lambda \vdash_s t(u{:}\mathbb{Q}) : \mathbb{P}$ is derived from $\Gamma \vdash_s t : \mathbb{Q} \to \mathbb{P}$ and $\Lambda \vdash_s u : \mathbb{Q}$ then define $[\![\Gamma, \Lambda \vdash_s t(u{:}\mathbb{Q}) : \mathbb{P}]\!]$ to be*

$$\mathbf{apply} \circ ([\![\Gamma \vdash_s t : \mathbb{Q} \to \mathbb{P}]\!] \,\&\, [\![\Lambda \vdash_s u : \mathbb{Q}]\!]),$$

where **apply** *is the counit of the exponential adjunction in* $\mathbf{FMCts}_s$ *as defined in 3.4.8.19.*

The semantics of function application can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.1.4 Lemma (Characterising Function Application). *Suppose that the judgement $\Gamma, \Lambda \vdash_s t(u{:}\mathbb{Q}) : \mathbb{P}$ is derived from $\Gamma \vdash_s t : \mathbb{Q} \to \mathbb{P}$ and $\Lambda \vdash_s u : \mathbb{Q}$. Let $\gamma \in [\![\Gamma]\!]'$, $\lambda \in [\![\Lambda]\!]'$ and $p \in \mathbb{P}$. Then*

$$p \in [\![\Gamma, \Lambda \vdash_s t(u{:}\mathbb{Q}) : \mathbb{P}]\!]\langle \gamma, \lambda \rangle_{\Gamma,\Lambda}$$

iff there exists $Q \in {!\mathbb{Q}}$ such that

$$Q \subseteq [\![\Lambda \vdash_s u : \mathbb{Q}]\!]\langle \lambda \rangle_\Lambda \text{ and } Q \mapsto p \in [\![\Gamma \vdash_s t : \mathbb{Q} \to \mathbb{P}]\!]\langle \gamma \rangle_\Gamma.$$

Proof. By 3.4.8.19. □

The denotational semantics of the higher-order action $u \mapsto p$ is closely related to that of function application to the argument u as follows.

5.2.1.5 Definition (Semantics of Higher-Order Actions). *If the judgement $\vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}'$ is derived from $\vdash_s \mathbb{P} : p : \mathbb{P}'$ and $\vdash_s u : \mathbb{Q}$ then define*

$$[\![\, \vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}']\!] =_{\mathrm{def}} [\![\, \vdash_s \mathbb{P} : p : \mathbb{P}']\!] \circ \mathbf{apply} \circ \big(\mathbf{1}_{\mathbb{Q} \to \mathbb{P}} \,\&\, [\![\, \vdash_s u : \mathbb{Q}]\!]\big).$$

The semantics of the higher-order action $u \mapsto p$ can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.1.6 Lemma (Characterising Higher-Order Actions). *Suppose that $\vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}'$ is derived from $\vdash_s \mathbb{P} : p : \mathbb{P}'$ and $\vdash_s u : \mathbb{Q}$. Let $X \in \widehat{\mathbb{Q} \to \mathbb{P}}$. Then*

$$\begin{aligned}&[\![\, \vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}']\!] X \\ &\quad = [\![\, \vdash_s \mathbb{P} : p : \mathbb{P}']\!] \{x \in \mathbb{P} \mid \exists Q \in {!\mathbb{Q}}.\ Q \subseteq [\![\, \vdash_s u : \mathbb{Q}]\!] \text{ and } Q \mapsto x \in X\}.\end{aligned}$$

Proof. By 3.4.8.19. □

5.2.2 Prefixing and Matching

The adjunction $\mathbf{FMLin}_s(!\mathbb{P},\mathbb{Q}) \cong \mathbf{FMCts}_s(\mathbb{P},\mathbb{Q})$ of 3.4.4.15 gives rise to a semantics for an anonymous prefix action, written `!`. The unit η acts as a constructor for this action, taking a term t to the prefixed term $\texttt{!}t$ as follows.

5.2.2.1 Definition (Semantics of Prefix).

$$[\![\Gamma \vdash_s \texttt{!}t : !\mathbb{P}]\!] =_{\text{def}} \eta_{\mathbb{P}} \circ [\![\Gamma \vdash_s t : \mathbb{P}]\!].$$

The semantics of prefixing can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.2.2 Lemma (Characterising Prefixing). *Suppose that $\Gamma \vdash_s \texttt{!}t : !\mathbb{P}$ is derived from $\Gamma \vdash_s t : \mathbb{P}$. Let $\gamma \in [\![\Gamma]\!]'$ and $P \in !\mathbb{P}$. Then*

$$P \in [\![\Gamma \vdash_s \texttt{!}t : !\mathbb{P}]\!]\langle\gamma\rangle_\Gamma \textit{ iff } P \subseteq [\![\Gamma \vdash_s t : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma.$$

Proof. By the definition of η. □

For example, the process $\vdash_\varnothing \texttt{nil} : \mathbb{O}$ can perform no action and has empty denotation, but $[\![\vdash_\varnothing \texttt{!nil} : !\mathbb{O}]\!] = \eta_{\mathbb{O}} \circ [\![\texttt{nil}]\!] = \{\varnothing\}$ is not empty and indeed the operational semantics shows that `!nil` can perform an action. The denotation of the judgement $\vdash_\varnothing !\mathbb{P} : \texttt{!} : \mathbb{P}$ is simply the identity map, as follows.

5.2.2.3 Definition (Semantics of Prefix Action).

$$[\![\vdash_\varnothing !\mathbb{P} : \texttt{!} : \mathbb{P}]\!] =_{\text{def}} \mathbf{1}_{!\mathbb{P}}.$$

The counit ϵ of the adjunction of 3.4.4.15 acts as a destructor for the `!` action, intuitively 'matching' a `!` action in the output of a term u and passing the resumption after performing the `!` to a term t. More precisely, if $\Lambda \vdash_s u : !\mathbb{Q}$ and $\texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$ then

$$\begin{aligned}&[\![\Lambda \vdash_s \texttt{[}u\texttt{ > !(x:}\mathbb{Q}\texttt{ \# }\varnothing\texttt{) => }t\texttt{]} : \mathbb{P}]\!] \\ &\quad =_{\text{def}} \epsilon_{\mathbb{P}} \circ ![\![\texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] \circ [\![\Lambda \vdash_s u : !\mathbb{Q}]\!].\end{aligned} \tag{5.2.2.4}$$

For example, $[\![\vdash_\varnothing \texttt{[!nil > !(x:}\mathbb{O}\texttt{ \# }\varnothing\texttt{) => x]} : \mathbb{O}]\!] = \epsilon_{\mathbb{O}} \circ \eta_{\mathbb{O}} \circ [\![\texttt{nil}]\!] = \varnothing$, and the operational semantics shows that after `!nil` has performed the action `!` it can perform no further actions. This covers the case where `x` is the only free variable of t. More generally, if $\Gamma, \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$ then it is necessary to use the strength map S of 3.4.8.23 to pass the extra parameters to t as follows.

$$\begin{aligned}&[\![\Gamma, \Lambda \vdash_s \texttt{[}u\texttt{ > !(x:}\mathbb{Q}\texttt{ \# }\varnothing\texttt{) => }t\texttt{]} : \mathbb{P}]\!] \\ &\quad =_{\text{def}} \epsilon_{\mathbb{P}} \circ ![\![\Gamma, \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] \circ S_{\Gamma,\mathbb{Q}} \circ (\mathbf{1}_\Gamma \,\&[\![\Lambda \vdash_s u : !\mathbb{Q}]\!]).\end{aligned} \tag{5.2.2.5}$$

Even more generally, the type of u may be $\mathbb{Q}'$ and not be in the form $!\mathbb{Q}$ so that u may perform actions other than ! and it is useful to be able to match against any action q that u may perform. More precisely, suppose now that $\Lambda \vdash_s u : \mathbb{Q}'$ and also that $\vdash_s \mathbb{Q}' : q : \mathbb{Q}$, then matching the output of u against the action q can be achieved using the map $\llbracket \vdash_s \mathbb{Q}' : q : \mathbb{Q} \rrbracket : \mathbb{Q}' \underset{\mathbf{C}}{\to} !\mathbb{Q}$ as follows.

$$\begin{aligned}
&\llbracket \Gamma, \Lambda \vdash_s \texttt{[}u \texttt{ > } q\texttt{(x:}\mathbb{Q}' \texttt{ \# } \varnothing\texttt{) => } t\texttt{]} : \mathbb{P} \rrbracket \\
&\quad =_{\text{def}} \epsilon_{\mathbb{P}} \circ !\llbracket \Gamma, \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P} \rrbracket \circ S_{\Gamma,\mathbb{Q}} \\
&\quad \circ (\mathbf{1}_\Gamma \,\&(\llbracket \vdash_s \mathbb{Q}' : q : \mathbb{Q} \rrbracket \circ \llbracket \Lambda \vdash_s u : \mathbb{Q}' \rrbracket)).
\end{aligned} \tag{5.2.2.6}$$

In full generality, the free variable x of t may have some freshness assumptions imposed on it, which leads to the following definition.

5.2.2.7 Definition (Semantics of Matching). *Suppose that typing judgements $\Gamma, \texttt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}$, $\Lambda \vdash_{s''} u : \mathbb{Q}'$ and $\vdash_{s''} \mathbb{Q}' : q : \mathbb{Q}$ hold, where $s'' \subseteq s \setminus s'$. Let $\phi^{!(s')} : (!-)^{\#s'} \cong !((-)^{\#s'})$ be as defined in 3.4.8.36, then*

$$\begin{aligned}
&\llbracket \Gamma, \Lambda \vdash_s \texttt{[}u \texttt{ > } q\texttt{(x:}\mathbb{Q}' \texttt{ \# } s'\texttt{) => } t\texttt{]} : \mathbb{P} \rrbracket \\
&\quad =_{\text{def}} \epsilon_{\mathbb{P}} \circ !\llbracket \Gamma, \texttt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P} \rrbracket \circ S_{\Gamma,\mathbb{Q}^{\#s'}} \\
&\quad \circ \left(\mathbf{1}_\Gamma \,\& \left(\widehat{\phi^{!(s')}_{\mathbb{Q}}} \circ \llbracket \vdash_{s''} \mathbb{Q}' : q : \mathbb{Q} \rrbracket^{\#s'++} \circ \llbracket \Lambda \vdash_{s''} u : \mathbb{Q}' \rrbracket^{\#s'++} \right) \right).
\end{aligned}$$

The semantics of prefix matching can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.2.8 Lemma (Characterising Matching). *Suppose that $\gamma \in \llbracket \Gamma \rrbracket'$ and $\lambda \in \llbracket \Lambda^{\#s'} \rrbracket'$ and $p \in \mathbb{P}$. Then*

$$p \in \llbracket \Gamma, \Lambda^{\#s'} \vdash_s \texttt{[}u \texttt{ > } q\texttt{(x:}\mathbb{Q}' \texttt{ \# } s'\texttt{) => } t\texttt{]} : \mathbb{P} \rrbracket \langle \gamma, \lambda \rangle_{\Gamma,\Lambda^{\#s'}}$$

iff there exists $Q \in !\mathbb{Q}'$ such that $p \in \llbracket t \rrbracket \langle \gamma, Q \rangle_{\Gamma,\texttt{x}:\mathbb{Q}'^{\#s'}}$, $Q \in \big(\llbracket q \rrbracket \circ \llbracket u \rrbracket \big) \langle \lambda \rangle_\Lambda$ and $Q \# s'$.

Proof. Let

$$f = \widehat{\phi^{!(s')}_{\mathbb{Q}'}} \circ \llbracket \vdash_{s''} \mathbb{Q} : q : \mathbb{Q}' \rrbracket^{\#s'++} \circ \llbracket \Lambda \vdash_{s''} u : \mathbb{Q} \rrbracket^{\#s'++}, \tag{5.2.2.9}$$

then

$$\begin{aligned}
f\langle \lambda \rangle_{\Lambda^{\#s'}} &= \left(\widehat{\phi^{!(s')}_{\mathbb{Q}'}} \circ \big(\llbracket q \rrbracket \circ \llbracket u \rrbracket \big)^{\#s'++} \circ \phi^{(s')}_{\llbracket \Lambda \rrbracket} \right) \langle \lambda \rangle_\Lambda \\
&= \left(\widehat{\phi^{!(s')}_{\mathbb{Q}'}} \circ \phi^{(s')}_{!\mathbb{Q}'} \circ \big(\llbracket q \rrbracket \circ \llbracket u \rrbracket \big)^{\#s'} \right) \langle \lambda \rangle_\Lambda \\
&= \widehat{\phi^{!(s')}_{\mathbb{Q}'}} \{ Q \in (\llbracket q \rrbracket \circ \llbracket u \rrbracket) \langle \lambda \rangle_\Lambda \mid Q \# s' \} \\
&= \{ \phi^{!(s')}_{\mathbb{Q}'} Q \mid Q \in (\llbracket q \rrbracket \circ \llbracket u \rrbracket) \langle \lambda \rangle_\Lambda \mid Q \# s' \}_{\downarrow}
\end{aligned} \tag{5.2.2.10}$$

so that $Q' \in f\langle\lambda\rangle_{\Lambda^{\# s'}}$ if and only if there exists $Q \in ([\![q]\!] \circ [\![u]\!])\langle\lambda\rangle_\Lambda$ such that $Q \# s'$ and $Q' \leq_{!\mathbb{Q}'} \phi^{!(s')}_{\mathbb{Q}'} Q$. By definition this last condition is equivalent to $Q' \subseteq \left(\phi^{!(s')}_{\mathbb{Q}'} Q\right)_\downarrow$ and by 3.4.5.2 this is equivalent to $Q' \subseteq \phi^{(s')}_{\mathbb{Q}'}(Q_\downarrow)$.

Unwinding definitions,

$$\begin{aligned}
&[\![\Gamma, \Lambda^{\# s'} \vdash_s \texttt{[}u \texttt{ > } q\texttt{(x:}\mathbb{Q}' \texttt{ \# } s'\texttt{) => } t\texttt{]} : \mathbb{P}]\!]\langle\gamma,\lambda\rangle_{\Gamma,\Lambda^{\# s'}} && (5.2.2.11)\\
&= \left(\epsilon_{\mathbb{P}} \circ ![\![t]\!] \circ S_{\Gamma,\mathbb{Q}'^{\# s'}} \circ (\mathbf{1}_\Gamma \,\&\, f)\right)\langle\gamma,\lambda\rangle_{\Gamma,\Lambda^{\# s'}}\\
&= \bigcup\left(![\![t]\!]\left(S_{\Gamma,\mathbb{Q}'^{\# s'}}(\langle\gamma\rangle_\Gamma \uplus f\langle\lambda\rangle_{\Lambda^{\# s'}})\right)\right)\\
&= \bigcup\left(![\![t]\!]\{G \uplus Q \mid G \subseteq \langle\gamma\rangle_\Gamma, G \in ![\![\Gamma]\!] \text{ and } Q \in f\langle\lambda\rangle_{\Lambda^{\# s'}}\}\right)\\
&= \bigcup\{P \in !\mathbb{P} \mid \exists G \in ![\![\Gamma]\!], Q \in f\langle\lambda\rangle_{\Lambda^{\# s'}}\\
&\qquad \text{such that } G \subseteq \langle\gamma\rangle_\Gamma \text{ and } P \subseteq [\![t]\!](G \uplus Q)_\downarrow\}.
\end{aligned}$$

If $p \in [\![\texttt{[}u \texttt{ > } q\texttt{(x:}\mathbb{Q}' \texttt{ \# } s'\texttt{) => } t\texttt{]}]\!]$ then there exists P, G and Q' such that $p \in P \subseteq [\![t]\!](G \uplus Q')_\downarrow$ where $G \in ![\![\Gamma]\!]$, $G \subseteq \langle\gamma\rangle_\Gamma$ and $Q' \in f\langle\lambda\rangle_{\Lambda^{\# s'}}$ so that by the discussion above there exists $Q \in ([\![q]\!] \circ [\![u]\!])\langle\lambda\rangle_\Lambda$ such that $Q \# s'$ and $Q' \subseteq \phi^{(s')}_{\mathbb{Q}'} Q_\downarrow$. By the monotonicity of $[\![t]\!]$ this means that as required $p \in [\![t]\!]\left(\langle\gamma\rangle_\Gamma \uplus \phi^{(s')}_{\mathbb{Q}'} Q_\downarrow\right) = [\![t]\!]\langle\gamma, Q_\downarrow\rangle_{\Gamma,\mathtt{x}:\mathbb{Q}'^{\# s'}}$.

Conversely if there exists a $Q \in ([\![q]\!] \circ [\![u]\!])\langle\lambda\rangle_\Lambda$ such that $Q \# s'$ and also $p \in [\![t]\!]\langle\gamma, Q_\downarrow\rangle_{\Gamma,\mathtt{x}:\mathbb{Q}'^{\# s'}}$ then by algebraicity there exists G and Q' such that $G \in ![\![\Gamma]\!]$ and $Q' \in !(\mathbb{Q}'^{\# s'})$ and $G \uplus Q' \subseteq \langle\gamma, Q_\downarrow\rangle_{\Gamma,\mathtt{x}:\mathbb{Q}'^{\# s'}}$ and $p \in [\![t]\!](G \uplus Q')_\downarrow$. It follows from the discussion above that $Q' \in f\langle\lambda\rangle_{\Lambda^{\# s'}}$ so that as required $p \in [\![\texttt{[}u \texttt{ > } q\texttt{(x:}\mathbb{Q}' \texttt{ \# } s'\texttt{) => } t\texttt{]}]\!]$. □

5.2.3 Labelled Processes

Generalised biproducts $\bigoplus_{\ell \in L} \mathbb{P}_\ell$, described in 3.4.8.16, give rise to a denotational semantics for labelling. Injection into the biproduct corresponds to tagging the outputs of a process with a particular label as follows.

5.2.3.1 Definition (Semantics of Labelling). *If $\Gamma \vdash_s \ell_0 : t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ is derived from $\Gamma \vdash_s t : \mathbb{P}_{\ell_0}$ then by the typing rules it must be that s supports ℓ_0 so that the injection $\mathbf{in}_{\ell_0} : \mathbb{P}_{\ell_0} \underset{\mathbf{C}}{\to} \bigoplus_{\ell \in L} \mathbb{P}_\ell$ is an arrow of $\mathbf{FMCts}_s$. Therefore define*

$$[\![\Gamma \vdash_s \ell_0 : t : \textstyle\bigoplus_{\ell \in L} \mathbb{P}_\ell]\!] =_{\text{def}} \mathbf{in}_{\ell_0} \circ [\![\Gamma \vdash_s t : \mathbb{P}_{\ell_0}]\!].$$

This definition can be given a concrete, set theoretic, characterisation in terms of paths which highlight how it captures the semantics of labelling as follows.

5.2.3.2 Lemma (Characterising Labelling). *Suppose that the judgement $\Gamma \vdash_s \ell_0 : t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ is derived from $\Gamma \vdash_s t : \mathbb{P}_{\ell_0}$. Let $\gamma \in [\![\Gamma]\!]'$, $\ell \in L$ and $p \in \mathbb{P}_{\ell_0}$. Then*

$$\ell : p \in [\![\Gamma \vdash_s \ell_0 : t : \textstyle\bigoplus_{\ell \in L} \mathbb{P}_\ell]\!]\langle\gamma\rangle_\Gamma \text{ iff } \ell = \ell_0 \text{ and } p \in [\![\Gamma \vdash_s t : \mathbb{P}_{\ell_0}]\!]\langle\gamma\rangle_\Gamma.$$

Proof. By the properties of the biproduct. □

On the other hand projection from the biproduct corresponds to matching the outputs of a process against a particular label as follows.

5.2.3.3 Definition (Semantics of Label Projection). *If the judgement $\Gamma \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}$ is derived from $\Gamma \vdash_s t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ then by the typing rules it must be that s supports ℓ_0 so that the projection $\mathbf{out}_{\ell_0} : \bigoplus_{\ell \in L} \mathbb{P}_\ell \underset{\mathbf{C}}{\to} \mathbb{P}_{\ell_0}$ is an arrow of $\mathbf{FMCts}_s$. Therefore define*

$$[\![\Gamma \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}]\!] =_{\text{def}} \mathbf{out}_{\ell_0} \circ [\![\Gamma \vdash_s t : \textstyle\bigoplus_{\ell \in L} \mathbb{P}_\ell]\!].$$

This definition can be given a concrete, set theoretic, characterisation in terms of paths which highlight how it captures the semantics of label-matching as follows.

5.2.3.4 Lemma (Characterising Label Projection). *Suppose that the typing judgement $\Gamma \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}$ is derived from $\Gamma \vdash_s t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$. Let $\gamma \in [\![\Gamma]\!]'$ and $p \in \mathbb{P}_{\ell_0}$. Then*

$$p \in [\![\Gamma \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}]\!]\langle\gamma\rangle_\Gamma \text{ iff } \ell_0 : p \in [\![\Gamma \vdash_s t : \textstyle\bigoplus_{\ell \in L} \mathbb{P}_\ell]\!]\langle\gamma\rangle_\Gamma.$$

Proof. By the properties of the biproduct. □

The denotational semantics of the labelled action $\ell_0 : p$ is closely related to that of label projection at the label ℓ_0 as follows.

5.2.3.5 Definition (Semantics of Labelled Actions). *If the typing judgement $\vdash_s \bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0 : p : \mathbb{P}'$ is derived from $\vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}'$ then by the typing rules it must be that s supports ℓ_0 so that the projection $\mathbf{out}_{\ell_0} : \bigoplus_{\ell \in L} \mathbb{P}_\ell \underset{\mathbf{C}}{\to} \mathbb{P}_{\ell_0}$ is an arrow of $\mathbf{FMCts}_s$. Therefore define*

$$[\![\ \vdash_s \textstyle\bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0 : p : \mathbb{P}']\!] =_{\text{def}} [\![\ \vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}']\!] \circ \mathbf{out}_{\ell_0}.$$

The semantics of the labelled action $\ell_0 : p$ can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.3.6 Lemma (Characterising Labelled Actions). *Suppose that the typing judgement* $\vdash_s \bigoplus_{\ell\in L}\mathbb{P}_\ell : \ell_0{:}p : \mathbb{P}'$ *is derived from* $\vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}'$*. Let* $X \in \widehat{\bigoplus_{\ell\in L}\mathbb{P}_\ell}$*. Then*

$$\begin{aligned}&[\![\vdash_s \textstyle\bigoplus_{\ell\in L}\mathbb{P}_\ell : \ell_0{:}p : \mathbb{P}']\!]X \\ &\qquad = [\![\vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}']\!]\{x \in \mathbb{P}_{\ell_0} \mid \ell_0{:}x \in X\}.\end{aligned}$$

Proof. By the properties of the biproduct. □

5.2.4 Recursion

By 3.4.8.2 hom-sets of $\mathbf{FMCts}_s$ are posets which have all joins of ω-chains. Let $f : \mathbb{Q} \mathbin{\&} \mathbb{P} \underset{\mathbf{C}}{\to} \mathbb{P}$ be an arrow of $\mathbf{FMCts}_s$ and consider the operation on hom-sets $f^* : \mathbf{FMCts}_s(\mathbb{Q},\mathbb{P}) \to \mathbf{FMCts}_s(\mathbb{Q},\mathbb{P})$ given by

$$f^*(g) = f \circ (\mathbf{1}_{\mathbb{Q}} \mathbin{\&} g) \circ \Delta_{\mathbb{Q}}. \tag{5.2.4.1}$$

Composition preserves joins of ω-chains in both its arguments as noted in 3.4.8.1 and the functor $\mathbf{1} \mathbin{\&} (-)$ preserves joins of ω-chains as noted in 3.4.8.3, so the operation f^* is continuous. Therefore by Kleene's fixpoint theorem it has a least fixed point $\mathrm{fix}(f^*)$ defined as follows. Let $g_0 = \varnothing \in \mathbf{FMCts}_s([\![\Gamma]\!],\mathbb{P})$ and for each $n \in \omega$ define $g_{n+1} = f \circ (\mathbf{1}_\Gamma \mathbin{\&} g_n) \circ \Delta_{[\![\Gamma]\!]}$, then $\mathrm{fix}(f^*)\langle\gamma\rangle_\Gamma = \bigcup_{n\in\omega} g_n\langle\gamma\rangle_\Gamma$. This fixed point gives a semantics for terms of the form $\mathtt{rec\,x}.t$ as follows.

5.2.4.2 Definition (Semantics of Recursive Processes). *If the judgement* $\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}$ *is derived from* $\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}$ *then define*

$$[\![\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}]\!] =_{\mathrm{def}} \mathrm{fix}([\![\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]^*).$$

As expected, the denotation of a recursive process is equivalent to its unfolding, as follows.

5.2.4.3 Lemma (Characterising Recursion). *Suppose that* $\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}$ *is derived from* $\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}$*. Let* $\gamma \in [\![\Gamma]\!]'$*. Then*

$$[\![\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma = [\![\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]\langle\gamma, [\![\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma\rangle_{\Gamma,\mathtt{x}:\mathbb{P}^{\#\varnothing}}.$$

Proof.

$$\begin{aligned}
&[\![\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma \\
&\quad = \mathrm{fix}([\![\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]^*)\langle\gamma\rangle_\Gamma \\
&\quad = \big([\![\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_\Gamma \mathbin{\&} \mathrm{fix}([\![\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]^*))\big)(\langle\gamma\rangle_\Gamma \uplus \langle\gamma\rangle_\Gamma) \\
&\quad = \big([\![\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_\Gamma \mathbin{\&} [\![\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}]\!])\big)(\langle\gamma\rangle_\Gamma \uplus \langle\gamma\rangle_\Gamma) \\
&\quad = [\![\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]\big(\langle\gamma\rangle_\Gamma \uplus [\![\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma\big) \\
&\quad = [\![\Gamma, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]\langle\gamma, [\![\Gamma \vdash_s \mathtt{rec\,x}.t : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma\rangle_{\Gamma,\mathtt{x}:\mathbb{P}^{\#\varnothing}}
\end{aligned}$$

□

Recursively-defined processes need recursively-defined types. The isomorphism $\mathbf{rep} : \mu_j\vec{P}.\ \vec{\mathbb{P}} \cong \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}] : \mathbf{abs}$, relating a recursive type to its unfolding, gives rise to the denotational semantics for terms of recursive types as follows.

5.2.4.4 Definition (Semantics of Folding). *If $\Gamma \vdash_s \mathtt{abs}\, t : \mu_j\vec{P}.\ \vec{\mathbb{P}}$ is derived from $\Gamma \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$ then define*

$$[\![\Gamma \vdash_s \mathtt{abs}\, t : \mu_j\vec{P}.\ \vec{\mathbb{P}}]\!] =_{\mathrm{def}} \mathbf{abs} \circ [\![\Gamma \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]]\!].$$

The semantics of folding can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.4.5 Lemma (Characterising Folding). *Suppose that the typing judgement $\Gamma \vdash_s \mathtt{abs}\, t : \mu_j\vec{P}.\ \vec{\mathbb{P}}$ is derived from $\Gamma \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$. Let $\gamma \in [\![\Gamma]\!]'$ and $p \in \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$. Then*

$$\mathtt{abs}\ p \in [\![\Gamma \vdash_s \mathtt{abs}\, t : \mu_j\vec{P}.\ \vec{\mathbb{P}}]\!]\langle\gamma\rangle_\Gamma \textit{ iff } p \in [\![\Gamma \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]]\!]\langle\gamma\rangle_\Gamma.$$

Similarly, unfolding a recursively-defined type uses the inverse **rep** in its semantics.

5.2.4.6 Definition (Semantics of Unfolding). *If $\Gamma \vdash_s \mathtt{rep}\, t : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$ is derived from $\Gamma \vdash_s t : \mu_j\vec{P}.\ \vec{\mathbb{P}}$ then define*

$$[\![\Gamma \vdash_s \mathtt{rep}\, t : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]]\!] =_{\mathrm{def}} \mathbf{rep} \circ [\![\Gamma \vdash_s t : \mu_j\vec{P}.\ \vec{\mathbb{P}}]\!].$$

The semantics of unfolding can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.4.7 Lemma (Characterising Unfolding). *Suppose that the typing judgement $\Gamma \vdash_s \mathtt{rep}\, t : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$ is derived from $\Gamma \vdash_s t : \mu_j\vec{P}.\ \vec{\mathbb{P}}$. Let $\gamma \in [\![\Gamma]\!]'$ and $p \in \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]$. Then*

$$p \in [\![\Gamma \vdash_s \mathtt{rep}\, t : \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}]]\!]\langle\gamma\rangle_\Gamma \textit{ iff } \mathtt{abs}\ p \in [\![\Gamma \vdash_s t : \mu_j\vec{P}.\ \vec{\mathbb{P}}]\!]\langle\gamma\rangle_\Gamma.$$

The denotational semantics of the action $\mathtt{abs}\ p$ is closely related to that of type unfolding as follows.

5.2.4.8 Definition (Semantics of Actions at Recursive Types). *If the judgement $\vdash_s \mu_j\vec{P}.\ \vec{\mathbb{P}} : \mathtt{abs}\ p : \mathbb{P}'$ is derived from $\vdash_s \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}'$ then define*

$$[\![\vdash_s \mu_j\vec{P}.\ \vec{\mathbb{P}} : \mathtt{abs}\ p : \mathbb{P}']\!] =_{\mathrm{def}} [\![\vdash_s \mathbb{P}_j[\mu\vec{P}.\ \vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}']\!] \circ \mathbf{rep}.$$

The semantics of the action $\mathtt{abs}\ p$ can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.4.9 Lemma (Characterising Actions at Recursive Types). *Suppose that* $\vdash_s \mu_j\vec{P}.\,\vec{\mathbb{P}} : \mathtt{abs}\ p : \mathbb{P}'$ *is derived from* $\vdash_s \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}'$*. Let* $X \in \widehat{\mu_j\vec{P}.\,\vec{\mathbb{P}}}$*. Then*

$$\begin{aligned}&[\![\, \vdash_s \mu_j\vec{P}.\,\vec{\mathbb{P}} : \mathtt{abs}\ p : \mathbb{P}']\!]X \\ &\qquad = [\![\, \vdash_s \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}']\!]\{x \in \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] \mid \mathtt{abs}\ x \in X\}.\end{aligned}$$

5.2.5 Nondeterminism

As demonstrated by 3.4.8.2, each hom-set $\mathbf{FMCts}_s([\![\Gamma]\!], \mathbb{P})$ can be seen as a subset of the complete FM-preorder $\widehat{![\![\Gamma]\!]^{\mathrm{op}} \times \mathbb{P}}$. Therefore, given a collection of arrows $f_i : [\![\Gamma]\!] \underset{\mathbf{C}}{\to} \mathbb{P}$ where the mapping $i \mapsto f_i$ is supported by s, the join $\sum_{i\in I} f_i$ of the f_i (in $\widehat{![\![\Gamma]\!]^{\mathrm{op}} \times \mathbb{P}}$) is an arrow of $\mathbf{FMCts}_s$. This join can be used to give a denotational semantics to nondeterministic sums as follows.

5.2.5.1 Definition (Semantics of Nondeterministic Sums). *Suppose that* $\Gamma \vdash_s \sum_{i\in I} t_i : \mathbb{P}$ *is derived from the collection of judgements* $\Gamma \vdash_{s_i} t_i : \mathbb{P}$ *for each* $i \in I$*. Then define*

$$[\![\Gamma \vdash_s \textstyle\sum_{i\in I} t_i : \mathbb{P}]\!] =_{\mathrm{def}} \displaystyle\sum_{i\in I} [\![\Gamma \vdash_{s_i} t_i : \mathbb{P}]\!].$$

This captures the semantics of nondeterminism as follows.

5.2.5.2 Lemma (Characterising Nondeterministic Sums). *Suppose that* $\Gamma \vdash_s \sum_{i\in I} t_i : \mathbb{P}$ *is derived from the collection of judgements* $\Gamma \vdash_{s_i} t_i : \mathbb{P}$ *for each* $i \in I$*. Let* $\gamma \in [\![\Gamma]\!]'$*. Then*

$$[\![\Gamma \vdash_s \textstyle\sum_{i\in I} t_i : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma = \displaystyle\bigcup_{i\in I} \big([\![\Gamma \vdash_{s_i} t_i : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma\big).$$

Proof. Since the join in $\widehat{![\![\Gamma]\!]^{\mathrm{op}} \times \mathbb{P}}$ that defines $[\![\Gamma \vdash_s \sum_{i\in I} t_i : \mathbb{P}]\!]$ is given by pointwise union, this follows immediately from 5.2.5.1. □

5.2.6 Names and Binding

The adjunction $(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s\dot{\cup}\{a\}}$ of 3.4.8.27 gives rise to the denotational semantics for terms of the form $\mathtt{new}\, a\mathtt{.}t$ and $t\mathtt{[}a\mathtt{]}$ as follows.

5.2.6.1 Definition (Semantics of Name Abstraction). *If the judgement $\Gamma \vdash_s \mathtt{new}\, a\mathtt{.}t : \delta\mathbb{P}$ is derived from $\Gamma^{\#a} \vdash_{s\dot{\cup}\{a\}} t : \mathbb{P}$ where $a \notin s$ then define $[\![\Gamma \vdash_s \mathtt{new}\, a\mathtt{.}t : \delta\mathbb{P}]\!]$ to be the transpose of $[\![\Gamma^{\#a} \vdash_{s\dot{\cup}\{a\}} t : \mathbb{P}]\!]$ in the adjunction $(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s\dot{\cup}\{a\}}$. In other words,*

$$[\![\Gamma \vdash_s \mathtt{new}\, a\mathtt{.}t : \delta\mathbb{P}]\!] =_{\mathrm{def}} \delta_a^{++}[\![\Gamma^{\#a} \vdash_{s\dot{\cup}\{a\}} t : \mathbb{P}]\!] \circ \widehat{\xi_{[\![\Gamma]\!]}}.$$

The semantics of name abstraction can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.6.2 Lemma (Characterising Name Abstraction). *Suppose that the judgement $\Gamma \vdash_s \mathtt{new}\, a\mathtt{.}t : \delta\mathbb{P}$ is derived from $\Gamma^{\#a} \vdash_{s\dot{\cup}\{a\}} t : \mathbb{P}$ where $a \notin s$. Let $\gamma \in [\![\Gamma]\!]'$, let b be a fresh name and let $p \in \mathbb{P}$. Then*

$$\mathtt{new}\, b\mathtt{.}\, p \in [\![\Gamma \vdash_s \mathtt{new}\, a\mathtt{.}t : \delta\mathbb{P}]\!]\langle\gamma\rangle_\Gamma \text{ iff } p \in [\![\Gamma^{\#b} \vdash_{s\dot{\cup}\{b\}} (ab) \cdot t : \mathbb{P}]\!]\langle\gamma\rangle_{\Gamma^{\#b}}.$$

Proof. Firstly, note that since $b \# \gamma$ it follows that $\gamma \in [\![\Gamma^{\#b}]\!]'$ so that $\langle\gamma\rangle_{\Gamma^{\#b}}$ is well-defined. Also it is the case that $\langle\gamma\rangle_{\Gamma^{\#b}} = \Big(\big(\theta^{-1}_{[\![\Gamma]\!]^{\#a}} \circ \widehat{\xi_{[\![\Gamma]\!]}}\big)\langle\gamma\rangle_\Gamma\Big) \mathbin{@} b$ as follows. Let $x \in \langle\gamma\rangle_{\Gamma^{\#b}}$, then $x \in \langle\gamma\rangle_\Gamma$ and $x \# b$ so that $[b].x \in \widehat{\xi_{[\![\Gamma]\!]}}\langle\gamma\rangle_\Gamma$. Let c be a fresh name, then $[b].x = (bc) \cdot ([b].x) = [c].((bc) \cdot x)$. By the definition of θ^{-1} it follows that

$$\big(\theta^{-1}_{[\![\Gamma]\!]^{\#a}} \circ \widehat{\xi_{[\![\Gamma]\!]}}\big)\langle\gamma\rangle_\Gamma = \mathbf{fresh}\, c\, \mathbf{in}\, [c].\{x \mid [c].x \in \widehat{\xi_{[\![\Gamma]\!]}}\langle\gamma\rangle_\Gamma\} \tag{5.2.6.3}$$

so that

$$\Big(\big(\theta^{-1}_{[\![\Gamma]\!]^{\#a}} \circ \widehat{\xi_{[\![\Gamma]\!]}}\big)\langle\gamma\rangle_\Gamma\Big) \mathbin{@} b = (bc) \cdot \big([c].\{x \mid [c].x \in \widehat{\xi_{[\![\Gamma]\!]}}\langle\gamma\rangle_\Gamma\}\big) \tag{5.2.6.4}$$

and hence $x \in \Big(\big(\theta^{-1}_{[\![\Gamma]\!]^{\#a}} \circ \widehat{\xi_{[\![\Gamma]\!]}}\big)\langle\gamma\rangle_\Gamma\Big) \mathbin{@} b$ as required. Conversely, let

$$x \in \Big(\big(\theta^{-1}_{[\![\Gamma]\!]^{\#a}} \circ \widehat{\xi_{[\![\Gamma]\!]}}\big)\langle\gamma\rangle_\Gamma\Big) \mathbin{@} b \tag{5.2.6.5}$$

and let c be a fresh name, then $(bc) \cdot x \in \{x \mid [c].x \in \widehat{\xi_{[\![\Gamma]\!]}}\langle\gamma\rangle_\Gamma\}$ so that

$$[c].((bc) \cdot x) = (bc) \cdot ([b].x) = [b].x \in \widehat{\xi_{[\![\Gamma]\!]}}\langle\gamma\rangle_\Gamma. \tag{5.2.6.6}$$

Therefore there exists $x' \in \langle\gamma\rangle_\Gamma$ such that $[b].x \leq_{\delta_a([\![\Gamma]\!]^{\#a})} \mathbf{fresh}\, d\, \mathbf{in}\, [d].x'$ and hence, for any fresh name d, $x \leq_{[\![\Gamma]\!]^{\#b}} (bd) \cdot x'$. Also $b \# (bd) \cdot x'$ so that $(bd) \cdot x' \in \langle\gamma\rangle_{\Gamma^{\#b}}$ and hence $x \in \langle\gamma\rangle_{\Gamma^{\#b}}$ as required. Thus as claimed,

$$\Big(\big(\theta^{-1}_{[\![\Gamma]\!]^{\#a}} \circ \widehat{\xi_{[\![\Gamma]\!]}}\big)\langle\gamma\rangle_\Gamma\Big) \mathbin{@} b = \langle\gamma\rangle_{\Gamma^{\#b}}. \tag{5.2.6.7}$$

Now $\mathtt{new}\, b.\, p \in \llbracket \Gamma \vdash_s \mathtt{new}\, a\,.\, t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma$ if and only if

$$\mathtt{new}\, b.\, p \in \llbracket \Gamma \vdash_s \mathtt{new}\, a\,.\, t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma = \big(\theta_{\mathbb{P}} \circ \delta_a \llbracket \Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t : \mathbb{P} \rrbracket \circ \theta^{-1}_{\llbracket\Gamma\rrbracket^{\#a}} \circ \widehat{\xi_{\llbracket\Gamma\rrbracket}}\big)\langle\gamma\rangle_\Gamma \tag{5.2.6.8}$$

and since b is fresh by the definition of θ it follows that this is equivalent to

$$\begin{aligned} p \quad &\in \quad \Big(\big(\delta_a \llbracket \Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t : \mathbb{P} \rrbracket \circ \theta^{-1}_{\llbracket\Gamma\rrbracket^{\#a}} \circ \widehat{\xi_{\llbracket\Gamma\rrbracket}}\big)\langle\gamma\rangle_\Gamma\Big) @\, b \qquad (5.2.6.9) \\ &= \quad \big((ab) \cdot \llbracket \Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t : \mathbb{P} \rrbracket\big) \Big(\Big(\big(\theta^{-1}_{\llbracket\Gamma\rrbracket^{\#a}} \circ \widehat{\xi_{\llbracket\Gamma\rrbracket}}\big)\langle\gamma\rangle_\Gamma\Big) @\, b\Big) \\ &= \quad \llbracket \Gamma^{\#b} \vdash_{s \dot\cup \{b\}} (ab) \cdot t : \mathbb{P} \rrbracket \Big(\Big(\big(\theta^{-1}_{\llbracket\Gamma\rrbracket^{\#a}} \circ \widehat{\xi_{\llbracket\Gamma\rrbracket}}\big)\langle\gamma\rangle_\Gamma\Big) @\, b\Big) \\ &= \quad \llbracket \Gamma^{\#b} \vdash_{s \dot\cup \{b\}} (ab) \cdot t : \mathbb{P} \rrbracket \langle\gamma\rangle_{\Gamma^{\#b}} \end{aligned}$$

as required. □

The adjunction $(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s \dot\cup \{a\}}$ also gives rise to a denotational semantics for terms of the form $t\,\mathtt{[}a\mathtt{]}$ as follows.

5.2.6.10 Definition (Semantics of Name Application). *If the judgement $\Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\,\mathtt{[}a\mathtt{]} : \mathbb{P}$ is derived from $\Gamma \vdash_s t : \delta\mathbb{P}$ where $a \notin s$ then define $\llbracket \Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\,\mathtt{[}a\mathtt{]} : \mathbb{P} \rrbracket$ to be the transpose of $\llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket$ in the adjunction $(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s \dot\cup \{a\}}$. In other words,*

$$\llbracket \Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\,\mathtt{[}a\mathtt{]} : \mathbb{P} \rrbracket =_{\mathrm{def}} \widehat{\zeta_{\mathbb{P}}} \circ \llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket^{\#a++}$$

The semantics of name abstraction can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.6.11 Lemma (Characterising Name Application). *Suppose that the typing judgement $\Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\,\mathtt{[}a\mathtt{]} : \mathbb{P}$ is derived from $\Gamma \vdash_s t : \delta\mathbb{P}$ where $a \notin s$. Let $\gamma \in \llbracket \Gamma^{\#a} \rrbracket'$ and let $p \in \mathbb{P}$. Then*

$$\mathtt{new}\, a.\, p \in \llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma \text{ iff } p \in \llbracket \Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\,\mathtt{[}a\mathtt{]} : \mathbb{P} \rrbracket \langle\gamma\rangle_{\Gamma^{\#a}}.$$

Proof. Firstly note that

$$\begin{aligned} &\llbracket \Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\,\mathtt{[}a\mathtt{]} : \mathbb{P} \rrbracket \langle\gamma\rangle_{\Gamma^{\#a}} \qquad (5.2.6.12) \\ &\quad = \quad \big(\widehat{\zeta_{\mathbb{P}}} \circ \phi_{\delta_a\mathbb{P}} \circ \llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket^{\#a} \circ \phi^{-1}_{\llbracket\Gamma\rrbracket}\big)\langle\gamma\rangle_{\Gamma^{\#a}} \\ &\quad = \quad \big(\widehat{\zeta_{\mathbb{P}}} \circ \phi_{\delta_a\mathbb{P}} \circ \llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket^{\#a}\big)\langle\gamma\rangle_\Gamma \\ &\quad = \quad \big(\widehat{\zeta_{\mathbb{P}}} \circ \phi_{\delta_a\mathbb{P}}\big)\Big(\llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma\Big) \end{aligned}$$

Notice that $a \,\#\, \mathtt{new}\, a.\, p$ so that if $\mathtt{new}\, a.\, p \in \llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma$ then it follows that $\mathtt{new}\, a.\, p \in \phi_{\delta_a\mathbb{P}}\Big(\llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma\Big)$ and hence

$$\begin{aligned} p \quad &= \quad (\mathtt{new}\, a.\, p) @ a \\ &\in \quad \big(\widehat{\zeta_{\mathbb{P}}} \circ \phi_{\delta_a\mathbb{P}}\big)\Big(\llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma\Big) \qquad (5.2.6.13) \\ &= \quad \llbracket \Gamma^{\#a} \vdash_{s \dot\cup \{a\}} t\,\mathtt{[}a\mathtt{]} : \mathbb{P} \rrbracket \langle\gamma\rangle_{\Gamma^{\#a}} \end{aligned}$$

as required. Conversely suppose that

$$\begin{array}{rcl} p & \in & [\![\Gamma^{\#a} \vdash_{s\dot{\cup}\{a\}} t\texttt{[}a\texttt{]} : \mathbb{P}]\!]\langle\gamma\rangle_{\Gamma^{\#a}} \\ & = & \big(\widehat{\zeta_{\mathbb{P}}} \circ \phi_{\delta_a\mathbb{P}}\big)\Big([\![\Gamma \vdash_s t : \delta\mathbb{P}]\!]\langle\gamma\rangle_\Gamma\Big) \end{array} \qquad (5.2.6.14)$$

then there exists $p' \in \phi_{\delta_a\mathbb{P}}\Big([\![\Gamma \vdash_s t : \delta\mathbb{P}]\!]\langle\gamma\rangle_\Gamma\Big)$ such that $p \leq_{\mathbb{P}} \zeta_{\mathbb{P}}(p') = p'@a$. Therefore $\texttt{new}\, a.\, p \leq_{\delta\mathbb{P}} \texttt{new}\, a.\, (p'@a) = p'$ and $p' \in [\![\Gamma \vdash_s t : \delta\mathbb{P}]\!]\langle\gamma\rangle_\Gamma$ so that $\texttt{new}\, a.\, p \in [\![\Gamma \vdash_s t : \delta\mathbb{P}]\!]\langle\gamma\rangle_\Gamma$ as required. □

The semantics of the action $\texttt{new}\, a.\, p$ is given by the functor δ_a^{++} together with the isomorphism $\theta^!$ as follows.

5.2.6.15 Definition (Semantics of Name-Abstracted Actions). *If the judgement* $\vdash_s \delta\mathbb{P} : \texttt{new}\, a.\, p : \delta\mathbb{P}'$ *is derived from* $\vdash_{s\dot{\cup}\{a\}} \mathbb{P} : p : \mathbb{P}'$ *where* $a \notin s$ *then define*

$$[\![\vdash_s \delta\mathbb{P} : \texttt{new}\, a.\, p : \delta\mathbb{P}']\!] =_{\text{def}} \widehat{\theta^!_{\mathbb{P}'}} \circ \delta_a^{++}[\![\vdash_{s\dot{\cup}\{a\}} \mathbb{P} : p : \mathbb{P}']\!].$$

The denotational semantics of the action $\texttt{new}\, a.\, p$ can be given a concrete, set theoretic, characterisation in terms of paths as follows.

5.2.6.16 Lemma (Characterising Name-Abstracted Actions). *Suppose that* $\vdash_s \delta\mathbb{P} : \texttt{new}\, a.\, p : \delta\mathbb{P}'$ *is derived from* $\vdash_{s\dot{\cup}\{a\}} \mathbb{P} : p : \mathbb{P}'$ *where* $a \notin s$*. Let* $X \in \widehat{\delta\mathbb{P}}$*. Then*

$$\begin{array}{l} [\![\vdash_s \delta\mathbb{P} : \texttt{new}\, a.\, p : \delta\mathbb{P}']\!]X \\ \qquad = \{\theta^!_{\mathbb{P}'}Y \mid \textbf{fresh}\, b\, \textbf{in}\, Y@b \in [\![(ab)\cdot p]\!]\{x \mid \texttt{new}\, b.\, x \in X\}\}_\downarrow \end{array}$$

Proof.

$$\begin{array}{rl} & [\![\vdash_s \delta\mathbb{P} : \texttt{new}\, a.\, p : \delta\mathbb{P}']\!]X \\ = & \big(\widehat{\theta^!_{\mathbb{P}'}} \circ \theta_{!\mathbb{P}'} \circ \delta_a[\![p]\!] \circ \theta_{\mathbb{P}}^{-1}\big)X \\ = & \big(\widehat{\theta^!_{\mathbb{P}'}} \circ \theta_{!\mathbb{P}'} \circ \delta_a[\![p]\!]\big)\big(\textbf{fresh}\, b\, \textbf{in}\, [b].\{x \mid \texttt{new}\, b.\, x \in X\}\big) \\ = & \big(\widehat{\theta^!_{\mathbb{P}'}} \circ \theta_{!\mathbb{P}'}\big)\big(\textbf{fresh}\, b\, \textbf{in}\, [b].[\![(ab)\cdot p]\!]\{x \mid \texttt{new}\, b.\, x \in X\}\big) \\ = & \widehat{\theta^!_{\mathbb{P}'}}\{Y \mid \textbf{fresh}\, b\, \textbf{in}\, Y@b \in [\![(ab)\cdot p]\!]\{x \mid \texttt{new}\, b.\, x \in X\}\} \\ = & \{\theta^!_{\mathbb{P}'}Y \mid \textbf{fresh}\, b\, \textbf{in}\, Y@b \in [\![(ab)\cdot p]\!]\{x \mid \texttt{new}\, b.\, x \in X\}\}_\downarrow \end{array}$$

□

5.2.7 Structural Rules

The denotational semantics associated with the usual structural rules make use of the cartesian structure of each $\mathbf{FMCts}_s$: variables denote the identity map on their underlying types, weakening corresponds to projection, exchange corresponds to a twist map $\langle \pi_2, \pi_1 \rangle$, and contraction corresponds to the diagonal map $\langle \mathbf{1}, \mathbf{1} \rangle$. The new structural rule of fresh-weakening,

$$\frac{\Gamma, \mathtt{x} : \mathbb{Q}^{\#s''} \vdash_s t : \mathbb{P}}{\Gamma, \mathtt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}} \; (s'' \subseteq s' \subseteq s) \tag{5.2.7.1}$$

comes from the map $\tau^{(s' \backslash s'')++}$ defined in 3.4.8.38, which in turn arises from the map $\tau : (-)^{\#a} \to \mathbf{1}$ defined in 3.2.1.21. Notice that the discussion of 6.1.2.8 shows that this is ultimately given by the projection $(-) \otimes \mathbb{A} \to \mathbf{1}$. The other new structural rule of support-weakening,

$$\frac{\Gamma \vdash_{s'} t : \mathbb{P}}{\Gamma \vdash_s t : \mathbb{P}} \; (s' \subseteq s) \tag{5.2.7.2}$$

is simply given by the inclusion $\mathbf{FMCts}_{s'} \hookrightarrow \mathbf{FMCts}_s$.

It is useful to characterise the action of these structural rules in terms of their action on inputs of the form $\langle \gamma \rangle_\Gamma$ where $\gamma \in [\![\Gamma]\!]'$ is a tuple of paths in the appropriate environment Γ. It turns out that these characterisations are all as one might expect. However, the proofs are a little long and tedious, so the results are only stated alongside the corresponding definitions, and the proofs are collected at the end of this section.

5.2.7.3 Definition (Semantics of Variable). *The meaning of a free variable (with no freshness assumptions) is simply the identity map on the underlying type.*

$$[\![\mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_\varnothing \mathtt{x} : \mathbb{P}]\!] =_{\mathrm{def}} \mathbf{1}_\mathbb{P}$$

5.2.7.4 Definition (Semantics of Weakening). *Suppose that the judgement $\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$ is derived from $\Gamma \vdash_s t : \mathbb{P}$ by weakening, then*

$$[\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] =_{\mathrm{def}} [\![\Gamma \vdash_s t : \mathbb{P}]\!] \circ \mathbf{out}_1.$$

5.2.7.5 Lemma (Characterising Weakening). *Suppose that the judgement $\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$ is derived from $\Gamma \vdash_s t : \mathbb{P}$ by weakening. Let $\gamma \in [\![\Gamma]\!]'$ and $q \in \widehat{\mathbb{Q}}$. Then*

$$[\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] \langle \gamma, q \rangle_{\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing}} = [\![\Gamma \vdash_s t : \mathbb{P}]\!] \langle \gamma \rangle_\Gamma.$$

5.2.7.6 Definition (Semantics of Exchange). *Suppose that a typing judgement $\Gamma, \mathtt{x2} : \mathbb{Q}_2{}^{\#s_2}, \mathtt{x1} : \mathbb{Q}_1{}^{\#s_1}, \Lambda \vdash_s t : \mathbb{P}$ is derived from the typing judgement*

$\Gamma, \mathtt{x1} : \mathbb{Q}_1{}^{\#s_1}, \mathtt{x2} : \mathbb{Q}_2{}^{\#s_2}, \Lambda \vdash_s t : \mathbb{P}$ by exchange, let ς be the twist map defined in 3.4.8.11 and let

$$f = [\![\Gamma, \mathtt{x1} : \mathbb{Q}_1{}^{\#s_1}, \mathtt{x2} : \mathbb{Q}_2{}^{\#s_2}, \Lambda \vdash_s t : \mathbb{P}]\!],$$

then define

$$[\![\Gamma, \mathtt{x2} : \mathbb{Q}_2{}^{\#s_2}, \mathtt{x1} : \mathbb{Q}_1{}^{\#s_1}, \Lambda \vdash_s t : \mathbb{P}]\!] =_{\mathrm{def}} f \circ \big(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1}, \mathbb{Q}_2{}^{\#s_2}} \mathbin{\&} \mathbf{1}_\Lambda\big)$$

5.2.7.7 Lemma (Characterising Exchange). *Suppose the typing judgement $\Gamma, \mathtt{x2} : \mathbb{Q}_2{}^{\#s_2}, \mathtt{x1} : \mathbb{Q}_1{}^{\#s_1}, \Lambda \vdash_s t : \mathbb{P}$ is derived by exchange from the judgement $\Gamma, \mathtt{x1} : \mathbb{Q}_1{}^{\#s_1}, \mathtt{x2} : \mathbb{Q}_2{}^{\#s_2}, \Lambda \vdash_s t : \mathbb{P}$. Let $\gamma \in [\![\Gamma]\!]'$, $\lambda \in [\![\Lambda]\!]'$, $q_1 \in \widehat{\mathbb{Q}_1}^{\#s_1}$ and $q_2 \in \widehat{\mathbb{Q}_2}^{\#s_2}$. Then*

$$\begin{aligned} & [\![\Gamma, \mathtt{x2} : \mathbb{Q}_2{}^{\#s_2}, \mathtt{x1} : \mathbb{Q}_1{}^{\#s_1}, \Lambda \vdash_s t : \mathbb{P}]\!]\langle \gamma, q_2, q_1, \lambda\rangle_{\Gamma, \mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}, \mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}, \Lambda} \\ = \; & [\![\Gamma, \mathtt{x1} : \mathbb{Q}_1{}^{\#s_1}, \mathtt{x2} : \mathbb{Q}_2{}^{\#s_2}, \Lambda \vdash_s t : \mathbb{P}]\!]\langle \gamma, q_1, q_2, \lambda\rangle_{\Gamma, \mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}, \mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}, \Lambda}. \end{aligned}$$

5.2.7.8 Definition (Semantics of Contraction). *Suppose that the judgement $\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'} \vdash_s t[\mathtt{x2}/\mathtt{x1}] : \mathbb{P}$ is derived from $\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}$ by contraction, let Δ be the diagonal map defined in 3.4.8.13 and let*

$$f = [\![\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}]\!]$$

then define

$$[\![\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'} \vdash_s t[\mathtt{x2}/\mathtt{x1}] : \mathbb{P}]\!] =_{\mathrm{def}} f \circ \big(\mathbf{1}_\Gamma \mathbin{\&} \Delta_{\mathbb{Q}^{\#s'}}\big)$$

5.2.7.9 Lemma (Characterising Contraction). *Suppose that the typing judgement $\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'} \vdash_s t[\mathtt{x2}/\mathtt{x1}] : \mathbb{P}$ is derived by contraction from the judgement $\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}$. Let $\gamma \in [\![\Gamma]\!]'$ and $q \in \widehat{\mathbb{Q}}^{\#s'}$. Then*

$$\begin{aligned} & [\![\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'} \vdash_s t[\mathtt{x2}/\mathtt{x1}] : \mathbb{P}]\!]\langle \gamma, q\rangle_{\Gamma, \mathtt{x1}:\mathbb{Q}^{\#s'}} \\ & = [\![\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}]\!]\langle \gamma, q, q\rangle_{\Gamma, \mathtt{x1}:\mathbb{Q}^{\#s'}, \mathtt{x2}:\mathbb{Q}^{\#s'}} \end{aligned}$$

5.2.7.10 Definition (Semantics of Fresh-Weakening). *Let $s'' \subseteq s' \subseteq s$ and suppose that $\Gamma, \mathtt{x} : \mathbb{Q}^{\#s''} \vdash_s t : \mathbb{P}$ is derived from $\Gamma, \mathtt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}$ by fresh-weakening. Let $\tau^{(s' \backslash s'')++} : (-)^{\#(s' \backslash s'')++} \to \mathbf{1}_{\mathbf{FMCts}_s}$ be as defined in 3.4.8.38, then define*

$$[\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}]\!] = [\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#s''} \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_\Gamma \mathbin{\&} \tau^{(s' \backslash s'')++}_{\mathbb{Q}^{\#s''}}).$$

5.2.7.11 Lemma (Characterising Fresh-Weakening). *Suppose that the judgement $\Gamma, \mathtt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}$ is derived from $\Gamma, \mathtt{x} : \mathbb{Q}^{\#s''} \vdash_s t : \mathbb{P}$ by fresh-weakening, where $s'' \subseteq s' \subseteq s$. Let $\gamma \in [\![\Gamma]\!]'$ and $q \in \widehat{\mathbb{Q}}^{\#s'}$. Then*

$$[\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}]\!]\langle \gamma, q\rangle_{\Gamma, \mathtt{x}:\mathbb{Q}^{\#s'}} = [\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#s''} \vdash_s t : \mathbb{P}]\!]\langle \gamma, q\rangle_{\Gamma, \mathtt{x}:\mathbb{Q}^{\#s''}}.$$

5.2.7.12 Definition (Support-Weakening for Terms). *If $s' \subseteq s$ then let $J : \mathbf{FMCts}_{s'} \hookrightarrow \mathbf{FMCts}_s$ be the inclusion of categories and define*

$$[\![\Gamma \vdash_s t : \mathbb{P}]\!] =_{\mathrm{def}} J[\![\Gamma \vdash_{s'} t : \mathbb{P}]\!].$$

5.2.7.13 Lemma (Characterising Support-Weakening for Terms). *Suppose that $\Gamma \vdash_s t : \mathbb{P}$ is derived from $\Gamma \vdash_{s'} t : \mathbb{P}$ by support-weakening, where $s' \subseteq s$. Let $\gamma \in [\![\Gamma]\!]'$. Then*

$$[\![\Gamma \vdash_s t : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma = [\![\Gamma \vdash_{s'} t : \mathbb{P}]\!]\langle\gamma\rangle_\Gamma.$$

Proof. Immediate. □

5.2.7.14 Definition (Support-Weakening for Actions). *If $s' \subseteq s$ then let $J : \mathbf{FMCts}_{s'} \hookrightarrow \mathbf{FMCts}_s$ be the inclusion of categories and define*

$$[\![\, \vdash_s \mathbb{P} : p : \mathbb{P}']\!] =_{\mathrm{def}} J[\![\, \vdash_{s'} \mathbb{P} : p : \mathbb{P}']\!].$$

5.2.7.15 Lemma (Characterising Support-Weakening for Actions). *If $s' \subseteq s$ and the judgement $\vdash_s \mathbb{P} : p : \mathbb{P}'$ is derived from $\vdash_{s'} \mathbb{P} : p : \mathbb{P}'$ by support-weakening, and furthermore $X \in \widehat{\mathbb{P}}$, then*

$$[\![\, \vdash_s \mathbb{P} : p : \mathbb{P}']\!]X = [\![\, \vdash_{s'} \mathbb{P} : p : \mathbb{P}']\!]X.$$

Proof. Immediate. □

5.2.7.16 Proofs of characterisation lemmas

Proof of 5.2.7.5 (Characterising Weakening).

$$\begin{aligned}
&[\![\Gamma, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] \circ \langle\cdot\rangle_{\Gamma,\mathtt{x}:\mathbb{Q}^{\#\varnothing}} && (5.2.7.17)\\
&\quad = \quad [\![\Gamma \vdash_s t : \mathbb{P}]\!] \circ \mathbf{out}_1 \circ \langle\cdot\rangle_{\Gamma,\mathtt{x}:\mathbb{Q}^{\#\varnothing}} && \text{by } 5.2.7.4\\
&\quad = \quad [\![\Gamma \vdash_s t : \mathbb{P}]\!] \circ \mathbf{out}_1 \circ m_{[\![\Gamma]\!],\mathbb{P}} \circ (\langle\cdot\rangle_\Gamma \times \mathbf{1}_\mathbb{P}) && \text{by } 5.1.2.3\\
&\quad = \quad [\![\Gamma \vdash_s t : \mathbb{P}]\!] \circ \pi_1 \circ (\langle\cdot\rangle_\Gamma \times \mathbf{1}_\mathbb{P}) && \text{by } 3.4.8.8\\
&\quad = \quad [\![\Gamma \vdash_s t : \mathbb{P}]\!] \circ \langle\cdot\rangle_\Gamma \circ \pi_1.
\end{aligned}$$

□

Proof of 5.2.7.7 (Characterising Exchange). It clearly suffices to show that

$$\begin{aligned}
&\Big(\mathbf{1}_\Gamma \,\&\, \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}} \,\&\, \mathbf{1}_\Lambda\Big) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\Lambda}\\
&\qquad = \langle\cdot\rangle_{\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda} \circ \Big(\mathbf{1}_{[\![\Gamma]\!]'} \times \langle\pi_2,\pi_1\rangle \times \mathbf{1}_{[\![\Lambda]\!]'}\Big).
\end{aligned} \qquad (5.2.7.18)$$

The proof proceeds by induction on the structure of Λ. The interesting case is when $\Lambda = ()$ which is as follows. Firstly, note that

$$\begin{aligned}&\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}}\\ &\qquad = \langle \mathbf{out}_1, \mathbf{out}_3, \mathbf{out}_2\rangle^{\&} : [\![\Gamma]\!] \mathbin{\&} \mathbb{Q}_1^{\#s_1} \mathbin{\&} \mathbb{Q}_2^{\#s_2} \underset{\mathbf{C}}{\to} [\![\Gamma]\!] \mathbin{\&} \mathbb{Q}_2^{\#s_2} \mathbin{\&} \mathbb{Q}_1^{\#s_1}.\end{aligned} \tag{5.2.7.19}$$

Therefore by 5.1.2.3

$$\begin{aligned}&\left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}}\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}}\\ &\quad = \quad \langle \mathbf{out}_1, \mathbf{out}_3, \mathbf{out}_2\rangle^{\&} \circ m_{[\![\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}]\!],\mathbb{Q}_2^{\#s_2}}\\ &\qquad\qquad \circ \left(\langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}} \times \phi_{\mathbb{Q}_2}^{(s_2)}\right)\end{aligned} \tag{5.2.7.20}$$

so that by 3.4.8.8

$$\begin{aligned}&\langle \mathbf{out}_1, \mathbf{out}_3\rangle^{\&} \circ \left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}}\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}}\\ &\quad = \quad \mathbf{out}_{12} \circ m_{[\![\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}]\!],\mathbb{Q}_2^{\#s_2}} \circ \left(\langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}} \times \phi_{\mathbb{Q}_2}^{(s_2)}\right)\\ &\quad = \quad \pi_{12} \circ \left(\langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}} \times \phi_{\mathbb{Q}_2}^{(s_2)}\right)\\ &\quad = \quad \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}} \circ \pi_{12}\end{aligned} \tag{5.2.7.21}$$

and hence

$$\begin{aligned}&\mathbf{out}_1 \circ \left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}}\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}}\\ &\quad = \quad \mathbf{out}_1 \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}} \circ \pi_{12}\\ &\quad = \quad \mathbf{out}_1 \circ m_{[\![\Gamma]\!],\mathbb{Q}_1^{\#s_1}} \circ \left(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}_1}^{(s_1)}\right) \circ \pi_{12}\\ &\quad = \quad \pi_1 \circ \left(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}_1}^{(s_1)}\right) \circ \pi_{12}\\ &\quad = \quad \langle\cdot\rangle_\Gamma \circ \pi_1\end{aligned} \tag{5.2.7.22}$$

and similarly

$$\begin{aligned}&\mathbf{out}_3 \circ \left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}}\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}}\\ &\quad = \quad \phi_{\mathbb{Q}_1}^{(s_1)} \circ \pi_2.\end{aligned} \tag{5.2.7.23}$$

Furthermore,

$$\begin{aligned}&\mathbf{out}_2 \circ \left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}}\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}}\\ &\quad = \quad \mathbf{out}_3 \circ m_{[\![\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}]\!],\mathbb{Q}_2^{\#s_2}} \circ \left(\langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}} \times \phi_{\mathbb{Q}_2}^{(s_2)}\right)\\ &\quad = \quad \pi_3 \circ \left(\langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}} \times \phi_{\mathbb{Q}_2}^{(s_2)}\right)\\ &\quad = \quad \phi_{\mathbb{Q}_2}^{(s_2)} \circ \pi_3.\end{aligned} \tag{5.2.7.24}$$

Therefore by 3.4.8.5 and 3.4.8.9

$$\begin{aligned}
&\left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}}\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}} \\
&\quad = \quad \langle\langle\cdot\rangle_\Gamma \circ \pi_1, \phi^{(s_2)}_{\mathbb{Q}_2} \circ \pi_3, \phi^{(s_1)}_{\mathbb{Q}_1} \circ \pi_2\rangle^{\&} \\
&\quad = \quad m_{\llbracket\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}\rrbracket,\mathbb{Q}_1^{\#s_1}} \circ \left(m_{\llbracket\Gamma\rrbracket,\mathbb{Q}_2^{\#s_2}} \times \mathbf{1}_{\mathbb{Q}_1^{\#s_1}}\right) \\
&\qquad\qquad \circ \langle\langle\cdot\rangle_\Gamma \circ \pi_1, \phi^{(s_2)}_{\mathbb{Q}_2} \circ \pi_3, \phi^{(s_1)}_{\mathbb{Q}_1} \circ \pi_2\rangle \\
&\quad = \quad m_{\llbracket\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}\rrbracket,\mathbb{Q}_1^{\#s_1}} \circ \left(\left(m_{\llbracket\Gamma\rrbracket,\mathbb{Q}_2^{\#s_2}} \circ \left(\langle\cdot\rangle_\Gamma \times \phi^{(s_2)}_{\mathbb{Q}_2}\right)\right) \times \phi^{(s_1)}_{\mathbb{Q}_1}\right) \\
&\qquad\qquad \circ \left(\mathbf{1}_{\llbracket\Gamma\rrbracket'} \times \langle\pi_2,\pi_1\rangle\right) \\
&\quad = \quad m_{\llbracket\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}\rrbracket,\mathbb{Q}_1^{\#s_1}} \circ \left(\langle\cdot\rangle_{\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2}} \times \phi^{(s_1)}_{\mathbb{Q}_1}\right) \\
&\qquad\qquad \circ \left(\mathbf{1}_{\llbracket\Gamma\rrbracket'} \times \langle\pi_2,\pi_1\rangle\right) \\
&\quad = \quad \langle\cdot\rangle_{\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1}} \circ \left(\mathbf{1}_{\llbracket\Gamma\rrbracket'} \times \langle\pi_2,\pi_1\rangle\right)
\end{aligned} \tag{5.2.7.25}$$

as required.

Now for the inductive case assume that

$$\begin{aligned}
&\left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}} \mathbin{\&} \mathbf{1}_\Lambda\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\Lambda} \\
&\qquad = \langle\cdot\rangle_{\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda} \circ \left(\mathbf{1}_{\llbracket\Gamma\rrbracket'} \times \langle\pi_2,\pi_1\rangle \times \mathbf{1}_{\llbracket\Lambda\rrbracket'}\right).
\end{aligned} \tag{5.2.7.26}$$

then

$$\begin{aligned}
&\left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}} \mathbin{\&} \mathbf{1}_{\Lambda,\mathtt{y}:\mathbb{R}^{\#r}}\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\Lambda,\mathtt{y}:\mathbb{R}^{\#r}} \\
&\quad = \quad \left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}} \mathbin{\&} \mathbf{1}_\Lambda \mathbin{\&} \mathbf{1}_{\widehat{\mathbb{R}^{\#r}}}\right) \\
&\qquad\quad \circ m_{\llbracket\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\Lambda\rrbracket,\mathbb{R}^{\#r}} \circ (\langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\Lambda} \times \phi^{(r)}_{\mathbb{R}}) \\
&\quad = \quad m_{\llbracket\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda\rrbracket,\mathbb{R}^{\#r}} \\
&\qquad\quad \circ \left(\left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}} \mathbin{\&} \mathbf{1}_\Lambda\right) \times \mathbf{1}_{\widehat{\mathbb{R}^{\#r}}}\right) \\
&\qquad\quad \circ (\langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\Lambda} \times \phi^{(r)}_{\mathbb{R}}) \\
&\quad = \quad m_{\llbracket\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda\rrbracket,\mathbb{R}^{\#r}} \\
&\qquad\quad \circ \left(\left(\left(\mathbf{1}_\Gamma \mathbin{\&} \varsigma_{\mathbb{Q}_1{}^{\#s_1},\mathbb{Q}_2{}^{\#s_2}} \mathbin{\&} \mathbf{1}_\Lambda\right) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\Lambda}\right) \times \phi^{(r)}_{\mathbb{R}}\right) \\
&\quad = \quad m_{\llbracket\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda\rrbracket,\mathbb{R}^{\#r}} \\
&\qquad\quad \circ \left(\left(\langle\cdot\rangle_{\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda} \circ \left(\mathbf{1}_{\llbracket\Gamma\rrbracket'} \times \langle\pi_2,\pi_1\rangle \times \mathbf{1}_{\llbracket\Lambda\rrbracket'}\right)\right) \times \phi^{(r)}_{\mathbb{R}}\right) \\
&\quad = \quad m_{\llbracket\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda\rrbracket,\mathbb{R}^{\#r}} \circ \left(\langle\cdot\rangle_{\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda} \times \phi^{(r)}_{\mathbb{R}}\right) \\
&\qquad\quad \circ \left(\mathbf{1}_{\llbracket\Gamma\rrbracket'} \times \langle\pi_2,\pi_1\rangle \times \mathbf{1}_{\llbracket\Lambda\rrbracket'} \times \mathbf{1}_{\widehat{\mathbb{R}}^{\#r}}\right) \\
&\quad = \quad \langle\cdot\rangle_{\Gamma,\mathtt{x2}:\mathbb{Q}_2{}^{\#s_2},\mathtt{x1}:\mathbb{Q}_1{}^{\#s_1},\Lambda,\mathtt{y}:\mathbb{R}^{\#r}} \circ \left(\mathbf{1}_{\llbracket\Gamma\rrbracket'} \times \langle\pi_2,\pi_1\rangle \times \mathbf{1}_{\llbracket\Lambda,\mathtt{y}:\mathbb{R}^{\#r}\rrbracket'}\right)
\end{aligned} \tag{5.2.7.27}$$

as required. □

Proof of 5.2.7.9 (Characterising Contraction). Firstly note that by 3.4.8.9 and

3.4.8.13 it is the case that $\Delta_{\mathbb{Q}\#s'} = m_{\mathbb{Q}\#s',\mathbb{Q}\#s'} \circ \langle \mathbf{1}_{\mathbb{Q}\#s'}, \mathbf{1}_{\mathbb{Q}\#s'} \rangle$. Therefore

$$\begin{aligned}
&\big(\mathbf{1}_\Gamma \mathbin{\&} \Delta_{\mathbb{Q}\#s'}\big) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}\#s'} \\
&\quad= \big(\mathbf{1}_\Gamma \mathbin{\&} \Delta_{\mathbb{Q}\#s'}\big) \circ m_{[\![\Gamma]\!],\mathbb{Q}\#s'} \circ \Big(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}}^{(s')}\Big) \\
&\quad= m_{[\![\Gamma]\!],\mathbb{Q}\#s' \mathbin{\&} \mathbb{Q}\#s'} \circ \big(\mathbf{1}_\Gamma \times \Delta_{\mathbb{Q}\#s'}\big) \circ \Big(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}}^{(s')}\Big) \\
&\quad= m_{[\![\Gamma]\!],\mathbb{Q}\#s' \mathbin{\&} \mathbb{Q}\#s'} \circ \big(\mathbf{1}_\Gamma \times m_{\mathbb{Q}\#s',\mathbb{Q}\#s'}\big) \\
&\qquad\quad \circ \big(\mathbf{1}_\Gamma \times \langle \mathbf{1}_{\widehat{\mathbb{Q}\#s'}}, \mathbf{1}_{\widehat{\mathbb{Q}\#s'}} \rangle\big) \\
&\qquad\quad \circ \Big(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}}^{(s')}\Big) \\
&\quad= m_{[\![\Gamma,\mathtt{x1}:\mathbb{Q}\#s']\!],\mathbb{Q}\#s'} \circ \big(m_{[\![\Gamma]\!],\mathbb{Q}\#s'} \times \mathbf{1}_{\widehat{\mathbb{Q}\#s'}}\big) \\
&\qquad\quad \circ \big(\mathbf{1}_\Gamma \times \langle \mathbf{1}_{\widehat{\mathbb{Q}\#s'}}, \mathbf{1}_{\widehat{\mathbb{Q}\#s'}} \rangle\big) \circ \Big(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}}^{(s')}\Big) \\
&\quad= m_{[\![\Gamma,\mathtt{x1}:\mathbb{Q}\#s']\!],\mathbb{Q}\#s'} \circ \big(m_{[\![\Gamma]\!],\mathbb{Q}\#s'} \times \mathbf{1}_{\widehat{\mathbb{Q}\#s'}}\big) \\
&\qquad\quad \circ \Big(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}}^{(s')} \times \phi_{\mathbb{Q}}^{(s')}\Big) \\
&\qquad\quad \circ \big(\mathbf{1}_{[\![\Gamma]\!]'} \times \langle \mathbf{1}_{\widehat{\mathbb{Q}}\#s'}, \mathbf{1}_{\widehat{\mathbb{Q}}\#s'} \rangle\big) \\
&\quad= m_{[\![\Gamma,\mathtt{x1}:\mathbb{Q}\#s']\!],\mathbb{Q}\#s'} \circ \Big(\langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}\#s'} \times \phi_{\mathbb{Q}}^{(s')}\Big) \\
&\qquad\quad \circ \big(\mathbf{1}_{[\![\Gamma]\!]'} \times \langle \mathbf{1}_{\widehat{\mathbb{Q}}\#s'}, \mathbf{1}_{\widehat{\mathbb{Q}}\#s'} \rangle\big) \\
&\quad= \langle\cdot\rangle_{\Gamma,\mathtt{x1}:\mathbb{Q}\#s',\mathtt{x2}:\mathbb{Q}\#s'} \\
&\qquad\quad \circ \big(\mathbf{1}_{[\![\Gamma]\!]'} \times \langle \mathbf{1}_{\widehat{\mathbb{Q}}\#s'}, \mathbf{1}_{\widehat{\mathbb{Q}}\#s'} \rangle\big)
\end{aligned} \tag{5.2.7.28}$$

as required. □

Proof of 5.2.7.11 (Characterising Fresh-Weakening). Firstly,

$$\begin{aligned}
&\tau_{\mathbb{Q}\#s''}^{(s'\backslash s'')++} \circ \phi_{\mathbb{Q}}^{(s')} \\
&\quad= \tau_{\widehat{\mathbb{Q}\#s''}}^{(s'\backslash s'')} \circ {\phi_{\mathbb{Q}\#s''}^{(s'\backslash s'')}}^{-1} \circ \phi_{\mathbb{Q}}^{(s')} \\
&\quad= \tau_{\widehat{\mathbb{Q}\#s''}}^{(s'\backslash s'')} \circ {\phi_{\mathbb{Q}\#s''}^{(s'\backslash s'')}}^{-1} \circ \phi_{\mathbb{Q}\#s''}^{(s'\backslash s'')} \circ {\phi_{\mathbb{Q}}^{(s'')}}^{\#(s'\backslash s'')} \\
&\quad= \tau_{\widehat{\mathbb{Q}\#s''}}^{(s'\backslash s'')} \circ {\phi_{\mathbb{Q}}^{(s'')}}^{\#(s'\backslash s'')} \\
&\quad= \phi_{\mathbb{Q}}^{(s'')} \circ \tau_{\widehat{\mathbb{Q}}\#s''}^{(s'\backslash s'')}
\end{aligned} \tag{5.2.7.29}$$

so that

$$\begin{aligned}
&\big(\mathbf{1}_\Gamma \mathbin{\&} \tau_{\mathbb{Q}\#s''}^{(s'\backslash s'')++}\big) \circ \langle\cdot\rangle_{\Gamma,\mathtt{x}:\mathbb{Q}\#s'} \\
&\quad= \big(\mathbf{1}_\Gamma \mathbin{\&} \tau_{\mathbb{Q}\#s''}^{(s'\backslash s'')++}\big) \circ m_{[\![\Gamma]\!],\mathbb{Q}\#s'} \circ \big(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}}^{(s')}\big) \\
&\quad= m_{[\![\Gamma]\!],\mathbb{Q}\#s''} \circ \big(\mathbf{1}_\Gamma \times \tau_{\mathbb{Q}\#s''}^{(s'\backslash s'')++}\big) \circ \big(\langle\cdot\rangle_\Gamma \times \phi_{\mathbb{Q}}^{(s')}\big) \\
&\quad= m_{[\![\Gamma]\!],\mathbb{Q}\#s''} \circ \big(\langle\cdot\rangle_\Gamma \times \big(\tau_{\mathbb{Q}\#s''}^{(s'\backslash s'')++} \circ \phi_{\mathbb{Q}}^{(s')}\big)\big) \\
&\quad= m_{[\![\Gamma]\!],\mathbb{Q}\#s''} \circ \big(\langle\cdot\rangle_\Gamma \times \big(\phi_{\mathbb{Q}}^{(s'')} \circ \tau_{\widehat{\mathbb{Q}}\#s''}^{(s'\backslash s'')}\big)\big) \\
&\quad= \langle\cdot\rangle_{\Gamma,\mathtt{x}:\mathbb{Q}\#s''} \circ \big(\mathbf{1}_\Gamma \times \tau_{\widehat{\mathbb{Q}}\#s''}^{(s'\backslash s'')}\big)
\end{aligned} \tag{5.2.7.30}$$

as required. □

5.3 Substitution as Composition

Substitution effectively amounts to composition of denotations, as the following lemma shows. However, care must be taken to ensure that the denotations lie in the same $\mathbf{FMCts}_s$ and that all the relevant freshness assumptions are satisfied.

5.3.0.1 Lemma (Semantic Substitution Lemma). *Suppose that the judgement* $\Gamma, \mathtt{y} : \mathbb{R}^{\#r} \vdash_s t : \mathbb{P}$ *and* $\Delta \vdash_{s_1} v : \mathbb{R}$ *where* $s_1 \cap r = \varnothing$ *and the variables in* Γ *are distinct from those in* Δ*. Let* $i_1 : s_1 \dot{\cup} r \hookrightarrow s \cup s_1$ *and* $i_2 : s \hookrightarrow s \cup s_1$*. Then*

$$[\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v/\mathtt{y}] : \mathbb{P}]\!] = i_2^* [\![\Gamma, \mathtt{y} : \mathbb{R}^{\#r} \vdash_s t : \mathbb{P}]\!] \circ \left(\mathbf{1}_\Gamma \,\&\, i_1^* [\![\Delta \vdash_{s_1} v : \mathbb{R}]\!]^{\#r++}\right)$$

Proof. The proof is by induction over the typing rules. As with the proof of lemma 4.3.0.1, in the presence of exchange and contraction the induction hypothesis must be strengthened as follows.

Suppose that $\Gamma' \vdash_s t : \mathbb{P}$, $\Delta \vdash_{s_1} v : \mathbb{R}$, and Γ' is a reordering of the environment $\Gamma, \mathtt{y1} : \mathbb{R}^{\#r_1}, \ldots, \mathtt{yn} : \mathbb{R}^{\#r_n}$, and represent this reordering by the isomorphism $\sigma : \Gamma \,\&\, \mathbb{R}^{\#r_1} \,\&\, \ldots \,\&\, \mathbb{R}^{\#r_n} \underset{\mathbf{C}}{\to} \Gamma'$. Let $r = r_1 \cup \ldots \cup r_n$. Suppose also that the variables in Γ are distinct from those in Δ and that $s_1 \cap r = \varnothing$. Define

$$[\![v]\!]^n = (j_1^* \,\&\, \ldots \,\&\, j_n^*) \circ \Delta^n_{\mathbb{R}^{\#r}} \circ [\![\Delta \vdash_{s_1} v : \mathbb{R}]\!]^{\#r++} : \Delta^{\#r} \underset{\mathbf{C}}{\to} \mathbb{R}^{\#r_1} \,\&\, \ldots \,\&\, \mathbb{R}^{\#r_n}$$

where $j_i : r_i \hookrightarrow r$ for each i. The clash of Δs in the notation is unfortunate: where a Δ has a subscript this means the diagonal map on that object, otherwise it means the environment in $\Delta \vdash_{s_1} v : \mathbb{R}$. Ultimately, conclude that

$$[\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P}]\!] = i_2^* [\![\Gamma' \vdash_s t : \mathbb{P}]\!] \circ \sigma \circ (\mathbf{1}_\Gamma \,\&\, i_1^* [\![v]\!]^n)$$

where $t[v]$ is the term obtained by simultaneously substituting v for the variables $\mathtt{y1}, \ldots, \mathtt{yn}$ in t.

The proof now proceeds by induction on the derivation of $\Gamma' \vdash_s t : \mathbb{P}$. Throughout, variables beginning with `y` and `z` such as $\mathtt{y1}, \mathtt{ym}$ and `zn` represent those that are subject to substitution, whereas others such as `x` and `x1` stand for those that are not. Where it is unimportant, omit the map σ and the functors i_1^* and i_2^* to preserve the clarity of the proof. For a similar reason, the various maps $\tau_{\mathbb{P}'}^{(s')++}$ that appear in the proof will be written $\tau_{\mathbb{P}'}$ or τ where the decorations are clear from the context.

Variable There are two possibilities for the variable rule, depending on whether the variable in question is subject to substitution or not. If it is subject to substitution then $t = \mathtt{y}$, $n = 1$ and $s = r = \varnothing$ so i_1^* is the identity functor and

$$\begin{array}{rcl}[\![\Delta \vdash_{s_1} t[v/\mathtt{y}] : \mathbb{R}]\!] & = & [\![\Delta \vdash_{s_1} v : \mathbb{R}]\!] \\ & = & [\![\mathtt{x} : \mathbb{R}^{\#\varnothing} \vdash_{\varnothing} \mathtt{x} : \mathbb{R}]\!] \circ (\mathbf{1}_{()} \,\&[\![\Delta \vdash_{s_1} v : \mathbb{R}]\!]) \\ & = & [\![\mathtt{x} : \mathbb{R}^{\#\varnothing} \vdash_{\varnothing} \mathtt{x} : \mathbb{R}]\!] \circ (\mathbf{1}_{()} \,\&[\![v]\!]^n)\end{array}$$

as required. On the other hand if the variable in question is not subject to substitution then $t = \mathtt{x}$ and $n = 0$ so that

$$\begin{array}{rcl}[\![\mathtt{x} : \mathbb{P}^{\#\varnothing}, \Delta \vdash_{s_1} t[v/\mathtt{y}] : \mathbb{P}]\!] & = & [\![\mathtt{x} : \mathbb{P}^{\#\varnothing}, \Delta \vdash_{s_1} \mathtt{x} : \mathbb{P}]\!] \\ & = & [\![\mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_{\varnothing} \mathtt{x} : \mathbb{P}]\!] \circ \mathbf{out}_1 \\ & = & [\![\mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_{\varnothing} \mathtt{x} : \mathbb{P}]\!] \circ (\mathbf{1}_{\mathbb{P}} \,\&[\![v]\!]^n)\end{array}$$

as required.

Weakening There are two possibilities for the weakening rule, depending on whether the newly-added variable is subject to substitution or not. Suppose that $\Gamma', \mathtt{yn} : \mathbb{R}^{\#\varnothing} \vdash_s t : \mathbb{P}$ is derived from $\Gamma' \vdash_s t : \mathbb{P}$, then $\mathtt{yn}$ does not appear free in t and $n > 0$ and hence

$$\begin{array}{rll}\multicolumn{3}{l}{[\![\Gamma, \Delta^{\#r_1 \cup \ldots \cup r_{n-1} \cup \varnothing} \vdash_{s \cup s_1} t[v] : \mathbb{P}]\!]} \\ = & [\![\Gamma' \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_\Gamma \,\&[\![v]\!]^{n-1}) & \text{by induction} \\ = & [\![\Gamma' \vdash_s t : \mathbb{P}]\!] \circ \mathbf{out} \circ (\mathbf{1}_\Gamma \,\&[\![v]\!]^n) & \\ = & [\![\Gamma', \mathtt{yn} : \mathbb{R}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_\Gamma \,\&[\![v]\!]^n) & \end{array}$$

as required.

On the other hand, suppose that $\Gamma', \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$ is derived from the judgement $\Gamma' \vdash_s t : \mathbb{P}$, then

$$\begin{array}{rll}\multicolumn{3}{l}{[\![\Gamma, \Delta^{\#r}, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_{s \cup s_1} t[v] : \mathbb{P}]\!]} \\ = & [\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P}]\!] \circ \mathbf{out} & \\ = & [\![\Gamma' \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_\Gamma \,\&[\![v]\!]^n) \circ \mathbf{out} & \text{by induction} \\ = & [\![\Gamma' \vdash_s t : \mathbb{P}]\!] \circ \mathbf{out} \circ (\mathbf{1}_{\Gamma \,\&\, \mathbb{Q}} \,\&[\![v]\!]^n) & \\ = & [\![\Gamma', \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_{\Gamma \,\&\, \mathbb{Q}} \,\&[\![v]\!]^n) & \end{array}$$

as required.

Exchange This case is trivial, by reordering the environments appropriately.

Contraction There are two possibilities for this case, depending on whether the contracted variable is subject to substitution or not. Suppose that the judgement $\Gamma', \mathtt{yn} : \mathbb{R}^{\#r_n} \vdash_s t[\mathtt{yn}/\mathtt{ym}] : \mathbb{P}$ is derived from a judgement of the form

$\Gamma', \mathtt{yn} : \mathbb{R}^{\#r_n}, \mathtt{ym} : \mathbb{R}^{\#r_n} \vdash_s t : \mathbb{P}$, then

$$\begin{aligned}
&[\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[\mathtt{yn}/\mathtt{ym}][v] : \mathbb{P}]\!] \\
&\quad = \quad [\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P}]\!] \\
&\quad = \quad [\![\Gamma', \mathtt{yn} : \mathbb{R}^{\#r_n}, \mathtt{ym} : \mathbb{R}^{\#r_n} \vdash_s t : \mathbb{P}]\!] \circ \big(\mathbf{1}_\Gamma \,\&\, [\![v]\!]^{n+1}\big) \quad \text{by induction} \\
&\quad = \quad [\![\Gamma', \mathtt{yn} : \mathbb{R}^{\#r_n}, \mathtt{ym} : \mathbb{R}^{\#r_n} \vdash_s t : \mathbb{P}]\!] \circ \big(\mathbf{1}_{\Gamma'} \,\&\, \Delta_{\mathbb{R}^{\#r}}\big) \circ \big(\mathbf{1}_\Gamma \,\&\, [\![v]\!]^{n}\big) \\
&\quad = \quad [\![\Gamma', \mathtt{yn} : \mathbb{R}^{\#r_n} \vdash_s t[\mathtt{yn}/\mathtt{ym}] : \mathbb{P}]\!] \circ \big(\mathbf{1}_\Gamma \,\&\, [\![v]\!]^{n}\big)
\end{aligned}$$

as required.

On the other hand suppose that the judgement $\Gamma', \mathtt{x1} : \mathbb{Q}^{\#s'} \vdash_s t[\mathtt{x1}/\mathtt{x2}] : \mathbb{P}$ is derived from the judgement $\Gamma', \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}$, then

$$\begin{aligned}
&[\![\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'}, \Delta^{\#r} \vdash_{s \cup s_1} t[\mathtt{x1}/\mathtt{x2}][v] : \mathbb{P}]\!] \\
&\quad = \quad [\![\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'}, \Delta^{\#r} \vdash_{s \cup s_1} t[v][\mathtt{x1}/\mathtt{x2}] : \mathbb{P}]\!] \\
&\quad = \quad [\![\Gamma, \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'}, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P}]\!] \circ (\mathbf{1}_{\Gamma \,\&\, \Delta^{\#r}} \,\&\, \Delta_{\mathbb{Q}^{\#s'}}) \\
&\quad = \quad [\![\Gamma', \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_{\Gamma \,\&\, \mathbb{Q}^{\#s'} \,\&\, \mathbb{Q}^{\#s'}} \,\&\, [\![v]\!]^n) \\
&\qquad\qquad \circ (\mathbf{1}_{\Gamma \,\&\, \Delta^{\#r}} \,\&\, \Delta_{\mathbb{Q}^{\#s'}}) \quad \text{by induction} \\
&\quad = \quad [\![\Gamma', \mathtt{x1} : \mathbb{Q}^{\#s'}, \mathtt{x2} : \mathbb{Q}^{\#s'} \vdash_s t : \mathbb{P}]\!] \circ (\mathbf{1}_{\Gamma'} \,\&\, \Delta_{\mathbb{Q}^{\#s'}}) \circ (\mathbf{1}_{\Gamma \,\&\, \mathbb{Q}^{\#s'}} \,\&\, [\![v]\!]^n) \\
&\quad = \quad [\![\Gamma', \mathtt{x1} : \mathbb{Q}^{\#s'} \vdash_s t[\mathtt{x1}/\mathtt{x2}] : \mathbb{P}]\!] \circ (\mathbf{1}_{\Gamma \,\&\, \mathbb{Q}^{\#s'}} \,\&\, [\![v]\!]^n)
\end{aligned}$$

as required.

Fresh-Weakening There are two possibilities, depending on whether the variable with extra freshness assumptions is subject to substitution or not. If it is not subject to substitution then this case is a simple application of the inductive hypothesis. On the other hand, suppose that $r'_n \subseteq r_n \subseteq s$, that Γ' is a reordering of the environment $\Gamma, \mathtt{y1} : \mathbb{R}^{\#r_1}, \ldots, \mathtt{yn} : \mathbb{R}^{\#r_n}$ and also that Γ'' is a reordering of the environment $\Gamma, \mathtt{y1} : \mathbb{R}^{\#r_1}, \ldots, \mathtt{yn} : \mathbb{R}^{\#r'_n}$. If $\Gamma' \vdash_s t : \mathbb{P}$ is derived from $\Gamma'' \vdash_s t : \mathbb{P}$ then

$$\begin{aligned}
&[\![\Gamma, \Delta^{\#r_1 \cup \ldots \cup r_{n-1} \cup r_n} \vdash_{s \cup s_1} t[v] : \mathbb{P}]\!] \\
&\quad = \quad [\![\Gamma, \Delta^{\#r_1 \cup \ldots \cup r_{n-1} \cup r'_n} \vdash_{s \cup s_1} t[v] : \mathbb{P}]\!] \circ \big(\mathbf{1}_\Gamma \,\&\, \tau_\Delta\big) \\
&\quad = \quad [\![\Gamma'' \vdash_s t : \mathbb{P}]\!] \circ \big(\mathbf{1}_\Gamma \,\&\, [\![v]\!]^n\big) \circ \big(\mathbf{1}_\Gamma \,\&\, \tau_\Delta\big) \quad \text{by induction} \\
&\quad = \quad [\![\Gamma'' \vdash_s t : \mathbb{P}]\!] \circ \big(\mathbf{1}_\Gamma \,\&\, \tau_{\mathbb{R}}^{(1)} \,\&\, \ldots \,\&\, \tau_{\mathbb{R}}^{(n)}\big) \circ \big(\mathbf{1}_\Gamma \,\&\, [\![v]\!]^n\big) \\
&\quad = \quad [\![\Gamma' \vdash_s t : \mathbb{P}]\!] \circ \big(\mathbf{1}_\Gamma \,\&\, [\![v]\!]^n\big)
\end{aligned}$$

as required.

Support-Weakening Suppose that $s' \subseteq s$ and that $\Gamma' \vdash_s t : \mathbb{P}$ is derived from $\Gamma' \vdash_{s'} t : \mathbb{P}$, then

$$\begin{aligned}
&\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P} \\
&\quad = \quad [\![\Gamma, \Delta^{\#r} \vdash_{s' \cup s_1} t[v] : \mathbb{P}]\!] \\
&\quad = \quad [\![\Gamma' \vdash_{s'} t : \mathbb{P}]\!] \circ \big(\mathbf{1}_\Gamma \,\&\, [\![v]\!]^n\big) \quad \text{by induction} \\
&\quad = \quad [\![\Gamma' \vdash_s t : \mathbb{P}]\!] \circ \big(\mathbf{1}_\Gamma \,\&\, [\![v]\!]^n\big)
\end{aligned}$$

as required.

Prefix Suppose that $\Gamma' \vdash_s \mathtt{!}t : !\mathbb{P}$ is derived from $\Gamma' \vdash_s t : \mathbb{P}$, then

$$\begin{array}{cl}
\multicolumn{2}{l}{\llbracket \Gamma, \Delta^{\#r} \vdash_{s \cup s_1} (\mathtt{!}t)[v] = \mathtt{!}(t[v]) : !\mathbb{P} \rrbracket} \\
= & \eta_{\mathbb{P}} \circ \llbracket \Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P} \rrbracket \\
= & \eta_{\mathbb{P}} \circ \llbracket \Gamma' \vdash_s t : \mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n) \quad \text{by induction} \\
= & \llbracket \Gamma' \vdash_s \mathtt{!}t : !\mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n)
\end{array}$$

as required.

Recursion Suppose that $\Gamma' \vdash_s \mathtt{rec}\,\mathtt{x}.t : \mathbb{P}$ is derived from the judgement $\Gamma', \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}$. Let

$$\begin{array}{rrcl}
 & f & = & \llbracket \Gamma', \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P} \rrbracket \\
\text{and} & f' & = & \llbracket \Gamma, \Delta^{\#r}, \mathtt{x} : \mathbb{P}^{\#\varnothing} \vdash_{s \cup s_1} t[v/\mathtt{y}] : \mathbb{P} \rrbracket,
\end{array}$$

then by the induction hypothesis $f' = f \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n \,\&\, \mathbf{1}_{\mathbb{P}})$. Let $g_0 = g'_0 = \varnothing$ and for each m let $g_{m+1} = f \circ (\mathbf{1}_{\Gamma'} \,\&\, g_m) \circ \Delta_{\Gamma'}$ and $g'_{m+1} = f' \circ (\mathbf{1}_{\Gamma \,\&\, \Delta^{\#r}} \,\&\, g'_m) \circ \Delta_{\Gamma \,\&\, \Delta^{\#r}}$. Now by induction on m it follows that $g'_m = g_m \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n)$: Certainly this holds for $m = 0$, since both sides are $\varnothing$. Furthermore

$$\begin{array}{rcl}
g'_{m+1} & = & f' \circ (\mathbf{1}_{\Gamma \,\&\, \Delta^{\#r}} \,\&\, g'_m) \circ \Delta_{\Gamma \,\&\, \Delta^{\#r}} \\
 & = & f \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n \,\&\, \mathbf{1}_{\mathbb{P}}) \circ (\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Delta^{\#r}} \,\&\, g'_m) \circ \Delta_{\Gamma \,\&\, \Delta^{\#r}} \\
 & = & f \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n \,\&\, g'_m) \circ \Delta_{\Gamma \,\&\, \Delta^{\#r}} \\
 & = & f \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n \,\&\, (g_m \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n))) \circ \Delta_{\Gamma \,\&\, \Delta^{\#r}} \\
 & = & f \circ (\mathbf{1}_{\Gamma'} \,\&\, g_m) \\
 & & \circ\, (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n \,\&\, \mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n) \circ \Delta_{\Gamma \,\&\, \Delta^{\#r}} \\
 & = & f \circ (\mathbf{1}_{\Gamma'} \,\&\, g_m) \circ \Delta_{\Gamma'} \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n) \\
 & = & g_{m+1} \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n)
\end{array}$$

by, respectively, the definition of g'_{m+1}, the relationship between f and f' established above, the universal property of products, the inductive hypothesis, the universal property of products, the universal property of diagonals and finally the definition of g_{m+1}. This completes the induction, so that

$$\begin{array}{cl}
\multicolumn{2}{l}{\llbracket \Gamma, \Delta^{\#r} \vdash_{s \cup s_1} (\mathtt{rec}\,\mathtt{x}.t)[v] = \mathtt{rec}\,\mathtt{x}.(t[v]) : \mathbb{P} \rrbracket} \\
= & \bigsqcup_{m \in \omega} g'_m \\
= & \bigsqcup_{m \in \omega} \big(g_m \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n) \big) \\
= & \big(\bigsqcup_{m \in \omega} g_m \big) \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n) \qquad \text{by continuity} \\
= & \llbracket \Gamma' \vdash_s \mathtt{rec}\,\mathtt{x}.t : \mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \,\&\, \llbracket v \rrbracket^n)
\end{array}$$

as required.

Match Suppose that $\Gamma', \Lambda'^{\#s'} \vdash_s \texttt{[}u\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ \# }s'\texttt{) => }t\texttt{]} : \mathbb{P}$ is derived from $\Gamma', \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_s t : \mathbb{P}$, $\Lambda' \vdash_{s''} u : \mathbb{Q}$ and $\vdash_{s''} \mathbb{Q} : q : \mathbb{Q}'$ where $s'' \subseteq s \setminus s'$. Suppose that Γ' is a reordering of $\Gamma, \texttt{y1} : \mathbb{R}^{\#r_1}, \ldots, \texttt{yn} : \mathbb{R}^{\#r_n}$ and that Λ' is a reordering of $\Lambda, \texttt{z1} : \mathbb{R}^{\#r'_1}, \ldots, \texttt{zm} : \mathbb{R}^{\#r'_m}$. Let $r = r_1 \cup \ldots \cup r_n$ and $r' = r'_1 \cup \ldots \cup r'_m$. There are now two possibilities, depending on whether $m = 0$ or not. Suppose that $m \neq 0$. Let Δ' and v' be Δ and v with all the variables renamed. Then

$$\begin{aligned}
&\llbracket \Gamma, \Lambda^{\#s'}, \Delta^{\#r\cup r'\cup s'} \vdash_{s\cup s_1} \texttt{[}u\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ \# }s'\texttt{) => }t\texttt{]}[v] : \mathbb{P} \rrbracket \\
&\quad = \quad \llbracket \Gamma, \Delta^{\#r\cup r'\cup s'}, \Lambda^{\#s'}, \Delta'^{\#r\cup r'\cup s'} \vdash_{s\cup s_1} \texttt{[}u[v']\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ \# }s'\texttt{) => }t[v]\texttt{]} : \mathbb{P} \rrbracket \\
&\qquad\qquad \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \Delta_{\Delta^{\#r\cup r'\cup s'}} \big) \\
&\quad = \quad \llbracket \Gamma, \Delta^{\#r}, \Lambda^{\#s'}, \Delta'^{\#r'\cup s'} \vdash_{s\cup s_1} \texttt{[}u[v']\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ \# }s'\texttt{) => }t[v]\texttt{]} : \mathbb{P} \rrbracket \\
&\qquad\qquad \circ \big(\mathbf{1}_\Gamma \,\&\, \tau^{(1)}_\Delta \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \tau^{(2)}_\Delta \big) \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \Delta_{\Delta^{\#r\cup r'\cup s'}} \big) \\
&\quad = \quad \epsilon_\mathbb{P} \circ {!}\llbracket \Gamma, \Delta^{\#r}, \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_{s\cup s_1} t[v] : \mathbb{P} \rrbracket \circ S_{(\Gamma, \Delta^{\#r}), \mathbb{Q}'^{\#s'}} \\
&\qquad\qquad \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Delta^{\#r}} \,\&\widehat{(\phi_{\mathbb{Q}'}} \circ \llbracket \vdash_{s''\cup s_1} \mathbb{Q} : q : \mathbb{Q}' \rrbracket^{\#s'++} \\
&\qquad\qquad\qquad\qquad \circ \llbracket \Lambda, \Delta^{\#r'} \vdash_{s''\cup s_1} u[v] : \mathbb{Q} \rrbracket^{\#s'++}) \big) \\
&\qquad\qquad \circ \big(\mathbf{1}_\Gamma \,\&\, \tau^{(1)}_\Delta \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \tau^{(2)}_\Delta \big) \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \Delta_{\Delta^{\#r\cup r'\cup s'}} \big)
\end{aligned}$$

by contraction and fresh-weakening. Writing

$$\begin{aligned}
(\cdots 1 \cdots) \quad =_{\text{def}} \quad & \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Delta^{\#r}} \,\&\widehat{(\phi_{\mathbb{Q}'}} \circ \llbracket \vdash_{s''\cup s_1} \mathbb{Q} : q : \mathbb{Q}' \rrbracket^{\#s'++} \\
& \qquad\qquad \circ \llbracket \Lambda, \Delta^{\#r'} \vdash_{s''\cup s_1} u[v] : \mathbb{Q} \rrbracket^{\#s'++}) \big) \\
& \circ \big(\mathbf{1}_\Gamma \,\&\, \tau^{(1)}_\Delta \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \tau^{(2)}_\Delta \big) \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \Delta_{\Delta^{\#r\cup r'\cup s'}} \big) \\
= \quad & \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Delta^{\#r}} \,\&\widehat{(\phi_{\mathbb{Q}'}} \circ \llbracket \vdash_{s''} \mathbb{Q} : q : \mathbb{Q}' \rrbracket^{\#s'++} \\
& \qquad \circ \llbracket \Lambda' \vdash_{s''} u : \mathbb{Q} \rrbracket^{\#s'++} \circ (\mathbf{1}_{\Lambda^{\#s'}} \,\&\, \llbracket v \rrbracket^{m\#s'++})) \big) \\
& \circ \big(\mathbf{1}_\Gamma \,\&\, \tau^{(1)}_\Delta \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \tau^{(2)}_\Delta \big) \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \Delta_{\Delta^{\#r\cup r'\cup s'}} \big)
\end{aligned}$$

by induction, and $\llbracket t \rrbracket$ for $\llbracket \Gamma', \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_s t : \mathbb{P} \rrbracket$,

$$\begin{aligned}
&\llbracket \Gamma, \Lambda^{\#s'}, \Delta^{\#r\cup r'\cup s'} \vdash_{s\cup s_1} \texttt{[}u\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ \# }s'\texttt{) => }t\texttt{]}[v] : \mathbb{P} \rrbracket \\
&\quad = \quad \epsilon_\mathbb{P} \circ {!}\llbracket \Gamma, \Delta^{\#r}, \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_{s\cup s_1} t[v] : \mathbb{P} \rrbracket \circ S_{(\Gamma, \Delta^{\#r}), \mathbb{Q}'^{\#s'}} \circ (\cdots 1 \cdots) \\
&\quad = \quad \epsilon_\mathbb{P} \circ {!}\llbracket t \rrbracket \circ {!}(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\mathbb{Q}'^{\#s'}} \,\&\llbracket v \rrbracket^n) \circ S_{(\Gamma, \Delta^{\#r}), \mathbb{Q}'^{\#s'}} \circ (\cdots 1 \cdots) \\
&\quad = \quad \epsilon_\mathbb{P} \circ {!}\llbracket t \rrbracket \circ S_{\Gamma', \mathbb{Q}'^{\#s'}} \circ (\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\mathbb{Q}'^{\#s'}} \,\&\llbracket v \rrbracket^n) \circ (\cdots 1 \cdots) \\
&\quad = \quad \epsilon_\mathbb{P} \circ {!}\llbracket t \rrbracket \circ S_{\Gamma', \mathbb{Q}'^{\#s'}} \circ \big(\mathbf{1}_{\Gamma'} \\
&\qquad\qquad\qquad \&\widehat{(\phi_{\mathbb{Q}'}} \circ \llbracket \vdash_{s''} \mathbb{Q} : q : \mathbb{Q}' \rrbracket^{\#s'++} \circ \llbracket \Lambda' \vdash_{s''} u : \mathbb{Q} \rrbracket^{\#s'++}) \big) \\
&\qquad\quad \circ (\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda'^{\#s'}} \,\&\llbracket v \rrbracket^n) \circ (\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Delta^{\#r}} \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \llbracket v \rrbracket^{m\#s'++}) \\
&\qquad\quad \circ \big(\mathbf{1}_\Gamma \,\&\, \tau^{(1)}_\Delta \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \tau^{(2)}_\Delta \big) \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\, \Delta_{\Delta^{\#r\cup r'\cup s'}} \big) \\
&\quad = \quad \llbracket \Gamma', \Lambda'^{\#s'} \vdash_s \texttt{[}u\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ \# }s'\texttt{) => }t\texttt{]} : \mathbb{P} \rrbracket \\
&\qquad\quad \circ \Big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\big((\llbracket v \rrbracket^n \,\&\, \llbracket v \rrbracket^{m\#s'++}) \circ (\tau^{(1)}_\Delta \,\&\, \tau^{(2)}_\Delta) \circ \Delta_{\Delta^{\#r\cup r'\cup s'}} \big) \Big) \\
&\quad = \quad \llbracket \Gamma', \Lambda'^{\#s'} \vdash_s \texttt{[}u\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ \# }s'\texttt{) => }t\texttt{]} : \mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \,\&\, \mathbf{1}_{\Lambda^{\#s'}} \,\&\llbracket v \rrbracket^{m+n})
\end{aligned}$$

as required. The proof in the case $m = 0$ is similar, albeit simpler.

Function Abstraction Suppose that $\Gamma' \vdash_s \lambda\,\mathtt{x}.t : \mathbb{Q} \to \mathbb{P}$ is derived from the judgement $\Gamma', \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}$, then

$$\begin{aligned}
&\llbracket \Gamma, \Delta^{\#r} \vdash_{s \cup s_1} (\lambda\,\mathtt{x}.t)[v] = \lambda\,\mathtt{x}.(t[v]) : \mathbb{Q} \to \mathbb{P} \rrbracket \\
&= \quad \mathbf{curry}\big(\llbracket \Gamma, \Delta^{\#r}, \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_{s \cup s_1} t[v] : \mathbb{P} \rrbracket\big) \\
&= \quad \mathbf{curry}\big(\llbracket \Gamma', \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \,\&\, \mathbf{1}_\mathbb{Q} \,\&\llbracket v \rrbracket^n)\big) \quad \text{by induction} \\
&= \quad \mathbf{curry}\big(\llbracket \Gamma', \mathtt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P} \rrbracket\big) \circ \big(\mathbf{1}_\Gamma \,\&\llbracket v \rrbracket^n\big) \\
&= \quad \llbracket \Gamma' \vdash_s \lambda\,\mathtt{x}.t : \mathbb{P} \rrbracket \circ \big(\mathbf{1}_\Gamma \,\&\llbracket v \rrbracket^n\big)
\end{aligned}$$

as required.

Function Application Suppose that $\Gamma', \Lambda' \vdash_s t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)} : \mathbb{P}$ is derived from the judgement $\Gamma' \vdash_s t : \mathbb{Q} \to \mathbb{P}$ and $\Lambda' \vdash_s u : \mathbb{Q}$. Suppose that Γ' is a reordering of the environment $\Gamma, \mathtt{y1} : \mathbb{R}^{\#r_1}, \ldots, \mathtt{yn} : \mathbb{R}^{\#r_n}$ and that Λ' is a reordering of $\Lambda, \mathtt{z1} : \mathbb{R}^{\#r'_1}, \ldots, \mathtt{zm} : \mathbb{R}^{\#r'_m}$. Let $r = r_1 \cup \ldots \cup r_n$ and $r' = r'_1 \cup \ldots \cup r'_m$. Then

$$\begin{aligned}
&\llbracket \Gamma, \Lambda, \Delta^{\#r \cup r'} \vdash_{s \cup s_1} (t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)})[v] = (t[v])\mathtt{(}(u[v])\mathtt{:}\mathbb{Q}\mathtt{)} : \mathbb{P} \rrbracket \\
&= \quad \llbracket \Gamma, \Delta^{\#r}, \Lambda, \Delta'^{\#r'} \vdash_{s \cup s_1} (t[v])\mathtt{(}(u[v'])\mathtt{:}\mathbb{Q}\mathtt{)} : \mathbb{P} \rrbracket \\
&\qquad \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_\Lambda \,\&((\tau^{(1)}_\Delta \,\&\, \tau^{(2)}_\Delta) \circ \Delta_\Delta) \\
&= \quad \mathbf{apply} \circ \big(\llbracket \Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{Q} \to \mathbb{P} \rrbracket \,\&\llbracket \Lambda, \Delta^{\#r'} \vdash_{s \cup s_1} u[v] : \mathbb{Q} \rrbracket\big) \\
&\qquad \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_\Lambda \,\&((\tau^{(1)}_\Delta \,\&\, \tau^{(2)}_\Delta) \circ \Delta_\Delta) \\
&= \quad \mathbf{apply} \circ \big(\llbracket \Gamma' \vdash_s t : \mathbb{Q} \to \mathbb{P} \rrbracket \,\&\llbracket \Lambda' \vdash_s u : \mathbb{Q} \rrbracket\big) \circ \big(\mathbf{1}_\Gamma \,\&\llbracket v \rrbracket^n \,\&\, \mathbf{1}_\Lambda \,\&\llbracket v \rrbracket^m\big) \\
&\qquad \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_\Lambda \,\&((\tau^{(1)}_\Delta \,\&\, \tau^{(2)}_\Delta) \circ \Delta_\Delta) \\
&= \quad \mathbf{apply} \circ \big(\llbracket \Gamma' \vdash_s t : \mathbb{Q} \to \mathbb{P} \rrbracket \,\&\llbracket \Lambda' \vdash_s u : \mathbb{Q} \rrbracket\big) \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_\Lambda \,\&\llbracket v \rrbracket^{n+m}\big) \\
&= \quad \llbracket \Gamma', \Lambda' \vdash_s t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)} : \mathbb{P} \rrbracket \circ \big(\mathbf{1}_\Gamma \,\&\, \mathbf{1}_\Lambda \,\&\llbracket v \rrbracket^{n+m}\big)
\end{aligned}$$

as required.

Name Abstraction Suppose that $\Gamma' \vdash_s \mathtt{new}\,a.t : \delta\mathbb{P}$ is derived from the judgement $\Gamma'^{\#a} \vdash_{s \dot\cup \{a\}} t : \mathbb{P}$ where a is a fresh name and Γ' is a reordering of $\Gamma, \mathtt{y1} : \mathbb{R}^{\#r_1}, \ldots, \mathtt{yn} : \mathbb{R}^{\#r_n}$. Notice that as a is fresh, therefore $a \notin r$ and $a \notin s_1$ so that $a \# v$. Then

$$\begin{aligned}
&\llbracket \Gamma, \Delta^{\#r} \vdash_{s \cup s_1} (\mathtt{new}\,a.t)[v] = \mathtt{new}\,a.(t[v]) : \delta\mathbb{P} \rrbracket \\
&= \quad \delta^{++}_a \llbracket \Gamma^{\#a}, \Delta^{\#r \dot\cup \{a\}} \vdash_{s \cup s_1 \cup \{a\}} t[v] : \mathbb{P} \rrbracket \circ \widehat{\xi_{\llbracket \Gamma, \Delta^{\#r} \rrbracket}} \\
&= \quad \delta^{++}_a \llbracket \Gamma'^{\#a} \vdash_{s \cup \{a\}} t : \mathbb{P} \rrbracket \circ \delta^{++}_a (\mathbf{1}_{\Gamma^{\#a}} \,\&\, \llbracket v \rrbracket^{n\#a++}) \circ \widehat{\xi_{\llbracket \Gamma, \Delta^{\#r} \rrbracket}} \quad \text{by ind.} \\
&= \quad \delta^{++}_a \llbracket \Gamma'^{\#a} \vdash_{s \cup \{a\}} t : \mathbb{P} \rrbracket \circ \widehat{\xi_{\llbracket \Gamma' \rrbracket}} \circ (\mathbf{1}_\Gamma \,\&\llbracket v \rrbracket^n) \\
&= \quad \llbracket \Gamma' \vdash_s \mathtt{new}\,a.t : \delta\mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \,\&\llbracket v \rrbracket^n)
\end{aligned}$$

as required.

Name Application Suppose that $a \notin s$ and that $\Gamma'^{\#a} \vdash_{s \dot\cup \{a\}} t\mathtt{[}a\mathtt{]} : \mathbb{P}$ is derived from $\Gamma' \vdash_s t : \delta\mathbb{P}$ where Γ' is a reordering of $\Gamma, \mathtt{y1} : \mathbb{R}^{\#r_1}, \ldots, \mathtt{yn} : \mathbb{R}^{\#r_n}$. By hypothesis, $s_1 \cap (r \mathbin{\dot\cup} \{a\}) = \varnothing$ so that $a \notin s_1$ and hence $a \mathrel{\#} v$. Then

$$\begin{array}{rl} \multicolumn{2}{l}{\llbracket \Gamma^{\# a}, \Delta^{\# r \dot\cup \{a\}} \vdash_{s \cup s_1 \dot\cup \{a\}} (t\,[a]\,)[v] = (t[v])\,[a]\, : \mathbb{P} \rrbracket} \\ = & \widehat{\zeta_{\mathbb{P}}} \circ \llbracket \Gamma, \Delta^{\# r} \vdash_{s \cup s_1} t[v] : \delta\mathbb{P} \rrbracket^{\# a{+}{+}} \\ = & \widehat{\zeta_{\mathbb{P}}} \circ \big(\llbracket \Gamma' \vdash_s t : \delta\mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \mathbin{\&} \llbracket v \rrbracket^n) \big)^{\# a{+}{+}} \\ = & \widehat{\zeta_{\mathbb{P}}} \circ \llbracket \Gamma' \vdash_s t : \delta\mathbb{P} \rrbracket^{\# a{+}{+}} \circ (\mathbf{1}_{\Gamma^{\# a}} \mathbin{\&} \llbracket v \rrbracket^{n \# a{+}{+}}) \\ = & \llbracket \Gamma'^{\# a} \vdash_{s \dot\cup \{a\}} t\,[a]\, : \mathbb{P} \rrbracket \circ (\mathbf{1}_{\Gamma^{\# a}} \mathbin{\&} \llbracket v \rrbracket^{n \# a{+}{+}}) \end{array}$$

as required.

Labelling Suppose that $\Gamma' \vdash_s \ell_0 {:} t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$ is derived from $\Gamma' \vdash_s t : \mathbb{P}_{\ell_0}$, then

$$\begin{array}{rll} \multicolumn{3}{l}{\llbracket \Gamma, \Delta^{\# r} \vdash_{s \cup s_1} (\ell_0 {:} t)[v] = \ell_0 {:} (t[v]) : \bigoplus_{\ell \in L} \mathbb{P}_\ell \rrbracket} \\ = & \mathbf{in}_{\ell_0} \circ \llbracket \Gamma, \Delta^{\# r} \vdash_{s \cup s_1} t[v] : \mathbb{P}_{\ell_0} \rrbracket & \\ = & \mathbf{in}_{\ell_0} \circ \llbracket \Gamma' \vdash_s t : \mathbb{P}_{\ell_0} \rrbracket \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n) & \text{by induction} \\ = & \llbracket \Gamma' \vdash_s \ell_0 {:} t : \bigoplus_{\ell \in L} \mathbb{P}_\ell \rrbracket \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n) & \end{array}$$

as required.

Label Projection Suppose that $\Gamma' \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}$ is derived from the judgement $\Gamma' \vdash_s t : \bigoplus_{\ell \in L} \mathbb{P}_\ell$, then

$$\begin{array}{rll} \multicolumn{3}{l}{\llbracket \Gamma, \Delta^{\# r} \vdash_{s \cup s_1} (\pi_{\ell_0} t)[v] = \pi_{\ell_0}(t[v]) : \mathbb{P}_{\ell_0} \rrbracket} \\ = & \mathbf{out}_{\ell_0} \circ \llbracket \Gamma, \Delta^{\# r} \vdash_{s \cup s_1} t[v] : \bigoplus_{\ell \in L} \mathbb{P}_\ell \rrbracket & \\ = & \mathbf{out}_{\ell_0} \circ \llbracket \Gamma' \vdash_s t : \bigoplus_{\ell \in L} \mathbb{P}_\ell \rrbracket \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n) & \text{by induction} \\ = & \llbracket \Gamma' \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0} \rrbracket \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n) & \end{array}$$

as required.

Nondeterministic Sum Suppose that $\Gamma' \vdash_s \sum_{i \in I} t_i : \mathbb{P}$ is derived from the judgement $\Gamma' \vdash_{s_i} t_i : \mathbb{P}$ for all $i \in I$, then

$$\begin{array}{rll} \multicolumn{3}{l}{\llbracket \Gamma, \Delta^{\# r} \vdash_{s \cup s_1} (\sum_{i \in I} t_i)[v] = \sum_{i \in I} (t_i[v]) : \mathbb{P} \rrbracket} \\ = & \sum_{i \in I} \llbracket \Gamma, \Delta^{\# r} \vdash_{s \cup s_1} t_i[v] : \mathbb{P} \rrbracket & \\ = & \sum_{i \in I} \big(\llbracket \Gamma' \vdash_s t_i : \mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n) \big) & \text{by induction} \\ = & \big(\sum_{i \in I} \llbracket \Gamma' \vdash_s t_i : \mathbb{P} \rrbracket \big) \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n) & \\ = & \llbracket \Gamma' \vdash_s \sum_{i \in I} t_i : \mathbb{P} \rrbracket \circ (\mathbf{1}_\Gamma \circ \llbracket v \rrbracket^n) & \end{array}$$

as required.

Recursive Type Folding Suppose that $\Gamma' \vdash_s \mathtt{abs}\, t : \mu_j \vec{P}.\, \vec{\mathbb{P}}$ is derived from $\Gamma' \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\, \vec{\mathbb{P}}/\vec{P}]$, then

$$\begin{array}{rl} & [\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} (\mathtt{abs}\, t)[v] = \mathtt{abs}\,(t[v]) : \mu_j \vec{P}.\, \vec{\mathbb{P}}]\!] \\ = & \mathbf{abs} \circ [\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mathbb{P}_j[\mu\vec{P}.\, \vec{\mathbb{P}}/\vec{P}]]\!] \\ = & \mathbf{abs} \circ [\![\Gamma' \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\, \vec{\mathbb{P}}/\vec{P}]]\!] \circ (\mathbf{1}_\Gamma \circ [\![v]\!]^n) \quad \text{by induction} \\ = & [\![\Gamma' \vdash_s \mathtt{abs}\, t : \mu_j \vec{P}.\, \vec{\mathbb{P}}]\!] \circ (\mathbf{1}_\Gamma \circ [\![v]\!]^n) \end{array}$$

as required.

Recursive Type Unfolding Suppose that $\Gamma' \vdash_s \mathtt{rep}\, t : \mathbb{P}_j[\mu\vec{P}.\, \vec{\mathbb{P}}/\vec{P}]$ is derived from $\Gamma' \vdash_s t : \mu_j \vec{P}.\, \vec{\mathbb{P}}$, then

$$\begin{array}{rl} & [\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} (\mathtt{rep}\, t)[v] = \mathtt{rep}\,(t[v]) : \mathbb{P}_j[\mu\vec{P}.\, \vec{\mathbb{P}}/\vec{P}]]\!] \\ = & \mathbf{rep} \circ [\![\Gamma, \Delta^{\#r} \vdash_{s \cup s_1} t[v] : \mu_j \vec{P}.\, \vec{\mathbb{P}}]\!] \\ = & \mathbf{rep} \circ [\![\Gamma' \vdash_s t : \mu_j \vec{P}.\, \vec{\mathbb{P}}]\!] \circ (\mathbf{1}_\Gamma \circ [\![v]\!]^n) \quad \text{by induction} \\ = & [\![\Gamma' \vdash_s \mathtt{rep}\, t : \mathbb{P}_j[\mu\vec{P}.\, \vec{\mathbb{P}}/\vec{P}]]\!] \circ (\mathbf{1}_\Gamma \circ [\![v]\!]^n) \end{array}$$

as required. □

5.4 Soundness and Adequacy

This section demonstrates that the denotational semantics described above corresponds closely with the operational semantics given in chapter 4.

The operational semantics gives rise to a notion of observation that can be made about a process: it is possible to observe an action $\vdash \mathbb{P} : p : \mathbb{P}'$ by deriving a judgement of the form $\mathbb{P} : t \xrightarrow{p} t'$. In fact the match operator reduces these general observations to observations of just `!` actions, because to observe the action p in the term t is the same as to observe a `!` action in the term `[`t `>` p`(x:`$\mathbb{P}$ `#` s`) => !nil]`. Although it might seem at first glance that this setup would give a full abstraction result 'for free' as in the development of HOPLA, it is not the case. Section 7.1 demonstrates a counterexample to full abstraction which relies on the fact that although it is possible to define every path using terms of the language, it is not possible to distinguish them all.

Firstly it is shown in section 5.4.1 that the denotational semantics is *sound*: the term t may perform a `!` action only if a corresponding path exists in the denotation $[\![t]\!]$.

Then section 5.4.2 introduces a logical relation between paths and terms and finally this logical relation is used in section 5.4.3 to demonstrate that the denotational semantics is *adequate*: as a converse to soundness, the term t may perform a `!` action whenever a corresponding path exists in the denotation $[\![t]\!]$.

5.4.1 Soundness

This section demonstrates that the denotational semantics of Nominal HOPLA is *sound*: the term t may perform a `!` action only if a corresponding path exists in the denotation $[\![t]\!]$. In fact, the path to this result is via a slightly more general lemma, which uses the intuition that the denotation $[\![p]\!]$ of an action p acts as a kind of 'destructor' for p, matching its input against the action p and returning a set of resumptions after performing p.

5.4.1.1 Lemma. *If* $\mathbb{P} : t \xrightarrow{p} t'$ *and* $\vdash \mathbb{P} : p : \mathbb{P}'$ *and* s *is a finite set of names such that* $\mathrm{supp}(t, p, t') \subseteq s$*, then*

$$[\![\vdash_s \mathtt{!}t' : !\mathbb{P}']\!] \sqsubseteq [\![\vdash_s \mathbb{P} : p : \mathbb{P}']\!] \circ [\![\vdash_s t : \mathbb{P}']\!].$$

Proof. By induction on the derivation of $\mathbb{P} : t \xrightarrow{p} t'$ as follows.

Recursion Suppose that the judgement $\mathbb{P} : \mathtt{rec\,x}.t \xrightarrow{p} t'$ is derived from $\mathbb{P} : t[\mathtt{rec\,x}.t/\mathtt{x}] \xrightarrow{p} t'$. Let t^* be defined by $t^*(f) = [\![t]\!] \circ f$, then

$$\begin{array}{lll}
\multicolumn{3}{l}{[\![\vdash_s \mathbb{P} : p : \mathbb{P}']\!] \circ [\![\vdash_s \mathtt{rec\,x}.t : \mathbb{P}]\!]} \\
\quad = & [\![p]\!] \circ \mathrm{fix}(t^*) & \text{by the denotational semantics} \\
\quad = & [\![p]\!] \circ t^*\big(\mathrm{fix}(t^*)\big) & \text{by the properties of fix} \\
\quad = & [\![p]\!] \circ [\![t]\!] \circ \mathrm{fix}(t^*) & \text{by definition of } t^* \\
\quad = & [\![p]\!] \circ [\![t]\!] \circ [\![\mathtt{rec\,x}.t]\!] & \text{by the denotational semantics} \\
\quad = & [\![p]\!] \circ [\![t[\mathtt{rec\,x}.t/\mathtt{x}]]\!] & \text{by the substitution lemma} \\
\quad \sqsupseteq & [\![\vdash_s \mathtt{!}t' : !\mathbb{P}']\!] & \text{by induction}
\end{array}$$

as required.

Prefix If $!\mathbb{P} : \mathtt{!}t \xrightarrow{!} t$ then

$$[\![\vdash_s !\mathbb{P} : \mathtt{!} : \mathbb{P}]\!] \circ [\![\vdash_s \mathtt{!}t : !\mathbb{P}]\!] = [\![\vdash_s \mathtt{!}t : !\mathbb{P}]\!]$$

as required.

Match Suppose that the judgement $\mathbb{P} : \texttt{[}u\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ # }s'\texttt{) => }t\texttt{]} \xrightarrow{p} t'$ is derived from $\mathbb{P} : t[u'/\mathtt{x}] \xrightarrow{p} t'$ and $\mathbb{Q} : u \xrightarrow{q} u'$, then

$$\begin{aligned}
&\llbracket \vdash_s \mathbb{P} : p : \mathbb{P}' \rrbracket \circ \llbracket \vdash_s \texttt{[}u\texttt{ > }q\texttt{(x:}\mathbb{Q}'\texttt{ # }s'\texttt{) => }t\texttt{]} : \mathbb{P} \rrbracket \\
&\quad = \llbracket p \rrbracket \circ \epsilon_{\mathbb{P}} \circ !\llbracket t \rrbracket \circ \widehat{\phi^{!(s')}_{\mathbb{Q}'}} \circ \big(\llbracket q \rrbracket \circ \llbracket u \rrbracket\big)^{\#s'++} \\
&\quad \sqsupseteq \llbracket p \rrbracket \circ \epsilon_{\mathbb{P}} \circ !\llbracket t \rrbracket \circ \widehat{\phi^{!(s')}_{\mathbb{Q}'}} \circ \llbracket \mathtt{!}\, u' \rrbracket^{\#s'++} && \text{by induction on } u \\
&\quad = \llbracket p \rrbracket \circ \epsilon_{\mathbb{P}} \circ !\llbracket t \rrbracket \circ \widehat{\phi^{!(s')}_{\mathbb{Q}'}} \circ \eta_{\mathbb{Q}'}^{\#s'++} \circ \llbracket u' \rrbracket^{\#s'++} \\
&\quad \sqsupseteq \llbracket p \rrbracket \circ \epsilon_{\mathbb{P}} \circ !\llbracket t \rrbracket \circ \eta_{\mathbb{Q}'\#s'} \circ \llbracket u' \rrbracket^{\#s'++} && \text{by lemma 3.4.8.37} \\
&\quad = \llbracket p \rrbracket \circ \epsilon_{\mathbb{P}} \circ \eta_{\mathbb{P}} \circ \llbracket t \rrbracket \circ \llbracket u' \rrbracket^{\#s'++} && \text{by naturality of } \eta \\
&\quad = \llbracket p \rrbracket \circ \llbracket t \rrbracket \circ \llbracket u' \rrbracket^{\#s'++} && \text{by triangular identity} \\
&\quad = \llbracket p \rrbracket \circ \llbracket t[u'/\mathtt{x}] \rrbracket && \text{by substitution lemma} \\
&\quad \sqsupseteq \llbracket \vdash_s \mathtt{!}\, t' : !\mathbb{P}' \rrbracket && \text{by induction on } t
\end{aligned}$$

as required.

Name Abstraction Suppose that $\delta\mathbb{P} : \mathtt{new}\, a\,\mathtt{.}\, t \xrightarrow{\mathtt{new}\, a\mathtt{.}\, p} \mathtt{new}\, a\,\mathtt{.}\, t'$ is derived from $\mathbb{P} : t \xrightarrow{p} t'$, then

$$\begin{aligned}
&\llbracket \vdash_s \delta\mathbb{P} : \mathtt{new}\, a\mathtt{.}\, p : \delta\mathbb{P}' \rrbracket \circ \llbracket \vdash_s \mathtt{new}\, a\,\mathtt{.}\, t : \delta\mathbb{P} \rrbracket \\
&\quad = \big(\widehat{\theta^{!}_{\mathbb{P}'}} \circ \delta_a^{++}\llbracket p \rrbracket\big) \circ \delta_a^{++}\llbracket t \rrbracket \circ \widehat{\xi_{\mathbb{O}}} && \text{by the denotational semantics} \\
&\quad = \widehat{\theta^{!}_{\mathbb{P}'}} \circ \delta_a^{++}\big(\llbracket p \rrbracket \circ \llbracket t \rrbracket\big) \circ \widehat{\xi_{\mathbb{O}}} \\
&\quad \sqsupseteq \widehat{\theta^{!}_{\mathbb{P}'}} \circ \delta_a^{++}\llbracket \mathtt{!}\, t' \rrbracket \circ \widehat{\xi_{\mathbb{O}}} && \text{by induction} \\
&\quad = \widehat{\theta^{!}_{\mathbb{P}'}} \circ \delta_a^{++}\eta_{\mathbb{P}'} \circ \delta_a^{++}\llbracket t' \rrbracket \circ \widehat{\xi_{\mathbb{O}}} \\
&\quad = \eta_{\delta_a\mathbb{P}'} \circ \delta_a^{++}\llbracket t' \rrbracket \circ \widehat{\xi_{\mathbb{O}}} && \text{by lemma 3.4.7.6} \\
&\quad = \llbracket \vdash_s \mathtt{!new}\, a\,\mathtt{.}\, t' : !\delta\mathbb{P}' \rrbracket && \text{by the denotational semantics}
\end{aligned}$$

as required.

Name Application Suppose that the judgement $\mathbb{P} : t\,\mathtt{[}a\mathtt{]} \xrightarrow{p} t'$ is derived from $\delta\mathbb{P} : t \xrightarrow{\mathtt{new}\, a\mathtt{.}\, p} \mathtt{new}\, a\,\mathtt{.}\, t'$, then

$$\begin{aligned}
&\llbracket \vdash_{s\dot\cup\{a\}} \mathbb{P} : p : \mathbb{P}' \rrbracket \circ \llbracket \vdash_{s\dot\cup\{a\}} t\,\mathtt{[}a\mathtt{]} : \mathbb{P} \rrbracket \\
&\quad = \llbracket p \rrbracket \circ \widehat{\zeta_{\mathbb{P}}} \circ \llbracket t \rrbracket^{\#a++} && \text{by the denotational semantics} \\
&\quad = \widehat{\zeta_{!\mathbb{P}'}} \circ (\delta_a^{++}\llbracket p \rrbracket \circ \llbracket t \rrbracket)^{\#a++} && \text{by naturality} \\
&\quad = \widehat{\zeta_{!\mathbb{P}'}} \circ (\widehat{\theta^{!}_{\mathbb{P}'}}^{-1} \circ \widehat{\theta^{!}_{\mathbb{P}'}} \circ \delta_a^{++}\llbracket p \rrbracket \circ \llbracket t \rrbracket)^{\#a++} \\
&\quad = \widehat{\zeta_{!\mathbb{P}'}} \circ (\widehat{\theta^{!}_{\mathbb{P}'}}^{-1} \circ \llbracket \mathtt{new}\, a\mathtt{.}\, p \rrbracket \circ \llbracket t \rrbracket)^{\#a++} \\
&\quad \sqsupseteq \widehat{\zeta_{!\mathbb{P}'}} \circ (\widehat{\theta^{!}_{\mathbb{P}'}}^{-1} \circ \llbracket \mathtt{!}\, t' \rrbracket)^{\#a++} && \text{by induction} \\
&\quad = \widehat{\zeta_{!\mathbb{P}'}} \circ (\widehat{\theta^{!}_{\mathbb{P}'}}^{-1} \circ \eta_{\delta_a\mathbb{P}'} \circ \llbracket t' \rrbracket)^{\#a++} && \text{by the denotational semantics} \\
&\quad = \widehat{\zeta_{!\mathbb{P}'}} \circ (\delta_a^{++}\eta_{\mathbb{P}'} \circ \llbracket t' \rrbracket)^{\#a++} && \text{by lemma 3.4.7.6} \\
&\quad = \eta_{\mathbb{P}'} \circ \widehat{\zeta_{\mathbb{P}'}} \circ \llbracket t' \rrbracket^{\#a++} && \text{by naturality} \\
&\quad = \llbracket \vdash_{s\dot\cup\{a\}} \mathtt{!}\, t' : !\mathbb{P}' \rrbracket
\end{aligned}$$

by the denotational semantics as required.

Function Abstraction Suppose that the judgement $\mathbb{Q} \to \mathbb{P} : \lambda \mathtt{x}.t \xrightarrow{u \mapsto p} t'$ is derived from $\mathbb{P} : t[u/\mathtt{x}] \xrightarrow{p} t'$, then

$$\begin{array}{lll} \multicolumn{3}{l}{[\![\vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}']\!] \circ [\![\vdash_s \lambda \mathtt{x}.t : \mathbb{Q} \to \mathbb{P}]\!]} \\ = & [\![p]\!] \circ \mathbf{apply} \circ \big(\mathbf{1}_{\mathbb{Q} \to \mathbb{P}} \,\&\, [\![u]\!]\big) \circ \mathbf{curry}[\![t]\!] & \text{by the denotational semantics} \\ = & [\![p]\!] \circ \mathbf{apply} \circ \big(\mathbf{curry}[\![t]\!] \,\&\, \mathbf{1}_{\mathbb{Q}}\big) \circ [\![u]\!] & \text{by naturality} \\ = & [\![p]\!] \circ [\![t]\!] \circ [\![u]\!] & \\ = & [\![p]\!] \circ [\![t[u/\mathtt{x}]]\!] & \text{by the substitution lemma} \\ \sqsupseteq & [\![\vdash_s {!t'} : !\mathbb{P}']\!] & \text{by induction} \end{array}$$

as required.

Function Application Suppose that the judgement $\mathbb{P} : t(u{:}\mathbb{Q}) \xrightarrow{p} t'$ is derived from $\mathbb{Q} \to \mathbb{P} : t \xrightarrow{u \mapsto p} t'$, then

$$\begin{array}{lll} \multicolumn{3}{l}{[\![\vdash_s \mathbb{P} : p : \mathbb{P}']\!] \circ [\![\vdash_s t(u{:}\mathbb{Q}) : \mathbb{P}]\!]} \\ = & [\![p]\!] \circ \mathbf{apply} \circ \big([\![t]\!] \,\&\, [\![u]\!]\big) & \text{by the denotational semantics} \\ = & [\![p]\!] \circ \mathbf{apply} \circ \big(\mathbf{1}_{\mathbb{Q} \to \mathbb{P}} \,\&\, [\![u]\!]\big) \circ [\![t]\!] & \text{by naturality} \\ = & [\![u \mapsto p]\!] \circ [\![t]\!] & \text{by the denotational semantics} \\ \sqsupseteq & [\![\vdash_s {!t'} : !\mathbb{P}']\!] & \text{by induction} \end{array}$$

as required.

Labelling Suppose that $\bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0{:}t \xrightarrow{\ell_0 : p} t'$ is derived from $\mathbb{P}_{\ell_0} : t \xrightarrow{p} t'$, then

$$\begin{array}{lll} \multicolumn{3}{l}{[\![\vdash_s \bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0{:}p : \mathbb{P}']\!] \circ [\![\vdash_s \ell_0{:}t : \bigoplus_{\ell \in L} \mathbb{P}_\ell]\!]} \\ = & [\![p]\!] \circ \mathbf{out}_{\ell_0} \circ \mathbf{in}_{\ell_0} \circ [\![t]\!] & \text{by the denotational semantics} \\ = & [\![p]\!] \circ [\![t]\!] & \text{by properties of the biproduct} \\ \sqsupseteq & [\![\vdash_s {!t'} : !\mathbb{P}']\!] & \text{by induction} \end{array}$$

as required

Label Projection Suppose that the judgement $\mathbb{P}_{\ell_0} : \pi_{\ell_0} t \xrightarrow{p} t'$ is derived from $\bigoplus_{\ell \in L} \mathbb{P}_\ell : t \xrightarrow{\ell_0 : p} t'$, then

$$\begin{array}{lll} \multicolumn{3}{l}{[\![\vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}']\!] \circ [\![\vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0}]\!]} \\ = & [\![p]\!] \circ \mathbf{out}_{\ell_0} \circ [\![t]\!] & \text{by the denotational semantics} \\ = & [\![\ell_0{:}p]\!] \circ [\![t]\!] & \text{by the denotational semantics} \\ \sqsupseteq & [\![\vdash_s {!t'} : !\mathbb{P}']\!] & \text{by induction} \end{array}$$

as required

Recursive Type Folding Suppose that $\mu_j\vec{P}.\,\vec{\mathbb{P}} : \texttt{abs}\,t \xrightarrow{\texttt{abs}\ p} t'$ is derived from $\mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : t \xrightarrow{p} t'$, then

$$\begin{array}{lll} \multicolumn{3}{l}{[\![\,\vdash_s \mu_j\vec{P}.\,\vec{\mathbb{P}} : \texttt{abs}\ p : \mathbb{P}']\!] \circ [\![\,\vdash_s \texttt{abs}\,t : \mu_j\vec{P}.\,\vec{\mathbb{P}}]\!]} \\ \quad = & [\![p]\!] \circ \textbf{rep} \circ \textbf{abs} \circ [\![t]\!] & \text{by the denotational semantics} \\ \quad = & [\![p]\!] \circ [\![t]\!] & \text{as } \textbf{rep} = \textbf{abs}^{-1} \\ \quad \sqsupseteq & [\![\,\vdash_s \texttt{!}\,t' : !\mathbb{P}']\!] & \text{by induction} \end{array}$$

as required

Recursive Type Unfolding Suppose that $\mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : \texttt{rep}\,t \xrightarrow{p} t'$ is derived from $\mu_j\vec{P}.\,\vec{\mathbb{P}} : t \xrightarrow{\texttt{abs}\ p} t'$, then

$$\begin{array}{lll} \multicolumn{3}{l}{[\![\,\vdash_s \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}']\!] \circ [\![\,\vdash_s \texttt{rep}\,t : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]]\!]} \\ \quad = & [\![p]\!] \circ \textbf{rep} \circ [\![t]\!] & \text{by the denotational semantics} \\ \quad = & [\![\texttt{abs}\ p]\!] \circ [\![t]\!] & \text{by the denotational semantics} \\ \quad \sqsupseteq & [\![\,\vdash_s \texttt{!}\,t' : !\mathbb{P}']\!] & \text{by induction} \end{array}$$

as required

Nondeterministic Sum Suppose that the judgement $\mathbb{P} : \sum_{i\in I} t_i \xrightarrow{p} t'$ is derived from $\mathbb{P} : t_{i_0} \xrightarrow{p} t'$, then

$$\begin{array}{lll} \multicolumn{3}{l}{[\![\,\vdash_s \mathbb{P} : p : \mathbb{P}']\!] \circ [\![\,\vdash_s \textstyle\sum_{i\in I} t_i : \mathbb{P}]\!]} \\ \quad \sqsupseteq & [\![p]\!] \circ [\![t_i]\!] & \text{by the denotational semantics} \\ \quad \sqsupseteq & [\![\,\vdash_s \texttt{!}\,t' : !\mathbb{P}']\!] & \text{by induction} \end{array}$$

as required □

5.4.1.2 Corollary (Soundness). *If* $!\mathbb{P} : t \xrightarrow{!} t'$ *and* s *is a finite set of names such that* $\mathrm{supp}(t, t') \subseteq s$ *then*

$$[\![\,\vdash_s \texttt{!}\,t' : !\mathbb{P}]\!] \sqsubseteq [\![\,\vdash_s t : !\mathbb{P}]\!].$$

Proof. By lemma 5.4.1.1, letting $p = \texttt{!}$ and noting that $[\![\texttt{!}]\!] = \mathbf{1}_{!\mathbb{P}}$. □

5.4.2 A Logical Relation

Define a relation $X \trianglelefteq_{\mathbb{P}} t$ between subsets $X \subseteq \mathbb{P}$ and terms such that $\vdash t : \mathbb{P}$ by way of an auxiliary relation $p \in_{\mathbb{P}} t$ between paths $p \in \mathbb{P}$ and terms such that $\vdash t : \mathbb{P}$ as shown in 5.4.2.1. The intuition behind the statement that $p \in_{\mathbb{P}} t$ is that p is a computation path of type $\mathbb{P}$ that the process t may perform. With this intuition in mind, lemma 5.4.2.2 demonstrates that if t can perform the path p_0 then it can certainly perform the shorter path p_1, and lemma 5.4.2.3 demonstrates that if t_0 can perform the path p and t_1 operationally subsumes t_0 then t_1 can also follow the path p. These technical lemmas are important building blocks in demonstrating that if $p \in [\![t]\!]$ then t can perform the path p which is the subject of the next section.

$$\begin{aligned}
X \trianglelefteq_{\mathbb{P}} t &\iff \forall p \in X.\ p \in_{\mathbb{P}} t \\
P \in_{!\mathbb{P}} t &\iff \exists t'.\ !\mathbb{P} : t \xrightarrow{!} t' \text{ and } P \trianglelefteq_{\mathbb{P}} t' \\
Q \mapsto p \in_{\mathbb{Q}\to\mathbb{P}} t &\iff \forall u.\ (Q \trianglelefteq_{\mathbb{Q}} u \Rightarrow p \in_{\mathbb{P}} t\,\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)}) \\
\texttt{new}\,a.\,p \in_{\delta\mathbb{P}} t &\iff \textbf{fresh}\,a\,\textbf{in}\,p \in_{\mathbb{P}} t\,\texttt{[}a\texttt{]} \\
\ell_0 : p \in_{\bigoplus_{\ell\in L}\mathbb{P}_\ell} t &\iff p \in_{\mathbb{P}_{\ell_0}} \pi_{\ell_0} t \\
\texttt{abs}\,p \in_{\mu_j \vec{P}.\,\vec{\mathbb{P}}} t &\iff p \in_{\mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]} \texttt{rep}\,t
\end{aligned} \tag{5.4.2.1}$$

5.4.2.2 Lemma. *If $p_0 \in_{\mathbb{P}} t$ and $p_1 \leq_{\mathbb{P}} p_0$ then $p_1 \in_{\mathbb{P}} t$.*

Proof. The proof is by induction on the derivation of the statement $p_1 \leq_{\mathbb{P}} p_0$.

Path Set By assumption $P_0 \in_{!\mathbb{P}} t$ so that from the definition of the logical relation there exists t' such that $!\mathbb{P} : t \xrightarrow{!} t'$ and $P_0 \trianglelefteq_{\mathbb{P}} t'$. Since $P_1 \leq_{!\mathbb{P}} P_0$ it follows that for each $p \in P_1$ there exists $p' \in P_0$ such that $p \leq_{\mathbb{P}} p'$ so that by induction it follows that $p \in_{\mathbb{P}} t'$. Therefore $P_1 \trianglelefteq_{\mathbb{P}} t'$ so that $P_1 \in_{!\mathbb{P}} t$ as required.

Function Space Path The induction hypothesis permits the assumption that $Q_0 \mapsto p_0 \in_{\mathbb{Q}\to\mathbb{P}} t$ and $Q_1 \mapsto p_1 \leq_{\mathbb{Q}\to\mathbb{P}} Q_0 \mapsto p_0$ so that $p_1 \leq_{\mathbb{P}} p_0$ and $Q_0 \leq_{!\mathbb{Q}} Q_1$. In order to show that $Q_1 \mapsto p_1 \in_{\mathbb{Q}\to\mathbb{P}} t$ let u be such that $Q_1 \trianglelefteq_{\mathbb{Q}} u$. Since $Q_0 \leq_{!\mathbb{Q}} Q_1$ it follows that for each $q \in Q_0$ there exists $q' \in Q_1$ such that $q \leq_{\mathbb{Q}} q'$ so that by induction it follows that $q \in_{\mathbb{Q}} u$ and hence $Q_0 \trianglelefteq_{\mathbb{Q}} u$. Now since $Q_0 \mapsto p_0 \in_{\mathbb{Q}\to\mathbb{P}} t$ it must be the case that $p_0 \in_{\mathbb{P}} t\,\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)}$ and hence by induction $p_1 \in_{\mathbb{P}} t\,\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)}$ as required.

New Name Path By assumption, $\texttt{new}\,b.\,p_0 \in_{\delta\mathbb{P}} t$ and $\texttt{new}\,b.\,p_0 \leq_{\delta\mathbb{P}} \texttt{new}\,b.\,p_1$ where b is a fresh name, and hence $p_0 \leq_{\mathbb{P}} p_1$. In order to show that $\texttt{new}\,b.\,p_1 \in_{\delta\mathbb{P}} t$ it is sufficient to show that $p_1 \in_{\mathbb{P}} t\,\texttt{[}b\texttt{]}$ and hence by induction that $p_0 \in_{\mathbb{P}} t\,\texttt{[}b\texttt{]}$; this is immediate since $\texttt{new}\,b.\,p_0 \in_{\delta\mathbb{P}} t$.

Labelled Path By assumption, $\ell_0 : p_0 \in_{\bigoplus_{\ell\in L}\mathbb{P}_\ell} t$ and $\ell_0 : p_0 \leq_{\bigoplus_{\ell\in L}\mathbb{P}_\ell} \ell_1 : p_1$ so that $\ell_0 = \ell_1$ and $p_0 \leq_{\mathbb{P}_{\ell_0}} p_1$. In order to show that $\ell_0 : p_1 \in_{\bigoplus_{\ell\in L}\mathbb{P}_\ell} t$ it is sufficient to show that $p_1 \in_{\mathbb{P}_{\ell_0}} \pi_{\ell_0} t$ and hence by induction that $p_0 \in_{\mathbb{P}_{\ell_0}} \pi_{\ell_0} t$; this is immediate since $\ell_0 : p_0 \in_{\bigoplus_{\ell\in L}\mathbb{P}_\ell} t$.

Recursive Type Path The inductive hypothesis permits the assumption that $\mathtt{abs}\ p_0 \in_{\mu_j \vec{P}.\ \vec{\mathbb{P}}} t$ and $\mathtt{abs}\ p_0 \leq_{\mu_j \vec{P}.\ \vec{\mathbb{P}}} \mathtt{abs}\ p_1$ so that $p_0 \leq_{\mathbb{P}_j[\mu \vec{P}.\ \vec{\mathbb{P}}/\vec{P}]} p_1$. To show that $\mathtt{abs}\ p_1 \in_{\mu_j \vec{P}.\ \vec{\mathbb{P}}} t$ it is sufficient to show that $p_1 \in_{\mathbb{P}_j[\mu \vec{P}.\ \vec{\mathbb{P}}/\vec{P}]} \mathtt{rep}\, t$ and hence by induction that $p_0 \in_{\mathbb{P}_j[\mu \vec{P}.\ \vec{\mathbb{P}}/\vec{P}]} \mathtt{rep}\, t$; this is immediate since by assumption $\mathtt{abs}\ p_0 \in_{\mu_j \vec{P}.\ \vec{\mathbb{P}}} t$. □

Define a relation between closed terms of type $\mathbb{P}$ by $t_0 \mathrel{\underset{\sim}{\sqsubset}}_{\mathbb{P}} t_1$ if for all t' and p,

$$\mathbb{P} : t_0 \xrightarrow{p} t' \qquad \Rightarrow \qquad \mathbb{P} : t_1 \xrightarrow{p} t'$$

5.4.2.3 Lemma. *If $p \in_{\mathbb{P}} t_0$ and $t_0 \mathrel{\underset{\sim}{\sqsubset}}_{\mathbb{P}} t_1$ then $p \in_{\mathbb{P}} t_1$.*

Proof. The proof is by induction on the path p.

Path Set By assumption, $P \in_{!\mathbb{P}} t_0$, so there exists t' such that $P \trianglelefteq_{\mathbb{P}} t'$ and $!\mathbb{P} : t_0 \xrightarrow{!} t'$. Since $t_0 \mathrel{\underset{\sim}{\sqsubset}}_{\mathbb{P}} t_1$, $!\mathbb{P} : t_1 \xrightarrow{!} t'$ too, so that $P \in_{!\mathbb{P}} t_1$ as required.

Function Space Path By assumption, $Q \mapsto p \in_{\mathbb{Q} \to \mathbb{P}} t_0$, so that for all u such that $Q \trianglelefteq_{\mathbb{Q}} u$ it follows that $p \in_{\mathbb{P}} t_0(u{:}\mathbb{Q})$. Also $t_0 \mathrel{\underset{\sim}{\sqsubset}}_{\mathbb{Q} \to \mathbb{P}} t_1$ implies that $t_0(u{:}\mathbb{Q}) \mathrel{\underset{\sim}{\sqsubset}}_{\mathbb{P}} t_1(u{:}\mathbb{Q})$ as follows. Suppose that $\mathbb{P} : t_0(u{:}\mathbb{Q}) \xrightarrow{p'} t'$, then by lemma 4.4.1.4(i) $\mathbb{Q} \to \mathbb{P} : t_0 \xrightarrow{u \mapsto p'} t'$ and hence $\mathbb{Q} \to \mathbb{P} : t_1 \xrightarrow{u \mapsto p'} t'$ so that finally $\mathbb{P} : t_1(u{:}\mathbb{Q}) \xrightarrow{p'} t'$. Thus by induction $p \in_{\mathbb{P}} t_1(u{:}\mathbb{Q})$, so that $Q \mapsto p \in_{\mathbb{Q} \to \mathbb{P}} t_1$ as required.

New Name Path By assumpton, $\mathtt{new}\, b.\, p \in_{\delta\mathbb{P}} t_0$ where b is a fresh name, so that $p \in_{\mathbb{P}} t_0[b]$. Also $t_0 \mathrel{\underset{\sim}{\sqsubset}}_{\delta\mathbb{P}} t_1$ implies that $t_0[b] \mathrel{\underset{\sim}{\sqsubset}}_{\mathbb{P}} t_1[b]$ as follows. Suppose that $\mathbb{P} : t_0[b] \xrightarrow{p'} t'$, then by lemma 4.4.1.4(iii) $\delta\mathbb{P} : t_0 \xrightarrow{\mathtt{new}\, b.\, p'} \mathtt{new}\, b.t'$ and hence $\delta\mathbb{P} : t_1 \xrightarrow{\mathtt{new}\, b.\, p'} \mathtt{new}\, b.t'$ so that finally $\mathbb{P} : t_1[b] \xrightarrow{p'} t'$. Thus by induction $p \in_{\mathbb{P}} t_1[b]$, so that $\mathtt{new}\, b.\, p \in_{\delta\mathbb{P}} t_1$ as required.

Labelled Path By assumpton, $\ell_0{:}p \in_{\bigoplus_{\ell \in L} \mathbb{P}_\ell} t_0$ so that $p \in_{\mathbb{P}_{\ell_0}} \pi_{\ell_0} t_0$. Also $t_0 \mathrel{\underset{\sim}{\sqsubset}}_{\bigoplus_{\ell \in L} \mathbb{P}_\ell} t_1$ implies that $\pi_{\ell_0} t_0 \mathrel{\underset{\sim}{\sqsubset}}_{\mathbb{P}_{\ell_0}} \pi_{\ell_0} t_1$ as follows. Suppose it is the case that $\mathbb{P}_{\ell_0} : \pi_{\ell_0} t_0 \xrightarrow{p'} t'$, then by lemma 4.4.1.4(ii) $\bigoplus_{\ell \in L} \mathbb{P}_\ell : t_0 \xrightarrow{\ell_0 : p'} t'$ and hence $\bigoplus_{\ell \in L} \mathbb{P}_\ell : t_1 \xrightarrow{\ell_0 : p'} t'$ so that finally $\mathbb{P}_{\ell_0} : \pi_{\ell_0} t_1 \xrightarrow{p'} t'$. Thus by induction $p \in_{\mathbb{P}_{\ell_0}} \pi_{\ell_0} t_1$, so that $\ell_0{:}p \in_{\bigoplus_{\ell \in L} \mathbb{P}_\ell} t_1$ as required.

Recursive Type Path By assumpton, $\mathtt{abs}\, p \in_{\mu_j \vec{P}.\, \vec{\mathbb{P}}} t_0$, so that by definition $p \in_{\mathbb{P}_j[\mu_j \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]} \mathtt{rep}\, t_0$. Also $t_0 \underset{\sim}{\sqsubset}_{\mu_j \vec{P}.\, \vec{\mathbb{P}}} t_1$ implies that $\mathtt{rep}\, t_0 \underset{\sim}{\sqsubset}_{\mathbb{P}_j[\mu_j \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]} \mathtt{rep}\, t_1$ as follows. Suppose that $\mathbb{P}_j[\mu_j \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : \mathtt{rep}\, t_0 \xrightarrow{p'} t'$, then by lemma 4.4.1.4(iv) $\mu_j \vec{P}.\, \vec{\mathbb{P}} : t_0 \xrightarrow{\mathtt{abs}\ p'} t'$ and hence $\mu_j \vec{P}.\, \vec{\mathbb{P}} : t_1 \xrightarrow{\mathtt{abs}\ p'} t'$ so that finally by the operational semantics it follows that $\mathbb{P}_j[\mu_j \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] : \mathtt{rep}\, t_1 \xrightarrow{p'} t'$. Thus by induction $p \in_{\mathbb{P}_j[\mu_j \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]} \mathtt{rep}\, t_1$, so that $\mathtt{abs}\, p \in_{\mu_j \vec{P}.\, \vec{\mathbb{P}}} t_1$ as required. □

5.4.3 Adequacy

The logical relation $X \trianglelefteq_{\mathbb{P}} t$ of the previous section is now used to demonstrate that if a path p appears — semantically — in the denotation $[\![t]\!]$ then the term t can — operationally — perform the path p. This result is complicated a little by the potential presence of free variables in t, leading to the statement of lemma 5.4.3.1. Furthermore, lemma 5.4.3.2 captures the intuition that an action p denotes a map $[\![p]\!]$ which matches its input against the action p and returns the resumptions of any matching paths. These two lemmas are proved simultaneously by a mutual induction on the structure of typing judgements.

5.4.3.1 Lemma. *Suppose $\Gamma \vdash_s t : \mathbb{P}$ where $\Gamma = \mathtt{x}_1 : \mathbb{P}_1{}^{\#s_1}, \ldots, \mathtt{x}_n : \mathbb{P}_n{}^{\#s_n}$. For each $i \in \{1, \ldots, n\}$ let $\gamma_i \in \widehat{\mathbb{P}_i}^{\#s_i}$ and let v_i be a closed term such that $\vdash_{s \setminus s_i} v_i : \mathbb{P}_i$ and $\gamma_i \trianglelefteq_{\mathbb{P}_i} v_i$. Then*

$$[\![\Gamma \vdash_s t : \mathbb{P}]\!]\langle \gamma_1, \ldots, \gamma_n \rangle_\Gamma \trianglelefteq_{\mathbb{P}} t[v]$$

where $t[v]$ is the term obtained by simultaneously substituting each $\mathtt{x}_i$ with v_i.

5.4.3.2 Lemma. *If $\vdash_s \mathbb{P} : p : \mathbb{P}'$ and $X \trianglelefteq_{\mathbb{P}} t$ and $P \in [\![p]\!]X$ then there exists t' such that $\mathbb{P} : t \xrightarrow{p} t'$ and $P \trianglelefteq_{\mathbb{P}'} t'$.*

Proof. The proof is by mutual induction on the respective derivations of the judgements $\Gamma \vdash_s t : \mathbb{P}$ and $\vdash_s \mathbb{P} : p : \mathbb{P}'$. Where it is unambiguous, write $\langle \cdot \rangle_\Gamma$ as $\langle \cdot \rangle$.

Variable Here $n = 1$, $\mathbb{P}_1 = \mathbb{P}$, $t = \mathtt{x}_1$, $\gamma = \gamma_1$ and $s = s_1 = \varnothing$.

$$\begin{array}{rcll} [\![\mathtt{x}_1 : \mathbb{P}^{\#\varnothing} \vdash_\varnothing \mathtt{x}_1 : \mathbb{P}]\!]\langle \gamma \rangle & = & \mathbf{1}_{\mathbb{P}}\gamma_1 = \gamma_1 & \text{by the denotational semantics} \\ & \trianglelefteq_{\mathbb{P}_1} & v_1 & \text{by hypothesis} \\ & = & t[v] & \text{by hypothesis,} \end{array}$$

as required.

Weakening

$$\begin{array}{lll}
\multicolumn{3}{l}{[\![\Gamma, \mathtt{x}_{n+1} : \mathbb{P}_{n+1}{}^{\# s_{n+1}} \vdash_s t : \mathbb{P}]\!]\langle \gamma_1, \ldots, \gamma_{n+1}\rangle} \\
= & ([\![\Gamma \vdash_s t : \mathbb{P}]\!] \circ \pi_1)\langle \gamma_1, \ldots, \gamma_{n+1}\rangle & \text{by the denotational semantics} \\
= & [\![\Gamma \vdash_s t : \mathbb{P}]\!]\langle \gamma_1, \ldots, \gamma_n\rangle & \\
\trianglelefteq_{\mathbb{P}} & t[v'] & \text{by induction,}
\end{array}$$

where $[v']$ is the simultaneous substitution of v_i for x_i for $i \in \{1, \ldots, n\}$ only. However $t[v'] = t[v]$ since x_{n+1} does not appear free in t which yields the desired result.

Exchange

$$\begin{array}{lll}
\multicolumn{3}{l}{[\![\Gamma, \mathtt{x1} : \mathbb{Q}_1{}^{\# s_1}, \mathtt{x2} : \mathbb{Q}_2{}^{\# s_2}, \Lambda \vdash_s t : \mathbb{P}]\!]\langle \gamma, q_1, q_2, \lambda\rangle} \\
= & [\![\Gamma, \mathtt{x2} : \mathbb{Q}_2{}^{\# s_2}, \mathtt{x1} : \mathbb{Q}_1{}^{\# s_1}, \Lambda \vdash_s t : \mathbb{P}]\!]\langle \gamma, q_2, q_1, \lambda\rangle & \text{by den. sem.} \\
\trianglelefteq_{\mathbb{P}} & t[v'] & \text{by induction} \\
= & t[v] &
\end{array}$$

where $[v']$ is the substitution $[v]$ appropriately reordered, as required.

Fresh-Weakening Since the freshness conditions have been strengthened, the terms v_i satisfy the weaker conditions required for the induction hypothesis to hold. Therefore

$$\begin{array}{lll}
\multicolumn{3}{l}{[\![\Gamma, \mathtt{x} : \mathbb{Q}^{\# s'} \vdash_s t : \mathbb{P}]\!]\langle \gamma, q\rangle} \\
= & [\![\Gamma, \mathtt{x} : \mathbb{Q}^{\# s''} \vdash_s t : \mathbb{P}]\!]\langle \gamma, q\rangle & \text{from the denotational semantics} \\
\trianglelefteq_{\mathbb{P}} & t[v] & \text{by induction,}
\end{array}$$

as required.

Support-Weakening (Terms) Suppose that $s' \subseteq s$. Therefore

$$\begin{array}{rcll}
[\![\Gamma \vdash_s t : \mathbb{P}]\!]\langle \gamma\rangle & = & [\![\Gamma \vdash_{s'} t : \mathbb{P}]\!]\langle \gamma\rangle & \text{by the denotational semantics} \\
& \trianglelefteq_{\mathbb{P}} & t[v] & \text{by induction,}
\end{array}$$

as required.

Prefix Let $P \in [\![\Gamma \vdash_s \mathtt{!}t : !\mathbb{P}]\!]\langle \gamma\rangle$. By the denotational semantics therefore $P \subseteq [\![\Gamma \vdash_s t : \mathbb{P}]\!]\langle \gamma\rangle$. By induction $[\![\Gamma \vdash_s t : \mathbb{P}]\!]\langle \gamma\rangle \trianglelefteq_{\mathbb{P}} t[v]$ so that $P \trianglelefteq_{\mathbb{P}} t[v]$; by the operational semantics, $!\mathbb{P} : \mathtt{!}t[v] \xrightarrow{!} t[v]$ so that $P \in_{!\mathbb{P}} \mathtt{!}t[v]$ as required.

Recursion Let $p \in [\![\Gamma \vdash_s \texttt{rec}\,\texttt{x}.t : \mathbb{P}]\!]\langle\gamma\rangle$. By the denotational semantics therefore there exists $n \in \omega$ such that $p \in \big([\![\Gamma, \texttt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]^*\big)^n\langle\gamma\rangle$. By induction on n, each $\big([\![\Gamma, \texttt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]^*\big)^n\langle\gamma\rangle \trianglelefteq_{\mathbb{P}} \texttt{rec}\,\texttt{x}.t[v]$, as follows. Certainly $\big([\![\Gamma, \texttt{x} : \mathbb{P}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]^*\big)^0\langle\gamma\rangle = \varnothing \trianglelefteq_{\mathbb{P}} \texttt{rec}\,\texttt{x}.t[v]$. Suppose that the result is true at n, then

$$\begin{array}{rcll}
\big([\![t]\!]^*\big)^{n+1}\langle\gamma\rangle & = & [\![t]\!] \circ (\mathbf{1}_{[\![\Gamma]\!]} \,\&\, \big([\![t]\!]^*\big)^n) \circ \Delta_{[\![\Gamma]\!]}\langle\gamma\rangle & \\
& = & [\![t]\!] \circ (\mathbf{1}_{[\![\Gamma]\!]} \,\&\, \big([\![t]\!]^*\big)^n)\langle\gamma,\gamma\rangle & \\
& = & [\![t]\!]\langle\gamma, \big([\![t]\!]^*\big)^n\gamma\rangle & \\
& \trianglelefteq_{\mathbb{P}} & t[v][\texttt{rec}\,\texttt{x}.t[v]/\texttt{x}] & \text{by induction} \\
& \precsim_{\mathbb{P}} & \texttt{rec}\,\texttt{x}.t[v] & \text{by the op. semantics}
\end{array}$$

so that $p \in_{\mathbb{P}} \texttt{rec}\,\texttt{x}.t[v]$ as required.

Match Let $p \in [\![\Gamma, \Lambda^{\#s'} \vdash_s \texttt{[}u\,\texttt{>}\,q\texttt{(x:}\mathbb{Q}'\,\texttt{\#}\,s'\texttt{)}\,\texttt{=>}\,t\texttt{]} : \mathbb{P}]\!]\langle\gamma,\lambda\rangle$, then by the denotational semantics there exists $Q \in !\mathbb{Q}'$ such that

$$\begin{array}{rcl}
p & \in & [\![\Gamma, \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_s t : \mathbb{P}]\!]\langle\gamma, Q_\downarrow\rangle \\
\text{and } Q & \in & \big([\![\,\vdash_{s''} \mathbb{Q} : q : \mathbb{Q}']\!] \circ [\![\Lambda \vdash_{s''} u : \mathbb{Q}]\!]\big)\langle\lambda\rangle.
\end{array}$$

By induction on u, $[\![\Lambda \vdash_{s''} u : \mathbb{Q}]\!]\langle\lambda\rangle \trianglelefteq_{\mathbb{Q}} u[v]$ so that by induction on q there exists u' such that $\mathbb{Q} : u[v] \xrightarrow{q} u'$ and $Q \trianglelefteq_{\mathbb{Q}'} u'$. By induction on t it follows that $[\![\Gamma, \texttt{x} : \mathbb{Q}'^{\#s'} \vdash_s t : \mathbb{P}]\!]\langle\gamma, Q_\downarrow\rangle \trianglelefteq_{\mathbb{P}} t[u'/\texttt{x}][v]$ and by the operational semantics $t[u'/\texttt{x}][v] \precsim_{\mathbb{P}} \texttt{[}u\,\texttt{>}\,q\texttt{(x:}\mathbb{Q}'\,\texttt{\#}\,s'\texttt{)}\,\texttt{=>}\,t\texttt{]}[v]$ so that $p \in_{\mathbb{P}} \texttt{[}u\,\texttt{>}\,q\texttt{(x:}\mathbb{Q}'\,\texttt{\#}\,s'\texttt{)}\,\texttt{=>}\,t\texttt{]}[v]$ as required.

Function Abstraction Let $Q \mapsto p \in [\![\Gamma \vdash_s \lambda\,\texttt{x}.t : \mathbb{Q} \to \mathbb{P}]\!]\langle\gamma\rangle$, and let u be such that $Q \trianglelefteq_{\mathbb{Q}} u$. Then

$$\begin{array}{rcll}
p & \in & [\![\Gamma, \texttt{x} : \mathbb{Q}^{\#\varnothing} \vdash_s t : \mathbb{P}]\!]\langle\gamma, Q\rangle & \text{by the denotational semantics} \\
& \trianglelefteq_{\mathbb{P}} & t[v][u/\texttt{x}] & \text{by induction} \\
& \precsim_{\mathbb{P}} & \lambda\,\texttt{x}.t[v]\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)} &
\end{array}$$

so that $p \in_{\mathbb{P}} \lambda\,\texttt{x}.t[v]\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)}$ and hence $Q \mapsto p \in_{\mathbb{Q}\to\mathbb{P}} \lambda\,\texttt{x}.t[v]$ as required.

Function Application Let $p \in [\![\Gamma, \Lambda \vdash_s t\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)} : \mathbb{P}]\!]\langle\gamma,\lambda\rangle$, then by the denotational semantics there is $Q \in !\mathbb{Q}$ such that $Q \mapsto p \in [\![\Gamma \vdash_s t : \mathbb{Q} \to \mathbb{P}]\!]\langle\gamma\rangle$ and $Q \subseteq [\![\Lambda \vdash_s u : \mathbb{Q}]\!]\langle\lambda\rangle$. By induction on u, $[\![\Lambda \vdash_s u : \mathbb{Q}]\!]\langle\lambda\rangle \trianglelefteq_{\mathbb{Q}} u[v]$ so that $Q \trianglelefteq_{\mathbb{Q}} u[v]$. By induction on t, $[\![\Gamma \vdash_s t : \mathbb{Q} \to \mathbb{P}]\!]\langle\gamma\rangle \trianglelefteq_{\mathbb{Q}\to\mathbb{P}} t[v]$ so that $Q \mapsto p \in_{\mathbb{Q}\to\mathbb{P}} t[v]$ and hence $p \in_{\mathbb{P}} t[v]\texttt{(}u[v]\texttt{:}\mathbb{Q}\texttt{)} = t\texttt{(}u\texttt{:}\mathbb{Q}\texttt{)}[v]$ as required.

Name Abstraction Let $\texttt{new}\,b.\,p \in \llbracket \Gamma \vdash_s \texttt{new}\,a\,.\,t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma$, where b is fresh, then by the denotational semantics $p \in \llbracket \Gamma^{\# b} \vdash_{s \dot\cup \{b\}} (ab)\cdot t : \mathbb{P} \rrbracket \langle\gamma\rangle_{\Gamma^{\# b}}$. Since b was fresh,

$$\begin{array}{lll} p \in_{\mathbb{P}} & ((ab)\cdot t)[v] & \text{by induction} \\ \underset{\sim}{\sqsubset}_{\mathbb{P}} & \big(\texttt{new}\,b.\,((ab)\cdot t)[v]\big)\,\texttt{[}b\texttt{]} & \text{by the operational semantics} \\ = & \big(\texttt{new}\,b.\,((ab)\cdot t)\big)[v]\,\texttt{[}b\texttt{]} & \text{as } b \# v \\ = & (\texttt{new}\,a\,.\,t)[v]\,\texttt{[}b\texttt{]} & \text{by } \alpha\text{-equivalence} \end{array}$$

so that $p \in_{\mathbb{P}} (\texttt{new}\,a\,.\,t)[v]\,\texttt{[}b\texttt{]}$ and hence $\texttt{new}\,b.\,p \in_{\delta\mathbb{P}} (\texttt{new}\,a\,.\,t)[v]$ as required.

Name Application Let $p \in \llbracket \Gamma^{\# a} \vdash_{s \dot\cup \{a\}} t\,\texttt{[}a\texttt{]} : \mathbb{P} \rrbracket \langle\gamma\rangle_{\Gamma^{\# a}}$ then by the denotational semantics $\texttt{new}\,a.\,p \in \llbracket \Gamma \vdash_s t : \delta\mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma$ and hence by induction $\texttt{new}\,a.\,p \in_{\delta\mathbb{P}} t[v]$. Let b be a fresh name, then $\texttt{new}\,a.\,p = \texttt{new}\,b.\,((ab)\cdot p)$, so that $(ab)\cdot p \in_{\delta\mathbb{P}} t[v]\,\texttt{[}b\texttt{]} = t\,\texttt{[}b\texttt{]}\,[v]$. By assumption $a \# t$ and $a \# v$ so that by equivariance $p \in_{\delta\mathbb{P}} t\,\texttt{[}a\texttt{]}\,[v]$ as required.

Labelling Let $\ell{:}p \in \llbracket \Gamma \vdash_s \ell_0{:}t : \bigoplus_{\ell\in L}\mathbb{P}_\ell \rrbracket \langle\gamma\rangle_\Gamma$, then by the denotational semantics $\ell = \ell_0$ and $p \in \llbracket \Gamma \vdash_s t : \mathbb{P}_{\ell_0} \rrbracket$. By induction $\llbracket \Gamma \vdash_s t : \mathbb{P}_\ell \rrbracket \langle\gamma\rangle \trianglelefteq_{\mathbb{P}_{\ell_0}} t[v]$ and by the operational semantics $t[v] \underset{\sim}{\sqsubset}_{\mathbb{P}_{\ell_0}} \pi_{\ell_0}\ell_0{:}t$ so that $p \in_{\mathbb{P}_{\ell_0}} \pi_{\ell_0}\ell_0{:}t$ and hence $\ell{:}p \in_{\bigoplus_{\ell\in L}\mathbb{P}_\ell} \ell_0{:}t[v]$ as required.

Label Projection Let $p \in \llbracket \Gamma \vdash_s \pi_{\ell_0} t : \mathbb{P}_{\ell_0} \rrbracket \langle\gamma\rangle_\Gamma$, then by the denotational semantics $\ell_0{:}p \in \llbracket \Gamma \vdash_s t : \bigoplus_{\ell\in L}\mathbb{P}_\ell \rrbracket \langle\gamma\rangle_\Gamma$. Also by induction it follows that $\llbracket \Gamma \vdash_s t : \bigoplus_{\ell\in L}\mathbb{P}_\ell \rrbracket \gamma \trianglelefteq_{\bigoplus_{\ell\in L}\mathbb{P}_\ell} t[v]$ so that $\ell_0{:}p \in_{\bigoplus_{\ell\in L}\mathbb{P}_\ell} t[v]$ and hence as required $p \in_{\mathbb{P}_{\ell_0}} \pi_{\ell_0} t[v]$.

Nondeterministic Sum Let $p \in \llbracket \Gamma \vdash_s \sum_{i\in I} t_i : \mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma$, then by the denotational semantics there exists $i_0 \in I$ such that $p \in \llbracket \Gamma \vdash_{s_{i_0}} t_{i_0} : \mathbb{P} \rrbracket \langle\gamma\rangle_\Gamma$. By induction $p \in_{\mathbb{P}} t_{i_0}[v]$ and by the operational semantics $t_{i_0} \underset{\sim}{\sqsubset}_{\mathbb{P}} \sum_{i\in I} t_i$ so that $p \in_{\mathbb{P}} \sum_{i\in I} t_i$ as required.

Recursive Type Folding Let $\texttt{abs}\ p \in \llbracket \Gamma \vdash_s \texttt{abs}\,t : \mu_j \vec{P}.\,\vec{\mathbb{P}} \rrbracket \langle\gamma\rangle_\Gamma$, then by the denotational semantics $p \in \llbracket \Gamma \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] \rrbracket \langle\gamma\rangle_\Gamma$. By induction $\llbracket \Gamma \vdash_s t : \mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}] \rrbracket \langle\gamma\rangle_\Gamma \trianglelefteq_{\mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]} t[v]$ and by the operational semantics $t[v] \underset{\sim}{\sqsubset}_{\mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]} \texttt{rep}\,\texttt{abs}\,t$ so that $p \in_{\mathbb{P}_j[\mu\vec{P}.\,\vec{\mathbb{P}}/\vec{P}]} \texttt{rep}\,\texttt{abs}\,t$ and hence it follows that $\texttt{abs}\ p \in_{\mu_j\vec{P}.\,\vec{\mathbb{P}}} \texttt{abs}\,t[v]$ as required.

Recursive Type Unfolding Let $p \in \llbracket \Gamma \vdash_s \mathtt{rep}\, t : \mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}] \rrbracket \langle \gamma \rangle_\Gamma$, then by the denotational semantics $\mathtt{abs}\ p \in \llbracket \Gamma \vdash_s t : \mu_j \vec{P}.\, \vec{\mathbb{P}} \rrbracket \langle \gamma \rangle_\Gamma$. By induction $\llbracket \Gamma \vdash_s t : \mu_j \vec{P}.\, \vec{\mathbb{P}} \rrbracket \langle \gamma \rangle_\Gamma \trianglelefteq_{\mu_j \vec{P}.\, \vec{\mathbb{P}}} t[v]$ so that $\mathtt{abs}\, p \in_{\mu_j \vec{P}.\, \vec{\mathbb{P}}} t[v]$ and hence $p \in_{\mathbb{P}_j[\mu \vec{P}.\, \vec{\mathbb{P}}/\vec{P}]} \mathtt{rep}\, t[v]$ as required.

Prefix Action Suppose that $X \trianglelefteq_{!\mathbb{P}} t$ and $P \in \llbracket \vdash_s !\mathbb{P} : \mathtt{!} : \mathbb{P} \rrbracket X = X$, then $P \in_{!\mathbb{P}} t$ and hence there exists t' such that $P \trianglelefteq_{\mathbb{P}} t'$ and $!\mathbb{P} : t \xrightarrow{!} t'$ as required.

Higher-Order Action Let $X' = \{p' \mid \exists Q \subseteq \llbracket \vdash_s u : \mathbb{Q} \rrbracket.\ Q \mapsto p' \in X\}$ then by the denotational semantics $\llbracket \vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}' \rrbracket X = \llbracket \vdash_s \mathbb{P} : p : \mathbb{P}' \rrbracket X'$. Also, if $p' \in X'$ then there is $Q \subseteq \llbracket \vdash_s u : \mathbb{Q} \rrbracket$ such that $Q \mapsto p' \in X$. By induction, $Q \subseteq \llbracket \vdash_s u : \mathbb{Q} \rrbracket$ implies that $Q \trianglelefteq_{\mathbb{Q}} u$, and $X \trianglelefteq_{\mathbb{Q} \to \mathbb{P}} t$ by hypothesis so $Q \mapsto p' \in_{\mathbb{Q} \to \mathbb{P}} t$ and hence $p' \in_{\mathbb{P}} t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)}$. Therefore $X' \trianglelefteq_{\mathbb{P}} t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)}$. Thus by induction if $P \in \llbracket \vdash_s \mathbb{Q} \to \mathbb{P} : u \mapsto p : \mathbb{P}' \rrbracket X = \llbracket \vdash_s \mathbb{P} : p : \mathbb{P}' \rrbracket X'$ then there exists t' such that $\mathbb{P} : t\mathtt{(}u\mathtt{:}\mathbb{Q}\mathtt{)} \xrightarrow{p} t'$ and $P \trianglelefteq_{\mathbb{P}'} t'$ and so by lemma 4.4.1.4(i) $\mathbb{Q} \to \mathbb{P} : t \xrightarrow{u \mapsto p} t'$ as required.

Labelled Action Let $X' = \{p' \mid \ell_0 : p' \in X\}$ then by the denotational semantics $\llbracket \vdash_s \bigoplus_{\ell \in L} \mathbb{P}_\ell : \ell_0 : p : \mathbb{P}' \rrbracket X = \llbracket \vdash_s \mathbb{P}_{\ell_0} : p : \mathbb{P}' \rrbracket X'$ and if $X \trianglelefteq_{\bigoplus_{\ell \in L} \mathbb{P}_\ell} t$ then $X' \trianglelefteq_{\mathbb{P}_{\ell_0}} \pi_{\ell_0} t$. Therefore by induction there exists t' such that $\mathbb{P}_{\ell_0} : \pi_{\ell_0} t \xrightarrow{p} t'$ and $P \trianglelefteq_{\mathbb{P}'} t'$ and hence by lemma 4.4.1.4(ii) $\bigoplus_{\ell \in L} \mathbb{P}_\ell : t \xrightarrow{\ell_0 : p} t'$ as required.

New Name Action The induction premises are that $\vdash_s \delta\mathbb{P} : \mathtt{new}\, a.\, p : \delta\mathbb{P}'$, $X \trianglelefteq_{\delta\mathbb{P}} t$ and $P \in \llbracket \vdash_s \delta\mathbb{P} : \mathtt{new}\, a.\, p : \delta\mathbb{P}' \rrbracket X$. By the denotational semantics, $P \in \llbracket \vdash_s \delta\mathbb{P} : \mathtt{new}\, a.\, p : \delta\mathbb{P}' \rrbracket X$ implies that $P = \theta^!_{\mathbb{P}'} P'$ and for fresh b it is the case that $P'@b \in \llbracket \vdash_{s \dot{\cup} \{b\}} \mathbb{P} : (ab) \cdot p : \mathbb{P}' \rrbracket X'$ where $X' = \{x \mid \mathtt{new}\, b.\, x \in X\}$.

Let $x' \in X'$, then $\mathtt{new}\, b.\, x' \in X \trianglelefteq_{\delta\mathbb{P}} t$ so that $x' \in_{\mathbb{P}} t\mathtt{[}b\mathtt{]}$. Therefore $X' \trianglelefteq_{\mathbb{P}} t\mathtt{[}b\mathtt{]}$; also $P'@b \in \llbracket \vdash_{s \dot{\cup} \{b\}} \mathbb{P} : (ab) \cdot p : \mathbb{P}' \rrbracket X'$ and $\vdash_{s \dot{\cup} \{b\}} \mathbb{P} : (ab) \cdot p : \mathbb{P}'$ so that by induction there exists t' such that $\mathbb{P} : t\mathtt{[}b\mathtt{]} \xrightarrow{(ab) \cdot p} t'$ and $P'@b \trianglelefteq_{\mathbb{P}'} t'$. By lemma 4.4.1.4(iii), $\mathbb{P} : t \xrightarrow{\mathtt{new}\, b.\, (ab) \cdot p} \mathtt{new}\, b\mathtt{.}t'$, and by α-equivalence it is the case that $\mathtt{new}\, b.\, (ab) \cdot p = \mathtt{new}\, a.\, p$ so that finally $\mathbb{P} : t \xrightarrow{\mathtt{new}\, a.\, p} \mathtt{new}\, b\mathtt{.}t'$.

It remains to show that $P \trianglelefteq_{\delta\mathbb{P}'} \mathtt{new}\, b\mathtt{.}t'$. Let $x \in P$ and recall from above that $P = \theta^!_{\mathbb{P}'} P' = \{x \mid \mathbf{fresh}\, b\, \mathbf{in}\, x@b \in P'@b\}$. Let b be fresh, then it follows that $x = \mathtt{new}\, b.\, x'$ and $x' \in P'@b$. Furthermore, $P'@b \trianglelefteq_{\mathbb{P}'} t'$ so that $x' \in_{\mathbb{P}'} t'$. By the operational semantics, $t' \mathrel{\underset{\sim}{\sqsubset}}_{\mathbb{P}'} \mathtt{new}\, b\mathtt{.}t'\mathtt{[}b\mathtt{]}$ so that $x' \in_{\mathbb{P}'} \mathtt{new}\, b\mathtt{.}t'\mathtt{[}b\mathtt{]}$ and hence $x \in_{\delta\mathbb{P}'} \mathtt{new}\, b\mathtt{.}t'$, as required.

Recursive Type Action Let

$$X' = \{p' \mid \texttt{abs}\ p' \in X\}$$

then $[\![\vdash_s \mu_j \vec{P}.\ \vec{\mathbb{P}} : \texttt{abs}\ p : \mathbb{P}']\!] X = [\![\vdash_s \mathbb{P}_j[\mu \vec{P}.\ \vec{\mathbb{P}}/\vec{P}] : p : \mathbb{P}']\!] X'$ and $X \trianglelefteq_{\mu_j \vec{P}.\ \vec{\mathbb{P}}} t$ implies that $X' \trianglelefteq_{\mathbb{P}_j[\mu \vec{P}.\ \vec{\mathbb{P}}/\vec{P}]} \texttt{rep}\, t$. Therefore by induction there exists t' such that $\mathbb{P}_j[\mu \vec{P}.\ \vec{\mathbb{P}}/\vec{P}] : \texttt{rep}\, t \xrightarrow{p} t'$ and $P \trianglelefteq_{\mathbb{P}'} t'$ and hence by lemma 4.4.1.4(iv), $\mu_j \vec{P}.\ \vec{\mathbb{P}} : t \xrightarrow{\texttt{abs}\ p} t'$ as required.

Support-Weakening (Actions) This case is clear from the given denotational semantics. □

It immediately follows that the denotation $[\![t]\!]$ consists of paths that the term t perform.

5.4.3.3 Corollary. *Suppose* $\vdash_s t : \mathbb{P}$*. Then* $[\![\vdash_s t : \mathbb{P}]\!] \trianglelefteq_{\mathbb{P}} t$*.*

Proof. Let Γ be the empty environment and apply lemma 5.4.3.1. □

It is now possible to demonstrate the main theorem of this thesis, namely the computational adequacy of the given semantics of Nominal HOPLA with respect to observations of `!` actions.

5.4.3.4 Theorem (Adequacy). $[\![\vdash t : !\mathbb{P}]\!] = \varnothing$ *if and only if there exists no* t' *such that* $!\mathbb{P} : t \xrightarrow{!} t'$*.*

Proof. Suppose that there exists a t' such that $!\mathbb{P} : t \xrightarrow{!} t'$. Then by soundness $[\![\vdash_s \texttt{!}t' : !\mathbb{P}]\!] \sqsubseteq [\![\vdash_s !\mathbb{P} : \texttt{!} : \mathbb{P}]\!] \circ [\![\vdash_s t : !\mathbb{P}]\!] = [\![\vdash t : !\mathbb{P}]\!]$. But $[\![\texttt{!}t']\!] = \eta_{\mathbb{P}} \circ [\![t']\!]$ and $\eta_{\mathbb{P}}(X) \neq \varnothing$ for any $X \subseteq_{\downarrow} \mathbb{P}$ and therefore $[\![\vdash t : !\mathbb{P}]\!] \neq \varnothing$ as required.

Conversely, suppose $[\![\vdash t : !\mathbb{P}]\!] \neq \varnothing$. Then since $[\![\vdash t : !\mathbb{P}]\!] \subseteq_{\downarrow} !\mathbb{P}$, $\varnothing \in [\![\vdash t : !\mathbb{P}]\!]$. Hence by lemma 5.4.3.3, $\varnothing \in_{!\mathbb{P}} t$ and so there is a t' such that $!\mathbb{P} : t \xrightarrow{!} t'$ as required. □

As discussed in the introduction to 5.4, the observation of general actions of the form $\vdash \mathbb{P} : p : \mathbb{P}'$ may be reduced to the observation of primitive actions of the form $\vdash !\mathbb{P}' : \texttt{!} : \mathbb{P}'$ by the matching operator `[`t `>` p`(x:`$\mathbb{P}'$ `#` s`) => !nil]`. Because of this, a consequence of theorem 5.4.3.4 is that two terms with equal denotations must also be operationally indistinguishable. This could be summarised by a slogan saying that *denotational equivalence implies contextual equivalence*, although the details of the definition of contextual equivalence are not included here. In the original version of HOPLA[20] contextual equivalence coincided precisely with denotational equivalence: HOPLA is fully abstract. This full abstraction property is well-known for its fragility and in particular it is not a property of Nominal HOPLA as shown in section 7.1.1 below.

Chapter 6

A Universal View

This chapter introduces a more universal approach to the nominal domain theory introduced above. It provides a justification that the definition of the semantics of Nominal HOPLA is the result of sensible and canonical choices.

In 6.1 it is shown that the collection $(\mathbf{FMPre}_s)_{s \subseteq_{\text{fin}} \mathbb{A}}$ of categories arises in a natural way from a dependent type theory in $\mathbf{FMPre}_\varnothing$. The category $\mathbf{FMPre}_\varnothing$ can be seen to be similar to **NPre** or equivalently the category of sheaves in $\mathbf{Pre}^{\mathbb{I}}$, which shows that the set-theoretical foundations of the $(\mathbf{FMPre}_s)_{s \subseteq_{\text{fin}} \mathbb{A}}$ are not a key factor in this discussion. The replacement of $(\mathbf{FMPre}_s)_{s \subseteq_{\text{fin}} \mathbb{A}}$ with a dependent type theory in the more canonical **NPre** is not studied in detail, but it is shown here that key constructions in $(\mathbf{FMPre}_s)_{s \subseteq_{\text{fin}} \mathbb{A}}$ have a categorical — as well as a set-theoretic — description. In particular, it is shown that the adjunction $(-)^{\# a} \dashv \delta_a : \mathbf{FMPre}_s \leftrightarrows \mathbf{FMPre}_{s \dot{\cup} \{a\}}$ of 3.2.1.15 arises from the adjunction $(-) \otimes \mathbb{A} \dashv \delta : \mathbf{FMPre}_\varnothing \leftrightarrows \mathbf{FMPre}_\varnothing$ in a purely categorical fashion, which justifies the simple set-theoretic definitions of $(-)^{\# a}$ and δ_a in 3.2.1.2 and 3.2.1.6.

Then, since $\mathbf{FMPre}_s(\mathbb{P}, \widehat{\mathbb{Q}}) \cong \mathbf{FMLin}_s(\mathbb{P}, \mathbb{Q})$ by 3.3.2.1 it follows that $\mathbf{FMLin}_s$ is isomorphic to the Kleisli category of the monad $\widehat{(-)} : \mathbf{FMPre}_s \to \mathbf{FMPre}_s$. Section 6.2 develops some abstract machinery regarding adjunctions on Kleisli categories, and then section 6.3 uses this machinery to lift the binding adjunction $(-)^{\# a} \dashv \delta_a : \mathbf{FMPre}_s \leftrightarrows \mathbf{FMPre}_{s \dot{\cup} \{a\}}$ to the Kleisli category, and demonstrates that the result of this abstract lifting process coincides with the adjunction $(-)^{\# a+} \dashv \delta_a^+ : \mathbf{FMLin}_s \leftrightarrows \mathbf{FMLin}_{s \dot{\cup} \{a\}}$ of 3.3.5.9.

By dualising this argument, since $\mathbf{FMLin}_s(!\mathbb{P}, \mathbb{Q}) \cong \mathbf{FMCts}_s(\mathbb{P}, \mathbb{Q})$ by 3.4.4.15 it follows that $\mathbf{FMCts}_s$ is isomorphic to the coKleisli category of the comonad $! : \mathbf{FMLin}_s \to \mathbf{FMLin}_s$ so that in section 6.4 it is shown that that the abstract machinery of section 6.2 lifts the adjunction $(-)^{\#a+} \dashv \delta_a^+$ on the FM-linear categories to the adjunction $(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s\dot{\cup}\{a\}}$.

To summarise, the process described above is as follows and each step can be described very abstractly.

$$\begin{tikzcd}
(-) \otimes \mathbb{A} \dashv \delta : \mathbf{FMPre}_\varnothing \leftrightarrows \mathbf{FMPre}_\varnothing \arrow[d, "\text{section 6.1.2}"] \\
(-)^{\#a} \dashv \delta_a : \mathbf{FMPre}_s \leftrightarrows \mathbf{FMPre}_{s\dot{\cup}\{a\}} \arrow[d, "\text{section 6.3}"] \\
(-)^{\#a+} \dashv \delta_a^+ : \mathbf{FMLin}_s \leftrightarrows \mathbf{FMLin}_{s\dot{\cup}\{a\}} \arrow[d, "\text{section 6.4}"] \\
(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s\dot{\cup}\{a\}}
\end{tikzcd}$$

In short, the key structure on the FM-continuous categories that makes them a suitable setting for a domain theory that supports name generation (i.e. the adjunction $(-)^{\#a++} \dashv \delta_a^{++}$) has a canonical description. It is therefore hoped that this chapter points a way towards a more general theory of semantics with names.

6.1 $\mathbf{FMPre}_s$ in Dependent Type Theory

The purist may wonder whether it is necessary that the foundations of this dissertation need to be as precisely specified as the cumulative hierarchy $\mathcal{V}_{\mathrm{FM}}$ of FM sets. After all, **NSet** can be constructed as $Sh(\mathbf{Set}^{\mathbb{I}})$ where **Set** is some category of sets, not necessarily one with a cumulative hierarchy out of which one can construct $\mathcal{V}_{\mathrm{FM}}$. Nor does **NSet** insist that the permutation action on any particular set is given by $\in$-recursion as FM set theory does. More practically, conventional domain theory does not really require all the fiddly details of ZF set theory, since it can be phrased in a much more universal fashion than by talking of sets and elements. Such concerns are justified. The use of FM set theory here is solely to give a semantics to a rich enough type theory for the purposes of this discussion. A further advantage is that the language of FM set theory is close enough to that of conventional set theories that the presentation given in this dissertation should be accessible to conventional domain theorists.

This section gives a flavour of a less prescriptive foundational basis for this thesis than $\mathcal{V}_{\mathrm{FM}}$. The theory is not developed fully here as to do so is beyond the scope of this dissertation, but it is intended that there is enough information here to persuade the purist that this work is applicable more generally it might first appear.

Secretly, this discussion has been taking place in a dependent type theory. Nominal sets supports a very rich dependent type theory as studied by Schöpp and Stark[26], but for the sake of clarity it is preferable to consider a simpler setup here. Roughly speaking, the available types are dependent on a 'current' set of names: if the current set of names is s then the available types are objects of $\mathbf{FMPre}_s$. The current set of names can change, by inserting a new name or by permuting the current names, and in general if $i : s \rightarrowtail s'$ is an arrow of $\mathbb{I}$ then the current set of names can be changed from s to s' using i.

6.1.1 A Fibration

In order to give this account in terms of dependent type theory, the discussion above indicates that a fibration over $\mathbb{I}^{\mathrm{op}}$ should be sought. Recall from lemma 2.2.4.2 that each nominal preorder can be seen as a pullback-preserving functor from $\mathbb{I}$ to $\mathbf{Pre}$. The Yoneda lemma therefore embeds a copy of $\mathbb{I}^{\mathrm{op}}$ in $\mathbf{NPre}$, taking each finite set s to the functor $ys =_{\mathrm{def}} \mathbb{I}(s, -)$. Following through the equivalence of lemma 2.2.4.2, the nominal preorder ys consists of all injections $s \rightarrowtail \mathbb{A}$ ordered with the discrete order, but note carefully that the permutation action is given by post-composition and not conjugation. Therefore each element $i : s \rightarrowtail \mathbb{A}$ of ys has support given by $\{ia \mid a \in s\}$. Isomorphically, ys consists of all s-ary tuples of distinct names, i.e.

$$ys \cong \underbrace{\mathbb{A} \otimes \ldots \otimes \mathbb{A}}_{|s| \text{ times}} \tag{6.1.1.1}$$

where the isomorphism is specified by the choice of an order of the names in s. If $|s| > 1$ then there are many such isomorphisms. There is also an important distinguished element of ys: the inclusion $i_s : s \hookrightarrow \mathbb{A}$, which has support s.

There is a similarly-defined embedding $y : \mathbb{I}^{\mathrm{op}} \to \mathbf{FMPre}_\varnothing$, and the codomain fibration on $(\mathbf{FMPre}_\varnothing \downarrow y)$ gives rise to the desired dependent type theory as follows.

6.1.1.2 Lemma. *For all finite sets of names s there is an equivalence of categories $\mathbf{FMPre}_s \simeq \mathbf{FMPre}_\varnothing / ys$.*

Proof. Let X be an object of $\mathbf{FMPre}_s$ and define

$$\overline{X} =_{\mathrm{def}} \bigcup_{i \in ys} \{i\} \times (\sigma_i \cdot X), \tag{6.1.1.3}$$

with the product order, where σ_i is any permutation extending i. This is well-defined: if σ_i and σ'_i both extend i then $\sigma_i^{-1}\sigma'_i \mathrel{\#} s$ and since X is supported by s it follows that $X = \sigma_i^{-1} \cdot \sigma'_i \cdot X$ so that $\sigma_i \cdot X = \sigma'_i \cdot X$ as required. Furthermore if σ is any permutation and $\langle i, x\rangle \in \overline{X}$ then $x \in \sigma_i \cdot X$ and $\sigma \cdot \langle i, x\rangle = \langle \sigma \circ i, \sigma \cdot x\rangle$. However, $\sigma \cdot x \in \sigma \cdot \sigma_i \cdot X = \sigma_{\sigma\circ i} \cdot X$ so that $\sigma \cdot \langle i, x\rangle \in \overline{X}$. Therefore $\overline{X} = \sigma \cdot \overline{X}$ and hence $\overline{X}$ is an object of $\mathbf{FMPre}_\varnothing$.

Define

$$F_s X =_{\text{def}} \langle \overline{X}, \pi_1\rangle. \tag{6.1.1.4}$$

Note that $\pi_1 : \overline{X} \to ys$ is equivariant and monotone so that $F_s X$ is an object of $\mathbf{FMPre}_\varnothing/ys$.

Let $f : X \to Y$ be an arrow of $\mathbf{FMPre}_s$ and define $\overline{f} : \overline{X} \to \overline{Y}$ by

$$\overline{f}\langle i, x\rangle =_{\text{def}} \langle i, (\sigma_i \cdot f)x\rangle, \tag{6.1.1.5}$$

where σ_i is any permutation extending i. This is well-defined: if σ_i and σ'_i both extend i then $\sigma_i^{-1}\sigma'_i \mathrel{\#} s$ and since f is supported by s it follows that $f = \sigma_i^{-1} \cdot \sigma'_i \cdot f$ so that $(\sigma_i \cdot f)x = (\sigma'_i \cdot f)x$ as required. Also, if $\langle i, x\rangle \in \overline{X}$ then $x \in \sigma_i \cdot X$ and hence $(\sigma_i \cdot f)x \in \sigma_i \cdot Y$ so that $\overline{f}\langle i, x\rangle \in \overline{Y}$ as required. Furthermore if σ is any permutation and $\langle i, x\rangle \in \overline{X}$ then

$$\begin{aligned}
\sigma \cdot \overline{f}\langle i, x\rangle &= \sigma \cdot \langle i, (\sigma_i \cdot f)x\rangle \\
&= \langle \sigma \circ i, (\sigma \cdot \sigma_i \cdot f)(\sigma \cdot x)\rangle \\
&= \langle \sigma \circ i, (\sigma_{\sigma\circ i} \cdot f)(\sigma \cdot x)\rangle \\
&= \overline{f}\langle \sigma \circ i, \sigma \cdot x\rangle \\
&= \overline{f}(\sigma \cdot \langle i, x\rangle).
\end{aligned} \tag{6.1.1.6}$$

This shows that $\overline{f}$ is equivariant, and it is clear that $\overline{f}$ is monotone, so that $\overline{f}$ is an arrow of $\mathbf{FMPre}_\varnothing$. Furthermore $\pi_1 \circ \overline{f} = \pi_1$ so that it is possible to define

$$F_s f =_{\text{def}} \overline{f} : F_s X \to F_s Y \tag{6.1.1.7}$$

as an arrow of $\mathbf{FMPre}_\varnothing/ys$. It is now not hard to see that F_s so defined is a functor $\mathbf{FMPre}_s \to \mathbf{FMPre}_\varnothing/ys$.

Let $\langle X, f\rangle$ be an object of $\mathbf{FMPre}_\varnothing/ys$ and define

$$G_s\langle X, f\rangle =_{\text{def}} \{x \in X \mid f(x) = i_s\} = f^{-1}\{i_s\}. \tag{6.1.1.8}$$

If $\sigma \mathrel{\#} s$ is a permutation then $\sigma \cdot f^{-1}\{i_s\} = f^{-1}\{\sigma \circ i_s\} = f^{-1}\{i_s\}$ so that s supports $G_s\langle X, f\rangle$ and hence $G_s\langle X, f\rangle$ is an object of $\mathbf{FMPre}_s$.

Let $h : \langle X, f\rangle \to \langle Y, g\rangle$ be an arrow of $\mathbf{FMPre}_\varnothing/ys$, then $h : X \to Y$ is an arrow of $\mathbf{FMPre}_\varnothing$ such that $f = g \circ h$. Define

$$G_s h =_{\text{def}} h \Big|_{G_s\langle X, f\rangle}. \tag{6.1.1.9}$$

If $x \in G_s\langle X, f\rangle$ then $f(x) = i_s$ and $G_sh(x) = h(x)$, but it is also the case that $(g \circ G_sh)(x) = (g \circ h)(x) = f(x) = i_s$ so that $G_sh(x) \in G_s\langle Y, g\rangle$ and hence $G_sh : G_s\langle X, f\rangle \to G_s\langle Y, f\rangle$. If $\sigma \# s$ is a permutation and $x \in G_s\langle X, f\rangle$ then $\sigma \cdot (G_sh(x)) = \sigma \cdot (h(x)) = h(\sigma \cdot x) = G_sh(\sigma \cdot x)$ so that s supports G_sh. It is clear that G_sh is monotone too, and hence that G_sh is an arrow of $\mathbf{FMPre}_s$. It is not hard to see that G_s so defined is a functor $\mathbf{FMPre}_\varnothing/ys \to \mathbf{FMPre}_s$.

If X is an object of $\mathbf{FMPre}_s$ then there is an isomorphism $G_sF_sX \cong X$ as follows. Let $\langle i, x\rangle \in G_sF_sX$, then $i = \pi_1\langle i, x\rangle = i_s$ and hence $x \in X$. Conversely, if $x \in X$ then $\langle i_s, x\rangle \in G_sF_sX$. It is not hard to see that this relationship is monotone, supported by s, and natural in X.

If $\langle X, f\rangle$ is an object of $\mathbf{FMPre}_\varnothing/ys$ then define $\beta : F_sG_s\langle X, f\rangle \to \langle X, f\rangle$ by $\beta\langle i, x\rangle =_{\text{def}} x$. If $\langle i, x\rangle \in \overline{G_s\langle X, f\rangle}$ then

$$x \in \sigma_i \cdot G_s\langle X, f\rangle = \sigma_i \cdot f^{-1}\{i_s\} = f^{-1}\{\sigma_i \circ i_s\} \tag{6.1.1.10}$$

and hence $(f \circ \beta)\langle i, x\rangle = f(x) = \sigma_i \circ i_s = i = \pi_1\langle i, x\rangle$. Also, β is clearly an equivariant monotone map, so it is an arrow of $\mathbf{FMPre}_\varnothing/ys$ as required. Conversely, define β^{-1} by $\beta^{-1}(x) =_{\text{def}} \langle f(x), x\rangle$. Then

$$\begin{array}{rcl} x & \in & f^{-1}\{f(x)\} \\ & = & f^{-1}\{\sigma_{f(x)} \circ i_s\} \\ & = & \sigma_{f(x)} \cdot f^{-1}\{i_s\} \\ & = & \sigma_{f(x)} \cdot G_s\langle X, f\rangle \end{array} \tag{6.1.1.11}$$

and hence $\beta^{-1}(x) \in \overline{G_s\langle X, f\rangle}$. Furthermore β^{-1} is clearly equivariant and monotone and $\pi_1 \circ \beta^{-1} = f$ so that β^{-1} is an arrow of $\mathbf{FMPre}_\varnothing/ys$ as required.

Finally, β and β^{-1} are readily seen to be mutual inverses, and their definitions are natural in $\langle X, f\rangle$, which completes the equivalence of $\mathbf{FMPre}_\varnothing/ys$ and $\mathbf{FMPre}_s$ as required. □

Write cod for the forgetful 'codomain' functor $\text{cod} : (\mathbf{FMPre}_\varnothing \downarrow y) \to \mathbb{I}^{\text{op}}$.

6.1.1.12 Lemma. *An arrow $\langle h, i\rangle : \langle X, f\rangle \to \langle Y, g\rangle$ of $(\mathbf{FMPre}_\varnothing \downarrow y)$ is Cartesian over $i : s \rightarrowtail s'$ (with respect to* cod*) if the following diagram is a pullback square.*

$$\begin{tikzcd} X \arrow[r, "h"] \arrow[d, "f"'] \arrow[dr, phantom, "\lrcorner", very near start] & Y \arrow[d, "g"] \\ ys' \arrow[r, "-\circ i"'] & ys \end{tikzcd}$$

Proof. By definition of Cartesianness. □

6.1.1.13 Lemma. *The functor* cod : $(\mathbf{FMPre}_\varnothing \downarrow y) \to \mathbb{I}^{\mathrm{op}}$ *is a cloven fibration, with cleavage given by the usual choice of pullbacks.*

Proof. Using lemma 6.1.1.12 and the fact that $\mathbf{FMPre}_\varnothing$ has pullbacks. □

If $i : s \rightarrowtail s'$ is an arrow of $\mathbb{I}$ then there is a corresponding reindexing functor $i^* : \mathbf{FMPre}_\varnothing/ys \to \mathbf{FMPre}_\varnothing/ys'$ given by pullback. If σ_i is a permutation that extends i then this reindexing corresponds to the permutation functor $\sigma_i : \mathbf{FMPre}_s \to \mathbf{FMPre}_{s'}$ via the equivalence of lemma 6.1.1.2 as follows. Let $\langle X, f\rangle$ be an object of $\mathbf{FMPre}_\varnothing/ys$, then

$$\begin{aligned} G_{s'} i^* \langle X, f\rangle &= G_{s'}\langle X \times_{ys} ys', \pi_2\rangle \\ &= \{\langle x, i'\rangle \in X \times ys' \mid f(x) = i' \circ i \wedge i' = i_{s'}\} \\ &\cong \{x \in X \mid f(x) = i_{s'} \circ i\} \\ &= \sigma_i \cdot \{x \in X \mid f(\sigma_i \cdot x) = i_{s'} \circ i\} \\ &= \sigma_i \cdot \{x \in X \mid f(x) = \sigma_i^{-1} \circ i_{s'} \circ i\} \\ &= \sigma_i \cdot \{x \in X \mid f(x) = i_s\} \\ &= \sigma_i \cdot G_s\langle X, f\rangle, \end{aligned} \tag{6.1.1.14}$$

and the action on arrows is straightforward. In particular, the inclusion $s \hookrightarrow s'$ gives rise to the inclusion $\mathbf{FMPre}_s \hookrightarrow \mathbf{FMPre}_{s'}$ in this way.

A parallel construction in the 'codomain' fibration on $(\mathbf{NPre} \downarrow y)$ gives rise to a more universal setting for this discussion. As mentioned above, the details of this construction are outside the scope of this thesis, but it is worth highlighting how some aspects of the structure of **NPre** manifest themselves within the dependent type theory.

6.1.2 Binding in $(\mathbf{FMPre}_\varnothing \downarrow y)$

This section demonstrates that the functor $(-) \otimes \mathbb{A}$ and its right adjoint δ give rise to the operation $(-)^{\#a}$ and its right adjoint δ_a defined in 3.2.1.15. In fact, this latter pair of functors are themselves dependent (on the name a) but it would require too much notation and too much complexity for the purposes of this section to capture this fact in full generality. For the sake of clarity, it is simpler here to notice that the isomorphism 6.1.1.1 gives rise to an isomorphism

$$\nu : y(s \mathbin{\dot\cup} \{a\}) \cong ys \otimes \mathbb{A}. \tag{6.1.2.1}$$

In detail, ν maps $i : s \mathbin{\dot\cup} \{a\} \rightarrowtail \mathbb{A}$ to the pair $\langle i|_s, i(a)\rangle$. Importantly, ν is equivariant, and hence an arrow of $\mathbf{FMPre}_\varnothing$, and $\nu(i_{s \dot\cup \{a\}}) = \langle i_s, a\rangle$.

6.1.2.2 The functors $(-) \otimes \mathbb{A}$ and $(-)^{\# a}$. Consider the object $\langle X, f\rangle$ of $\mathbf{FMPre}_{\varnothing}/ys$. The action of $(-) \otimes \mathbb{A}$ on $f : X \to ys$ results in the arrow $f \otimes \mathbf{1}_{\mathbb{A}} : X \otimes \mathbb{A} \to ys \otimes \mathbb{A}$. From the equivalence in lemma 6.1.1.2 the object $\langle X \otimes \mathbb{A}, \nu^{-1} \circ (f \otimes \mathbf{1}_{\mathbb{A}})\rangle$ corresponds to the FM set

$$\begin{aligned}
& G_{s \dot\cup \{a\}}\langle X \otimes \mathbb{A}, \nu^{-1} \circ (f \otimes \mathbf{1}_{\mathbb{A}})\rangle \\
&\quad = \quad \{\langle x, a'\rangle \in X \otimes \mathbb{A} \mid \nu^{-1}((f \otimes \mathbf{1}_{\mathbb{A}})\langle x, a'\rangle) = i_{s \dot\cup \{a\}}\} \\
&\quad = \quad \{\langle x, a'\rangle \in X \otimes \mathbb{A} \mid \langle f(x), a'\rangle = \langle i_s, a\rangle\} \\
&\quad = \quad \{\langle x, a\rangle \in X \otimes \mathbb{A} \mid f(x) = i_s\} \\
&\quad \cong \quad \{x \in X \mid f(x) = i_s \wedge a \# x\} \\
&\quad = \quad \{x \in X \mid f(x) = i_s\}^{\# a} \\
&\quad = \quad (G_s\langle X, f\rangle)^{\# a}.
\end{aligned} \tag{6.1.2.3}$$

It is straightforward to see that the action of $(-) \otimes \mathbb{A}$ on arrows gives rise to the action of $(-)^{\# a}$ on arrows in the same fashion.

6.1.2.4 The functors δ and δ_a. Now consider the object $\langle X, f\rangle$ of the category $\mathbf{FMPre}_{\varnothing}/y(s \dot\cup \{a\})$. The action of δ on $f : X \to y(s \dot\cup \{a\})$ results in the arrow $\delta f : \delta X \to \delta(y(s \dot\cup \{a\}))$, and using the isomorphism 6.1.2.1 this corresponds to an arrow $\delta\nu \circ \delta f : \delta X \to \delta(ys \otimes \mathbb{A})$. Form the pullback against the unit $\xi_{ys} : ys \to \delta(ys \otimes \mathbb{A})$ as follows:

$$\begin{tikzcd}
\delta X \times_{\delta(ys \otimes \mathbb{A})} ys \arrow[rr, "\pi_2"] \arrow[d, "\pi_1"'] \arrow[dr, phantom, "\lrcorner", very near start] & & ys \arrow[d, "\xi_{ys}"] \\
\delta X \arrow[r, "\delta f"'] & \delta(y(s \dot\cup \{a\})) \arrow[r, "\delta\nu"'] & \delta(ys \otimes \mathbb{A})
\end{tikzcd} \tag{6.1.2.5}$$

Via lemma 6.1.1.2 the object $\langle \delta X \times_{\delta(ys \otimes \mathbb{A})} ys, \pi_2\rangle$ of $\mathbf{FMPre}_{\varnothing}/ys$ corresponds to

$$\begin{aligned}
& G_s\langle \delta X \times_{\delta(ys \otimes \mathbb{A})} ys, \pi_2\rangle \\
&\quad = \quad \{\langle x, i\rangle \in \delta X \times ys \mid \delta\nu(\delta f(x)) = \xi_{ys}(i) \wedge i = i_s\} \\
&\quad = \quad \{\langle x, i_s\rangle \in \delta X \times ys \mid \mathbf{fresh}\, b\, \mathbf{in}\, [b].\nu(f(x@b)) = \mathbf{fresh}\, b\, \mathbf{in}\, [b].\langle i_s, b\rangle\} \\
&\quad \cong \quad \{x \in \delta X \mid \mathbf{fresh}\, b\, \mathbf{in}\, \nu(f(x@b)) = \langle i_s, b\rangle\} \\
&\quad = \quad \{x \in \delta X \mid \mathbf{fresh}\, b\, \mathbf{in}\, f(x@b) = (ab) \cdot i_{s \dot\cup \{a\}}\} \\
&\quad = \quad \{x \in \delta X \mid \mathbf{fresh}\, b\, \mathbf{in}\, x@b \in (ab) \cdot G_{s \dot\cup \{a\}}\langle X, f\rangle\} \\
&\quad = \quad \delta_a G_{s \dot\cup \{a\}}\langle X, f\rangle
\end{aligned} \tag{6.1.2.6}$$

It is straightforward to see that the action of δ on arrows gives rise to the action of δ_a on arrows in the same fashion.

6.1.2.7 The transformations $\pi_1 : (-) \otimes \mathbb{A} \to \mathbf{1}$ and $\tau_a : (-)^{\#a} \to \mathbf{1}$. Let $i : s \hookrightarrow s \mathbin{\dot\cup} \{a\}$ and consider the following situation.

$$\begin{tikzcd}
X \otimes \mathbb{A} \arrow[drr, bend left, "\pi_1"] \arrow[dr, dotted, "\tau"] \arrow[ddr, bend right, "\nu^{-1} \circ (f \otimes \mathbf{1}_{\mathbb{A}})"'] & & \\
& i^* X \arrow[r] \arrow[d] \arrow[dr, phantom, "\lrcorner", very near start] & X \arrow[d, "f"] \\
& y(s \mathbin{\dot\cup} \{a\}) \arrow[r, "yi"'] & ys
\end{tikzcd} \tag{6.1.2.8}$$

As i^*X is formed by pullback, there is a unique map $\tau : X \otimes \mathbb{A} \to i^*X$ which is an arrow of $\mathbf{FMPre}_{\varnothing}/y(s \mathbin{\dot\cup} \{a\})$. By 6.1.1.14 the object i^*X corresponds to $G_s\langle X, f\rangle$ as an object of $\mathbf{FMPre}_{s \dot\cup \{a\}}$, and by 6.1.2.3 the object $\langle X \otimes \mathbb{A}, \nu^{-1} \circ (f \otimes \mathbf{1}_{\mathbb{A}})\rangle$ corresponds to $(G_s\langle X, f\rangle)^{\#a}$. Therefore by lemma 6.1.1.2 the arrow τ corresponds to an arrow $(G_s\langle X, f\rangle)^{\#a} \to G_s\langle X, f\rangle$ defined for all $x \in (G_s\langle X, f\rangle)^{\#a}$ by

$$(G_{s \dot\cup a}\tau)x =_{\mathrm{def}} x. \tag{6.1.2.9}$$

In other words, the τ^a from 3.2.1.21 is the dependently-typed analogue of the projection $\pi_1 : X \otimes \mathbb{A} \to X$.

6.2 Adjunctions and Kleisli Categories

It can be seen from section 2.1 that a key feature of the Domain Theory for Concurrency is the use of Kleisli and co-Kleisli constructions, and from 2.2 that a key feature of the theory of nominal sets is the adjunction $(-) \otimes \mathbb{A} \dashv \delta$. Since this dissertation aims to merge these two theories, it is important that these structures interplay well. Abstractly, and subject to certain coherence conditions, they do indeed interplay well, and this fact gives rise to the binding structure in the FM-linear and FM-continuous categories defined in 3.3.5.9 and 3.4.8.26 respectively.

6.2.1 Adjoints to Inclusions

Firstly, notice that in both 3.3.2.1 and 3.4.4.15 one of the adjoints is the identity on objects. By the abstract argument below it follows that each of these adjunctions is isomorphic to the appropriate (co-)Kleisli construction.

6.2.1.1 Lemma. *Suppose that there is an adjunction*

$$\begin{tikzcd}
\mathcal{C} \arrow[r, bend right, "G"'] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend right, "F"']
\end{tikzcd}$$

with unit η and counit ϵ. If the left adjoint F is a bijection on objects then $\mathcal{C}$ is isomorphic to the Kleisli category of the monad $(GF, \eta, G\epsilon_F)$. Dually, if the right adjoint G is a bijection on objects then $\mathcal{D}$ is isomorphic to the coKleisli category of the comonad $(FG, \epsilon, F\eta_G)$.

Proof. The final two sentences of the statement of this lemma are dual to each other, so by duality it is sufficient to show just one. Therefore, suppose that the right adjoint G is a bijection on objects. Let $\mathcal{E}$ be the coKleisli category of FG. Define $L : \mathcal{E} \to \mathcal{D}$ as the (unique) coKleisli comparison functor. In detail, if A is an object of $\mathcal{E}$ then $LA =_{\text{def}} GA$ and if $f : A \to B$ is an arrow of $\mathcal{E}$ then $Lf =_{\text{def}} Gf \circ \eta_{GA}$.

Define $K : \mathcal{D} \to \mathcal{E}$ as follows. If A is an object of $\mathcal{D}$ then define $KA =_{\text{def}} G^{-1}A$ and if $f : A \to B$ is an arrow of $\mathcal{D}$ then define $Kf =_{\text{def}} \epsilon_{G^{-1}B} \circ Ff$.

To see that K is a functor, note that $K\mathbf{1}_A = \epsilon_{G^{-1}A}$ which is the identity on

KA in $\mathcal{E}$, and if $f : A \to B$ and $g : B \to C$ are arrows of $\mathcal{D}$ then

$$\begin{aligned}
K(g \circ f) &= \epsilon_{G^{-1}C} \circ F(g \circ f) \\
&= \epsilon_{G^{-1}C} \circ Fg \circ Ff \\
&= \epsilon_{G^{-1}C} \circ Fg \circ FG\epsilon_{G^{-1}B} \circ F\eta_{GG^{-1}B} \circ Ff \quad \text{by triangle identity} \\
&= \epsilon_{G^{-1}C} \circ Fg \circ FG\epsilon_{G^{-1}B} \circ FGFf \circ F\eta_{GG^{-1}A} \quad \text{by nat. of } \eta \\
&= \epsilon_{G^{-1}C} \circ Fg \circ FG(\epsilon_{G^{-1}B} \circ Ff) \circ F\eta_{GG^{-1}A} \\
&= Kg \circ FGKf \circ F\eta_{GKA}
\end{aligned} \tag{6.2.1.2}$$

which is the composition of Kg and Kf in $\mathcal{E}$ as required.

Let $f : A \to B$ in $\mathcal{D}$, then

$$\begin{aligned}
LKf &= L\big(\epsilon_{G^{-1}B} \circ Ff\big) \qquad (6.2.1.3) \\
&= G\epsilon_{G^{-1}B} \circ GFf \circ \eta_{GG^{-1}A} \\
&= G\epsilon_{G^{-1}B} \circ \eta_{GG^{-1}B} \circ f \quad \text{by naturality of } \eta \\
&= f \quad \text{by triangular identity.}
\end{aligned}$$

Conversely, let $f : A \to B$ in $\mathcal{E}$, then

$$\begin{aligned}
KLf &= K\big(Gf \circ \eta_{GA}\big) \qquad (6.2.1.4) \\
&= \epsilon_{G^{-1}GB} \circ FGf \circ F\eta_{GA} \\
&= \epsilon_{B} \circ FGf \circ F\eta_{GA} \\
&= f \circ \epsilon_{FGA} \circ F\eta_{GA} \quad \text{by naturality of } \epsilon \\
&= f \quad \text{by triangular identity.}
\end{aligned}$$

Therefore L and K are mutual inverses, which completes the proof. □

In particular, $K_{\mathbf{Lin}} : \mathbf{FMLin}_s \to \mathrm{Kl}\big(\widehat{(-)}$ on $\mathbf{FMPre}_s\big)$ and its inverse $L_{\mathbf{Lin}}$ are both defined as the identity on objects, and if $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{P}'$ is an arrow of $\mathbf{FMLin}_s$ and $g : \mathbb{Q} \to \widehat{\mathbb{Q}'}$ is an arrow of $\mathbf{FMPre}_s$ then

$$K_{\mathbf{Lin}}f =_{\mathrm{def}} f \circ \{\cdot\}_{\mathbb{P}} \quad \text{and} \quad L_{\mathbf{Lin}}g =_{\mathrm{def}} g^{\dagger}. \tag{6.2.1.5}$$

Similarly, $K_{\mathbf{Cts}} : \mathbf{FMCts}_s \to \mathrm{Kl}\big(!$ on $\mathbf{FMLin}_s\big)$ and its inverse $L_{\mathbf{Cts}}$ are both defined as the identity on objects, and if $f : \mathbb{P} \underset{\mathbf{C}}{\to} \mathbb{P}'$ is an arrow of $\mathbf{FMCts}_s$ and $g : !\mathbb{Q} \underset{\mathbf{L}}{\to} \mathbb{Q}'$ is an arrow of $\mathbf{FMLin}_s$ then

$$K_{\mathbf{Cts}}f =_{\mathrm{def}} (f \circ i_{\mathbb{P}})^{\dagger} \quad \text{and} \quad L_{\mathbf{Cts}}g =_{\mathrm{def}} g \circ \eta_{\mathbb{Q}}. \tag{6.2.1.6}$$

6.2.2 Adjunctions in Kleisli Categories

In a situation

$$\begin{tikzcd} \mathcal{C} \arrow[loop left, "T"] \arrow[r, bend left, "F"] \arrow[r, phantom, "\perp"] & \mathcal{D} \arrow[l, bend left, "G"] \arrow[loop right, "S"] \end{tikzcd} \tag{6.2.2.1}$$

where T and S are monads and all the functors satisfy certain coherence conditions, it is possible to 'lift' the adjunction $F \dashv G$ to the respective Kleisli categories of T and S. This follows from abstract results on the formal theory of monads[32] but the proof is elementary diagram-chasing so it is reproduced here. In 6.3 it is shown that this lifting process, combined with the isomorphism $K_{\mathbf{Lin}}$ defined in 6.2.1.5, gives rise to the adjunction $(-)^{\#a+} \dashv \delta_a^+$ described in 3.3.5.9. Furthermore it is shown in 6.4 that this lifting process applied to the adjunction $(-)^{\#a+} \dashv \delta_a^+$, combined with the isomorphism $K_{\mathbf{Cts}}$ defined in 6.2.1.6, gives rise to the adjunction $(-)^{\#a++} \dashv \delta_a^{++}$ described in 3.4.8.26.

6.2.2.2 Lemma. *Let $\mathcal{C}$ and $\mathcal{D}$ be categories. Let (T, η, μ) be a monad on $\mathcal{C}$ and let (S, θ, ν) be a monad on $\mathcal{D}$. Let $F \dashv G : \mathcal{C} \leftrightarrows \mathcal{D}$ be an adjunction with unit ξ and counit ϵ. Let $k : FT \to SF$ and $h : GS \to TG$ be natural transformations such that the following diagrams commute.*

(a)
$$\begin{tikzcd} F \arrow[r, "F\eta"] \arrow[dr, "\theta_F"'] & FT \arrow[d, "k"] \\ & SF \end{tikzcd}$$

(b)
$$\begin{tikzcd} FTT \arrow[r, "F\mu"] \arrow[d, "k_T"] & FT \arrow[dd, "k"] \\ SFT \arrow[d, "Sk"] & \\ SSF \arrow[r, "\nu_F"] & SF \end{tikzcd}$$

(c)
$$\begin{tikzcd} T \arrow[r, "T\xi"] \arrow[d, "\xi_T"] & TGF \\ GFT \arrow[r, "Gk"] & GSF \arrow[u, "h_F"'] \end{tikzcd}$$

(d)
$$\begin{tikzcd} G \arrow[r, "G\theta"] \arrow[dr, "\eta_G"'] & GS \arrow[d, "h"] \\ & TG \end{tikzcd}$$

(e)
$$\begin{tikzcd} GSS \arrow[r, "G\nu"] \arrow[d, "h_S"] & GS \arrow[dd, "h"] \\ TGS \arrow[d, "Th"] & \\ TTG \arrow[r, "\mu_G"] & TG \end{tikzcd}$$

(f)
$$\begin{tikzcd} FGS \arrow[r, "Fh"] \arrow[d, "\epsilon_S"] & FTG \arrow[d, "k_G"] \\ S & SFG \arrow[l, "S\epsilon"'] \end{tikzcd}$$

For all objects A and all arrows $f : A \to TB$ of $\mathcal{C}$ define

$$F^+A =_{\text{def}} FA \quad \textit{and} \quad F^+f =_{\text{def}} k_B \circ Ff. \tag{6.2.2.3}$$

Similarly, for all objects A and all arrows $g : A \to B$ of $\mathcal{D}$ define

$$G^+A =_{\text{def}} GA \quad \textit{and} \quad G^+f =_{\text{def}} h_B \circ Gf. \tag{6.2.2.4}$$

Then F^+ and G^+ are functors and there is an adjunction

$$F^+ \dashv G^+ : \mathrm{Kl}(T) \leftrightarrows \mathrm{Kl}(S) \tag{6.2.2.5}$$

with unit ξ^+ and counit ϵ^+ where

$$\xi^+ =_{\text{def}} \eta_{GF} \circ \xi \quad \textit{and} \quad \epsilon^+ =_{\text{def}} \theta \circ \epsilon. \tag{6.2.2.6}$$

Proof. The proof proceeds by unwinding definitions and diagram-chasing as follows.

$F^+ : \mathrm{Kl}(T) \to \mathrm{Kl}(S)$ is a functor: Let A be an object of $\mathrm{Kl}(T)$, then the identity on A is given by η_A in $\mathcal{C}$. Then by (a), $F^+\eta_A = k_A \circ F\eta_A = \theta_{FA}$ which is the identity on $FA = F^+A$ in $\mathrm{Kl}(S)$ as required. Now let $f : A \to B$ and $g : B \to C$ be arrows in $\mathrm{Kl}(T)$, then the following diagram commutes in $\mathcal{D}$ by naturality of k and (b).

$$\begin{tikzcd}
FA \arrow[r, "Ff"] & FTB \arrow[r, "FTg"] \arrow[d, "k_B"] & FTTC \arrow[rr, "F\mu_C"] \arrow[d, "k_{TC}"] & & FTC \arrow[d, "k_C"] \\
& SFB \arrow[r, "SFg"] & SFTC \arrow[r, "Sk_C"] & SSFC \arrow[r, "\nu_{FC}"] & SFC
\end{tikzcd} \tag{6.2.2.7}$$

In $\mathrm{Kl}(S)$ the compositions clockwise and anticlockwise around the outside of this diagram are the arrows $F^+(g \circ f)$ and $F^+g \circ F^+f$ respectively, which shows that F^+ is a functor.

$G^+ : \mathrm{Kl}(S) \to \mathrm{Kl}(T)$ is a functor: Let A be an object of $\mathrm{Kl}(S)$, then the identity on A is given by θ_A in $\mathcal{D}$. Then by (d), $G^+\theta_A = h_A \circ G\theta_A = \eta_{GA}$ which is the identity on $GA = G^+A$ in $\mathrm{Kl}(T)$ as required. Now let $f : A \to B$ and $g : B \to C$ be arrows in $\mathrm{Kl}(S)$, then the following diagram commutes in $\mathcal{C}$ by naturality of h and (e).

$$\begin{tikzcd}
GA \arrow[r, "Gf"] & GSB \arrow[r, "GSg"] \arrow[d, "h_B"] & GSSC \arrow[rr, "G\nu_C"] \arrow[d, "h_{SC}"] & & GSC \arrow[d, "h_C"] \\
& TGB \arrow[r, "TGg"] & TGSC \arrow[r, "Th_C"] & TTGC \arrow[r, "\mu_{GC}"] & TGC
\end{tikzcd} \tag{6.2.2.8}$$

In $\mathrm{Kl}(T)$ the compositions clockwise and anticlockwise around the outside of this diagram are the arrows $G^+(g \circ f)$ and $G^+g \circ G^+f$ respectively, which shows that G^+ is a functor.

$\xi^+ : \mathbf{1} \to G^+F^+$ is a natural transformation: Let $f : A \to B$ be an arrow of $\mathrm{Kl}(T)$. The following diagram commutes in $\mathcal{C}$ by naturality of η and ξ, the monad laws for T, and (c).

$$\begin{tikzcd}
A \arrow[r, "f"] \arrow[d, "\xi_A"] & TB \arrow[d, "\xi_{TB}"] \arrow[drr, "T\xi_B"] & & & \\
GFA \arrow[r, "GFf"] \arrow[d, "\eta_{GFA}"] & GFTB \arrow[r, "Gk_B"] & GSFB \arrow[r, "h_{FB}"] & TGFB \arrow[r, "T\eta_{GFB}"] \arrow[d, "\eta_{TGFB}"'] \arrow[dr, equal] & TTGFB \arrow[d, "\mu_{GFB}"] \\
TGFA \arrow[r, "TGFf"] & TGFTB \arrow[r, "TGk_B"] & TGSFB \arrow[r, "Th_{FB}"] & TTGFB \arrow[r, "\mu_{GFB}"] & TGFB
\end{tikzcd} \tag{6.2.2.9}$$

In $\mathrm{Kl}(T)$ the compositions clockwise and anticlockwise around the outside of this diagram are the arrows $\xi_B^+ \circ f$ and $G^+F^+f \circ \xi_A^+$ respectively, which shows that ξ^+ is natural.

$\epsilon^+ : F^+G^+ \to \mathbf{1}$ is a natural transformation: Let $f : A \to B$ be an arrow of $\mathrm{Kl}(S)$. The following diagram commutes in $\mathcal{D}$ by naturality of ϵ and θ, the monad laws for S and (f).

$$\begin{tikzcd}
FGA \arrow[r, "FGf"] \arrow[d, "\epsilon_A"] & FGSB \arrow[r, "Fh_B"] \arrow[d, "\epsilon_{SB}"'] & FTGB \arrow[r, "k_{GB}"] & SFGB \arrow[dll, bend left, "S\epsilon_B"'] \\
A \arrow[r, "f"] \arrow[d, "\theta_A"] & SB \arrow[r, "S\theta_B"'] \arrow[d, "\theta_{SB}"'] \arrow[dr, equal] & SSB \arrow[d, "\nu_B"] & \\
SA \arrow[r, "Sf"] & SSB \arrow[r, "\nu_B"'] & SB &
\end{tikzcd} \tag{6.2.2.10}$$

In $\mathrm{Kl}(S)$ the compositions around the outside of this diagram clockwise and anticlockwise are the arrows $\epsilon_B^+ \circ F^+G^+f$ and $f \circ \epsilon_A^+$ respectively, which shows that ϵ^+ is natural.

ξ^+ and ϵ^+ satisfy the triangular identities: The following diagram commutes in $\mathcal{C}$ by naturality of η, the monad laws for T, the triangular identity for G and (c).

$$\begin{tikzcd}
 & TGFGA \arrow[r, "TG\epsilon_A"] & TGA \arrow[r, "TG\theta_A"] & TGSA \arrow[r, "Th_A"] & TTGA \arrow[r, "\mu_{GA}"] & TGA \\
GA \arrow[r, "\xi_{GA}"] \arrow[rr, equal, bend right] & GFGA \arrow[r, "G\epsilon_A"] \arrow[u, "\eta_{GFGA}"] & GA \arrow[r, "G\theta_A"] \arrow[rr, bend right, "\eta_{GA}"'] & GSA \arrow[r, "h_A"] & TGA \arrow[u, "\eta_{TGA}"'] \arrow[ur, equal] &
\end{tikzcd} \tag{6.2.2.11}$$

In $\mathrm{Kl}(T)$ the compositions clockwise and anticlockwise around the outside of this diagram are the arrows $G^+\epsilon_A^+ \circ \xi_{G^+A}$ and the identity on GA respectively, which demonstrates the triangular identity for G^+.

The following diagram commutes in $\mathcal{D}$ by the naturality of θ, the monad laws for S, the triangular identity for F and (a).

$$\begin{tikzcd}
FTGFA \arrow[r, "k_{GFA}"] & SFGFA \arrow[r, "S\epsilon_{FA}"] & SFA \arrow[r, "S\theta_{FA}"] \arrow[dr, equal] & SSFA \arrow[d, "\nu_{FA}"] \\
FA \arrow[r, "F\xi_A"'] \arrow[rr, equal, bend right] & FGFA \arrow[ul, "F\eta_{GFA}"'] \arrow[u, "\theta_{FGFA}"'] \arrow[r, "\epsilon_{FA}"] & FA \arrow[u, "\theta_{FA}"] & SFA
\end{tikzcd} \tag{6.2.2.12}$$

In $\mathrm{Kl}(S)$ the compositions clockwise and anticlockwise around the outside of this diagram are the arrows $\epsilon^+_{F^+A} \circ F^+\xi_A$ and the identity on FA respectively, which demonstrates the triangular identity for F^+. This completes the proof. $\square$

The commuting diagrams (a) – (f) of lemma 6.2.2.2 may be referred to respectively as the 'left triangle', 'left pentagon', 'unit square', 'right triangle', 'right pentagon' and 'counit square'.

6.3 Binding in FM-Linear Categories

As discussed in 3.3.5.9 there is an adjunction on the FM-linear categories that is analogous to the adjunction $(-)^{\#a} \dashv \delta_a : \mathbf{FMPre}_s \leftrightarrows \mathbf{FMPre}_{s\dot{\cup}\{a\}}$ described in lemma 3.2.1.15. Abstractly, this adjunction arises from lemma 6.2.2.2 making use of the natural transformations $\phi : \widehat{(-)}^{\#a} \to \widehat{(-)^{\#a}}$ and $\theta : \delta_a\widehat{(-)} \to \widehat{\delta_a(-)}$, and lemma 6.2.1.1 which shows that $\mathbf{FMLin}_s$ is the appropriate Kleisli category. This section demonstrates that this abstract description coincides with the concrete definitions given in 3.3.5.9. Firstly it is shown in 6.3.1 that ϕ and θ satisfy the premises of lemma 6.2.2.2 and then 6.3.2 unwinds definitions to demonstrate that the result of this abstract process coincides with the concrete description given in 3.3.5.9.

6.3.1 Binding in $\mathbf{FMLin}_s$, Abstractly

6.3.1.1 Lemma. *If $\mathbb{P}$ is an object of $\mathbf{FMPre}_s$ then define $F^+\mathbb{P} = \mathbb{P}^{\#a}$, and if $f : \mathbb{P} \to \widehat{\mathbb{Q}}$ is an arrow of $\mathbf{FMPre}_s$ then define $F^+f = \phi_{\mathbb{Q}} \circ f^{\#a}$. If $\mathbb{P}$ is an object of $\mathbf{FMPre}_{s\dot{\cup}\{a\}}$ then define $G^+\mathbb{P} = \delta_a\mathbb{P}$, and if $f : \mathbb{P} \to \widehat{\mathbb{Q}}$ is an arrow of $\mathbf{FMPre}_{s\dot{\cup}\{a\}}$ then define $G^+f = \theta_{\mathbb{Q}} \circ \delta_a f$. Define $\xi^+ = \{\cdot\}_{\delta_a((-)^{\#a})} \circ \xi$ and $\zeta^+ = \{\cdot\}_{\downarrow} \circ \zeta$. Then ξ^+ and ζ^+ are respectively the unit and counit of an adjunction*

$$F^+ \dashv G^+ : \mathrm{Kl}\big(\widehat{(-)} \text{ on } \mathbf{FMPre}_s\big) \leftrightarrows \mathrm{Kl}\big(\widehat{(-)} \text{ on } \mathbf{FMPre}_{s\dot{\cup}\{a\}}\big).$$

Proof. By lemma 6.2.2.2 it is sufficient to show that the diagrams below commute.

$$\begin{tikzcd}
\mathbb{P}^{\#a} \arrow[r, "\{\cdot\}^{\#a}_{\mathbb{P}}"] \arrow[dr, "\{\cdot\}_{\mathbb{P}^{\#a}}"'] & \widehat{\mathbb{P}}^{\#a} \arrow[d, "\phi_{\mathbb{P}}"] \\
 & \widehat{\mathbb{P}^{\#a}}
\end{tikzcd}
\qquad
\begin{tikzcd}
\widehat{\widehat{\mathbb{P}}}^{\#a} \arrow[r, "\cup^{\#a}_{\mathbb{P}}"] \arrow[d, "\phi_{\widehat{\mathbb{P}}}"] & \widehat{\mathbb{P}}^{\#a} \arrow[dd, "\phi_{\mathbb{P}}"] \\
\widehat{\widehat{\mathbb{P}}^{\#a}} \arrow[d, "\widehat{\phi_{\mathbb{P}}}"] & \\
\widehat{\widehat{\mathbb{P}^{\#a}}} \arrow[r, "\cup_{\mathbb{P}^{\#a}}"] & \widehat{\mathbb{P}^{\#a}}
\end{tikzcd}
\qquad
\begin{tikzcd}
\widehat{\mathbb{P}} \arrow[r, "\widehat{\xi}"] \arrow[d, "\xi_{\widehat{\mathbb{P}}}"] & \widehat{\delta_a(\mathbb{P}^{\#a})} \\
\delta_a(\widehat{\mathbb{P}}^{\#a}) \arrow[r, "\delta_a\phi_{\mathbb{P}}"] & \delta_a\widehat{\mathbb{P}^{\#a}} \arrow[u, "\theta_{\mathbb{P}^{\#a}}"']
\end{tikzcd}$$

$$\begin{tikzcd}
\delta_a\mathbb{P} \arrow[r, "\delta_a\{\cdot\}_{\mathbb{P}}"] \arrow[dr, "\{\cdot\}_{\delta_a\mathbb{P}}"'] & \delta_a\widehat{\mathbb{P}} \arrow[d, "\theta_{\mathbb{P}}"] \\
 & \widehat{\delta_a\mathbb{P}}
\end{tikzcd}
\qquad
\begin{tikzcd}
\delta_a\widehat{\widehat{\mathbb{P}}} \arrow[r, "\delta_a\cup_{\mathbb{P}}"] \arrow[d, "\theta_{\widehat{\mathbb{P}}}"] & \delta_a\widehat{\mathbb{P}} \arrow[dd, "\theta_{\mathbb{P}}"] \\
\widehat{\delta_a\widehat{\mathbb{P}}} \arrow[d, "\widehat{\theta_{\mathbb{P}}}"] & \\
\widehat{\widehat{\delta_a\mathbb{P}}} \arrow[r, "\cup_{\delta_a\mathbb{P}}"] & \widehat{\delta_a\mathbb{P}}
\end{tikzcd}
\qquad
\begin{tikzcd}
(\delta_a\widehat{\mathbb{P}})^{\#a} \arrow[r, "\theta^{\#a}_{\mathbb{P}}"] \arrow[d, "\zeta_{\widehat{\mathbb{P}}}"] & \widehat{\delta_a\mathbb{P}}^{\#a} \arrow[d, "\phi_{\delta_a\mathbb{P}}"] \\
\widehat{\mathbb{P}} & \widehat{(\delta_a\mathbb{P})^{\#a}} \arrow[l, "\widehat{\zeta}_{\mathbb{P}}"']
\end{tikzcd}$$

Left Triangle. By lemma 3.3.3.6.

Left Pentagon. Let $X \in \widehat{\widehat{\mathbb{P}}}^{\#a}$. First, let $p \in \big(\phi_{\mathbb{P}} \circ \cup^{\#a}_{\mathbb{P}}\big)X$, then $a \# p$ and there exists $x \in X$ such that $p \in x$. Let b be a fresh name, then it follows that $p = (ab) \cdot p \in (ab) \cdot x \in (ab) \cdot X = X$ and $a \# (ab) \cdot x$. Therefore $(ab) \cdot x \in \phi_{\widehat{\mathbb{P}}} X$ and $p \in \phi_{\mathbb{P}}(ab) \cdot x$ so that $p \in \big(\cup_{\mathbb{P}^{\#a}} \circ \widehat{\phi_{\mathbb{P}}} \circ \phi_{\widehat{\mathbb{P}}}\big)X$. Conversely, let $p \in \big(\cup_{\mathbb{P}^{\#a}} \circ \widehat{\phi_{\mathbb{P}}} \circ \phi_{\widehat{\mathbb{P}}}\big)X$, then $a \# p$ and there exists $x \in X$ such that $a \# x$ and $p \in x$ so that $p \in \big(\phi_{\mathbb{P}} \circ \cup^{\#a}_{\mathbb{P}}\big)X$ as required.

Unit Square. Let $x \in \widehat{\mathbb{P}}$, then

$$\begin{aligned}
\big(\theta_{\mathbb{P}^{\#a}} \circ \delta_a\phi_{\mathbb{P}} \circ \xi_{\widehat{\mathbb{P}}}\big)x &= \big(\theta_{\mathbb{P}^{\#a}} \circ \delta_a\phi_{\mathbb{P}}\big)\, \mathbf{fresh}\, b\, \mathbf{in}\, [b].x \\
&= \theta_{\mathbb{P}^{\#a}}\, \mathbf{fresh}\, b\, \mathbf{in}\, [b].\big(((ab) \cdot \phi)x\big) \\
&= \theta_{\mathbb{P}^{\#a}}\, \mathbf{fresh}\, b\, \mathbf{in}\, [b].\{p \in x \mid b \# p\} \\
&= \{p' \mid \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in \{p \in x \mid b \# p\}\} \\
&= \{p' \mid \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in x \wedge b \# p'@b\} \\
&= \widehat{\xi}_{\mathbb{P}} x.
\end{aligned} \tag{6.3.1.2}$$

Right Triangle. By lemma 3.3.4.5.

Right Pentagon. Let $X' \in \delta_a\widehat{\widehat{\mathbb{P}}}$. First, let $p' \in \big(\theta_{\mathbb{P}} \circ \delta_a\cup_{\mathbb{P}}\big)X'$ and let b be a fresh name, then $p'@b \in \big(\delta_a\cup_{\mathbb{P}}X'\big)@b = \bigcup(X'@b)$ and hence there exists some $x \in X'@b$ such that $p'@b \in x$. It follows that $([b].x)@b \in X'@b$ and $p'@b \in ([b].x)@b$ and hence $[b].x \in \theta_{\widehat{\mathbb{P}}}X'$ and $p' \in \theta_{\mathbb{P}}[b].x$ so it follows that $p' \in \big(\cup_{\delta_a\mathbb{P}} \circ \widehat{\theta_{\mathbb{P}}} \circ \theta_{\widehat{\mathbb{P}}}\big)X'$. Conversely, let $p' \in \big(\cup_{\delta_a\mathbb{P}} \circ \widehat{\theta_{\mathbb{P}}} \circ \theta_{\widehat{\mathbb{P}}}\big)X'$, then there exists x' such that for a fresh name b it is the case that $p'@b \in x'@b$ and $x'@b \in X'@b$ so that $p'@b \in \bigcup(X'@b) = \big(\delta_a\cup_{\mathbb{P}}X'\big)@b$ and hence $p' \in \big(\theta_{\mathbb{P}} \circ \delta_a\cup_{\mathbb{P}}\big)X'$ as required.

Counit Square. Let $x' \in (\delta_a\widehat{\mathbb{P}})^{\#a}$, then

$$\begin{array}{rcll}
\big(\widehat{\zeta_{\mathbb{P}}} \circ \phi_{\delta_a\mathbb{P}} \circ \theta^{\#a}_{\mathbb{P}}\big)x' & = & \big(\widehat{\zeta_{\mathbb{P}}} \circ \phi_{\delta_a\mathbb{P}}\big)\{p' \mid \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in x'@b\} & (6.3.1.3)\\
& = & \widehat{\zeta_{\mathbb{P}}}\{p' \mid a \# p' \wedge \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in x'@b\} & \\
& = & \{p'@a \mid a \# p' \wedge \mathbf{fresh}\, b\, \mathbf{in}\, p'@b \in x'@b\} & \\
& = & \{p'@a \mid a \# p' \wedge p'@a \in x'@a\} & \\
& = & x'@a = \zeta_{\widehat{\mathbb{P}}}x'. &
\end{array}$$

This completes the proof. □

6.3.2 Binding in FMLin$_s$, Concretely

Via the isomorphism $L_{\mathbf{Lin}} = K^{-1}_{\mathbf{Lin}}$ of 6.2.1.5 the adjunction $F^+ \dashv G^+$ of lemma 6.3.1.1 gives rise to the adjunction $(-)^{\#a+} \dashv \delta^+_a : \mathbf{FMLin}_s \leftrightarrows \mathbf{FMLin}_{s\dot{\cup}\{a\}}$ of 3.3.5.9 as follows.

6.3.2.1 The Left Adjoint. If $\mathbb{P}$ is an object of $\mathbf{FMLin}_s$ then certainly $L_{\mathbf{Lin}}F^+K_{\mathbf{Lin}}\mathbb{P} = \mathbb{P}^{\#a} = \mathbb{P}^{\#a+}$. If $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_s$ then

$$\begin{array}{rcll}
L_{\mathbf{Lin}}F^+K_{\mathbf{Lin}}f & = & \big(\phi_{\mathbb{Q}} \circ f^{\#a} \circ \{\cdot\}^{\#a}_{\mathbb{P}}\big)^{\dagger} & (6.3.2.2)\\
& = & \big(\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}} \circ \{\cdot\}_{\mathbb{P}\#a}\big)^{\dagger} & \text{by 3.3.3.6}\\
& = & \big(\phi_{\mathbb{Q}} \circ f^{\#a} \circ \phi^{-1}_{\mathbb{P}}\big) & \text{by 3.3.1.4}\\
& = & f^{\#a+} & \text{by 3.3.5.13.}
\end{array}$$

6.3.2.3 The Right Adjoint. Similarly, if $\mathbb{P}$ is an object of $\mathbf{FMLin}_{s\dot{\cup}\{a\}}$ then $L_{\mathbf{Lin}}G^+K_{\mathbf{Lin}}\mathbb{P} = \delta_a\mathbb{P} = \delta^+_a\mathbb{P}$. If $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_{s\dot{\cup}\{a\}}$ then

$$\begin{array}{rcll}
L_{\mathbf{Lin}}G^+K_{\mathbf{Lin}}f & = & \big(\theta_{\mathbb{Q}} \circ \delta_a f \circ \delta_a\{\cdot\}_{\mathbb{P}}\big)^{\dagger} & (6.3.2.4)\\
& = & \big(\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}} \circ \{\cdot\}_{\delta_a\mathbb{P}}\big)^{\dagger} & \text{by 3.3.4.5}\\
& = & \big(\theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}}\big) & \text{by 3.3.1.4}\\
& = & \delta^+_a f & \text{by 3.3.5.14.}
\end{array}$$

6.3.2.5 The Unit. First note that the following diagram commutes by naturality of ξ, 3.3.3.6 and 3.3.4.5.

$$\begin{tikzcd}
\mathbf{1} \arrow[r, "\xi"] \arrow[d, "\{\cdot\}_{\downarrow}"'] & \delta_a((-)^{\#a}) \arrow[r, "\{\cdot\}_{\delta_a((-)^{\#a})}"] \arrow[d, "\delta_a\{\cdot\}^{\#a}_{\downarrow}"'] \arrow[rd, "\delta_a\{\cdot\}_{(-)^{\#a}}"] & \widehat{\delta_a((-)^{\#a})} \\
\widehat{(-)} \arrow[r, "\xi_{\widehat{(-)}}"'] & \delta_a\widehat{(-)}^{\#a} \arrow[r, "\delta_a\phi"'] & \delta_a\widehat{(-)^{\#a}} \arrow[u, "\theta_{(-)^{\#a}}"']
\end{tikzcd} \tag{6.3.2.6}$$

Therefore

$$\begin{aligned}
L_{\mathbf{Lin}}\xi^{+}_{K_{\mathbf{Lin}}} &= \left(\{\cdot\}_{\delta_a((-)^{\#a})} \circ \xi\right)^{\dagger} && \\
&= \left(\theta_{(-)^{\#a}} \circ \delta_a\phi \circ \xi_{\widehat{(-)}} \circ \{\cdot\}_{\downarrow}\right)^{\dagger} && \text{by } 6.3.2.6 \\
&= \theta_{(-)^{\#a}} \circ \delta_a\phi \circ \xi_{\widehat{(-)}} && \text{by } 3.3.1.4 \\
&= \widehat{\xi} && \text{by } 6.3.1.2.
\end{aligned} \tag{6.3.2.7}$$

6.3.2.8 The Counit. Similarly, note first that the following diagram commutes by naturality of ζ, 3.3.3.6 and 3.3.4.5.

$$\begin{tikzcd}
\widehat{(\delta_a -)^{\#a}} \arrow[d, "\phi^{-1}_{\delta_a}"'] & (\delta_a(-))^{\#a} \arrow[l, "\{\cdot\}_{(\delta_a(-))^{\#a}}"'] \arrow[r, "\zeta"] \arrow[ld, "\{\cdot\}^{\#a}_{\delta_a}"'] \arrow[d, "(\delta_a\{\cdot\})^{\#a}"] & \mathbf{1} \arrow[d, "\{\cdot\}_{\downarrow}"] \\
\widehat{\delta_a(-)}^{\#a} \arrow[r, "\theta^{-1\#a}"'] & (\delta_a\widehat{(-)})^{\#a} \arrow[r, "\zeta_{\widehat{(-)}}"'] & \widehat{(-)}
\end{tikzcd} \tag{6.3.2.9}$$

Therefore

$$\begin{aligned}
L_{\mathbf{Lin}}\zeta^{+}_{K_{\mathbf{Lin}}} &= \left(\{\cdot\}_{\downarrow} \circ \zeta\right)^{\dagger} && \\
&= \left(\zeta_{\widehat{(-)}} \circ \theta^{-1\#a} \circ \phi^{-1}_{\delta_a} \circ \{\cdot\}_{(\delta_a(-))^{\#a}}\right)^{\dagger} && \text{by } 6.3.2.9 \\
&= \zeta_{\widehat{(-)}} \circ \theta^{-1\#a} \circ \phi^{-1}_{\delta_a} && \text{by } 3.3.1.4 \\
&= \widehat{\zeta} && \text{by } 6.3.1.3.
\end{aligned} \tag{6.3.2.10}$$

Therefore the adjunction $F^{+} \dashv G^{+}$ of lemma 6.3.1.1 gives rise to the adjunction $(-)^{\#a+} \dashv \delta^{+}_{a} : \mathbf{FMLin}_{s} \leftrightarrows \mathbf{FMLin}_{s\dot{\cup}\{a\}}$ of 3.3.5.9 by way of the isomorphism $L_{\mathbf{Lin}} = K^{-1}_{\mathbf{Lin}}$ of 6.2.1.5 as required.

6.4 Binding in FM-Continuous Categories

As discussed in 3.4.8.26 there is an adjunction on the FM-continuous categories that is analogous to the adjunctions $(-)^{\#a} \dashv \delta_a : \mathbf{FMPre}_{s} \leftrightarrows \mathbf{FMPre}_{s\dot{\cup}\{a\}}$ and $(-)^{\#a+} \dashv \delta^{+}_{a} : \mathbf{FMLin}_{s} \leftrightarrows \mathbf{FMLin}_{s\dot{\cup}\{a\}}$ of 3.2.1.15 and 3.3.5.9 respectively. Abstractly, this adjunction arises from the dual of lemma 6.2.2.2 since the FM-continuous categories are isomorphic to coKleisli categories via the isomorphism $K_{\mathbf{Cts}}$ of 6.2.1.6. This section demonstrates that this abstract description coincides with the concrete definitions given in 3.4.8.26. The natural transformations required to apply lemma 6.2.2.2 are $\widehat{\phi^{!\,-1}}$ and $\widehat{\theta^{!\,-1}}$. Firstly it is shown in 6.4.1 that these natural transformations satisfy the appropriate coherence conditions to apply lemma 6.2.2.2, and then 6.4.2 unwinds definitions to demonstrate that the result of this abstract process coincides with the concrete route given in 3.4.8.26.

6.4.1 Binding in FMCts$_s$, Abstractly

This section uses the abstract machinery of lemma 6.2.2.2 to lift the adjunction $(-)^{\#a+} \dashv \delta_a^+ : \mathbf{FMLin}_s \leftrightarrows \mathbf{FMLin}_{s\dot\cup\{a\}}$ to a corresponding adjunction on the coKleisli categories of the respective ! comonads. Firstly, note that $\widehat{\phi^{!\,-1}}$ and $\widehat{\theta^{!\,-1}}$ are natural transformations of the appropriate types, as shown in lemmas 6.4.1.1 and 6.4.1.2. Then lemma 6.4.1.3 shows that they also satisfy the appropriate coherence conditions to apply lemma 6.2.2.2.

6.4.1.1 Lemma. *$\widehat{\phi^{!\,-1}}$ is a natural transformation $!((-)^{\#a+}) \to (!-)^{\#a+}$.*

Proof. Let $f : \mathbb{P} \underset{\mathbf{L}}{\to} \mathbb{Q}$ be an arrow of $\mathbf{FMLin}_s$, then

$$\begin{array}{lll}
\multicolumn{3}{l}{(!f)^{\#a+} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}} \circ \{\cdot\}_{!(\mathbb{P}^{\#a})}} \\
= & \phi_{!\mathbb{Q}} \circ (!f)^{\#a} \circ \phi_{!\mathbb{P}}^{-1} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}} \circ \{\cdot\}_{!(\mathbb{P}^{\#a})} & \text{by } 3.3.5.13 \\
= & \phi_{!\mathbb{Q}} \circ (!f)^{\#a} \circ \phi_{!\mathbb{P}}^{-1} \circ \{\cdot\}_{(!\mathbb{P})^{\#a}} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by naturality of } \{\cdot\}_{\downarrow} \\
= & \phi_{!\mathbb{Q}} \circ (!f)^{\#a} \circ \{\cdot\}_{!\mathbb{P}}^{\#a} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by } 3.3.3.6 \\
= & \phi_{!\mathbb{Q}} \circ \eta_{\mathbb{Q}}^{\#a} \circ f^{\#a} \circ i_{\mathbb{P}}^{\#a} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by } 3.4.4.14 \\
= & \phi_{!\mathbb{Q}} \circ \eta_{\mathbb{Q}}^{\#a} \circ f^{\#a} \circ \phi_{\mathbb{P}}^{-1} \circ i_{\mathbb{P}^{\#a}} & \text{by } 3.4.5.7 \\
= & \phi_{!\mathbb{Q}} \circ \eta_{\mathbb{Q}}^{\#a} \circ \phi_{\mathbb{Q}}^{-1} \circ f^{\#a+} \circ i_{\mathbb{P}^{\#a}} & \text{by } 3.3.5.13 \\
= & \widehat{\phi_{\mathbb{Q}}^{!\,-1}} \circ \eta_{\mathbb{Q}^{\#a}} \circ f^{\#a+} \circ i_{\mathbb{P}^{\#a}} & \text{by } 3.4.7.3 \\
= & \widehat{\phi_{\mathbb{Q}}^{!\,-1}} \circ !(f^{\#a+}) \circ \{\cdot\}_{!(\mathbb{P}^{\#a})} & \text{by } 3.4.4.14.
\end{array}$$

But $(!f)^{\#a+} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}}$ and $\widehat{\phi_{\mathbb{Q}}^{!\,-1}} \circ !(f^{\#a+})$ are both linear, so by 3.3.1.6 they are equal as required. □

6.4.1.2 Lemma. *$\widehat{\theta^{!\,-1}}$ is a natural transformation $!\delta_a^+ \to \delta_a^+!$.*

Proof. Let $f : \mathbb{P} \underset{\mathbf{L}}{\rightarrow} \mathbb{Q}$ be an arrow of $\mathbf{FMLin}_{s \dot\cup \{a\}}$, then

$$\begin{aligned}
&\delta_a^+ !f \circ \widehat{\theta^!_{\mathbb{P}}}^{-1} \circ \{\cdot\}_{!\delta_a \mathbb{P}} \\
&= \theta_{!\mathbb{Q}} \circ \delta_a !f \circ \theta^{-1}_{!\mathbb{P}} \circ \widehat{\theta^!_{\mathbb{P}}}^{-1} \circ \{\cdot\}_{!\delta_a \mathbb{P}} && \text{by 3.3.5.14} \\
&= \theta_{!\mathbb{Q}} \circ \delta_a !f \circ \theta^{-1}_{!\mathbb{P}} \circ \{\cdot\}_{\delta_a !\mathbb{P}} \circ {\theta^!_{\mathbb{P}}}^{-1} && \text{by naturality of } \{\cdot\}_{\downarrow} \\
&= \theta_{!\mathbb{Q}} \circ \delta_a !f \circ \delta_a \{\cdot\}_{!\mathbb{P}} \circ {\theta^!_{\mathbb{P}}}^{-1} && \text{by 3.3.4.5} \\
&= \theta_{!\mathbb{Q}} \circ \delta_a \eta_{\mathbb{Q}} \circ \delta_a f \circ \delta_a i_{\mathbb{P}} \circ {\theta^!_{\mathbb{P}}}^{-1} && \text{by 3.4.4.14} \\
&= \theta_{!\mathbb{Q}} \circ \delta_a \eta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}} \circ i_{\delta_a \mathbb{P}} && \text{by 3.4.6.11} \\
&= \theta_{!\mathbb{Q}} \circ \delta_a \eta_{\mathbb{Q}} \circ \theta^{-1}_{\mathbb{Q}} \circ \delta_a^+ f \circ i_{\delta_a \mathbb{P}} && \text{by 3.3.5.14} \\
&= \widehat{\theta^!_{\mathbb{Q}}}^{-1} \circ \eta_{\delta_a \mathbb{Q}} \circ \delta_a^+ f \circ i_{\delta_a \mathbb{P}} && \text{by 3.4.7.6} \\
&= \widehat{\theta^!_{\mathbb{Q}}}^{-1} \circ !\delta_a^+ f \circ \{\cdot\}_{!\delta_a \mathbb{P}} && \text{by 3.4.4.14.}
\end{aligned}$$

But $\delta_a^+ !f \circ \widehat{\theta^!_{\mathbb{P}}}^{-1}$ and $\widehat{\theta^!_{\mathbb{Q}}}^{-1} \circ !\delta_a^+ f$ are both linear, so by 3.3.1.6 they are equal as required. □

It is now possible to show the main result of this section, namely that the abstract machinery of lemma 6.2.2.2 can be used to lift the binding adjunction $(-)^{\#a+} \dashv \delta_a^+ : \mathbf{FMLin}_s \leftrightarrows \mathbf{FMLin}_{s \dot\cup \{a\}}$ to a corresponding adjunction on the coKleisli categories of the respective ! comonads.

6.4.1.3 Lemma. *If $\mathbb{P}$ is an object of $\mathbf{FMLin}_s$ then define $F^{++}\mathbb{P} = \mathbb{P}^{\#a}$, and if $f : !\mathbb{P} \underset{\mathbf{L}}{\rightarrow} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_s$ then define $F^{++}f = f^{\#a+} \circ \widehat{\phi^!_{\mathbb{P}}}^{-1}$. If $\mathbb{P}$ is an object of $\mathbf{FMLin}_{s \dot\cup \{a\}}$ then define $G^{++}\mathbb{P} = \delta_a \mathbb{P}$, and if $f : !\mathbb{P} \underset{\mathbf{L}}{\rightarrow} \mathbb{Q}$ is an arrow of $\mathbf{FMLin}_{s \dot\cup \{a\}}$ then define $G^{++}f = \delta_a^+ f \circ \widehat{\theta^!_{\mathbb{P}}}^{-1}$. Define $\xi^{++} = \widehat{\xi} \circ \epsilon$ and $\zeta^{++} = \widehat{\zeta} \circ \epsilon_{(\delta_a -)^{\#a}}$. Then ξ^{++} and ζ^{++} are respectively the unit and counit of an adjunction*

$$F^{++} \dashv G^{++} : \mathrm{coKl}\big(! \text{ on } \mathbf{FMLin}_s\big) \leftrightarrows \mathrm{coKl}\big(! \text{ on } \mathbf{FMLin}_{s \dot\cup \{a\}}\big).$$

Proof. By the dual of lemma 6.2.2.2, the following argument is sufficient.

Left Triangle.

$$\begin{aligned}
&\epsilon^{\#a+}_{\mathbb{P}} \circ \widehat{\phi^!_{\mathbb{P}}}^{-1} \circ \{\cdot\}_{!(\mathbb{P}^{\#a})} \\
&= \epsilon^{\#a+}_{\mathbb{P}} \circ \{\cdot\}_{(!\mathbb{P})^{\#a}} \circ {\phi^!_{\mathbb{P}}}^{-1} && \text{by naturality of } \{\cdot\}_{\downarrow} \\
&= \phi_{\mathbb{P}} \circ \epsilon^{\#a}_{\mathbb{P}} \circ \phi^{-1}_{!\mathbb{P}} \circ \{\cdot\}_{(!\mathbb{P})^{\#a}} \circ {\phi^!_{\mathbb{P}}}^{-1} && \text{by 3.3.5.13} \\
&= \phi_{\mathbb{P}} \circ \epsilon^{\#a}_{\mathbb{P}} \circ \{\cdot\}^{\#a}_{!\mathbb{P}} \circ {\phi^!_{\mathbb{P}}}^{-1} && \text{by 3.3.3.6} \\
&= \phi_{\mathbb{P}} \circ i^{\#a}_{\mathbb{P}} \circ {\phi^!_{\mathbb{P}}}^{-1} && \text{by 3.4.4.16} \\
&= \phi_{\mathbb{P}} \circ \phi^{-1}_{\mathbb{P}} \circ i_{\mathbb{P}^{\#a}} && \text{by 3.4.5.7} \\
&= i_{\mathbb{P}^{\#a}} \\
&= \epsilon_{\mathbb{P}^{\#a}} \circ \{\cdot\}_{!(\mathbb{P}^{\#a})} && \text{by 3.4.4.16}
\end{aligned}$$

But $\epsilon_{\mathbb{P}\#a}$ and $\epsilon_{\mathbb{P}}^{\#a+} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}}$ are both linear, so they are equal by 3.3.1.6.

Right Triangle.

$$
\begin{array}{rll}
\multicolumn{2}{l}{\delta_a^{+}\epsilon_{\mathbb{P}} \circ \widehat{\theta_{\mathbb{P}}^{!\,-1}} \circ \{\cdot\}_{!\delta_a\mathbb{P}}} & \\
= & \delta_a^{+}\epsilon_{\mathbb{P}} \circ \{\cdot\}_{\delta_a!\mathbb{P}} \circ \theta_{\mathbb{P}}^{!\,-1} & \text{by naturality of } \{\cdot\}_{\downarrow} \\
= & \theta_{\mathbb{P}} \circ \delta_a\epsilon_{\mathbb{P}} \circ \theta_{!\mathbb{P}}^{-1} \circ \{\cdot\}_{\delta_a!\mathbb{P}} \circ \theta_{\mathbb{P}}^{!\,-1} & \text{by 3.3.5.14} \\
= & \theta_{\mathbb{P}} \circ \delta_a\epsilon_{\mathbb{P}} \circ \delta_a\{\cdot\}_{!\mathbb{P}} \circ \theta_{\mathbb{P}}^{!\,-1} & \text{by 3.3.4.5} \\
= & \theta_{\mathbb{P}} \circ \delta_a i_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{!\,-1} & \text{by 3.4.4.16} \\
= & \theta_{\mathbb{P}} \circ \theta_{\mathbb{P}}^{-1} \circ i_{\delta_a\mathbb{P}} & \text{by 3.4.6.11} \\
= & i_{\delta_a\mathbb{P}} & \\
= & \epsilon_{\delta_a\mathbb{P}} \circ \{\cdot\}_{!\delta_a\mathbb{P}} & \text{by 3.4.4.16}
\end{array}
$$

But $\epsilon_{\delta_a\mathbb{P}}$ and $\delta_a^{+}\epsilon_{\mathbb{P}} \circ \widehat{\theta_{\mathbb{P}}^{!\,-1}}$ are both linear, so they are equal by 3.3.1.6.

Left Pentagon.

$$
\begin{array}{rll}
\multicolumn{2}{l}{(!\eta_{\mathbb{P}})^{\#a+} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}} \circ \{\cdot\}_{!(\mathbb{P}^{\#a})}} & \\
= & (!\eta_{\mathbb{P}})^{\#a+} \circ \{\cdot\}_{(!\mathbb{P})^{\#a}} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by naturality of } \eta \\
= & \phi_{!!\mathbb{P}} \circ (!\eta_{\mathbb{P}})^{\#a} \circ \phi_{!\mathbb{P}}^{-1} \circ \{\cdot\}_{(!\mathbb{P})^{\#a}} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by 3.3.5.13} \\
= & \phi_{!!\mathbb{P}} \circ (!\eta_{\mathbb{P}})^{\#a} \circ \{\cdot\}_{!\mathbb{P}}^{\#a} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by 3.3.3.6} \\
= & \phi_{!!\mathbb{P}} \circ \eta_{!\mathbb{P}}^{\#a} \circ \eta_{\mathbb{P}}^{\#a} \circ i_{\mathbb{P}}^{\#a} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by 3.4.4.14} \\
= & \phi_{!!\mathbb{P}} \circ \eta_{!\mathbb{P}}^{\#a} \circ \{\cdot\}_{!\mathbb{P}}^{\#a} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by 3.4.4.10} \\
= & \phi_{!!\mathbb{P}} \circ \eta_{!\mathbb{P}}^{\#a} \circ \phi_{!\mathbb{P}}^{-1} \circ \{\cdot\}_{(!\mathbb{P})^{\#a}} \circ \phi_{\mathbb{P}}^{!\,-1} & \text{by 3.3.3.6} \\
= & \phi_{!!\mathbb{P}} \circ \eta_{!\mathbb{P}}^{\#a} \circ \phi_{!\mathbb{P}}^{-1} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}} \circ \{\cdot\}_{!(\mathbb{P}^{\#a})} & \text{by naturality of } \eta \\
= & \phi_{!!\mathbb{P}} \circ \eta_{!\mathbb{P}}^{\#a} \circ \phi_{!\mathbb{P}}^{-1} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}} \circ \eta_{\mathbb{P}^{\#a}} \circ i_{\mathbb{P}^{\#a}} & \text{by 3.4.4.10} \\
= & \widehat{\phi_{!\mathbb{P}}^{!\,-1}} \circ \eta_{(!\mathbb{P})^{\#a}} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}} \circ \eta_{\mathbb{P}^{\#a}} \circ i_{\mathbb{P}^{\#a}} & \text{by 3.4.7.3} \\
= & \widehat{\phi_{!\mathbb{P}}^{!\,-1}} \circ !\widehat{\phi_{\mathbb{P}}^{!\,-1}} \circ !\eta_{\mathbb{P}^{\#a}} \circ \{\cdot\}_{!(\mathbb{P}^{\#a})} & \text{by 3.4.4.14}
\end{array}
$$

But $(!\eta_{\mathbb{P}})^{\#a+} \circ \widehat{\phi_{\mathbb{P}}^{!\,-1}}$ and $\widehat{\phi_{!\mathbb{P}}^{!\,-1}} \circ !\widehat{\phi_{\mathbb{P}}^{!\,-1}} \circ !\eta_{\mathbb{P}^{\#a}}$ are both linear, so by 3.3.1.6 they are equal.

Right Pentagon.

$$\begin{aligned}
&\delta_a^+ !\eta_{\mathbb{P}} \circ \widehat{\theta^!_{\mathbb{P}}{}^{-1}} \circ \{\cdot\}_{!\delta_a\mathbb{P}} && \\
&= \delta_a^+ !\eta_{\mathbb{P}} \circ \{\cdot\}_{\delta_a!\mathbb{P}} \circ \theta^!_{\mathbb{P}}{}^{-1} && \text{by naturality of } \eta \\
&= \theta_{!!\mathbb{P}} \circ \delta_a !\eta_{\mathbb{P}} \circ \theta^{-1}_{!\mathbb{P}} \circ \{\cdot\}_{\delta_a!\mathbb{P}} \circ \theta^!_{\mathbb{P}}{}^{-1} && \text{by 3.3.5.14} \\
&= \theta_{!!\mathbb{P}} \circ \delta_a !\eta_{\mathbb{P}} \circ \delta_a\{\cdot\}_{!\mathbb{P}} \circ \theta^!_{\mathbb{P}}{}^{-1} && \text{by 3.3.4.5} \\
&= \theta_{!!\mathbb{P}} \circ \delta_a \eta_{!\mathbb{P}} \circ \delta_a \eta_{\mathbb{P}} \circ \delta_a i_{\mathbb{P}} \circ \theta^!_{\mathbb{P}}{}^{-1} && \text{by 3.4.4.14} \\
&= \theta_{!!\mathbb{P}} \circ \delta_a \eta_{!\mathbb{P}} \circ \delta_a\{\cdot\}_{!\mathbb{P}} \circ \theta^!_{\mathbb{P}}{}^{-1} && \text{by 3.4.4.10} \\
&= \theta_{!!\mathbb{P}} \circ \delta_a \eta_{!\mathbb{P}} \circ \theta^{-1}_{!\mathbb{P}} \circ \{\cdot\}_{\delta_a!\mathbb{P}} \circ \theta^!_{\mathbb{P}}{}^{-1} && \text{by 3.3.4.5} \\
&= \theta_{!!\mathbb{P}} \circ \delta_a \eta_{!\mathbb{P}} \circ \theta^{-1}_{!\mathbb{P}} \circ \widehat{\theta^!_{\mathbb{P}}{}^{-1}} \circ \{\cdot\}_{!\delta_a\mathbb{P}} && \text{by naturality of } \eta \\
&= \theta_{!!\mathbb{P}} \circ \delta_a \eta_{!\mathbb{P}} \circ \theta^{-1}_{!\mathbb{P}} \circ \widehat{\theta^!_{\mathbb{P}}{}^{-1}} \circ \eta_{\delta_a\mathbb{P}} \circ i_{\delta_a\mathbb{P}} && \text{by 3.4.4.10} \\
&= \widehat{\theta^!_{!\mathbb{P}}{}^{-1}} \circ \eta_{\delta_a!\mathbb{P}} \circ \widehat{\theta^!_{\mathbb{P}}{}^{-1}} \circ \eta_{\delta_a\mathbb{P}} \circ i_{\delta_a\mathbb{P}} && \text{by 3.4.7.6} \\
&= \widehat{\theta^!_{!\mathbb{P}}{}^{-1}} \circ !\widehat{\theta^!_{\mathbb{P}}{}^{-1}} \circ !\eta_{\delta_a\mathbb{P}} \circ \{\cdot\}_{!\delta_a\mathbb{P}} && \text{by 3.4.4.14}
\end{aligned}$$

But $\delta_a^+ !\eta_{\mathbb{P}} \circ \widehat{\theta^!_{\mathbb{P}}{}^{-1}}$ and $\widehat{\theta^!_{!\mathbb{P}}{}^{-1}} \circ !\widehat{\theta^!_{\mathbb{P}}{}^{-1}} \circ !\eta_{\delta_a\mathbb{P}}$ are both linear, so by 3.3.1.6 they are equal.

Unit Square.

$$\begin{aligned}
&\delta_a^+ \widehat{\phi^!_{\mathbb{P}}{}^{-1}} \circ \widehat{\theta^!_{\mathbb{P}\#a}{}^{-1}} \circ !\widehat{\xi_{\mathbb{P}}} \circ \{\cdot\}_{!\mathbb{P}} && \\
&= \delta_a^+ \widehat{\phi^!_{\mathbb{P}}{}^{-1}} \circ \widehat{\theta^!_{\mathbb{P}\#a}{}^{-1}} \circ \eta_{\delta_a(\mathbb{P}\#a)} \circ \widehat{\xi_{\mathbb{P}}} \circ i_{\mathbb{P}} && \text{by 3.4.4.14} \\
&= \theta_{(!\mathbb{P})^{\#a}} \circ \delta_a \widehat{\phi^!_{\mathbb{P}}{}^{-1}} && \\
&\qquad \circ \theta^{-1}_{!(\mathbb{P}^{\#a})} \circ \widehat{\theta^!_{\mathbb{P}\#a}{}^{-1}} \circ \eta_{\delta_a(\mathbb{P}^{\#a})} \circ \widehat{\xi_{\mathbb{P}}} \circ i_{\mathbb{P}} && \text{by 3.3.5.14} \\
&= \theta_{(!\mathbb{P})^{\#a}} \circ \delta_a \widehat{\phi^!_{\mathbb{P}}{}^{-1}} && \\
&\qquad \circ \theta^{-1}_{!(\mathbb{P}^{\#a})} \circ \theta_{!(\mathbb{P}^{\#a})} \circ \delta_a \eta_{\mathbb{P}^{\#a}} \circ \theta^{-1}_{\mathbb{P}^{\#a}} \circ \widehat{\xi_{\mathbb{P}}} \circ i_{\mathbb{P}} && \text{by 3.4.7.6} \\
&= \theta_{(!\mathbb{P})^{\#a}} \circ \delta_a \widehat{\phi^!_{\mathbb{P}}{}^{-1}} \circ \delta_a \eta_{\mathbb{P}^{\#a}} \circ \theta^{-1}_{\mathbb{P}^{\#a}} \circ \widehat{\xi_{\mathbb{P}}} \circ i_{\mathbb{P}} && \\
&= \theta_{(!\mathbb{P})^{\#a}} \circ \delta_a \phi_{!\mathbb{P}} \circ \delta_a(\eta_{\mathbb{P}}^{\#a}) \circ \delta_a \phi^{-1}_{\mathbb{P}} \circ \theta^{-1}_{\mathbb{P}^{\#a}} \circ \widehat{\xi_{\mathbb{P}}} \circ i_{\mathbb{P}} && \text{by 3.4.7.3} \\
&= \theta_{(!\mathbb{P})^{\#a}} \circ \delta_a \phi_{!\mathbb{P}} \circ \delta_a(\eta_{\mathbb{P}}^{\#a}) \circ \xi_{\widehat{\mathbb{P}}} \circ i_{\mathbb{P}} && \text{by 6.3.1.2} \\
&= \theta_{(!\mathbb{P})^{\#a}} \circ \delta_a \phi_{!\mathbb{P}} \circ \delta_a(\eta_{\mathbb{P}}^{\#a}) \circ \delta_a(i_{\mathbb{P}}^{\#a}) \circ \xi_{!\mathbb{P}} && \text{by naturality of } \xi \\
&= \theta_{(!\mathbb{P})^{\#a}} \circ \delta_a \phi_{!\mathbb{P}} \circ \delta_a(\{\cdot\}_{!\mathbb{P}}^{\#a}) \circ \xi_{!\mathbb{P}} && \text{by 3.4.4.10} \\
&= \theta_{(!\mathbb{P})^{\#a}} \circ \delta_a \{\cdot\}_{(!\mathbb{P})^{\#a}} \circ \xi_{!\mathbb{P}} && \text{by 3.3.3.6} \\
&= \{\cdot\}_{\delta_a((!\mathbb{P})^{\#a})} \circ \xi_{!\mathbb{P}} && \text{by 3.3.4.5} \\
&= \widehat{\xi_{!\mathbb{P}}} \circ \{\cdot\}_{!\mathbb{P}} && \text{by naturality of } \{\cdot\}
\end{aligned}$$

But $\delta_a^+ \widehat{\phi^!_{\mathbb{P}}{}^{-1}} \circ \widehat{\theta^!_{\mathbb{P}\#a}{}^{-1}} \circ !\widehat{\xi_{\mathbb{P}}}$ and $\widehat{\xi_{!\mathbb{P}}}$ are both linear, so by 3.3.1.6 they are equal.

Counit Square.

$$\begin{array}{lll}
\multicolumn{2}{l}{\widehat{\zeta_{!\mathbb{P}}} \circ \widehat{\theta^{!\,-1}_{\mathbb{P}}}^{\#a+} \circ \widehat{\phi^{!\,-1}_{\delta_a\mathbb{P}}} \circ \{\cdot\}_{!((\delta_a\mathbb{P})^{\#a})}} & \\
= & \widehat{\zeta_{!\mathbb{P}}} \circ \phi_{\delta_a!\mathbb{P}} \circ \widehat{\theta^{!\,-1}_{\mathbb{P}}}^{\#a} \circ \phi^{-1}_{!\delta_a\mathbb{P}} \circ \widehat{\phi^{!\,-1}_{\delta_a\mathbb{P}}} \circ \{\cdot\}_{!((\delta_a\mathbb{P})^{\#a})} & \text{by } 3.3.5.13 \\
= & \widehat{\zeta_{!\mathbb{P}}} \circ \phi_{\delta_a!\mathbb{P}} \circ \widehat{\theta^{!\,-1}_{\mathbb{P}}}^{\#a} \circ \phi^{-1}_{!\delta_a\mathbb{P}} \circ \widehat{\phi^{!\,-1}_{\delta_a\mathbb{P}}} \circ \eta_{(\delta_a\mathbb{P})^{\#a}} \circ i_{(\delta_a\mathbb{P})^{\#a}} & \text{by } 3.4.4.10 \\
= & \widehat{\zeta_{!\mathbb{P}}} \circ \phi_{\delta_a!\mathbb{P}} \circ \widehat{\theta^{!\,-1}_{\mathbb{P}}}^{\#a} \circ \phi^{-1}_{!\delta_a\mathbb{P}} \circ \phi_{!\delta_a\mathbb{P}} \circ \eta^{\#a}_{\delta_a\mathbb{P}} \circ \phi^{-1}_{\delta_a\mathbb{P}} \circ i_{(\delta_a\mathbb{P})^{\#a}} & \text{by } 3.4.7.3 \\
= & \widehat{\zeta_{!\mathbb{P}}} \circ \phi_{\delta_a!\mathbb{P}} \circ \widehat{\theta^{!\,-1}_{\mathbb{P}}}^{\#a} \circ \eta^{\#a}_{\delta_a\mathbb{P}} \circ \phi^{-1}_{\delta_a\mathbb{P}} \circ i_{(\delta_a\mathbb{P})^{\#a}} & \\
= & \widehat{\zeta_{!\mathbb{P}}} \circ \phi_{\delta_a!\mathbb{P}} \circ \theta^{\#a}_{!\mathbb{P}} \circ (\delta_a\eta_{\mathbb{P}})^{\#a} \circ \theta^{-1\,\#a}_{\mathbb{P}} \circ \phi^{-1}_{\delta_a\mathbb{P}} \circ i_{(\delta_a\mathbb{P})^{\#a}} & \text{by } 3.4.7.6 \\
= & \zeta_{!\widehat{\mathbb{P}}} \circ (\delta_a\eta_{\mathbb{P}})^{\#a} \circ \theta^{-1\,\#a}_{\mathbb{P}} \circ \phi^{-1}_{\delta_a\mathbb{P}} \circ i_{(\delta_a\mathbb{P})^{\#a}} & \text{by } 6.3.1.3 \\
= & \eta_{\mathbb{P}} \circ \zeta_{\widehat{\mathbb{P}}} \circ \theta^{-1\,\#a}_{\mathbb{P}} \circ \phi^{-1}_{\delta_a\mathbb{P}} \circ i_{(\delta_a\mathbb{P})^{\#a}} & \text{by nat. of } \zeta \\
= & \eta_{\mathbb{P}} \circ \widehat{\zeta_{\mathbb{P}}} \circ i_{(\delta_a\mathbb{P})^{\#a}} & \text{by } 6.3.1.3 \\
= & !\widehat{\zeta_{\mathbb{P}}} \circ \{\cdot\}_{!((\delta_a\mathbb{P})^{\#a})} & \text{by } 3.4.4.14
\end{array}$$

But $\widehat{\zeta_{!\mathbb{P}}} \circ \widehat{\theta^{!\,-1}_{\mathbb{P}}}^{\#a+} \circ \widehat{\phi^{!\,-1}_{\delta_a\mathbb{P}}}$ and $!\widehat{\zeta_{\mathbb{P}}}$ are both linear, so by 3.3.1.6 they are equal. This completes the proof. □

6.4.2 Binding in $\mathbf{FMCts}_s$, Concretely

Via the isomorphism $L_{\mathbf{Cts}} = K^{-1}_{\mathbf{Cts}}$ of 6.2.1.6 the adjunction $F^{++} \dashv G^{++}$ of lemma 6.4.1.3 gives rise to the adjunction

$$(-)^{\#a++} \dashv \delta_a^{++} : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s\dot{\cup}\{a\}} \tag{6.4.2.1}$$

of 3.4.8.26 as follows.

6.4.2.2 The Left Adjoint. If $\mathbb{P}$ is an object of $\mathbf{FMCts}_s$ then certainly $L_{\mathbf{Cts}}F^{++}K_{\mathbf{Cts}}\mathbb{P} = \mathbb{P}^{\#a}$. Suppose that $f : \mathbb{P} \underset{\mathbf{C}}{\rightarrow} \mathbb{Q}$ is an arrow of $\mathbf{FMCts}_s$, then the following commutes in $\mathbf{FMPre}_{s\dot{\cup}\{a\}}$ by naturality of $\{\cdot\}_{\downarrow}$, 3.4.5.7, 3.3.3.6, 3.3.5.13 and the freeness property 3.3.1.4.

$$\begin{tikzcd}
\widehat{!(\mathbb{P}^{\#a})} \arrow[r, "\widehat{\phi^{!\,-1}_{\mathbb{P}}}"] & \widehat{(!\mathbb{P})^{\#a}} \arrow[dr, "\phi^{-1}_{!\mathbb{P}}"] \arrow[rrdd, "(f\circ i_{\mathbb{P}})^{\dagger\,\#a+}"] & & \\
!(\mathbb{P}^{\#a}) \arrow[u, "\{\cdot\}_{!(\mathbb{P}^{\#a})}"] \arrow[r, "\phi^{!\,-1}_{\mathbb{P}}"'] \arrow[d, "i_{\mathbb{P}^{\#a}}"'] & (!\mathbb{P})^{\#a} \arrow[u, "\{\cdot\}_{(!\mathbb{P})^{\#a}}"'] \arrow[r, "\{\cdot\}^{\#a}_{!\mathbb{P}}"'] \arrow[d, "i^{\#a}_{\mathbb{P}}"'] & \widehat{!\mathbb{P}}^{\#a} \arrow[d, "(f\circ i_{\mathbb{P}})^{\dagger\,\#a}"] & \\
\widehat{\mathbb{P}^{\#a}} \arrow[r, "\phi^{-1}_{\mathbb{P}}"'] & \widehat{\mathbb{P}}^{\#a} \arrow[r, "f^{\#a}"'] & \widehat{\mathbb{Q}}^{\#a} \arrow[r, "\phi_{\mathbb{Q}}"'] & \widehat{\mathbb{Q}^{\#a}}
\end{tikzcd} \tag{6.4.2.3}$$

Therefore

$$\begin{aligned}
L_{\mathbf{Cts}}F^{++}K_{\mathbf{Cts}}f \circ i_{\mathbb{P}\#^a} &= (f \circ i_{\mathbb{P}})^{\dagger\#^a+} \circ \widehat{\phi^{!\;-1}_{\mathbb{P}}} \circ \eta_{\mathbb{P}\#^a} \circ i_{\mathbb{P}\#^a} \\
&= (f \circ i_{\mathbb{P}})^{\dagger\#^a+} \circ \widehat{\phi^{!\;-1}_{\mathbb{P}}} \circ \{\cdot\}_{!(\mathbb{P}\#^a)} && \text{by 3.4.4.10} \\
&= \phi_{\mathbb{Q}} \circ f^{\#^a} \circ \phi^{-1}_{\mathbb{P}} \circ i_{\mathbb{P}\#^a} && \text{by 6.4.2.3}
\end{aligned} \tag{6.4.2.4}$$

so that $L_{\mathbf{Cts}}F^{++}K_{\mathbf{Cts}}f = \phi_{\mathbb{Q}} \circ f^{\#^a} \circ \phi^{-1}_{\mathbb{P}} = f^{\#^a++}$ by 3.4.4.9 and 3.4.8.28.

6.4.2.5 The Right Adjoint. Similarly, if $\mathbb{P}$ is an object of $\mathbf{FMCts}_{s\dot\cup\{a\}}$ then $L_{\mathbf{Cts}}G^{++}K_{\mathbf{Cts}}\mathbb{P} = \delta_a\mathbb{P}$. Suppose that $f : \mathbb{P} \underset{\mathbf{C}}{\to} \mathbb{Q}$ is an arrow of $\mathbf{FMCts}_{s\dot\cup\{a\}}$, then the following diagram commutes in $\mathbf{FMPre}_s$ by naturality of $\{\cdot\}_{\downarrow}$, 3.4.6.11, 3.3.4.5, 3.3.5.14 and the freeness property 3.3.1.4.

$$\begin{tikzcd}
\widehat{!\delta_a\mathbb{P}} \arrow[r, "\widehat{\theta^{!\;-1}_{\mathbb{P}}}"] & \widehat{\delta_a!\mathbb{P}} \arrow[dr, "\theta^{-1}_{!\mathbb{P}}"] \arrow[rrdd, "\delta^+_a(f\circ i_{\mathbb{P}})^\dagger"] & & \\
!\delta_a\mathbb{P} \arrow[u, "\{\cdot\}_{!\delta_a\mathbb{P}}"] \arrow[r, "\theta^{!\;-1}_{\mathbb{P}}"'] \arrow[d, "i_{\delta_a\mathbb{P}}"'] & \delta_a!\mathbb{P} \arrow[u, "\{\cdot\}_{\delta_a!\mathbb{P}}"] \arrow[r, "\delta_a\{\cdot\}_{!\mathbb{P}}"'] \arrow[d, "\delta_a i_{\mathbb{P}}"'] & \delta_a\widehat{!\mathbb{P}} \arrow[d, "\delta_a(f\circ i_{\mathbb{P}})^\dagger"] & \\
\widehat{\delta_a\mathbb{P}} \arrow[r, "\theta^{-1}_{\mathbb{P}}"'] & \delta_a\widehat{\mathbb{P}} \arrow[r, "\delta_a f"'] & \delta_a\widehat{\mathbb{Q}} \arrow[r, "\theta_{\mathbb{Q}}"'] & \widehat{\delta_a\mathbb{Q}}
\end{tikzcd} \tag{6.4.2.6}$$

Therefore

$$\begin{aligned}
L_{\mathbf{Cts}}G^{++}K_{\mathbf{Cts}}f \circ i_{\delta_a\mathbb{P}} &= \delta^+_a(f \circ i_{\mathbb{P}})^\dagger \circ \widehat{\theta^{!\;-1}_{\mathbb{P}}} \circ \eta_{\delta_a\mathbb{P}} \circ i_{\delta_a\mathbb{P}} \\
&= \delta^+_a(f \circ i_{\mathbb{P}})^\dagger \circ \widehat{\theta^{!\;-1}_{\mathbb{P}}} \circ \{\cdot\}_{!\delta_a\mathbb{P}} && \text{by 3.4.4.10} \\
&= \theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}} \circ i_{\delta_a\mathbb{P}} && \text{by 6.4.2.6}
\end{aligned} \tag{6.4.2.7}$$

so that $L_{\mathbf{Cts}}G^{++}K_{\mathbf{Cts}}f = \theta_{\mathbb{Q}} \circ \delta_a f \circ \theta^{-1}_{\mathbb{P}} = \delta^{++}_a$ by 3.4.4.9 and 3.4.8.29.

6.4.2.8 The Unit.

$$L_{\mathbf{Cts}}\xi^{++}_{K_{\mathbf{Cts}}} = \widehat{\xi} \circ \epsilon \circ \eta = \widehat{\xi} \tag{6.4.2.9}$$

6.4.2.10 The Counit.

$$L_{\mathbf{Cts}}\zeta^{++}_{K_{\mathbf{Cts}}} = \widehat{\zeta} \circ \epsilon_{(\delta_a-)\#^a} \circ \eta_{(\delta_a-)\#^a} = \widehat{\zeta} \tag{6.4.2.11}$$

Therefore the adjunction $F^{++} \dashv G^{++}$ of lemma 6.4.1.3 gives rise to the adjunction $(-)^{\#^a++} \dashv \delta^{++}_a : \mathbf{FMCts}_s \leftrightarrows \mathbf{FMCts}_{s\dot\cup\{a\}}$ of 3.4.8.26 by way of the isomorphism $L_{\mathbf{Cts}} = K^{-1}_{\mathbf{Cts}}$ of 6.2.1.6 as required.

Chapter 7

Conclusion

7.1 Related and Future Work

7.1.1 Full Abstraction

The path semantics of (classical) HOPLA was shown to be fully abstract[20] by an argument which demonstrated inductively that all paths could be defined and distinguished by terms of the language. Initial suggestions towards a similar full abstraction theorem by Staton and Winskel (in personal correspondence[29, 30, 36]) has so far not yielded a positive result. As it stands it would appear that every path may be defined by a term of the language but they cannot all be distinguished. For example, it is suggested that it is impossible for a Nominal HOPLA term to tell apart the paths $P =_{\mathrm{def}} \langle\{a\}\rangle_{\varnothing}$ and $P_b =_{\mathrm{def}} \langle\{a\}\rangle_{\{b\}}$ (where $a \neq b$) unless it has b in its support. Since terms are always finitely-supported it follows that it is not possible to construct a term t which distinguishes P from all other paths: for any candidate t there exists a name $b \# t$ which means that t cannot tell the difference between P and P_b.

More precisely, the argument is to write $\tilde{\mathbb{A}} =_{\mathrm{def}} \bigoplus_{a\in\mathbb{A}} !\mathbb{O}$ and let

$$\begin{aligned} u_a &=_{\mathrm{def}} \textstyle\sum_{b\neq a\in\mathbb{A}} b\texttt{:!nil} \text{ for each } a \in \mathbb{A}, \\ v &=_{\mathrm{def}} \textstyle\sum_{b\in\mathbb{A}} b\texttt{:!nil}, \\ t_1 &=_{\mathrm{def}} \textstyle\sum_{a\in\mathbb{A}} \texttt{x(}u_a\texttt{:}\tilde{\mathbb{A}}\texttt{)} \text{ and} \\ t_2 &=_{\mathrm{def}} \texttt{x(}v\texttt{:}\tilde{\mathbb{A}}\texttt{)} + \textstyle\sum_{a\in\mathbb{A}} \texttt{x(}u_a\texttt{:}\tilde{\mathbb{A}}\texttt{)}. \end{aligned} \tag{7.1.1.1}$$

It is clear that it is possible to derive the following typing judgements.

$$\begin{array}{lll} \vdash u_a : \tilde{\mathbb{A}} & \qquad & \texttt{x} : \tilde{\mathbb{A}} \to !\mathbb{O} \vdash t_1 : !\mathbb{O} \\ \vdash v : \tilde{\mathbb{A}} & \qquad & \texttt{x} : \tilde{\mathbb{A}} \to !\mathbb{O} \vdash t_2 : !\mathbb{O} \end{array} \tag{7.1.1.2}$$

It is claimed that t_1 and t_2 are contextually equivalent: there is no context $C[-]$ such that the terms $\vdash C[t_1] : !\mathbb{O}$ and $\vdash C[t_2] : !\mathbb{O}$ behave differently. However the function space $\tilde{\mathbb{A}} \to !\mathbb{O}$ is isomorphic to $\mathbb{A} \to \{*\}$ and therefore to $\widehat{!\mathbb{A}^{\mathrm{op}}}$ by 3.4.8.2 and under this isomorphism the element $\{\mathbb{A}\}_\downarrow \in \widehat{!\mathbb{A}^{\mathrm{op}}}$ corresponds to an element $d \in \tilde{\mathbb{A}} \to !\mathbb{O}$ such that $[\![t_1]\!]d = \varnothing$ and $[\![t_2]\!]d = \{\varnothing\}$ so that $[\![t_1]\!] \neq [\![t_2]\!]$. It follows that d is not definable in Nominal HOPLA, and importantly that contextual equivalence does not coincide with denotational equivalence.

If it is unpalatable to take a sum over the set $\mathbb{A}$ of all names, notice that the argument above also applies to sums over any cofinite $B \subseteq_{\mathrm{fs}} \mathbb{A}$, since it rests only on the fact that B is infinite but finitely supported.

In general it is not computationally unreasonable to want to distinguish the paths $P_1 =_{\mathrm{def}} \langle F_1 \rangle_{s_1}$ and $P_2 =_{\mathrm{def}} \langle F_2 \rangle_{s_2}$ of type $!\mathbb{P}$. To see this, notice that $s_1 \cup s_2$ supports both P_1 and P_2 so that by lemma 3.4.3.10 there exist finite F_1' and F_2' such that $P_1 = \langle F_1' \rangle_{s_1 \cup s_2}$ and $P_2 = \langle F_2' \rangle_{s_1 \cup s_2}$. Distinguishing P_1 and P_2 therefore boils down to comparing the finite sets F_1' and F_2'.

Turning to the possibility of defining a distinguishing term, notice that

$$\begin{aligned} \langle F \rangle_s \subseteq x \quad &\Leftrightarrow \quad \forall \pi \mathrel{\#} s.\ \pi \cdot F \subseteq x \\ &\Leftrightarrow \quad \forall \pi \mathrel{\#} s.\ F \subseteq \pi \cdot x \\ &\Leftrightarrow \quad F \subseteq \textstyle\bigcap_{\pi \# s} \pi \cdot x \end{aligned} \tag{7.1.1.3}$$

which suggests that if t is a term then it might be worth defining a term $\texttt{[}t\texttt{]}_s$ whose denotational semantics is given by

$$[\![\,\texttt{[}t\texttt{]}_s]\!]\langle\gamma\rangle_\Gamma =_{\mathrm{def}} \bigcap_{\pi \# s} \pi \cdot [\![t]\!]\langle\gamma\rangle_\Gamma . \tag{7.1.1.4}$$

The proof that HOPLA was fully abstract constructed for each path p a 'producer' term t_p and a 'consumer' context $C_p[-]$ and it seems that that the same proof works for Nominal HOPLA extended by terms of the form $\texttt{[}t\texttt{]}_s$. The key alteration to the proof would be to define the 'consumer' of the path $\langle \{p_1, \ldots, p_n\} \rangle_s$ to be a context such as

$$\texttt{[}C_{p_1}\texttt{[[}-\texttt{]}_s\texttt{]} \;\texttt{>}\; \texttt{!(x}_1\texttt{:}\mathbb{P}\;\texttt{\#}\;\varnothing\texttt{)} \;\texttt{=>}\; \ldots \;\texttt{[}C_{p_n}\texttt{[[}-\texttt{]}_s\texttt{]} \;\texttt{>}\; \texttt{!(x}_n\texttt{:}\mathbb{P}\;\texttt{\#}\;\varnothing\texttt{)} \;\texttt{=>}\; \texttt{!nil]}\ldots\texttt{]} \tag{7.1.1.5}$$

although such a guess is not immediately obviously well-defined.

The denotational semantics given in 7.1.1.4 suggests an operational rule such as

$$\frac{\mathbb{P} : \pi \cdot t \xrightarrow{p} t' \text{ for all } \pi \mathrel{\#} s}{\mathbb{P} : \texttt{[}t\texttt{]}_s \xrightarrow{p} t'} \tag{7.1.1.6}$$

By an argument similar to that of 3.4.3.10 using the footprint lemma (lemma 2.2.1.6) it can be shown that this rule is effectively finitary: only finitely many

$\pi \# s$ must be checked before it is clear that the premise holds. Unfortunately this operational rule might break the adequacy result presented above: the term

$$t =_{\text{def}} u_a + a\texttt{:!}(\texttt{nil} + \texttt{nil}) \tag{7.1.1.7}$$

has $[\![t]\!] = \mathbb{A}$ and hence $[\![\texttt{[}t\texttt{]}_\varnothing]\!] = \mathbb{A}$ too. If the semantics were adequate, it would follow that for each $c \neq a$ it should be that $\tilde{\mathbb{A}} : \texttt{[}t\texttt{]}_\varnothing \xrightarrow{c\texttt{:!}} \texttt{nil}$; however $\tilde{\mathbb{A}} : (ac) \cdot t \xrightarrow{c\texttt{:!}} t'$ implies that $t' = \texttt{nil} + \texttt{nil}$ which is a contradiction.

It is not clear at this time how best to proceed to solve these discrepancies between the denotational and operational semantics. The full abstraction result for HOPLA gave rise to a characteristic modal logic for the language, and it is hoped that a similar result for Nominal HOPLA may suggest a similar characteristic *nominal* modal logic.

7.1.2 Relationships with New HOPLA

The language new-HOPLA of Zappa Nardelli and Winskel[38] was motivated by a similar idea to that of Nominal HOPLA: namely to design the language around universal constructions in a categorical setting that supported a notion of name generation. The emphasis of the work so far on new-HOPLA has been more concerned with its operational semantics and particularly its expressivity, which is in contrast with the present development of Nominal HOPLA. Indeed, Zappa Nardelli[39] demonstrates a number of results about operational equivalences for new-HOPLA and shows that this calculus is expressive enough to encode two variants of the π-calculus.

Because of the similarity between the origins of new-HOPLA and Nominal HOPLA, the languages have very similar operational semantics; the major difference is that new-HOPLA's judgements are indexed by a 'current' set of names whereas that information is unnecessary in the operational semantics of Nominal HOPLA. Because of this, much of the work on the expressivity of new-HOPLA should apply to Nominal HOPLA too. In particular it is believed that Nominal HOPLA can encode rich process calculi with name-generation such as the π-calculus, although this avenue of research has not yet been explored. Similarly, it would be interesting to check that the results about operational equivalences in new-HOPLA — such as that bisimilarity is a congruence[39] — also apply to Nominal HOPLA.

Attempts have been made to equip new-HOPLA with a denotational semantics by making use of the functor category $\mathbf{Lin}^{\mathbb{I}}$ in place of the category $\mathbf{Lin}$ that was used for HOPLA. Indeed, it is the structure of $\mathbf{Lin}^{\mathbb{I}}$ which motivated the design of the operational semantics for new-HOPLA. However it transpires that not all function spaces exist in $\mathbf{Lin}^{\mathbb{I}}$. It is possible that sufficiently many function spaces do exist in the functor-category setting but demonstrating this has proved remarkably delicate[35]. It is also possible that replacing $\mathbf{Lin}^{\mathbb{I}}$ with an internally-constructed version of **Lin** within the presheaf topos $\mathbf{Set}^{\mathbb{I}}$ would have provided a suitable setting for a denotational semantics for new-HOPLA, but the (notationally and conceptually) simpler setting of nominal set theory has helped to solve some of the problems associated with the functor-category-based domain theory. It would be interesting to characterise the type functors that correspond to the types of Nominal HOPLA via the equivalence of lemma 2.2.4.2, but this remains an open problem.

Nominal set theory also helped to clarify a suitable ! comonad that captured the notions of approximation and continuity for a domain theory with name-generation. In $\mathbf{Lin}^{\mathbb{I}}$ the ! functor was defined by

$$(!\mathbb{P})s =_{\text{def}} \{F \mid F \subseteq_{\text{fin}} \mathbb{P}s\} \tag{7.1.2.1}$$

but lemma 2.2.6.2 demonstrates that this picks out the internally *Dedekind* finite subsets of $\mathbb{P}$ which do not quite coincide with the internally isolated elements of $\widehat{\mathbb{P}}$ as desired. Because of this discrepancy it may be that the functor-category semantics also has a failure of continuity much like that described in section 3.4.1.

7.1.3 Even-More-Nominal HOPLA

A fundamental driving force in the study of nominal set theory is to avoid explicitly mentioning the 'current' set of names whenever possible, because it is usually not necessary to do so. Lemma 4.2.2.9 validates dropping the explicit supports in the typing rules for nominal HOPLA, but moreover it points the way towards dropping explicit supports throughout the whole denotational semantics developed here. Sadly, this lemma was formulated too late in the development of this thesis for it to have been possible to follow up this line of enquiry.

In personal correspondence, Pitts points out that it would be 'more nominal' to consider a categorical setting of the form $(\mathbf{FMPre}^{\#s})_{s\subseteq_{\text{fin}}\mathbb{A}}$, that is, categories indexed by the names that are fresh for — rather than that explicitly support — their objects and arrows. The key adjunction $(-)^{\#a} \dashv \delta_a$ appears in this setting too, in the form

$$(-)^{\#a} : \mathbf{FMPre}^{\#s\dot{\cup}\{a\}} \leftrightarrows \mathbf{FMPre}^{\#s} : \delta_a$$

and moreover here it is an *equivalence* of categories. Intuitively, this equivalence is related to the freedom in the choice of the primitive set $\mathbb{A}$ of names that was made at the very lowest level of the development of nominal set theory. It captures the idea that it is possible to add or remove a name from the chosen $\mathbb{A}$ without changing the resulting mathematical theory. Perhaps this equivalence also helps to explain the slightly mysterious δ_a functor: it describes the operation of removing the name a from sight.

7.1.4 Presheaf Semantics

The path semantics of HOPLA was motivated as a simplified version of the presheaf semantics of Cattani and Winskel[5]. The presheaf setting gives a much more detailed semantics: the denotation of a process records not only which paths it can perform, but also *how* those paths may be realised. For example, the presheaf semantics distinguishes the processes `!!nil` and `!nil` $+$ `!!nil` which are not bisimilar but which are nonetheless confounded by the path semantics. The cost of this extra detail is that it is more mathematically cumbersome to work directly with the presheaf semantics.

The development of a nominal path semantics suggests a similar approach to adjoining name generation to the presheaf semantics by working within a nominal setting. Of course it is currently far from clear what might be meant by a 'nominal setting' for the requisite category theory. Presumably it would be involve some notion of a permutation action on a category giving rise to a notion of finitely-supported objects and arrows and equivariant functors. It might then be necessary to design what is meant by the collection $\widehat{\mathcal{C}}$ of 'nominal presheaves' on a category $\mathcal{C}$, as well as to build an analogue of the key binding adjunction $(-) \otimes \mathbb{A} \dashv \delta$. Hopefully the important isomorphisms such as $\widehat{\delta\mathbb{P}} \cong \delta\widehat{\mathbb{P}}$ (cf. lemma 3.3.4.4) would also carry across to the categorical setting.

7.1.5 Nominal Domain Theory

The domain theory presented in this dissertation is rather simple compared with many modern domain theories, and it skirts around many of the subtleties that concern conventional practitioners of the subject. Perhaps from the viewpoint of domain theory the single most important lesson to be drawn from this thesis is that a great deal of care is required to capture a sensible notion of approximation in the presence of name generation, even in this simplified setting. Working within nominal set theory can help because it is sufficiently similar to conventional set theory that many standard arguments continue to apply in the nominal setting, so intuitions that have been developed without name generation are still valuable in this setting. It would be fascinating to develop a more general nominal domain theory, including results on solving recursive domain equations (including those that use a 'binding' domain constructor δ). It is likely that this, too, would be greatly helped by the development of a notion of nominal category theory. There are many more worthwhile avenues of research to explore!

7.2 Summary

A simple domain theory for concurrency has been developed within the theory of nominal sets. Elements of nominal sets are mathematical objects that have an intrinsic notion of 'free name' and 'bound name', so developing a theory using nominal sets has the effect of adjoining names to the mathematics. Here, processes denote the (nominal) sets of computation paths that they can follow, and the paths may be collected together into domains that support a Hoare or 'may do' style of nondeterminism. Because the processes denote nominal sets, the computation paths may mention names, and binding and unbinding the names gives a semantics for generating a new name as required.

The use of nominal sets mostly requires little alteration to the theory, but an important point where it makes a difference is that the conventional domain-theoretic notion of approximation — by directed sets — is not suitable in the presence of name generation because binding a name turns out not to preserve directed joins. The remedy is to consider approximation by directed sets that are also uniformly supported, i.e., whose elements only mention names drawn from some fixed finite set.

The described domain theory gives rise to an expressive metalanguage, Nominal HOPLA, based entirely on universal constructions in the model. Furthermore, the domain theory suggests an operational semantics for Nominal HOPLA which coincides closely with its denotational semantics. More precisely, it is shown that the denotational semantics is a sound and computationally adequate description of the operational semantics.

The relationship between the λ-calculus and the universal property of cartesian-closure is remarkably powerful because of its universality: the same machinery applies to give a notion of higher-order computation in any cartesian-closed setting. Similarly by making use of universal properties in the development of Nominal HOPLA it should be the case that analogous constructions should work similarly in different settings. It is also possible that the universality will help to unify various different approaches to giving a semantics to calculi that support concurrency and name generation by providing a common framework in which comparisons can be made.

Bibliography

[1] R. M. Amadio and P. L. Curien. *Domains and Lambda-Calculi.* Cambridge University Press, 1998.

[2] H. P. Barendregt. *The Lambda Calculus: its Syntax and Semantics.* North Holland, 1981.

[3] N. Benton, G. Bierman, V. de Paiva, and M. Hyland. Linear lambda-calculus and categorical models revisited. In E. Borgër, G. Jagër, Kleine H. Bunïng, S. Martini, and M. Richter, editors, *Proceedings of the Sixth Workshop on Computer Science Logic*, pages 61–84. Springer Verlag, 1993.

[4] G. L. Cattani, I. Stark, and G. Winskel. Presheaf models for the π-calculus. In *CTCS '97: Proceedings of the 7th International Conference on Category Theory and Computer Science*, pages 106–126, London, UK, 1997. Springer-Verlag.

[5] G. L. Cattani and G. Winskel. Profunctors, open maps and bisimulation. *Mathematical. Structures in Comp. Sci.*, 15(3):553–614, 2005.

[6] R. A. Clouston and A. M. Pitts. Nominal equational logic. *Electron. Notes Theor. Comput. Sci.*, 172:223–257, 2007.

[7] B. Day. On closed categories of functors. In Saunders M. Lane, editor, *Reports of the Midwest Category Seminar*, volume 137 of *Lecture Notes in Mathematics*, pages 1–38, Berlin–New York, 1970. Springer-Verlag.

[8] M. P. Fiore, E. Moggi, and D. Sangiorgi. A fully abstract model for the π-calculus. *Inf. Comput.*, 179(1):76–117, 2002.

[9] M. J. Gabbay. *A Theory of Inductive Definitions with Alpha-Equivalence.* PhD thesis, Cambridge University, 2001. Supervised by Andrew Pitts.

[10] M. J. Gabbay and A. M. Pitts. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing*, 13:341–363, 2001.

[11] Murdoch J. Gabbay and Aad Mathijssen. Nominal algebra. In *Proceedings of the 18th Nordic Workshop on Programming Theory (NWPT'06)*, 2006.

[12] B. Jacobs. Semantics of weakening and contraction. *Ann. Pure Appl. Logic*, 69(1):73–106, 1994.

[13] P. T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium vol. 1.* Oxford University Press, October 2002.

[14] P. T. Johnstone. *Sketches of an Elephant: A Topos Theory Compendium vol. 2.* Oxford University Press, November 2002.

[15] J. Lambek and P. J. Scott. *Introduction to Higher-Order Categorical Logic (Cambridge Studies in Advanced Mathematics).* Cambridge University Press, March 1988.

[16] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, parts I and II. Technical Report ECS-LFCS-89-85 and -86, University of Edinburgh, Laboratory for Foundations of Computer Science, 1989.

[17] E. Moggi. An abstract view of programming languages. Technical Report ECS-LFCS-90-113, University of Edinburgh, Laboratory for Foundations of Computer Science, 1990.

[18] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.

[19] M. Nygaard. *Domain Theory for Concurrency.* PhD thesis, University of Aarhus, Department of Computer Science, July 2003.

[20] M. Nygaard and G. Winskel. Domain theory for concurrency. *Theor. Comput. Sci.*, 316(1-3):153–190, 2004.

[21] F. J. Oles. Type algebras, functor categories and block structure. *Algebraic Methods in Semantics*, pages 543–573, 1985.

[22] A. M. Pitts. Alpha-structural recursion and induction. *Journal of the ACM*, 53:459–506, 2006.

[23] G. Plotkin. Pisa notes (on domain theory). 1983. Available from `http://homepages.inf.ed.ac.uk/gdp/publications/`.

[24] G. D. Plotkin. A powerdomain construction. *SIAM Journal on Computing*, 5(3):452–487, 1976.

[25] G. D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.

[26] U. Schöpp and I. Stark. A dependent type theory with names and binding. In *Computer Science Logic: Proceedings of the 18th International Workshop CSL 2004*, number 3210 in Lecture Notes in Computer Science, pages 235–249. Springer-Verlag, 2004.

[27] M. R. Shinwell. *The Fresh Approach: functional programming with names and binders.* PhD thesis, University of Cambridge, Computer Laboratory, February 2005. UCAM-CL-TR-618.

[28] I. Stark. A fully abstract domain model for the π-calculus. In *Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science*, pages 36–42. IEEE Computer Society Press, 1996.

[29] S. Staton. Question about Hopla. Personal communication by email, 22 September 2006.

[30] S. Staton. Op sem for full abstraction. Personal communication by email, 3 December 2008.

[31] J. Stoy. *Denotational semantics: The Scott-Strachey approach to programming language theory.* MIT Press.

[32] R. Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2:149–168, 1972.

[33] Christian Urban, Andrew M. Pitts, and Murdoch J. Gabbay. Nominal unification. *Theoretical Computer Science*, 323(1–3):473–497, 2004.

[34] G. Winskel. *The formal semantics of programming languages: an introduction.* MIT Press, Cambridge, MA, USA, 1993.

[35] G. Winskel. Name generation and linearity. In *LICS '05: Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science*, pages 301–310, Washington, DC, USA, 2005. IEEE Computer Society.

[36] G. Winskel. Personal communication, 3 December 2008.

[37] G. Winskel and K. G. Larsen. Using information systems to solve recursive domain equations effectively. Technical Report UCAM-CL-TR-51, University of Cambridge, Computer Laboratory, 1984.

[38] G. Winskel and F. Zappa Nardelli. new-HOPLA: a higher-order process language with name generation. In *Proc. of 3rd IFIP TCS*, pages 521–534, 2004.

[39] F. Zappa Nardelli. *De la sémantique des processus d'ordre supérieur.* PhD thesis, Université Paris, 2003.