Computing Science Group

Nominal Game Semantics

Nikos Tzevelekos

CS-RR-09-18

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford, OX1 3QD

Nominal Game Semantics

Nikos Tzevelekos

Brasenose College

University of Oxford

Trinity 2008

A thesis submitted for the degree of Doctor of Philosophy

Abstract

Game Semantics arguably stands for one of the most successful techniques in denotational semantics, having provided not only proper denotational, accurate models for a large variety of programming languages, but also new semantical tools for program verification and validation. Most of all, over the last couple of decades, game semantics has contributed a novel understanding of computations, namely as functions with inner structure, the latter being described as interaction between two players — the Program and the Environment.

On the other hand, Nominal Computation is a key theme within the Theory of Computation which has not been adressed semantically in a satisfactory manner. The significance of nominal computation is clearly depicted in the ubiquity of names in computational scenarios: names form the basis of many calculi of mobile processes; appear in network protocols and secure transactions; and are generally essential in programming for identifying variables, channels, threads, objects, codes, and many other sorts of name in disguise.

This thesis examines nominal game semantics, that is, game semantics *for* nominal computation. Our starting point is the basic nominal language, the ν-calculus, which we model in a basic category of nominal games. The construction of nominal games is based on recent advances in game semantics, and also on the theory of Nominal Sets, which serves as a general foundation for reasoning about names.

Our main focus is on languages extending the basic nominal language by use of names for general references and exceptions. These languages faithfully reflect the practice and reach the expressivity of programming languages such as ML; moreover, their full-abstraction problems had not been solved previously in a fully satisfactory manner. Such solutions we provide herein. We first devise abstract categorical models for these languages, and then construct fully abstract models in nominal games.

Preface to the Technical Report

This Report of December 2009 corrects discrepancies related to the terms M_2 and M_3 used in Chapters 4 and onwards (especially Section 5.2.6) in order to distinguish the examined nominal calculi. I am grateful to Andrzej Murawski for the many fruitful discussions, out of which those problems became apparent.

Acknowledgements

First, I would like to thank my supervisor, Samson Abramsky, for his constant encouragement, support and guidance, and his impeccable academic ethos. Furthermore, certain people have been particularly supportive of this work, thus greatly contributing to its successful completion; among them I would like to single out Andy Pitts, Luke Ong, Andrzej Murawski, Guy McCusker, Dan Ghica and Ian Stark. In addition, Andy was happy to offer advice on nominal matters; Guy readily advised on game-semantics; and Andrzej, Ian and Luke (the latter two acting as examiners) read the thesis and suggested several corrections and improvements. I would also like to thank Jim Laird, Paul Levy and Sam Sanjabi for fruitful discussions, suggestions and criticisms.

Many thanks go to my friends during these years in Oxford, and particularly to Elfy, Loukia, George, Nicholas, Antonis, Iris, Andria and Elina. I would also like to thank my family for their faith and support. I am deeply grateful to Note, for everything.

Finally, I would like to acknowledge the financial support of the Engineering and Physical Sciences Research Council, the Eugenides Foundation, the A. G. Leventis Foundation and Brasenose College.

Στη Νότα.

List of Figures

Contents

Chapter 1

Introduction

A focal point in Computer Science is the semantics of programs, i.e. *What does a program really mean?* A first answer to the question is given by means of the machine code produced by a compiler. However, this description is problematic if we are interested in a semantics independent of hardware and compiler design, revealing of the *essence of computation* hidden behind the implementation; a more abstract semantics is needed. In this direction, ***Operational Semantics*** considers programs as executing on an abstract, high-level computational environment. The semantics of a program is then its *observable behaviour* in this environment. Two programs are *observationally equivalent* if they have indistinguishable behaviours. This procedural description of computation at a level of abstraction that is both useful and intuitive is by and large thought of as giving the *intended semantics* of a programming language.

On the other hand, programs are expressive enough to be given a syntax-free description in an abstract mathematical domain. This method, called ***Denotational Semantics***, was pioneered by Strachey as a "mathematical semantics" of programming languages [Str66], and was substantiated through the work of Scott on Domain Theory [Sco70]. With operational semantics giving the intended program behaviour, a denotational model needs to capture both the programming language and its observational equivalence. The model is ***fully abstract*** if observational equivalence and denotational equality coincide through semantic translation.

The quest for fully abstract denotational semantics started with the purely functional language PCF, introduced by Plotkin [Plo77] and embodying the logic LCF of Scott [Sco93]. With PCF it was understood that for full-abstraction it was necessary to work in a domain of 'sequential' functions: the parallel nature of argument evaluation inherent in ordinary functions is simply impossible to capture with PCF. The problem was finally solved in the mid 90's independently by three teams of researchers: Abramsky, Jagadeesan and Malacaria [AJM00]; Hyland and Ong [HO00]; Nickau [Nic96]. Their models were based on ***Game Semantics***: computation was modelled by dynamic interaction between two participants, one of them representing the program and the other the environment. It was soon realised that the potential of game semantics was not confined to the semantics of PCF. The flexibility in applying and removing conditions from the rules of the games, along with the potentiality of altering the structure of the games themselves, allowed for the accurate modelling of a wide range of programming languages exhibiting various computational effects. This series of full-abstraction results established game models as a powerful paradigm within denotational semantics.

At around the same time that game semantics appeared on the scene, Pitts and Stark were focusing on a computational effect pervasive in computing, the use of ***names***, and examined a prototypical *nominal language*, the ν-calculus [PS93]. Names are syntactic atoms used to distinguish objects which are otherwise indistinguishable yet have distinct roles inside a computation; more importantly, names can be dynamically generated provoking a *local-state* effect. This latter feature along with mobility of names rendered the operational semantics of this seemingly simple language quite intricate.

The full-abstraction problem for the ν-calculus remained open for a decade. Meanwhile, Gabbay and Pitts [GP02] had introduced ***Nominal Sets*** as a general mathematical foundation for nominal structures, by revisiting the Fraenkel-Mostowski permutation models of ZFA discovered in the 20's and 30's. In 2004, Abramsky, Ghica, Murawski, Ong and Stark [AGM$^+$04], and independently Laird [Lai04], introduced ***Nominal Games*** for the semantical description of nominal computation; [AGM$^+$04] in particular proposed a fully abstract semantics for the ν-calculus. This thesis is a further investigation on nominal game semantics. We rectify the discrepancies arising in the original presentation of [AGM$^+$04] and then examine fully abstract semantics for languages with nominal general references and nominal exceptions.

1.1 Background Remarks

1.1.1 Nominal Languages

One of the most pervasive features in computation is the use of *names* to distinguish entities that are otherwise indistinguishable yet have distinct roles inside a computation. The names we focus on have no inner structure whatsoever: in Needham's taxonomy they correspond to 'pure names' [Nee93]. Moreover, following the *What's new* motto of Pitts and Stark [PS93], names can be

> *created with local scope, compared for equality, and passed around via function application.*

The above describes the basic nominal specification, which we may refer to as ***the nominal effect***. In programming languages, though, more specifications may be added so that names be used for channels, threads, references, codes, exceptions, etc. We refer to such languages generically as ***nominal languages***. The prototypical nominal language is the ν-calculus [PS93], which constitutes a call-by-value λ-calculus incorporating the basic nominal specification. Of the more sophisticated and more 'realistic' nominal languages, one that stands out is the π-calculus of Milner [SW01]. It is the paradigmatic language incorporating names-for-channels, providing a programming framework for concurrent processes intercommunicating through named channels.

Although constructed as simple computationally as possible, the ν-calculus exhibits a rather delicate behaviour, [Sta97]:

> *Functions may have local names that remain private and persist from one use of the function to the next; alternatively, names may be passed out of their original scope and can even outlive their creator. It is precisely this mobility of names that allows the nu-calculus to model issues of locality, privacy and non-interference.*

Hence, this seemingly plain language became of increasing importance to semanticists. Research focused primarily on the notion of observational equivalence, which resisted all attempts to be modelled accurately by use of ordinary (non-nominal) techniques, be they denotational or operational [Sta94, Sta96, Sta97, ZN03].

1.1.2 Nominal Sets

Invented in the 20's and 30's by Fraenkel and Mostowski as a model of set theory with atoms (ZFA), for showing its independence from the Axiom of Choice, nominal sets were re-introduced in the late 90's by Gabbay and Pitts [GP02, Pit03] as a general framework for the formal treatment of names and name-binding. The main objective was to exploit the rich structure of nominal sets for defining abstract syntaxes with variable binding which would incorporate 'clean' rules for structural recursion and induction. Nominal sets (and nominal abstract syntax) have been used extensively for building languages with symbolic-binding

constructors, for devising nominal theorem provers, and for studying programming language semantics: see [Che05] for a survey, and [Gab00, Che04, Shi05a] for thorough investigations.

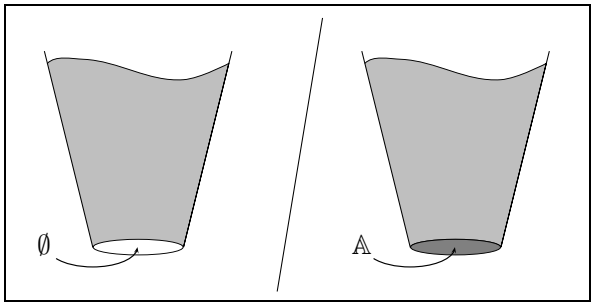


Figure 1.1: The von Neumann cumulative hierarchy of sets is built (in ZF) starting from the empty set and taking powersets, while for the Fraenkel-Mostowski hierarchy (the *basic Fraenkel model*, in ZFA) we start from the set of atoms $\mathbb{A}$ and take powersets constrained to elements of *finite support* (i.e. involving finitely many atoms).

Intuitively, nominal sets are sets whose elements involve a finite number of ***atoms***, and which can be acted upon by finite ***atom-permutations***. The expressivity thus obtained is remarkable: in the category of nominal sets, notions like atom-permutation, atom-freshness and atom-binding are essentially built inside the underlying structure. It is therefore self-suggesting to use nominal sets, with atoms playing the role of names, as **a general foundation for reasoning about names**.

1.1.3 Game Semantics

The first success of games in the semantics of programming languages was the fully abstract modelling of PCF (an idealised functional language with if-then-else, basic arithmetic and recursion) [AJM00, HO00, Nic96]. What distinguishes game semantics from traditional denotational semantics is its *intensional character*, which is expressed by the description of computation as a dynamic interaction between two *participants*: a Player and an Opponent. In particular, games are specified by ***plays***, that is, sequences of moves played in alternation by the two participants in relevant ***arenas*** of moves. Moves are in effect a representation of computation steps, and hence programs are modelled by ***strategies***, which are collections of instructions for Player on how to play a game on a specific arena.

Due to the intensional character of games and the great flexibility in applying and removing constraints from strategies, game semantics is able to capture accurately a wide range of computational effects and provide fully abstract, proper denotational models for a variety of languages. Some characteristic such constructions obtained from the models of PCF are the following. The first, second and fourth constructions, along with the model of PCF, produce what is known as *the semantic cube* [AM99].

1996. Removing the *innocence* condition from strategies, Abramsky and McCusker [AM97] were able to model fully abstractly Idealized Algol (IA) [Rey81], an extension of PCF with ground-type references. Moreover, the model of IA was shown to be *effectively presentable*, something that wasn't true for PCF — and for a good reason as shown by Loader [Loa01].

1997. Relaxing the *well-bracketing* condition, Laird [Lai97, Lai98] modelled fully abstractly PCF with non-local control flow.

1998. Abandoning the *visibility* condition, Abramsky, Honda and McCusker [AHM98] were able to provide a fully abstract model for a functional language with general, higher-order references.

1999. Abandoning the *determinacy* condition, Harmer and McCusker [HM99, Har99] produced a fully abstract model for finite-nondeterminism.

In addition to alterations to the constraints on strategies, variations to the notion of game itself proved also meaningful computationally.

1996. Departing from the PCF models, McCusker [McC96, McC98] introduced a game-setting with rich structure which allowed for the modelling of coproducts and the solution of domain equations on games. The result was a fully abstract model for FPC, an extension of PCF with coproducts and recursive types.

1997. Game models had thus far focused exclusively on call-by-name languages. At this point, Honda and Yoshida [HY99] showed that the current framework of games could be dualised appropriately, yielding the (equally primary) notion of call-by-value games. At the same time, Abramsky and McCusker [AM98] introduced a general categorical construction, the *family construction*, which built CBV models from CBN ones, and applied it to CBN games. The two constructions, which are essentially equivalent, yielded fully abstract semantics for the CBV version of PCF.

1997. The introduction by Hughes of the notion of *second-order move*, that is, a move introducing a new 'game-board', lead to the development of *hypergames* and to full-abstraction for system F [Hug97, Hug00].

The above results, which are by no means proposed as a complete enumeration of the achievements of games, built a significant momentum for game semantics and established it as a powerful paradigm in denotational semantics.

1.2 Thesis Outline

The thesis is structured as follows.

Chapter 2. In this chapter we present some background material necessary for the developments in the sequel. We start by presenting the theory of **nominal sets**, following the exposition of Pitts, and introducing the notion of **strong support**.
We continue by presenting the ν-calculus of Pitts and Stark, in a strongly supported version, and give some of its basic properties.
In the last part we give an exposition of the categorical notions of **monad** and **comonad**, and briefly examine the properties of monadic-comonadic (bi-Kleisli) categorical frameworks.

Chapter 3. In this chapter we present (AGMOS-style) **nominal games**. These are ordinary, call-by-value, stateful games cast inside the universe of strong nominal sets. We introduce the basic definitions of arenas, plays and strategies, and construct the basic category of nominal games $\mathcal{G}$. The rest of the chapter examines $\mathcal{G}$ and its subcategories $\mathcal{V}$ and $\mathcal{V}_t$ of innocent and total strategies respectively.

Chapter 4. This chapter introduces the $\nu\rho$-calculus, an extension of the ν-calculus with **nominal general references**, and models it fully abstractly in nominal games. The semantical part starts by presenting abstract categorical models, $\nu\rho$-models, which give correct interpretations of $\nu\rho$. We then build a concrete such model in the category $\mathcal{V}_t$, and finally obtain full-abstraction by restricting to **tidy strategies**, that is, strategies following a certain 'discipline' with regard to storage.

Chapter 5. In this chapter we examine fully abstract semantics for **nominal exceptions** in nominal games. We introduce the calculi $\nu\varepsilon$ and $\nu\varepsilon\rho$, which are extensions with nominal exceptions of the ν-calculus and the $\nu\rho$-calculus respectively. Categorical models for the calculi are presented: these are based on the fact that exceptions and local state

are separable effects, described abstractly by the notion of **precompound monad**. Finally, a specific fully abstract model for $\nu\varepsilon\rho$ is constructed in the subcategory of $\mathcal{V}_t$ containing **x-tidy strategies**, that is, tidy strategies following some extra discipline for exceptions.

1.2.1 Main Contributions

The contributions of this thesis, which have also appeared in [Tze07, Tze08], can be summarised as follows.

- The identification of strong nominal sets, that is, nominal sets with 'ordered involvement' of names, as the appropriate setting for nominal languages and (mainly) their semantics.
- The abstract categorical description of the nominal effect of nominal languages. Moreover, the categorical presentation of fully abstract models of languages with nominal references and exceptions, in the spirit of [Abr00].
- The formulation/rectification of nominal games and their use in constructing models of nominal references and exceptions.
- The introduction of game-disciplines to capture computation with names-as-references and names-as-exceptions, leading to definable and hence fully abstract game models.

Chapter 2

Names, Nu and Monads

In this chapter we present background material necessary for this thesis. In section 2.1 we present the theory of Nominal Sets of Gabbay and Pitts, which we use as a general foundation for constructions with names. In section 2.2 we present the basic nominal calculus, i.e. the ν-calculus of Pitts and Stark, and also a version of the latter with ordered local state, the $s\nu$-calculus. In the final section we expose some results regarding the categorical notions of monad and comonad.

2.1 Nominal Sets

The use of nominal sets in this thesis is limited, albeit essential. In particular, we express the intuitive notion of names by use of atoms, either in the syntax of our languages or in their denotational semantics. The features of nominal sets allowing this modelling are:

- all *finitely supported* constructions with atoms can be carried out in nominal sets,
- atom-equality is decidable,
- there is an infinite supply of (fresh) atoms.

Another appealing feature of nominal sets is the 'transparent' notion of atom-permutation, which we see as a 'clean' version of atom-substitution.

Perhaps it is not clear to the reader why nominal sets should be used — couldn't we simply model names by natural numbers? Indeed, numerals could be used for such semantical purposes (see e.g. [Lai08]), but they would constitute an over-specification: numerals carry a linear order and a bottom element which would need to be carefully nullified in the semantical definitions. Nominal sets factor out this burden by providing the minimal solution to specifying names; in this sense, nominal sets are *the intended model* for names.

Finally, note that nominal sets appear in the literature also as "FM-sets" (e.g. [GP02]), since they descend from Fraenkel–Mostowski permutation models of set theory with atoms. We will see more on that in section 2.1.3.

2.1.1 Definition

We are generally interested in languages having possibly infinitely many types of names, and hence we construct nominal sets over an ω-indexed family of sets of atoms. Thus, we generally follow the presentation of [Pit03], the only difference being that, since we are not interested in supplying "a first order theory of atoms and binding" (*Nominal Logic*), we base our presentation on finite permutations instead of swappings of atoms (following e.g. [PG00, Appendix]).

Let us fix a countably infinite family $(\mathbb{A}_i)_{i\in\omega}$ of pairwise disjoint, countably infinite sets of ***atoms***, and let us denote by $\mathrm{PERM}(\mathbb{A}_i)$ the group of finite permutations of $\mathbb{A}_i$. Atoms

are denoted by a, b, c and variants; permutations are denoted by π and variants; $\mathtt{id}$ is the identity permutation and $(a\ b)$ is the permutation swapping a and b (and fixing all others). We write $\mathbb{A}$ for the union of all the $\mathbb{A}_i$'s. We take

$$\mathrm{PERM}(\mathbb{A}) \triangleq \bigoplus_{i\in\omega} \mathrm{PERM}(\mathbb{A}_i) \tag{2.1}$$

to be the direct sum of the groups $\mathrm{PERM}(\mathbb{A}_i)$, so $\mathrm{PERM}(\mathbb{A})$ is a group of finite permutations of $\mathbb{A}$ which act separately on each constituent $\mathbb{A}_i$. For each $S \subseteq \mathbb{A}$ we let

$$\mathtt{fix}(S) \triangleq \{\pi \in \mathrm{PERM}(\mathbb{A}) \mid \forall a \in S.\, \pi(a) = a\} \tag{2.2}$$

and say that a permutation π *fixes* S if $\pi \in \mathtt{fix}(S)$.

Recall that $\mathrm{PERM}(\mathbb{A})$ being a direct sum means that each $\pi \in \mathrm{PERM}(\mathbb{A})$ is an ω-indexed list of permutations, $\pi \in \prod_{i\in\omega} \mathrm{PERM}(\mathbb{A}_i)$, and that $(\pi)_i \neq \mathtt{id}_{\mathbb{A}_i}$ holds for finitely many indices i. If $(\pi)_i \neq \mathtt{id}_{\mathbb{A}_i}$ holds for exactly one index i then we call π a *basic permutation*, and $(\pi)_i$ the *basic component* of π.

Fact 2.1 If $\pi \in \mathrm{PERM}(\mathbb{A})$ then,

- there exist basic permutations $\pi_1, ..., \pi_n$ such that $\pi = \mathtt{id} \circ \pi_1 \circ \cdots \circ \pi_n$,
- there exist basic permutations $\pi_1, ..., \pi_n$ such that the basic component of each π_i is a swapping, and $\pi = \mathtt{id} \circ \pi_1 \circ \cdots \circ \pi_n$,
- for any $S \subseteq \mathbb{A}$, if $\pi \in \mathtt{fix}(S)$ then there exist basic permutations $\pi_1, ..., \pi_n$ such that the basic component of each π_i is a swapping of atoms outside S, and $\pi = \mathtt{id} \circ \pi_1 \circ \cdots \circ \pi_n$.

We will therefore abandon the list-representation of permutations and — with a slight abuse of notation which identifies a basic permutation with its basic component — we will write (non-uniquely) each permutation π as a finite composition $\pi_1 \circ \cdots \circ \pi_n$ such that each π_i belongs to some $\mathrm{PERM}(\mathbb{A}_j)$.

We proceed to nominal sets. As seen in the following definition, the notion of finite support is central to our presentation. More general supports have been examined in [Gab02, Che04]; in the latter work it is shown that the notion of *support ideals* completely corresponds to the axioms of Nominal Logic. But these matters will not concern us here since all our constructions entail finitely many atoms.

Definition 2.2 (Nominal Set on $\mathbb{A}$) A nominal set X is a set $|X|$ (usually denoted X) equipped with an action of $\mathrm{PERM}(\mathbb{A})$, that is, a function $\_ \circ \_ : \mathrm{PERM}(\mathbb{A}) \times X \to X$ such that, for any $\pi, \pi' \in \mathrm{PERM}(\mathbb{A})$ and $x \in X$,

$$\pi \circ (\pi' \circ x) = (\pi \circ \pi') \circ x\,, \qquad \mathtt{id} \circ x = x\,.$$

Moreover, for any $x \in X$ there is a finite set $S \subseteq \mathbb{A}$ such that

$$\mathtt{fix}(S) \subseteq \{\pi \in \mathrm{PERM}(\mathbb{A}) \mid \pi \circ x = x\}\,.$$

We say that S *supports* x. ▲

Concretely, a set $S \subseteq \mathbb{A}$ supports some $x \in X$ if, for all permutations π,

$$(\forall a \in S.\, \pi(a) = a) \implies \pi \circ x = x\,.$$

For example, $\mathbb{A}$ with the action of permutations being simply permutation-application is a nominal set.

As shown below, finite support is closed under intersection. Hence, each element x of a nominal set X has least finite support, called ***the support of*** x:

$$\mathtt{S}(x) \triangleq \bigcap \{S \subseteq_{\mathsf{fin}} \mathbb{A} \mid S \text{ supports } x\}\,. \tag{2.3}$$

For example, for each atom $a \in \mathbb{A}$, $\mathtt{S}(a) = \{a\}$. We say that a ***is fresh for*** x, written $a \# x$, if $a \notin \mathtt{S}(x)$. x is called ***equivariant*** if it has empty support.

Proposition 2.3 *Let X be a nominal set and $x \in X$. For any finite $S \subseteq \mathbb{A}$, S supports x iff*

$$\forall a, a' \in (\mathbb{A} \setminus S).\ (a\ a') \circ x = x\,.$$

Moreover, if finite $S, S' \subseteq \mathbb{A}$ support x then $S \cap S'$ also supports x. Finally,

$$\mathtt{S}(x) = \{a \in \mathbb{A} \mid \text{for infinitely many } b.\ (a\ b) \circ x \neq x\}\,.$$

Proof: For the first claim we need only show that S supports x if $(a\ a') \circ x = x$ for all atoms a, a' outside S. Assume the latter condition holds and take any $\pi \in \mathtt{fix}(S)$. By fact 2.1, $\pi = \mathtt{id} \circ \pi_1 \circ \cdots \circ \pi_n$ with each π_i being a swapping of atoms outside S, and hence $\pi \circ x = x$. Now, if finite $S, S' \subseteq \mathbb{A}$ support x then take any distinct $a, a' \notin (S \cap S')$. For any $b \notin S \cup S' \cup \{a, a'\}$, $(a\ a') \circ x = (a\ b) \circ (a'\ b) \circ (a\ b) \circ x = x$ since $(a\ b) \circ x = x = (a'\ b) \circ x$. Hence, $S \cap S'$ supports x.
Finally, let

$$A \triangleq \{a \in \mathbb{A} \mid \text{for infinitely many } b.\ (a\ b) \circ x \neq x\}\,.$$

If $a \in A \setminus \mathtt{S}(x)$ then there are infinitely many b such that $(a\ b) \circ x \neq x$ and, since $\mathtt{S}(x)$ is finite, there is such a $b \notin \mathtt{S}(x)$, $\lightning$ as $\mathtt{S}(x)$ supports x. Hence, $A \subseteq \mathtt{S}(x)$. Conversely, it suffices to show that $(a\ a') \circ x = x$ for all distinct $a, a' \notin A$. But $a, a' \notin A$ implies that, for cofinitely many b, $(a\ b) \circ x = x = (a'\ b) \circ x$. Take some $b \neq a, a'$ of the cofinitely many; we have

$$(a\ a') \circ x = (a\ b) \circ (a'\ b) \circ (a\ b) \circ x = x\,.$$

■

From the last part of the proposition we have:

$$\begin{aligned} a \# x &\iff \text{for cofinitely many } b.\ (a\ b) \circ x = x \\ &\overset{\text{by defn}}{\iff} \mathrel{\text{И}} b \in \mathbb{A}.\ (a\ b) \circ x = x \end{aligned} \tag{2.4}$$

The "fresh" quantifier И, introduced in [GP02], quantifies over cofinitely many atoms, i.e.

$$\mathrel{\text{И}} a \in \mathbb{A}.\ \phi(a) \iff^{\triangle} \text{for cofinitely many } a \in \mathbb{A}.\ \phi(a)\,. \tag{И}$$

A subtlety here is that the holes in ϕ must all be of the same atom-type, say i, and that, in fact, we mean "for cofinitely many $a \in \mathbb{A}_i$".

Example 2.4 There are several ways to obtain new nominal sets from given nominal sets X and Y:

- The disjoint union $X \uplus Y$ with permutation-action inherited from X and Y is a nominal set. The construction easily extends to infinite disjoint union.
- The cartesian product $X \times Y$ with permutations acting componentwise is a nominal set; if $(x, y) \in X \times Y$ then $\mathtt{S}(x, y) = \mathtt{S}(x) \cup \mathtt{S}(y)$.
- The fs-powerset $\mathcal{P}_{\mathsf{fs}}(X)$, that is, the set of subsets of X which have finite support, with permutations acting elementwise.
- $X' \subseteq X$ is a ***nominal subset*** of X if X' is closed under permutations, these acting as on X.
- The fs-function space $X \to_{\mathsf{fs}} Y$, that is, the set of functions from X to Y with finite support: $X \to_{\mathsf{fs}} Y \triangleq \{f \in \mathcal{P}_{\mathsf{fs}}(X \times Y) \mid f \text{ a function with domain } X\}$.

Example 2.5 Apart from $\mathbb{A}$, some standard nominal sets are the following.

- Using products and infinite unions we obtain the nominal set:

$$\mathbb{A}^{\#} \triangleq \bigcup_{n\in\omega} \{\, a_1 \dots a_n \mid \forall i,j \in 1..n.\ a_i \in \mathbb{A} \wedge (i \neq j \implies a_i \neq a_j) \,\}\,, \tag{2.5}$$

  that is, the set of ***finite lists of distinct atoms***. Such lists we denote by $\vec{a}, \vec{b}, \vec{c}$ and variants. For notational economy, we write $a \in \vec{a}$ for $a \in \mathtt{S}(\vec{a})$. Moreover, for each $\vec{a} \in \mathbb{A}^{\#}$ we set:

$$\mathbb{A}^{\vec{a}} \triangleq \{\, \pi \circ \vec{a} \mid \pi \in \mathrm{PERM}(\mathbb{A}) \,\}\,. \tag{2.6}$$

  Finally, for $\vec{a}, \vec{b} \in \mathbb{A}^{\#}$ we write:

  - $\vec{a} \leq \vec{b}$ if $\vec{a}$ is a prefix of $\vec{b}$,
  - $\vec{a} \preceq \vec{b}$ if $\vec{a}$ is a (not necessarily contiguous) sublist of $\vec{b}$,
  - $\vec{a} \subseteq \vec{b}$ if $\mathtt{S}(\vec{a}) \subseteq \mathtt{S}(\vec{b})$.

- The fs-powerset $\mathcal{P}_{\mathsf{fs}}(\mathbb{A})$ is the set of finite and cofinite sets of atoms, and has $\mathcal{P}_{\mathsf{fin}}(\mathbb{A})$ as a nominal subset (the set of finite sets of atoms).

For X and Y nominal sets, a relation $\mathcal{R} \subseteq X \times Y$ is a ***nominal relation*** if it is a nominal subset of $X \times Y$. Concretely, $\mathcal{R}$ is a nominal relation iff, for any permutation π and $(x, y) \in X \times Y$,

$$x \mathcal{R} y \iff (\pi \circ x) \mathcal{R} (\pi \circ y)\,. \tag{2.7}$$

For example, $\# \subseteq \mathbb{A} \times X$ is a nominal relation: for all relevant a, x, π,

$$\begin{aligned} a \# x &\implies \mathit{\forall} b \in \mathbb{A}.\ (a\ b) \circ x = x \implies \mathit{\forall} b \in \mathbb{A}.\ \pi \circ (a\ b) \circ x = \pi \circ x \\ &\implies \mathit{\forall} b \in \mathbb{A}.\ (\pi(a)\ \pi(b)) \circ \pi \circ x = \pi \circ x \implies \mathit{\forall} b' \in \mathbb{A}.\ (\pi(a)\ b') \circ \pi \circ x = \pi \circ x \\ &\implies \pi(a) \# \pi \circ x\,. \end{aligned}$$

From nominal relations we proceed to *nominal functions* and the ***category of nominal sets***.

Definition 2.6 (The category Nom) A function $f : X \to Y$ is a ***nominal function*** if, for any $\pi \in \mathrm{PERM}(\mathbb{A})$ and $x \in X$,

$$f(\pi \circ x) = \pi \circ f(x)\,.$$

We let **Nom** be the category of nominal sets and nominal functions. ▲

Thus, nominal functions are fs-functions with empty support. For example, the support function $\mathtt{S}(\_) : X \longrightarrow \mathcal{P}_{\mathsf{fin}}(\mathbb{A})$ is a nominal function since $\mathtt{S}(\pi \circ x) = \pi \circ \mathtt{S}(x)$.

Nom inherits rich structure from **Set** and is in particular a topos. More importantly, it contains atom-abstraction mechanisms. The mechanism which triggered the study of nominal sets in programming is the following. For any nominal set X, any $x \in X$ and any $a \in \mathbb{A}$, we can abstract a from x by forming

$$\langle a \rangle x \triangleq \{(b, y) \in \mathbb{A} \times X \mid (b = a \vee b \# x) \wedge y = (a\ b) \circ x\}\,.$$

The abstraction takes the orbit of (a, x) under all swappings of a for fresh atoms. In λ-calculus terminology, $\langle a \rangle x$ is literally the α-equivalence class of (a, x) (that is, with regard to the abstraction of a). Hence, it is not difficult to see that $\mathtt{S}(\langle a \rangle x) = \mathtt{S}(x) \setminus \{a\}$. Moreover, $\pi \circ \langle a \rangle x = \langle \pi \circ a \rangle (\pi \circ x)$ and therefore we can define the nominal set $\langle \mathbb{A} \rangle X \subseteq \mathcal{P}_{\mathsf{fs}}(\mathbb{A} \times X)$ of abstracted elements as

$$\langle \mathbb{A} \rangle X \triangleq \{\langle a \rangle x \mid a \in \mathbb{A} \wedge x \in X\}\,,$$

and atom-abstraction as an arrow $\langle \_ \rangle \_ : \mathbb{A} \times X \longrightarrow \langle \mathbb{A} \rangle X$ in **Nom**.

However, in this thesis we are not interested in treating name- and variable-abstractions nominally, and therefore we will not use the above form of abstraction. The abstraction mechanism which is useful to us, instead of abstracting specified atoms from x, abstracts all atoms outside a specified subset of $\mathtt{S}(x)$. It is therefore similar to the abstaction mechanisms used in [AGM+04, Tze07].

Definition 2.7 (Support abstraction) Let X be a nominal set and $x \in X$. For any finite $S \subseteq \mathbb{A}$, we can *abstract x to S*, by forming

$$[x]_S \triangleq \{y \in X \mid \exists \pi \in \mathtt{fix}(S \cap \mathtt{S}(x)).\, y = \pi \circ x\}\,.$$

▲

This form of abstraction restricts the support of x to $S \cap \mathtt{S}(x)$ by appropriate orbiting of x (and note that $[x]_S \in \mathcal{P}_{\mathsf{fs}}(X)$). This is shown in the following lemma, along with the fact that $[\_]\_$ is itself nominal.

Lemma 2.8 *For any $x \in X$, $S \subseteq_{\mathsf{fin}} \mathbb{A}$ and $\pi \in \mathrm{PERM}(\mathbb{A})$,*

- $\pi \circ [x]_S = [\pi \circ x]_{\pi \circ S}$,
- $\mathtt{S}([x]_S) = \mathtt{S}(x) \cap S$.

Proof: For the first clause, we have:

$$\begin{aligned}
y \in \pi \circ [x]_S &\overset{\exists \pi'}{\Longrightarrow} y = \pi \circ \pi' \circ x \wedge \forall a \in S \cap \mathtt{S}(x).\, \pi' \circ a = a \\
&\Longrightarrow y = (\pi \circ \pi' \circ \pi^{-1}) \circ \pi \circ x \wedge \forall a \in S \cap \mathtt{S}(x).\, (\pi \circ \pi' \circ \pi^{-1}) \circ \pi \circ a = \pi \circ a \\
&\Longrightarrow y = (\pi \circ \pi' \circ \pi^{-1}) \circ \pi \circ x \wedge \forall a' \in \pi \circ (S \cap \mathtt{S}(x)).\, (\pi \circ \pi' \circ \pi^{-1}) \circ a' = a' \\
&\Longrightarrow y \in [\pi \circ x]_{\pi \circ S} \qquad (\textit{note } \pi \circ (S \cap \mathtt{S}(x)) = (\pi \circ S) \cap \mathtt{S}(\pi \circ x)\ )\,, \\
z \in [\pi \circ x]_{\pi \circ S} &\overset{\exists \pi'}{\Longrightarrow} z = \pi' \circ \pi \circ x \wedge \forall a' \in \pi \circ (S \cap \mathtt{S}(x)).\, \pi' \circ a' = a' \\
&\Longrightarrow z = \pi' \circ \pi \circ x \wedge \forall a \in S \cap \mathtt{S}(x).\, \pi' \circ \pi \circ a = \pi \circ a \\
&\Longrightarrow z = \pi \circ (\pi^{-1} \circ \pi' \circ \pi) \circ x \wedge \forall a \in S \cap \mathtt{S}(x).\, (\pi^{-1} \circ \pi' \circ \pi) \circ a = \pi^{-1} \circ \pi \circ a = a \\
&\Longrightarrow z \in \pi \circ [x]_S\,.
\end{aligned}$$

Note that $\forall \pi.\, \pi \circ [x]_S = [\pi \circ x]_{\pi \circ S}$ implies $\mathtt{S}([x]_S) \subseteq \mathtt{S}(x) \cup S$.
For the second clause, assume $a \in S \cap \mathtt{S}(x)$ and $a \# [x]_S$. Then, for any $b \# x, S$, $(a\ b) \circ [x]_S = [x]_S$, and hence $x \in (a\ b) \circ [x]_S$. This means there exists $\pi \in \mathtt{fix}(S \cap \mathtt{S}(x))$ such that $x = (a\ b) \circ \pi \circ x$, and therefore $a \in \mathtt{S}(x)$ implies

$$b = (a\ b) \circ \pi \circ a \in \mathtt{S}((a\ b) \circ \pi \circ x) = \mathtt{S}(x)\,,$$

↯ to $b \# x$. Hence, $S \cap \mathtt{S}(x) \subseteq \mathtt{S}([x]_S)$.
For the converse, for any $a, b \notin S \cap \mathtt{S}(x)$, we have

$$(a\ b) \circ [x]_S = \{(a\ b) \circ \pi \circ x \mid \pi \in \mathtt{fix}(S \cap \mathtt{S}(x))\} = \{\pi \circ x \mid \pi \in \mathtt{fix}(S \cap \mathtt{S}(x))\} = [x]_S\,,$$

and hence $\mathtt{S}([x]_S) \subseteq S \cap \mathtt{S}(x)$. ■

Two particular subcases of support abstraction are of interest. First, in case $S \subseteq \mathtt{S}(x)$, the abstraction becomes

$$[x]_S = \{y \in X \mid \exists \pi \in \mathtt{fix}(S).\, y = \pi \circ x\}\,. \qquad (*)$$

This is the mechanism used in [Tze07].[1] Note that if $S \not\subseteq \mathtt{S}(x) \wedge \mathtt{S}(x) \not\subseteq S$ then $(*)$ does not yield $\mathtt{S}([x]_S) = S \cap \mathtt{S}(x)$. Note also (proof left as exercise) that if $S \subseteq \mathtt{S}(x) \cap \mathtt{S}(y)$ then

$$[x]_S = [y]_S \iff y \in [x]_S\,. \qquad (2.8)$$

The other subcase is the simplest possible, that is, of S being empty; it turns out that this is all we need from support abstractions in this thesis. We define:

$$[x] \triangleq \{y \in X \mid \exists \pi.\, y = \pi \circ x\}\,. \qquad (2.9)$$

[1] The mechanism used in [AGM+04] is $[x]_S \triangleq \{(y, S) \mid \exists \pi \in \mathtt{fix}(S).\, y = \pi \circ x\}$, and is equivalent to the other two in case $S \subseteq \mathtt{S}(x)$, but not in general.

2.1.2 Strong support

Nominal sets describe a framework of objects built around a finite (or cofinite) amount of atoms. The framework does not specify how these atoms are present inside an object's structure, so atoms may appear in an 'unordered' fashion, as for example in the set $\{a, b\}$. The distinction between ordered and unordered involvement of atoms can be formally seen in the definition of support. In particular, we have seen that a set S supports x if

$$(\forall a \in S.\, \pi(a) = a) \implies \pi \circ x = x\,.$$

Ordered involvement then means that the reverse implication is also true. This notion we call ***strong support***.

Definition 2.9 For any nominal set X, any $x \in X$ and any $S \subseteq \mathbb{A}$, S ***strongly supports*** x if

$$\mathtt{fix}(S) = \{\, \pi \in \mathrm{PERM}(\mathbb{A}) \mid \pi \circ x = x \,\}\,.$$

We say that X is a ***strong nominal set*** if all its elements have strong support. ▲

Thus, the set $\{a, b\}$ does not support $\{a, b\}$ strongly, since the permutation $(a\ b)$ does not fix $\{a, b\}$,[2] but still $(a\ b) \circ \{a, b\} = \{a, b\}$. On the other hand, $\{a, b\}$ strongly supports the list ab. In fact, all finite lists of (distinct) atoms have strong support, and therefore $\mathbb{A}^{\#}$ is a strong nominal set.

The notion of strong support is stronger than that of support, as we saw in the example of $\{a, b\}$. Nonetheless, strong support coincides with weak support when the former exists.

Proposition 2.10 *If X is a nominal set and $x \in X$ has strong support S then $S = \mathtt{S}(x)$.*

Proof: By definition, S supports x, so $\mathtt{S}(x) \subseteq S$. Now suppose there exists $a \in S \setminus \mathtt{S}(x)$. For any fresh b, $(a\ b)$ fixes $\mathtt{S}(x)$ but not S, so it doesn't fix x, $\lightning$. ■

Hence, $\mathcal{P}_{\mathsf{fin}}(\mathbb{A})$ is not a strong nominal set (but $\mathbb{A}^{\#}$ is). The main reason for using strong nominal sets is the following result.

Lemma 2.11 (Strong support lemma) *Let X be a strong nominal set and $x_1, x_2, y_1, y_2, z_1, z_2 \in X$. Suppose also that, for some $S \subseteq_{\mathsf{fin}} \mathbb{A}$,*

$$S \subseteq \mathtt{S}(z_i) \cap \mathtt{S}(y_i) \subseteq \mathtt{S}(x_i)\,,$$

for $i = 1, 2$, and there exist $\pi_y, \pi_z \in \mathtt{fix}(S)$ such that

$$\pi_y \circ x_1 = \pi_z \circ x_1 = x_2\,, \quad \pi_y \circ y_1 = y_2\,, \quad \pi_z \circ z_1 = z_2\,.$$

Then, there exists some $\pi \in \mathtt{fix}(S)$ such that $\pi \circ x_1 = x_2$, $\pi \circ y_1 = y_2$ and $\pi \circ z_1 = z_2$.

Proof: Note that $\mathtt{S}(z_i) \cap \mathtt{S}(y_i) \subseteq \mathtt{S}(x_i)$ iff $(\mathtt{S}(z_i) \setminus \mathtt{S}(x_i)) \cap \mathtt{S}(y_i) = \varnothing$. Let $\Delta_i \triangleq \mathtt{S}(z_i) \setminus \mathtt{S}(x_i)$, $i = 1, 2$, so $\Delta_2 = \pi_z \circ \Delta_1$, and let $\pi' \triangleq \pi_y^{-1} \circ \pi_z$. By assumption, $\pi' \circ x_1 = x_1$, and therefore $\pi' \in \mathtt{fix}(\mathtt{S}(x_1))$ by strong support. Take any $b \in \Delta_1$. Then, $\pi'(b) \# \pi' \circ x_1 = x_1$ and $\pi_z(b) \in \pi_z \circ \Delta_1 = \Delta_2$, $\therefore\ \pi_z(b) \# y_2$, $\therefore\ \pi'(b) \# \pi_y^{-1} \circ y_2 = y_1$. Hence,

$$b \in \Delta_1 \implies b, \pi'(b) \# x_1, y_1\,.$$

Now assume $\Delta_1 = \{b_1, ..., b_N\}$ and define $\pi_0, \pi_1, ..., \pi_N$ by recursion:

$$\pi_0 \triangleq \mathtt{id}\,, \quad \pi_{i+1} \triangleq (b_{i+1}\ (\pi_i \circ \pi')(b_{i+1})) \circ \pi_i\,.$$

We claim that, for each $0 \leq i \leq N$ and $1 \leq j \leq i$, we have

$$\pi_i \circ \pi' \circ b_j = b_j\,, \quad \pi_i \circ x_1 = x_1\,, \quad \pi_i \circ y_1 = y_1\,.$$

[2] Recall that a permutation π fixes a set of atoms S if $\pi(a) = a$ for all $a \in S$.

We do induction on i; the case of $i = 0$ is trivial. For the inductive step, if $\pi_i \circ \pi' \circ b_{i+1} = b_{i+1}$ then $\pi_{i+1} = \pi_i$, and $\pi_{i+1} \circ \pi' \circ b_{i+1} = \pi_i \circ \pi' \circ b_{i+1} = b_{i+1}$. Moreover, by IH, $\pi_{i+1} \circ \pi' \circ b_j = b_j$ for all $1 \leq j \leq i$, and $\pi_{i+1} \circ x_1 = x_1$ and $\pi_{i+1} \circ y_1 = y_1$. If $\pi_i \circ \pi' \circ b_{i+1} = b'_{i+1} \neq b_{i+1}$ then, by construction, $\pi_{i+1} \circ \pi' \circ b_{i+1} = b_{i+1}$. Moreover, for each $1 \leq j \leq i$, by IH, $\pi_{i+1} \circ \pi' \circ b_j = (b_{i+1}\ b'_{i+1}) \circ b_j$, and the latter equals b_j since $b_{i+1} \neq b_j$ implies $b'_{i+1} \neq \pi_i \circ \pi' \circ b_j = b_j$. Finally, for any $a \in \mathtt{S}(x_1) \cup \mathtt{S}(y_1)$, $\pi_{i+1} \circ a = (b_{i+1}\ b'_{i+1}) \circ \pi_i \circ a = (b_{i+1}\ b'_{i+1}) \circ a$, by IH, with $a \neq b_{i+1}$. But the latter equals a since $\pi'(b_{i+1}) \neq a$ implies that $b'_{i+1} \neq \pi_i \circ a = a$, as required.
Hence, for each $1 \leq j \leq N$,

$$\pi_N \circ \pi' \circ b_j = b_j\,, \quad \pi_N \circ x_1 = x_1\,, \quad \pi_N \circ y_1 = y_1\,.$$

Moreover, $\pi_N \circ \pi' \circ z_1 = z_1$, as we also have

$$b \in \mathtt{S}(z_1) \cap \mathtt{S}(x_1) \implies \pi_N \circ \pi' \circ b = \pi_N \circ b = b$$

(again by strong support). Thus, taking $\pi \triangleq \pi_y \circ \pi_N^{-1}$ we have:

$$\pi_y \circ \pi_N^{-1} \circ x_1 = \pi_y \circ x_1 = x_2\,, \quad \pi_y \circ \pi_N^{-1} \circ y_1 = \pi_y \circ y_1 = y_2\,,$$
$$\pi_y \circ \pi_N^{-1} \circ z_1 = \pi_y \circ \pi_N^{-1} \circ \pi_N \circ \pi' \circ z_1 = \pi_y \circ \pi' \circ z_1 = \pi_y \circ \pi_y^{-1} \circ \pi_z \circ z_1 = z_2\,.$$

Finally, from $\pi_N \in \mathtt{fix}(\mathtt{S}(x_1)) \subseteq \mathtt{fix}(S)$ and $\pi_y \in \mathtt{fix}(S)$ we obtain $\pi \in \mathtt{fix}(S)$. ∎

A more enlightening formulation of the lemma can be given in terms of abstractions.

Let X be a strong nominal set and $x_1, x_2, y_1, y_2, z_1, z_2 \in X$. Suppose also that, for some $S \subseteq_{\mathsf{fin}} \mathbb{A}$,

$$S \subseteq \mathtt{S}(z_i) \cap \mathtt{S}(y_i) \subseteq \mathtt{S}(x_i)\,,$$

for $i = 1, 2$, and moreover that

$$[x_1, y_1]_S = [x_2, y_2]_S \quad \wedge \quad [x_1, z_1]_S = [x_2, z_2]_S\,.$$

Then $[x_1, y_1, z_1]_S = [x_2, y_2, z_2]_S$.

Figure 2.1: Strong Support Lemma

In the context of nominal games later on, where we will be dealing with abstractions of plays of this form (with $S = \varnothing$), the strong support lemma will guarantee us that composition of abstractions of plays can be reduced to composition of plays.

2.1.3 A historical note

In this section we briefly describe the permutation models of Fraenkel and Mostowski, which form the basis of what we call in this thesis "nominal sets". Our main reference here is the book by Jech [Jec73, Chapter 4]; for further references the reader is referred to the references therein.

Fraenkel–Mostowski (FM) permutation models of set theory were introduced by Fraenkel in the early 20's, and further developed by Mostowski in the late 30's, in order to prove the independence of the Axiom of Choice from the axioms of Zermelo–Fraenkel set theory with Atoms (ZFA). ZFA is an axiomatisation of set theory which allows for a set $\mathbb{A}$ the elements of which are not sets but *atoms* (*urelemente*). Atoms contain no elements, but are not the empty set. The usual axioms of ZF are present in ZFA with the necessary restrictions for atoms.

The universe of sets is constructed following the construction of the Cumulative Hierarchy, only from a different starting point: the set $\mathbb{A}$ of atoms, instead of $\varnothing$. Put formally, the

universe is now $V \triangleq \bigcup_{\alpha\in\mathtt{On}} V_\alpha$, where:

$$\begin{aligned} V_0 &\triangleq \mathbb{A} \\ V_{\alpha+1} &\triangleq \mathbb{A} \cup \mathcal{P}(V_\alpha) \\ V_\delta &\triangleq \bigcup_{\alpha<\delta} V_\alpha \end{aligned} \tag{V}$$

Note here that by convention the notion of subset applies only to sets, that is,

$$x \subseteq y \overset{\triangle}{\iff} x \notin \mathbb{A} \wedge \forall z.\, z \in x \implies z \in y\,.$$

Powersets are defined accordingly.

A genre of models for ZFA is that of *permutation models*. Within V, consider a group $\mathcal{G}$ of permutations of $\mathbb{A}$. Permutations are expanded to act on all sets in V elementwise, with $\pi \circ \varnothing = \varnothing$ for every $\pi \in \mathcal{G}$. We fix a *normal filter* $\mathcal{F}$ on $\mathcal{G}$, which is a set of subgroups of $\mathcal{G}$ such that, for all subgroups H, K of $\mathcal{G}$,

- $\mathcal{G} \in \mathcal{F}$,
- if $H \in \mathcal{F}$ and $H \subseteq K$ then $K \in \mathcal{F}$,
- if $H, K \in \mathcal{F}$ then $H \cap K \in \mathcal{F}$,
- if $\pi \in \mathcal{G}$ and $H \in \mathcal{F}$ then $\pi \circ H \circ \pi^{-1} \in \mathcal{F}$,
- for each $a \in \mathbb{A}$, $\mathtt{sym}(a) \in \mathcal{F}$,

where $\mathtt{sym}(x) \triangleq \{\pi \in \mathcal{G} \mid \pi \circ x = x\}$, for any x. The permutation model is constructed by taking the intersection of the set X of elements x such that $\mathtt{sym}(x) \in \mathcal{F}$, and of the transitive closure of X (atoms included). That is,

$$\mathcal{V} \triangleq \mathbb{A} \cup \{\, x \mid \mathtt{sym}(x) \in \mathcal{F} \wedge x \subseteq \mathcal{V} \,\}\,.$$

Analytically, $\mathcal{V} \triangleq \bigcup_{\alpha\in\mathtt{On}} \mathcal{V}_\alpha$, where:

$$\begin{aligned} \mathcal{V}_0 &\triangleq \mathbb{A} \\ \mathcal{V}_{\alpha+1} &\triangleq \mathbb{A} \cup \{\, x \subseteq \mathcal{V}_\alpha \mid \mathtt{sym}(x) \in \mathcal{F} \,\} \\ \mathcal{V}_\delta &\triangleq \bigcup_{\alpha\leq\delta} \mathcal{V}_\alpha \end{aligned}$$

One can show that $\mathcal{V}$ is a transitive model of ZFA.

The *basic Fraenkel model* is a simple permutation model that refutes the AC; hence, the AC is not provable from the axioms of ZFA. $\mathbb{A}$ is assumed to be countably infinite while the group $\mathcal{G}$ consists of all permutations of $\mathbb{A}$. Now, for each set x we define

$$\mathtt{fix}(x) \triangleq \{\pi \in \mathcal{G} \mid \forall y \in x.\, \pi \circ y = y\}\,.$$

We take the filter $\mathcal{F}$ to be the one generated by the subgroups $\mathtt{fix}(S)$, for finite $S \subseteq \mathbb{A}$, that is,

$$\mathcal{F} \triangleq \{H \subseteq \mathcal{G} \mid \exists S \subseteq_{\mathsf{fin}} \mathbb{A}.\, \mathtt{fix}(S) \subseteq H\}\,.$$

$\mathcal{F}$ is a normal filter and consists of subgroups of $\mathcal{G}$ that fix some finite set of atoms. We take $\mathcal{V}$ to be the resulting model. Concretely, we have that, for any x, $\mathtt{sym}(x) \in \mathcal{F}$ iff there is a finite $S \subseteq \mathbb{A}$ such that S *supports* x, that is,

$$\mathtt{fix}(S) \subseteq \mathtt{sym}(x)\,.$$

Hence, $x \in \mathcal{V}$ iff x has finite support and all its elements have finite support, and so on. Moreover, since supports are closed under intersection, for each $x \in \mathcal{V}$ there exists a least support $\mathtt{S}(x)$.

To see that the Axiom of Choice fails for $\mathcal{V}$, suppose that there is in $\mathcal{V}$ a choice function f for $\mathbb{A}$, i.e. an

$$f : \mathcal{P}_{\mathcal{V}}(\mathbb{A}) \setminus \{\varnothing\} \longrightarrow \mathbb{A}$$

such that, for any non-empty subset X of $\mathbb{A}$ in $\mathcal{V}$, $f(X) \in X$. Note that $\mathbb{A}$ is an element of $\mathcal{V}$, and its powerset in $\mathcal{V}$ contains all its finite and cofinite subsets. Then, we can define in $\mathcal{V}$

$$f' : \mathcal{P}_{\mathsf{fin}}(\mathbb{A}) \longrightarrow \mathbb{A} \triangleq X \mapsto f(\mathbb{A} \setminus X)\,.$$

f' is defined on the set of finite subsets of $\mathbb{A}$, and is itself supported by some finite such set, say S. Then, by definition, $f'(S) \notin S$. Since S is finite, there is some $a \in \mathbb{A} \setminus (S \cup \{f'(S)\})$. Let $(a\ f'(S))$ be the permutation swapping a and $f'(S)$ and leaving all other atoms stable; we then have that

$$(a\ f'(S)) \in \mathtt{fix}(S)\,,\ \ \therefore\ \ (a\ f'(S)) \circ f' = f',$$

and since $(a\ f'(S)) \circ (S, f'(S)) = (S, a)$, we get $(S, f'(S)), (S, a) \in f'$, i.e. $f'(S) = a$, a contradiction.

2.2 A paradigmatic nominal language

The ν-calculus of Pitts and Stark [PS93, Sta94] is a paradigmatic nominal language consisting of a call-by-value simply-typed λ-calculus with names. Names are constant terms of ground type which, according to the *What's new?* motto [PS93],

> "...are created with local scope, can be tested for equality and can be passed around via function application, but that is all."

The locality of creation and the possibility to communicate names add the feature of *local state* in an otherwise purely functional calculus. In each step of a program evaluation the local state is simply the set of available names, that is, the set of names created up to that step.

A specification that is implied by the previous motto is that an infinite supply of names is needed, so that a program can always create new names. However, the crucial specification that is hidden in the definition is that

> creation of fresh names is important as a feature, yet which names are specifically created is not important.

In other words, computation is impervious to name-permutation.

Pitts and Stark did not use the nominal framework for formulating their nominal language; after all, the (re)introduction of nominal sets occurred several years after the introduction of the ν-calculus. Nonetheless, such an approach is self-suggesting: the casting of syntactic constructions inside nominal sets, with atoms playing the role of names, results in a syntax which comes equipped with name-permutations, a name-freshness relation, etc. We upgrade this reasoning to a general guideline for modelling nominal languages, which we will strengthen in the next section and follow throughout this thesis:

> *Model names by atoms and cast all structure in nominal sets.*

Note that we do not use the full force of nominal sets in our approach, that is, we do not present binding constructors by nominal abstractions. In the languages we examine there are two forms of binding: variable-binding and name-binding. Both of these are presented in the usual way, using the *Barendregt convention* [Bar84]: terms are equal up to choice of bound names and variables, but we may also assume that our particular choices are sufficiently fresh. Although this approach introduces some amount of informality,[3] it is preferred for its simplicity, which allows us to concentrate on more pressing issues. In fact, it has been shown in [Pit06] (using nominal sets) that arguments in the style of the Barendregt convention are correct once a certain hygiene is followed.

2.2.1 The ν-calculus

The ν-calculus we present below is that of [PS93], only equipped with natural numbers instead of booleans. The calculus is cast inside **Nom**, by stipulating the existence of a set of atoms

$$\mathbb{A}_\nu \in (\mathbb{A}_i)_{i\in\omega}$$

from which names are drawn. We will briefly examine the syntax of the calculus and its operational semantics, experimenting with nominal versions of results proven in [PS93, Sta94].

The types of the calculus are given as follows. We have types for names, naturals and functions:

$$\mathrm{TY} \ni A, B ::= \ \nu \mid \mathbb{N} \mid A \to B$$

[3] In particular, the results obtained from these fresh choices are usually not shown to be independent of choice.

Terms form a strong nominal set TE:

$$\begin{array}{lll} \text{TE} \ni M, N ::= & x \mid \lambda x.M \mid M\ N & \lambda\text{-calculus} \\ & \mid n \mid \mathtt{pred}\, M \mid \mathtt{succ}\, N & \text{arithmetic} \\ & \mid \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 & \text{if\_then\_else} \\ & \mid a & \text{name, } a \in \mathbb{A}_\nu \\ & \mid [M = N] & \text{name-equality test} \\ & \mid \nu a.M & \nu\text{-abstraction} \end{array}$$

Of the terms above, the values are:

$$\text{VA} \ni V, W ::= n \mid a \mid x \mid \lambda x.M$$

Permutations act on TE componentwise, that is, for any $\pi \in \text{PERM}(\mathbb{A})$,

$$\pi \circ a = \pi(a) \quad \pi \circ \nu a.M = \nu(\pi \circ a).(\pi \circ M) \quad \pi \circ x = x \quad \pi \circ \lambda x.M = \lambda x.(\pi \circ M) \quad etc.$$

Note that there are two types of binding in the syntax, variable-binding and name-binding, and each of these yields its own notion of α-equivalence (note also that variables are *not* names). The set of ***free variables*** of a term is defined by:

$$\mathtt{fv}(x) \triangleq \{x\}, \quad \mathtt{fv}(\lambda x.M) \triangleq \mathtt{fv}(M) \setminus \{x\}, \quad \mathtt{fv}(\nu a.M) \triangleq \mathtt{fv}(M), \quad \mathtt{fv}(n) = \mathtt{fv}(a) \triangleq \varnothing,$$

plus standard rules for the other non-binding constructs. A term M is ***closed*** if $\mathtt{fv}(M)$ is empty. Similarly, the set of ***free names*** of a term is defined by:

$$\mathtt{fn}(a) \triangleq \{a\}, \quad \mathtt{fn}(\nu a.M) \triangleq \mathtt{fn}(M) \setminus \{a\}, \quad \mathtt{fn}(\lambda x.M) \triangleq \mathtt{fn}(M), \quad \mathtt{fn}(n) = \mathtt{fn}(x) \triangleq \varnothing,$$

plus standard rules for the other non-binding constructs. α-equivalence for variable-binding, henceforth called α_V-equivalence and written $=_{\alpha_V}$, is defined as usually.[4] α-equivalence for name-binding, henceforth called α_N-equivalence and written $=_{\alpha_N}$, is defined by recursion (on term size) as follows,

$$\frac{}{M =_{\alpha_N} M}\ {\scriptstyle M = x, a, n} \qquad \frac{\nu b \in \mathbb{A}_\nu.\ (a\ b) \circ M =_{\alpha_N} (a'\ b) \circ M'}{\nu a.M =_{\alpha_N} \nu a'.M'} \qquad \frac{M =_{\alpha_N} M'}{\lambda x.M =_{\alpha_N} \lambda x.M'}$$

plus standard rules for the other non-binding constructs. The definition is adapted from [GP02] and it captures the usual notion of α-equivalence, i.e. it equates terms up to choice of bound names (v. [GP02, proposition 2.2]).

The casting of our calculus in nominal sets equips us with a well-behaved action of name-permutation and a crisp notion of name-freshness. In the following proposition we give a couple of examples of results that can be shown with elegance using these mechanisms. Note that a consequence of the first result is that the second rule for $=_{\alpha_N}$ reduces to $\dfrac{M =_{\alpha_N} M'}{\nu a.M =_{\alpha_N} \nu a.M'}$ for $a = a'$.

Proposition 2.12 *For all terms M, N and $a, b \in \mathbb{A}_\nu$,*

- $M =_{\alpha_N} N \implies (a\ b) \circ M =_{\alpha_N} (a\ b) \circ N$,
- $a, b \notin \mathtt{fn}(M) \implies (a\ b) \circ M =_{\alpha_N} M$.

Proof: The first claim is shown by induction on M, and the only non-trivial case is that of ν-abstraction. So let $M = \nu a'.M'$ and $M =_{\alpha_N} N$. By definition, $N = \nu b'.N'$ and $\nu c.\ (a'\ c) \circ M' =_{\alpha_N} (b'\ c) \circ N'$. For any such $c \# a, b$, by IH, $(a\ b) \circ (a'\ c) \circ M' =_{\alpha_N} (a\ b) \circ (b'\ c) \circ N'$. Taking $a'' = (a\ b) \circ a'$ and $b'' = (a\ b) \circ b'$ we have $(a''\ c) \circ (a\ b) \circ M' =_{\alpha_N} (b''\ c) \circ (a\ b) \circ N'$,

[4] i.e. nominally! See [Kri90].

and therefore $\nu a''.(a\ b) \circ M' =_{\alpha_N} \nu b''.(a\ b) \circ N'$, as required.
For the second claim we do induction on M and assume $a \neq b$. Again, the only non-trivial case is that of ν-abstraction, so let $M = \nu a'.M'$. If $a' \# a, b$, then $(a\ b) \circ M = \nu a'.(a\ b) \circ M'$ and $a, b \notin \mathtt{fn}(M')$ so, by IH, $(a\ b) \circ M' =_{\alpha_N} M'$ and therefore, by use of first claim, $\nu a'.(a\ b) \circ M' =_{\alpha_N} \nu a'.M'$. If $a' \in \{a, b\}$, say $a' = a$, then $(a\ b) \circ M = \nu b.(a\ b) \circ M'$ and $b \notin \mathtt{fn}(M')$ so, by IH and for any fresh c, $(b\ c) \circ M' =_{\alpha_N} M'$, hence, by first claim, $(b\ c) \circ (a\ b) \circ M' = (a\ c) \circ (b\ c) \circ M' =_{\alpha_N} (a\ c) \circ M'$, and therefore $\nu b.(a\ b) \circ M' =_{\alpha_N} \nu a.M'$, as required. ∎

We now take the usual step of equating terms up to α-equivalence. It is true that the nominal setting allows us to work without α_N-equivalence with relevant elegance, but such a choice would undeservedly complicate our presentation.

> *We assume the set of terms is quotiented by α-equivalence for both binding mechanisms, that is, we equate terms up to choice of bound variables and bound names.*

We proceed to the typing system of the calculus. Terms are typed in environments $s \mid \Gamma$, where s is a finite subset of $\mathbb{A}_\nu$ and Γ a finite set of variable-type pairs.

$$\frac{}{s \mid \Gamma \vdash n : \mathbb{N}} \qquad \frac{}{s \mid \Gamma \vdash a : \nu}\ a \in s \qquad \frac{s \mid \Gamma \vdash M : \mathbb{N}}{s \mid \Gamma \vdash \mathtt{pred}\, M : \mathbb{N}} \qquad \frac{s \mid \Gamma \vdash M : \mathbb{N}}{s \mid \Gamma \vdash \mathtt{succ}\, M : \mathbb{N}}$$

$$\frac{}{s \mid \Gamma, x{:}A \vdash x : A} \qquad \frac{s \mid \Gamma \vdash M : \mathbb{N} \qquad s \mid \Gamma \vdash N_1 : A \quad s \mid \Gamma \vdash N_2 : A}{s \mid \Gamma \vdash \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 : A}$$

$$\frac{s \mid \Gamma, x{:}A \vdash M : B}{s \mid \Gamma \vdash \lambda x.M : A \to B} \qquad \frac{s \mid \Gamma \vdash M : A \to B \qquad s \mid \Gamma \vdash N : A}{s \mid \Gamma \vdash M\, N : B}$$

$$\frac{s, a \mid \Gamma \vdash M : B}{s \mid \Gamma \vdash \nu a.M : B}\ a \notin s \qquad \frac{s \mid \Gamma \vdash M : \nu \qquad s \mid \Gamma \vdash N : \nu}{s \mid \Gamma \vdash [M = N] : \mathbb{N}}$$

We can show the following equivariance and weakening properties.

Lemma 2.13 *Let $s \mid \Gamma \vdash M : A$ have a derivation $\mathcal{D}$. Then,*

- *$(a\ b) \circ (s \mid \Gamma \vdash M : A)$ has a derivation $\mathcal{D}'$ with $|\mathcal{D}| = |\mathcal{D}'|$, for any a, b,*
- *$s' \mid \Gamma \vdash M : A$ has a derivation $\mathcal{D}'$ with $|\mathcal{D}| = |\mathcal{D}'|$, for any $s \subseteq s'$.*

Proof: By (simultaneous) induction on $|\mathcal{D}|$. The base cases are straightforward. Of the cases in the inductive step, only ν-abstraction is non-standard. So let $s \mid \Gamma \vdash \nu c.M : A$ have derivation $\mathcal{D}$, and let the penultimate sequent in $\mathcal{D}$ be $s, c \mid \Gamma \vdash M : A$. For the first claim, let χ' be $(a\ b) \circ \chi$, for $\chi = c, s, M$. Then, by IH, $s', c' \mid \Gamma \vdash M' : A$ has a derivation $\mathcal{D}'$ with $|\mathcal{D}'| = |\mathcal{D}| - 1$, and by ν-abstracting we obtain a derivation of size $|\mathcal{D}|$ for $s' \mid \Gamma \vdash \nu c'.M' : A$, as required. For the second claim, take c' fresh for s'. By IH, $s, c' \mid \Gamma \vdash (c\ c') \circ M : A$ has a derivation $\mathcal{D}'$ with $|\mathcal{D}'| = |\mathcal{D}| - 1$. Now $s, c' \subseteq s', c'$, so, by IH, $s', c' \mid \Gamma \vdash (c\ c') \circ M : A$ has a derivation $\mathcal{D}''$ with $|\mathcal{D}''| = |\mathcal{D}| - 1$, and by ν-abstracting we obtain a derivation of size $|\mathcal{D}|$ for $s' \mid \Gamma \vdash \nu c'.(c\ c') \circ M : A = s' \mid \Gamma \vdash \nu c.M : A$, by α-equivalence. ∎

We proceed with the operational semantics, which is defined via a small-step reduction relation. Reduction occurs in local state environment s. We write $s \models_{\Gamma, A} M$ only if $s \mid \Gamma \vdash M : A$ is derivable, and usually we write simply $s \models M$. Reduction rules are as follows.

$$\text{LAM}\ \frac{}{s \vDash (\lambda x.M)\, V \longrightarrow s \vDash M\{V/x\}} \qquad \text{SUC}\ \frac{}{s \vDash \mathtt{succ}\, n \longrightarrow s \vDash n+1}$$

$$\text{PRD}\ \frac{}{s \vDash \mathtt{pred}\,(n+1) \longrightarrow s \vDash n} \qquad \text{PRD}\ \frac{}{s \vDash \mathtt{pred}\, 0 \longrightarrow s \vDash 0}$$

$$\text{IF0}\ \frac{}{s \vDash \mathtt{if0}\, n\, \mathtt{then}\, N_1\, \mathtt{else}\, N_2 \longrightarrow s \vDash N}\ \begin{array}{l} N{=}N_1 \text{ if } n{=}0 \\ N{=}N_2 \text{ if } n{>}0 \end{array}$$

$$\text{EQ}\ \frac{}{s \vDash [a=b] \longrightarrow s \vDash n}\ \begin{array}{l} n{=}0 \text{ if } a{=}b \\ n{=}1 \text{ if } a{\neq}b \end{array} \qquad \text{NEW}\ \frac{}{s \vDash \nu a.M \longrightarrow s, b \vDash (a\ b) \circ M}\ b \notin s$$

$$\text{CTX}\ \frac{s \vDash M \longrightarrow s' \vDash M'}{s \vDash \mathrm{E}[M] \longrightarrow s' \vDash \mathrm{E}[M']}$$

We let $\longrightarrow\!\!\!\!\!\rightarrow$ denote the reflexive-transitive closure of reduction, and $\underset{n}{\longrightarrow\!\!\!\!\!\rightarrow}$ denote its n-step restriction. Evaluation contexts E are of the forms:

$$[\_ = N]\ ,\ [a = \_]\ ,\ \mathtt{if0}\ \_\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2\ ,\ (\lambda x.N)\ \_\ ,\ \_\ N\ ,\ \mathtt{pred}\ \_\ ,\ \mathtt{succ}\ \_\ .$$

Note that, because of α-equivalence, the NEW rule can be also written as:

$$\text{NEW}\ \frac{}{s \vDash \nu a.M \longrightarrow s, a \vDash M}\ a \notin s$$

We observe that whenever $s \vDash M \longrightarrow s' \vDash M'$ then $s \subseteq s'$. Moreover, as the following result shows, the reduction relation is nominal and yields a reduction calculus which is deterministic up to choice of fresh names.

Proposition 2.14 *Let $s, s', s'' \subseteq_{\mathsf{fin}} \mathbb{A}_\nu$ and $M, M', M'' \in \mathrm{TE}$. Then,*

- *if $s \vDash M \longrightarrow s' \vDash M'$ then $\pi \circ s \vDash \pi \circ M \longrightarrow \pi \circ s' \vDash \pi \circ M'$,*
- *if $s \vDash M \longrightarrow s' \vDash M'$ and $s \vDash M \longrightarrow s'' \vDash M''$ then $(s'', M'') = (a\ b) \circ (s', M')$, for some $a, b \notin s$,*
- *if $s \vDash M \underset{n}{\longrightarrow\!\!\!\!\!\rightarrow} s' \vDash M'$ and $s \vDash M \underset{n}{\longrightarrow\!\!\!\!\!\rightarrow} s'' \vDash M''$ then there exists $\pi \in \mathtt{fix}(s)$ such that $(s'', M'') = \pi \circ (s', M')$.*

Proof: For the first claim we do induction on the size of the derivation of $s \vDash M \longrightarrow s' \vDash M'$. For the base case, the only non-trivial subcase is that of reducing by NEW, say $s \vDash \nu a.N \longrightarrow s, b \vDash (a\ b) \circ N$. We have that $\pi \circ s \vDash \pi \circ \nu a.N = \pi \circ s \vDash \nu(\pi \circ a).(\pi \circ N) \longrightarrow \pi \circ s, b' \vDash ((\pi \circ a)\ b') \circ \pi \circ N$, any $b' \# \pi \circ s$. Now, $b \# s$ implies that $\pi \circ b \# \pi \circ s$, hence we can take b' to be $\pi(b)$, and thus

$$\pi \circ s \vDash \pi \circ \nu a.N \longrightarrow \pi \circ s, \pi(b) \vDash (\pi(a)\ \pi(b)) \circ \pi \circ N = \pi \circ (s, b) \vDash \pi \circ (a\ b) \circ N$$

as required. For the induction step, assume $s \vDash \mathrm{E}[N] \longrightarrow s' \vDash \mathrm{E}[N']$ is derived from $s \vDash N \longrightarrow s' \vDash N'$. By IH, we can derive $\pi \circ s \vDash \pi \circ N \longrightarrow \pi \circ s' \vDash \pi \circ N'$, and by applying CTX with context $\pi \circ \mathrm{E}$ we obtain what required.
For the second claim we again do induction. The base case is by observation. For the inductive step, assume $s \vDash \mathrm{E}[N] \longrightarrow s' \vDash \mathrm{E}[N']$ is derived from $s \vDash N \longrightarrow s' \vDash N'$. Then it must be the case that $s \vDash M \longrightarrow s'' \vDash M''$ is $s \vDash \mathrm{E}[N] \longrightarrow s'' \vDash \mathrm{E}[N'']$, some N'', derived from $s \vDash N \longrightarrow s'' \vDash N''$. By IH, $s'' = (a\ b) \circ s'$ and $N'' = (a\ b) \circ N'$, some $a, b \# s$, and hence we nos that $\mathrm{E}[N''] = (a\ b) \circ \mathrm{E}[N']$. But $\mathrm{E}[N]$ being typed in s implies that a, b are not free in E, hence, by α-equivalence, $(a\ b) \circ \mathrm{E}[N''] = \mathrm{E}[(a\ b) \circ N''] = \mathrm{E}[N']$.
For the last claim we do induction on n. The base case is trivial. For the inductive step, assume

$$s \vDash M \longrightarrow s_1' \vDash M_1' \underset{n-1}{\longrightarrow\!\!\!\!\!\rightarrow} s' \vDash M' \quad \wedge \quad s \vDash M \longrightarrow s_1'' \vDash M_1'' \underset{n-1}{\longrightarrow\!\!\!\!\!\rightarrow} s'' \vDash M''.$$

By our previous claims we have that $(s_1'', m_1'') = (a\ b) \circ (s_1', m_1')$ and $(a\ b) \circ s_1' \vDash (a\ b) \circ M_1' \xrightarrow[n-1]{} (a\ b) \circ s' \vDash (a\ b) \circ M'$. By IH, $(s'', M'') = \pi \circ (a\ b) \circ (s', M')$, for some $\pi \in \mathtt{fix}(s_1'')$. But then $\pi \circ (a\ b) \in \mathtt{fix}(s)$, so we are done. ■

Note that the notion of determinism up to fresh names is succinctly captured by support abstraction as follows. We observe that, for every typed term $s \mid \Gamma \vdash M : A$ and any values $s' \mid \Gamma \vdash V' : A$ and $s'' \mid \Gamma \vdash V'' : A$,

- if $s \vDash M \longrightarrow\!\!\!\!\!\rightarrow s' \vDash V'$ and $s \vDash M \longrightarrow\!\!\!\!\!\rightarrow s'' \vDash V''$ then $[s', V']_s = [s'', V'']_s$,
- if $s \vDash M \longrightarrow\!\!\!\!\!\rightarrow s' \vDash V'$ and $[s', V']_s = [s'', V'']_s$ then $s \vDash M \longrightarrow\!\!\!\!\!\rightarrow s'' \vDash V''$.

This allows us to define an *abstract* evaluation relation between terms and abstracted values, as follows.

$$s \vDash M \xrightarrow[eval]{} [s' \vDash V']_{s''} \overset{\Delta}{\iff} s = s'' \wedge s \vDash M \longrightarrow\!\!\!\!\!\rightarrow s' \vDash V' \tag{2.10}$$

The previous proposition implies that $\xrightarrow[eval]{}$ is a well-defined partial function. More than that, it is a total function on closed terms, as the following theorem shows.

Theorem 2.15 (SN) *For any $s \mid \varnothing \vdash M : A$ there exist s', V' such that $s \vDash M \longrightarrow\!\!\!\!\!\rightarrow s' \vDash V'$.*

Proof: Shown as [Sta94, theorem 2.4]. ■

2.2.2 The $s\nu$-calculus

Modelling of local state in sets of names yields a notion of *unordered state*, which is inadequate for our intended denotational semantics. Nominal game semantics is based on plays of moves containing information about the current state. Programs are then modelled by strategies, that is, partial functions operating on plays. These strategies, however, are deterministic up to choice of fresh names, a feature which is in direct conflict to unordered state.[5]

Ordered state is therefore more appropriate for our purposes. One possible approach would be to use unordered state at the level of syntax and operational semantics of our nominal languages, and ordered state at the level of denotational semantics. In fact, this already happens with contexts: a context Γ is a set of premises, but $[\![\Gamma]\!]$ is an (ordered) product of type-translations. Another approach would be to use ordered state both for syntactic and semantic purposes. For the syntax this would mean to use lists of (distinct) names instead of sets of names in local state. As lemma 2.17 suggests, one should not expect substantial differences between the two approaches. In this thesis we choose to follow the latter: ordered state does not add much complication while it saves us from some informality.

Once we shift to ordered state, the presentation of the ν-calculus is given entirely inside strong nominal sets. For this reason we call this version of the calculus ***sν-calculus***, i.e. ***strong ν-calculus***. As mentioned above, all the nominal calculi we will examine in the sequel will de facto be "strong".

Definition 2.16 The $s\nu$-calculus shares the same syntax as the ν-calculus (page 24). Its typing system is given in environments $\vec{a} \mid \Gamma$ and its operational semantics in environments $\vec{a}$, where $\vec{a} \in \mathbb{A}_\nu^\#$. The rules for these are given in figure 2.2 (note we write "$a \in \vec{a}$" for "$a \in \mathtt{S}(\vec{a})$"); contexts E and the condition $(*)$ are as in page 27. ▲

The two calculi, ν and $s\nu$, are equivalent in the following sense.

Lemma 2.17 *For any $M, N \in \mathrm{TE}$ and any $\vec{a}, \vec{ac} \in \mathbb{A}_\nu^\#$,*

[5] The problematic behaviour of nominal games in weak support is discussed again in detail in remark 3.19.

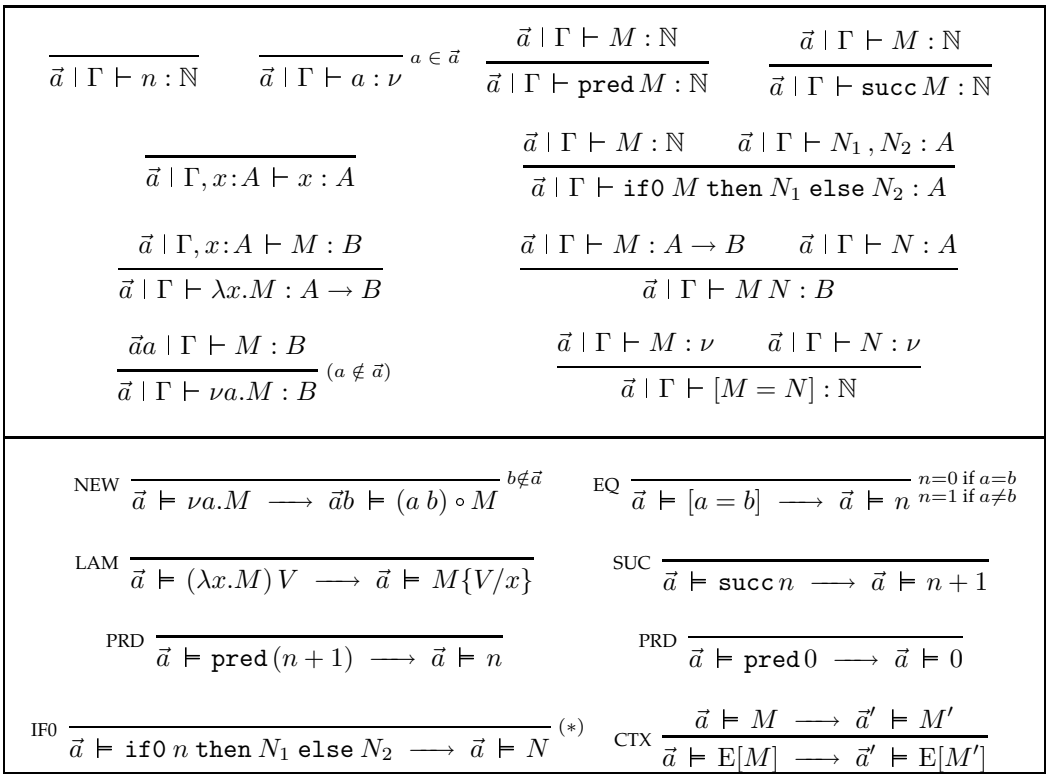


Figure 2.2: The $s\nu$-calculus: typing and reduction rules.

- $\vec{a} \mid \Gamma \vdash_{s\nu} M : A$ *iff* $\mathtt{S}(\vec{a}) \mid \Gamma \vdash_{\nu} M : A$,
- $\mathtt{S}(\vec{a}) \vDash M \underset{\nu}{\longrightarrow\!\!\!\!\!\rightarrow} \mathtt{S}(\vec{a}\vec{c}) \vDash N$ *implies* $\vec{a} \vDash M \underset{s\nu}{\longrightarrow\!\!\!\!\!\rightarrow} \vec{a}\vec{c}' \vDash N$, *for* $[\vec{c}'] = [\vec{c}]$,
- $\vec{a} \vDash M \underset{s\nu}{\longrightarrow\!\!\!\!\!\rightarrow} \vec{a}\vec{c} \vDash N$ *implies* $\mathtt{S}(\vec{a}) \vDash M \underset{\nu}{\longrightarrow\!\!\!\!\!\rightarrow} \mathtt{S}(\vec{a}\vec{c}) \vDash N$. ■

From the lemma it follows that $s\nu$ is strongly normalising. Moreover, we have that the calculi are essentially equal, meaning that their notions of observational approximation coincide:

$$\vec{a} \mid \Gamma \vdash_{s\nu} M \lessapprox N \quad \text{iff} \quad \mathtt{S}(\vec{a}) \mid \Gamma \vdash_{\nu} M \lessapprox N\,. \tag{2.11}$$

We do not wish to elaborate on this (easy) result, as we have not yet formally defined observational approximation; the interested reader can check its validity by referring to the definitions of observational approximation in the next chapters.

2.3 Monads and Comonads

In this section we present some basic results on the categorical constructions we will be using in the following chapters. Some basic category theory is assumed, covering notions such as products, coproducts, adjoints, etc. (see e.g. [Mac98]).

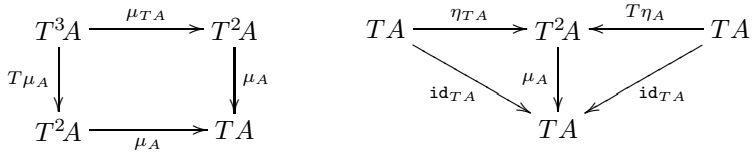
Monads and comonads are standard categorical notions (v. [Mac98] and [BW99, "triples"]) which have been used extensively in denotational semantics of programming languages in order to *encapsulate* computation. The success of these constructions is due to their conceptual simplicity: definitions involve nothing more than ome natural transformations and commuting diagrams. Combining monads is not always an easy (or even possible) task, and this is their main defect. However, the monads we use in this thesis combine relatively well.

2.3.1 Monads

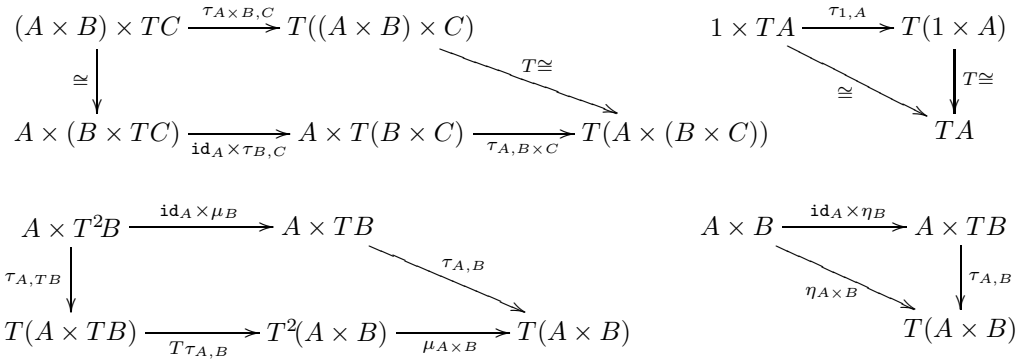
Monads were introduced in denotational semantics through the work of Moggi [Mog89, Mog91], who proposed them as a generic tool for encapsulating computational effects. Wadler [Wad92, Wad95] popularised monads in programming as a means of simulating effects in functional programs, and nowadays monads form part and parcel of the Haskell programming language [Jon03].

Definition 2.18 A ***strong monad*** on a category $\mathcal{C}$ with finite products is a quadruple (T, η, μ, τ), where:

- (T, η, μ) is a ***monad***, i.e. T is an endofunctor in $\mathcal{C}$ and $\eta : Id_{\mathcal{C}} \longrightarrow T$, $\mu : T^2 \longrightarrow T$ are natural transformations such that the following diagrams commute.



- $\tau : \_ \times T\_ \longrightarrow T(\_ \times \_)$ is a natural transformation such that the following diagrams commute.



We say that $\mathcal{C}$ has T***-exponentials*** if, for every pair B, C of objects in $\mathcal{C}$, there exists an object TC^B such that, for any object A, there exists a bijection

$$\Lambda^T_{A,B,C} : \mathcal{C}(A \times B, TC) \xrightarrow{\cong} \mathcal{C}(A, TC^B)$$

natural in A. ▲

Given a strong monad (T, η, μ, τ), we define $\tau' : T(\_) \times \_ \longrightarrow T(\_ \times \_)$ as follows.

$$\tau'_{A,B} : TA \times B \xrightarrow{\cong} B \times TA \xrightarrow{\tau_{B,A}} T(B \times A) \xrightarrow{\cong} T(A \times B) \tag{2.12}$$

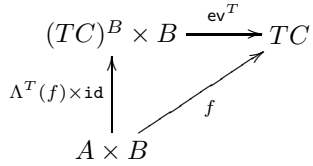
τ' satisfies the corresponding strength equations. We may refer to τ as *right strength* and to τ' as *left strength*. Combining strengths and multiplications we obtain natural transformations:

$$\begin{aligned} \psi_{A,B} &: TA \times TB \xrightarrow{\tau'_{A,TB}} T(A \times TB) \xrightarrow{T\tau_{A,B}} T^2(A \times B) \xrightarrow{\mu_{A\times B}} T(A \times B)\,, \\ \psi'_{A,B} &: TA \times TB \xrightarrow{\tau_{TA,B}} T(TA \times B) \xrightarrow{T\tau'_{A,B}} T^2(A \times B) \xrightarrow{\mu_{A\times B}} T(A \times B)\,. \end{aligned} \tag{2.13}$$

T-exponentials supply us with of ***T-evaluation arrows***, that is,

$$\mathsf{ev}^T_{B,C} : TC^{\,B} \times B \longrightarrow TC \triangleq \Lambda^{T\,-1}(\mathtt{id}_{TC^B}) \tag{2.14}$$

so that, for each $f : A \times B \longrightarrow TC$, the following diagram commutes.

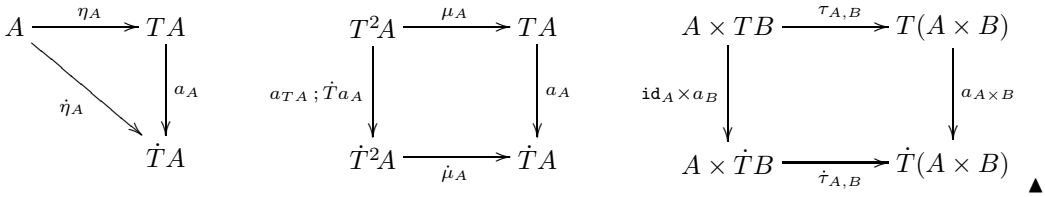


T-exponentiation is in fact a functor $(T\_)^- : \mathcal{C}^{\,op} \times \mathcal{C} \longrightarrow \mathcal{C}$ which takes each $f : A' \longrightarrow A$ and $g : B' \longrightarrow B$ to

$$Tg^{\,f} : TB'^{\,A} \longrightarrow TB^{\,A'} \triangleq \Lambda^T(TB'^{\,A} \times A' \xrightarrow{\mathtt{id}\,\times\, f} TB'^{\,A} \times A \xrightarrow{\mathsf{ev}^T} TB' \xrightarrow{Tg} TB)\,. \tag{2.15}$$

Finally, monads on a given category form a category of their own by use of the following notion.

Definition 2.19 Let (T, η, μ, τ) and $(\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ be strong monads on a category $\mathcal{C}$. A ***monad morphism*** $a : (T, \eta, \mu, \tau) \longrightarrow (\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ is a natural transformation $a : T \longrightarrow \dot{T}$ making the following diagrams commute.



▲

2.3.2 The Kleisli construction and the intrinsic preorder

Given a monad (T, η, μ) on a category $\mathcal{C}$ one may want to construct a category $\mathcal{C}^T$ including all objects of $\mathcal{C}$ but constraining its collection of arrows to those of types

$$A \longrightarrow TB.$$

This construction is called the ***Kleisli construction***. The reasons for applying it can be category-theoretical: the Kleisli construction provides a means for factorising the monad T into a pair of adjoint functors between $\mathcal{C}$ and $\mathcal{C}^T$. Most importantly for us, though, the category $\mathcal{C}^T$ represents the category of *T-computations* and is therefore the universe which holds our denotational translations.

Definition 2.20 Let $\mathcal{C}$ be a category and (T, η, μ) be a monad on $\mathcal{C}$. The ***Kleisli category*** $\mathcal{C}^T$ contains the same objects as $\mathcal{C}$ and, for all objects A, B,

$$\mathcal{C}^T(A, B) \triangleq \mathcal{C}(A, TB)\,.$$

Moreover, the identity arrow on A is η_A, and composition of arrows $f : A \longrightarrow TB$ and $g : B \longrightarrow TC$ is given by:

$$A \xrightarrow{f} TB \xrightarrow{Tg} T^2C \xrightarrow{\mu} TC\,.$$

▲

If $\mathcal{C}$ has finite products then T being a strong monad corresponds to $\mathcal{C}^T$ having a *symmetric premonoidal tensor*, that is, a non-bifunctorial tensor product (see [PR97]), which allows us to model computation products in $\mathcal{C}^T$. Furthermore, the requirement for T-exponentials makes the premonoidal structure of $\mathcal{C}^T$ closed (and corresponds to closure of the related *Freyd category*, see [Pow00]).

Intrinsic preorder The notion of equating programs modulo their observable behaviour can be modelled categorically by means of quotienting by the *intrinsic preorder*. So let us assume that T is a strong monad with exponentials on $\mathcal{C}$ and that there is a distinguished object o of $\mathcal{C}$ corresponding to a type of *observables*. We fix a collection

$$O \subseteq \mathcal{C}(1, To)$$

of arrows of *specific* observable behaviour and build the intrinsic preorder on arrows as follows.[6]

Definition 2.21 Let $\mathcal{C}, T, o, O$ be as above. We define the ***intrinsic preorder***, $\lesssim$, to be the union, over all objects A, B, of relations $\lesssim_{A,B} \subseteq \mathcal{C}(A, TB)^2$ defined by:

$$f \lesssim_{A,B} g \iff \forall \rho \in \mathcal{C}(TB^A, To).\ \Lambda^T(f); \rho \in O \implies \Lambda^T(g); \rho \in O\,.$$

▲

Note that our definition of the intrinsic preorder relates only arrows which correspond to computations, that is, arrows of $\mathcal{C}^T$. Clearly, on those arrows $\lesssim$ is a preorder. To make use of the intrinsic preorder in the semantical translation of programs, it is necessary that $\lesssim$ be coherent with the structure of $\mathcal{C}^T$, that is, it should preserve composition, premonoidal tensors and tensor exponentials.[7] In $\mathcal{C}$, these conditions are translated as follows.

Proposition 2.22 *Let $\mathcal{C}, T, o, O$ and $\lesssim$ be as above. For any $f, g : A \longrightarrow TB$ and any arrow h, if $f \lesssim g$ then:*

$$\begin{array}{llll}
\textit{if} & h : B \longrightarrow TB' & \textit{then} & f\,;Th\,;\mu \lesssim g\,;Th\,;\mu\,,\\
\textit{if} & h : A' \longrightarrow TA & \textit{then} & h\,;Tf\,;\mu \lesssim h\,;Tg\,;\mu\,,\\
\textit{if} & h : A \longrightarrow TC & \textit{then} & \langle f, h\rangle\,;\psi \lesssim \langle g, h\rangle\,;\psi \quad \textit{and} \quad \langle h, f\rangle\,;\psi \lesssim \langle h, g\rangle\,;\psi\,,\\
\textit{if} & A = A_1 \times A_2 & \textit{then} & \Lambda^T_{A_1,A_2,B}(f)\,;\eta \lesssim \Lambda^T_{A_1,A_2,B}(g)\,;\eta\,.
\end{array}$$

Proof: The claims follow from the following equations,

- $\Lambda^T(f\,;Th\,;\mu) = \Lambda^T(f)\,;\Lambda^T(TB^A \times A \xrightarrow{\mathsf{ev}^T} TB \xrightarrow{Th} T^2B' \xrightarrow{\mu} TB')$
- $\Lambda^T(h\,;Tf\,;\mu) = \Lambda^T(f)\,;\Lambda^T(TB^A \times A' \xrightarrow{\mathtt{id} \times h} TB^A \times TA \xrightarrow{\tau} T(TB^A \times A) \xrightarrow{T\mathsf{ev}^T\,;\mu} TB)$
- $\Lambda^T(\langle f, h\rangle\,;\psi) = \Lambda^T(f)\,;\Lambda^T(TB^A \times A \xrightarrow{\langle \mathsf{ev}^T,\, \pi_2\,;\, h\rangle} TB \times TC \xrightarrow{\psi} T(B \times C))$

[6] Note that, for an arrow $f : A \longrightarrow TB$, we may write (abusively) $\Lambda^T(f) : 1 \longrightarrow TB^A$ for the arrow $\Lambda^T(1 \times A \xrightarrow{\cong} A \xrightarrow{f} TB)$.

[7] In few words, $\lesssim$ should enrich $\mathcal{C}^T$.

- $\Lambda^T(\Lambda^T(f)\,;\eta) = \Lambda^T(f)\,;\Lambda^T(TB^{\,A_1\times A_2}\times A_1 \xrightarrow{\Lambda^T(\mathsf{ev}^T)} TB^{\,A_2} \xrightarrow{\eta} T(TB^{\,A_2}))$

which are true due to naturality of Λ^T. ■

Let us remark here that we will not be making actual use of the Kleisli construction in the semantical models of the following chapters. Rather, we will remain at the base semantical categories and make use of properties coming from the categories' Kleisli counterparts, such as the intrinsic preorder properties of the previous proposition.

2.3.3 Defining side-effects

Given a strong monad with exponentials and any object ξ of $\mathcal{C}$, we can form a ξ-side-effect monad on $\mathcal{C}$ as follows (cf. [Mog88]).

Proposition and Definition 2.23 *Let (T,η,μ,τ) be a strong monad with exponentials on $\mathcal{C}$ and let ξ be an object of $\mathcal{C}$. Form the quadruple $(\ddot{T},\ddot{\eta},\ddot{\mu},\ddot{\tau})$ by taking:*

- $\ddot{T} : \mathcal{C} \longrightarrow \mathcal{C} \triangleq T(\_ \times \xi)^{\xi}$,
- $\ddot{\eta}_A : A \longrightarrow \ddot{T}A \triangleq \Lambda^T(\tilde{\eta}_A)$,
- $\ddot{\mu}_A : \ddot{T}^2A \longrightarrow \ddot{T}A \triangleq \Lambda^T(\tilde{\mu}_A)$,
- $\ddot{\tau}_{A,B} : A \times \ddot{T}B \longrightarrow \ddot{T}(A\times B) \triangleq \Lambda^T(\tilde{\tau}_{A,B})$;
- $\tilde{\eta}_A \triangleq A \times \xi \xrightarrow{\eta} T(A\times\xi)$,
- $\tilde{\mu}_A \triangleq \ddot{T}^2A \times \xi \xrightarrow{\mathsf{ev}^T} T(\ddot{T}A\times\xi) \xrightarrow{T\mathsf{ev}^T} T^2(A\times\xi) \xrightarrow{\mu} T(A\times\xi)$,
- $\tilde{\tau}_{A,B} \triangleq A \times \ddot{T}B \times \xi \xrightarrow{\mathtt{id}\times\mathsf{ev}^T} A \times T(B\times\xi) \xrightarrow{\tau} T(A\times B\times\xi)$.

Then $(\ddot{T},\ddot{\eta},\ddot{\mu},\ddot{\tau})$ is a strong monad on $\mathcal{C}$. Moreover, we obtain $\ddot{T}$-exponentials by taking, for each A,B,C and any $f : A\times B \longrightarrow \ddot{T}C$, $g : A \longrightarrow \ddot{T}C^{\,B}$,

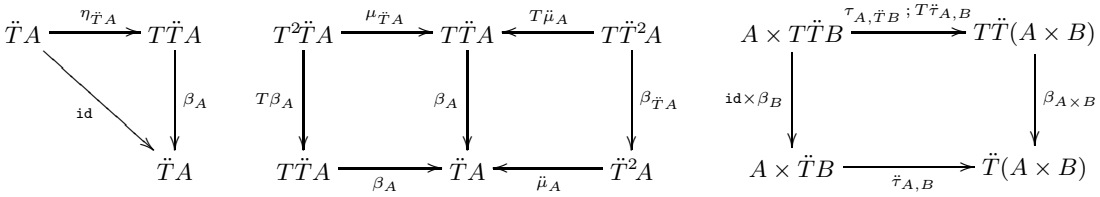
$$\begin{aligned}\ddot{T}A^{\,B} &\triangleq TA^{\,B\times\xi}\\ \Lambda^{\ddot{T}}(f) &\triangleq \Lambda^T_{A,B\times\xi,C\times\xi}({\Lambda^T_{A\times B,\xi,C\times\xi}}^{-1}(f))\\ {\Lambda^{\ddot{T}}}^{-1}(g) &\triangleq \Lambda^T_{A\times B,\xi,C\times\xi}({\Lambda^T_{A,B\times\xi,C\times\xi}}^{-1}(g))\,.\end{aligned}$$

Proof: Standard result [Mog88]. ■

We can now define a natural transformation $\beta : T\ddot{T} \longrightarrow \ddot{T}$ which *embeds* T inside $\ddot{T}$, by setting, for each object A, $\beta_A \triangleq \Lambda^T(\tilde{\beta}_A)$ and

$$\tilde{\beta}_A \triangleq T\ddot{T}A\times\xi \xrightarrow{\tau'} T(\ddot{T}A\times\xi) \xrightarrow{T\mathsf{ev}^T} T^2(A\times\xi) \xrightarrow{\mu} T(A\times\xi)\,. \tag{2.16}$$

Lemma 2.24 *For (T,η,μ,τ), $(\ddot{T},\ddot{\eta},\ddot{\mu},\ddot{\tau})$ and β defined as above, the following diagrams commute.*



■

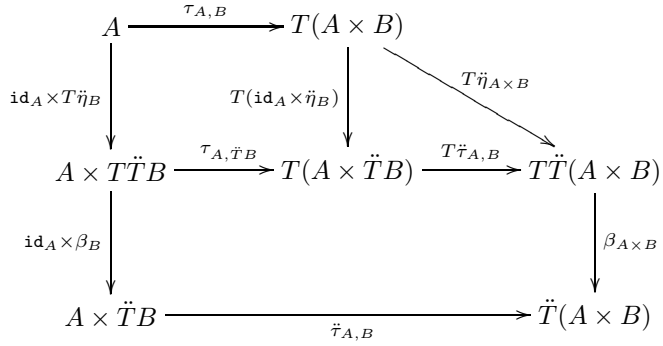
Interestingly, a natural transformation $\beta : T\ddot{T} \longrightarrow T$ satisfying the first two diagrams above corresponds to a *layering of $\ddot{T}$ over T*, in Filinski's terminology [Fil99]. We can show that β yields a monad morphism $\alpha : T \longrightarrow \ddot{T}$.

Proposition 2.25 *Let (T, η, μ, τ), $(\ddot{T}, \ddot{\eta}, \ddot{\mu}, \ddot{\tau})$ and β be as above, and let $\alpha : T \longrightarrow \ddot{T}$ be defined by:*

$$\alpha_A \triangleq TA \xrightarrow{T\ddot{\eta}_A} T\ddot{T}A \xrightarrow{\beta_A} \ddot{T}A\,.$$

Then, α is a monad morphism.

Proof: That α is a morphism between (possibly non-strong) monads is a corollary of the previous lemma, shown in [BW02, Section 3.6]. Regarding strengths, we have that the diagram



commutes, which completes the proof. ∎

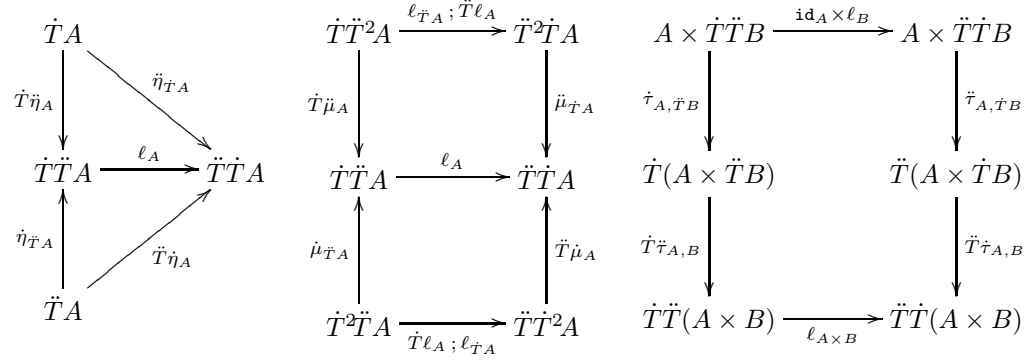
Moreover, α can be reduced as follows.

$$\begin{aligned} \alpha = T\ddot{\eta}\,;\beta = T\ddot{\eta}\,;\Lambda^T(\tau'\,;T\mathsf{ev}^T\,;\mu) = \Lambda^T(T\ddot{\eta} \times \mathtt{id}\,;\tau'\,;T\mathsf{ev}^T\,;\mu) \\ = \Lambda^T(\tau'\,;T(\ddot{\eta} \times \mathtt{id})\,;T\mathsf{ev}^T\,;\mu) = \Lambda^T(\tau'\,;T\eta\,;\mu) = \Lambda(\tau') \end{aligned} \tag{2.17}$$

2.3.4 Monad composition

Simple computational effects may be *composed* in a serial fashion, yielding more complex effects. In the monadic reading this corresponds to *monad composition*, that is, to monads $\dot{T}$ and $\ddot{T}$ being composed to the compound monad $\ddot{T}\dot{T}$. Although this construction yields a compound functor, it does not necessarily yield a monad: the resulting structure may fail to satisfy the monad axioms. Nevertheless, when $\dot{T}$ *distributes over* $\ddot{T}$ such a composition is successful.

Definition 2.26 Let $(\ddot{T}, \ddot{\eta}, \ddot{\mu}, \ddot{\tau})$ and $(\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ be strong monads on a category $\mathcal{C}$. A ***distributive law*** of $\dot{T}$ over $\ddot{T}$ is a natural transformation $\ell : \dot{T}\ddot{T} \longrightarrow \ddot{T}\dot{T}$ such that, for all objects A, B, the following diagrams commute.



If such an ℓ is given, define the ***compound monad*** (T, η, μ, τ) by:

$$\begin{aligned} T &\triangleq \ddot{T}\dot{T} \\ \eta_A &\triangleq A \xrightarrow{\ddot{\eta}_A} \ddot{T}A \xrightarrow{\ddot{T}\dot{\eta}_A} TA \\ \mu_A &\triangleq T^2A \xrightarrow{\ddot{T}\ell_{\dot{T}A}} \ddot{T}^2\dot{T}^2A \xrightarrow{\ddot{\mu}_{\dot{T}^2A}} \ddot{T}\dot{T}^2A \xrightarrow{\ddot{T}\dot{\mu}_A} TA \\ \tau_{A,B} &\triangleq A \times TB \xrightarrow{\ddot{\tau}_{A,\dot{T}B}} \ddot{T}(A \times \dot{T}B) \xrightarrow{\ddot{T}\dot{\tau}_{A,B}} T(A \times B)\,. \end{aligned}$$

▲

The notion of monad-distributivity was introduced by Beck [Bec69], who showed that it is a sufficient requirement for composing monads. The last diagram above allows the extension of Beck's result to strong monads. In the previous definition note that compound monads are by-products of distributivity laws, and hence the use of different laws can give distinct compound monads for the same pair of monads.

Proposition 2.27 *Let $\dot{T}$, $\ddot{T}$ and ℓ be as above. Then, T is a strong monad and the natural transformations*

$$\ddot{\eta} : \dot{T} \longrightarrow T\,, \quad \ddot{T}\dot{\eta} : \ddot{T} \longrightarrow T$$

are monad morphisms. ■

2.3.5 Defining exceptions

Let $\mathcal{C}$ be a category with binary products and coproducts, and let us use the following notation for coproducts.

$$A \xrightarrow{\iota_1} A + B \xleftarrow{\iota_2} B$$

Moreover, suppose $\mathcal{C}$ is *distributive*, i.e. the canonical arrow

$$\mathtt{dst}_{A,B,C} : A \times B + A \times C \longrightarrow A \times (B + C)$$

is an isomorphism, for all objects A, B, C. For any object E of $\mathcal{C}$, we can form the E-exception monad as follows.

Proposition and Definition 2.28 *For $\mathcal{C}$ and E as above, define the quadruple $(\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ as follows.*

$$\begin{aligned} \dot{T} &: \mathcal{C} \longrightarrow \mathcal{C} \triangleq \_ + E \\ \dot{\eta}_A &: A \longrightarrow \dot{T}A \triangleq \iota_1 \\ \dot{\mu}_A &: \dot{T}^2A \longrightarrow \dot{T}A \triangleq [\mathtt{id}_{\dot{T}A}, \iota_2] \\ \dot{\tau}_{A,B} &: A \times \dot{T}B \longrightarrow \dot{T}(A \times B) \triangleq \mathtt{dst}^{-1}_{A,B,E}\,;(\mathtt{id}_{A\times B} + \pi_2) \end{aligned}$$

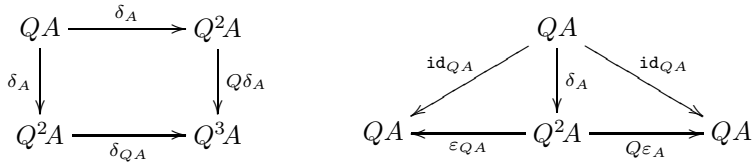
Then, $(\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ is a strong monad and, for any other monad $(\ddot{T}, \ddot{\eta}, \ddot{\mu}, \ddot{\tau})$ on $\mathcal{C}$, $\dot{T}$ distributes over $\ddot{T}$ via $\ell_A \triangleq \dot{T}\ddot{T}A \xrightarrow{[\ddot{T}\iota_1,\, \iota_2;\, \ddot{\eta}]} \ddot{T}\dot{T}A$.

Proof: This is a standard result [Mog88]. ■

2.3.6 Comonads

Comonads, the dual of monads, were proposed in denotational semantics by Brookes and Geva [BG92] for modelling programs *intensionally*: instead of abstracting away from computations and seeing programs as functions, one models programs as mechanisms which receive external computation data and decide on an output. The comonadic approach was further pursued by Brookes and van Stone [BvS93], who examined monadic-comonadic approaches, and others [Kie99, LSLM00, UV05, PW02], yet it never reached the popularity of monads due to its seemingly limited applicability.

Definition 2.29 A ***comonad*** on a category $\mathcal{C}$ is a triple (Q, ε, δ), where Q is an endofunctor in $\mathcal{C}$ and $\varepsilon : Q \longrightarrow Id_{\mathcal{C}}$, $\delta : Q \longrightarrow Q^2$ are natural transformations such that the following diagrams commute.



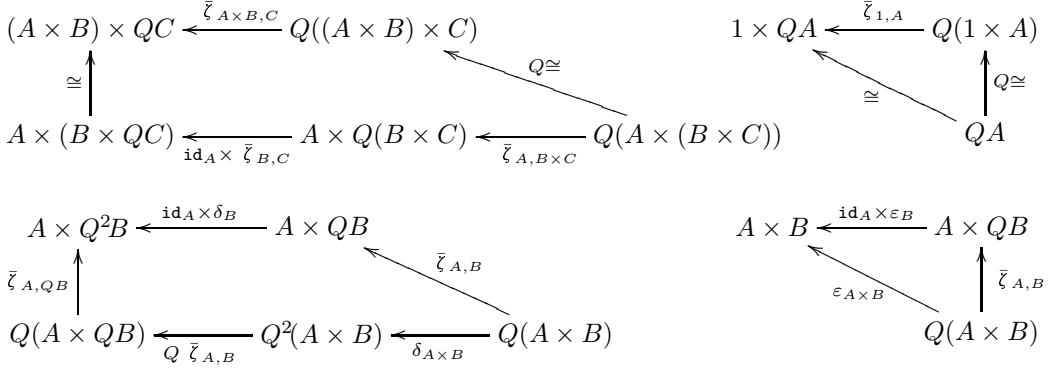
▲

In case $\mathcal{C}$ has products, we define a transformation $\bar{\zeta} : Q(\_ \times \_) \longrightarrow \_ \times Q(\_)$,

$$\bar{\zeta}_{A,B} : Q(A \times B) \xrightarrow{\langle Q\pi_1, Q\pi_2 \rangle} QA \times QB \xrightarrow{\varepsilon_A \times \mathtt{id}_{QB}} A \times QB \tag{2.18}$$

which makes the (comonadic) strength-diagrams commute.

Lemma 2.30 *Let (Q, ε, δ) be a comonad on a category $\mathcal{C}$ with finite products. Then, $\bar{\zeta}$ makes the following diagrams commute.*



Proof: This follows easily from the comonadic properties; we show the last two cases.

$$\begin{aligned}
\bar{\zeta}_{A,B} ; \mathtt{id}_A \times \varepsilon_B &= \langle Q\pi_1, Q\pi_2 \rangle ; \varepsilon_A \times \mathtt{id}_{QB} ; \mathtt{id}_A \times \varepsilon_B = \langle Q\pi_1 ; \varepsilon_A, Q\pi_2 ; \varepsilon_B \rangle \\
&= \langle \varepsilon_{A\times B} ; \pi_1, \varepsilon_{A\times B} ; \pi_2 \rangle = \varepsilon_{A\times B} \\
\delta_{A\times B} ; Q\ \bar{\zeta}_{A,B} ; \bar{\zeta}_{A,QB} &= \delta_{A\times B} ; Q\langle Q\pi_1, Q\pi_2 \rangle ; Q(\varepsilon_A \times \mathtt{id}_{QB}) ; \langle Q\pi_1, Q\pi_2 \rangle ; \varepsilon_A \times \mathtt{id}_{Q^2B} \\
&= \delta_{A\times B} ; Q\langle Q\pi_1, Q\pi_2 \rangle ; \langle Q\pi_1 ; Q\varepsilon_A, Q\pi_2 \rangle ; \varepsilon_A \times \mathtt{id}_{Q^2B} \\
&= \delta_{A\times B} ; Q\langle Q\pi_1, Q\pi_2 \rangle ; \langle Q\pi_1, Q\pi_2 \rangle ; (Q\varepsilon_A ; \varepsilon_A) \times \mathtt{id}_{Q^2B} \\
&= \delta_{A\times B} ; \langle Q^2\pi_1, Q^2\pi_2 \rangle ; (Q\varepsilon_A ; \varepsilon_A) \times \mathtt{id}_{Q^2B} \\
&= \langle Q\pi_1 ; \delta_A, Q\pi_2 ; \delta_B \rangle ; (Q\varepsilon_A ; \varepsilon_A) \times \mathtt{id}_{Q^2B} \\
&= \langle Q\pi_1 ; \varepsilon_A, Q\pi_2 ; \delta_B \rangle = \bar{\zeta}_{A\times B} ; \mathtt{id}_A \times \delta_B
\end{aligned}$$

■

Stronger comonads are obtained by stipulating a transformation ζ in the other direction, as in the case of *strong comonads* of [BvS93]. In our case, we stipulate even stronger conditions.

Definition 2.31 A comonad (Q, ε, δ) with transformation $\bar{\zeta}$ defined as above is called a ***product comonad*** if $\bar{\zeta}$ is a natural isomorphism. ▲

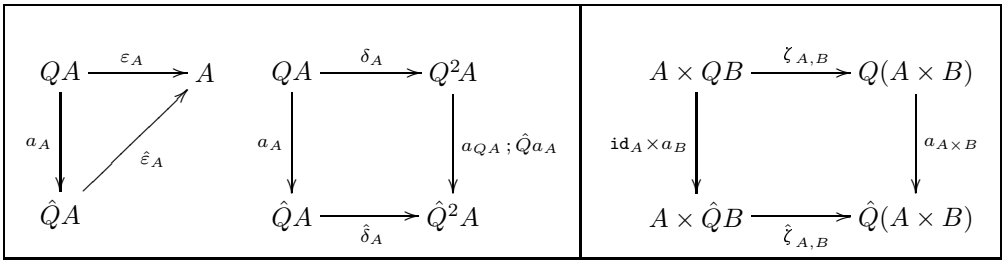
We write $\zeta : \_ \times Q(\_) \longrightarrow Q(\_ \times \_)$ for the inverse of $\bar{\zeta}$. Moreover, and as in the case of monadic strengths, we let ζ', $\bar{\zeta}'$ be their symmetric counterparts. Note that a product comonad Q can be written as

$$Q\_ \cong Q1 \times \_ \tag{2.19}$$

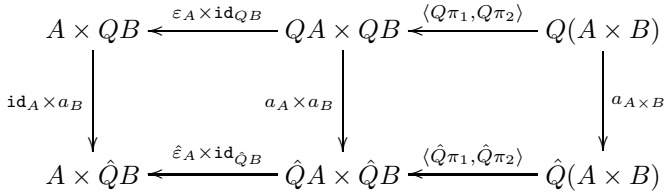
hence the name.[8] We say that $Q1$ is the ***basis of the comonad***.

[8] Note this is an isomorphism between comonads, not merely between functors.

Comonad morphisms Now let $(Q, \varepsilon, \delta), (\hat{Q}, \hat{\varepsilon}, \hat{\delta})$ be comonads on a category $\mathcal{C}$. A *comonad morphism* $a : (Q, \varepsilon, \delta) \longrightarrow (\hat{Q}, \hat{\varepsilon}, \hat{\delta})$ is a natural transformation $a : Q \longrightarrow \hat{Q}$ making the diagrams on the left below commute.



If $Q, \hat{Q}$ are product comonads then a necessarily respects coherence conditions for $\zeta, \hat{\zeta}$ (depicted on the right above). This follows from $\bar{\zeta}_{A,B}\,;\mathtt{id}_A \times a_B = a_{A\times B}\,;\bar{\hat{\zeta}}_{A,B}$, which is shown below.



2.3.7 Monadic-comonadic setting

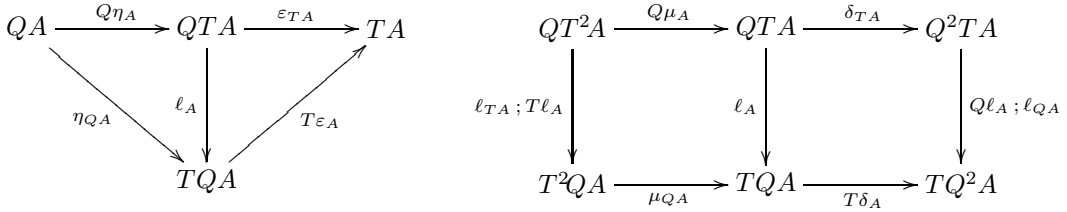
In the presence of both a strong monad (T, η, μ, τ) and a product comonad $(Q, \varepsilon, \delta, \zeta)$ in a cartesian category $\mathcal{C}$, one may want to consider solely arrows of types

$$QA \longrightarrow TB,$$

that is, arrows from some initial computation data (e.g. some initial state) of type A to some computation of type B. This amounts to applying the *biKleisli* construction on $\mathcal{C}$, i.e. to defining the category $\mathcal{C}_Q^T$ with the same objects as $\mathcal{C}$, and arrows

$$\mathcal{C}_Q^T(A, B) \triangleq \mathcal{C}(QA, TB)\,.$$

For arrow composition to work in the biKleisli category, we stipulate a distributive law between Q and T, that is, a natural transformation $\ell : QT \longrightarrow TQ$ making the following diagrams commute.



In this case, composition of $f : QA \longrightarrow TB$ and $g : QB \longrightarrow TC$ is performed as:

$$QA \xrightarrow{\delta_A} Q^2A \xrightarrow{Qf} QTB \xrightarrow{\ell_B} TQB \xrightarrow{Tg} T^2C \xrightarrow{\mu_C} TC\,. \tag{2.20}$$

Identities in the category are given by arrows of the form:

$$QA \xrightarrow{\varepsilon_A} A \xrightarrow{\eta_A} TA\,. \tag{2.21}$$

Recall we are examining a monadic-comonadic setting for strong monad T and product comonad Q, which means that a distributive law amounts to a natural transformation

$$\ell : Q1 \times T\_ \longrightarrow T(Q1 \times \_)$$

and which is therefore given for free: take $\ell \triangleq \tau_{Q1,\_}$. The distributivity equations follow straightforwardly from the strength equations.

Exponentials and the intrinsic preorder The notion of T-exponentials can be generalised to the monadic-comonadic setting as follows.

Definition 2.32 Let $\mathcal{C}$ be a category with finite products and let (T, η, μ, τ), (Q, ε, δ) be a strong monad and a comonad respectively on $\mathcal{C}$. We say that $\mathcal{C}$ has (Q,T)***-exponentials*** if, for each pair B, C of objects in $\mathcal{C}$ there exists an object $(Q,T)C^B$ such that, for each object A, there exists a bijection

$$\phi_{A,B,C} : \mathcal{C}(Q(A \times B), TC) \xrightarrow{\cong} \mathcal{C}(QA, (Q,T)C^B)$$

natural in A. ▲

In the particular case of Q being a product comonad and T having exponentials, (Q,T)-exponentials come for free.

Proposition 2.33 *In the setting of the previous definition, if $\mathcal{C}$ has T-exponentials and Q is a product comonad then $\mathcal{C}$ has (Q,T)-exponentials defined by:*

$$(Q,T)C^B \triangleq TC^B$$
$$\phi(f : Q(A \times B) \longrightarrow TC) \triangleq \Lambda^T(QA \times B \xrightarrow{\zeta'} Q(A \times B) \xrightarrow{f} TC),$$

in which case ϕ is a bijection with its inverse sending each $g : QA \longrightarrow TC^B$ to

$$Q(A \times B) \xrightarrow{\bar{\zeta}'} QA \times B \xrightarrow{g \times \mathtt{id}} TC^B \times B \xrightarrow{\mathsf{ev}^T} TC.$$

■

In this same setting, we can also define an extended notion of intrinsic preorder. So, assuming an observable object o and a collection $O \subseteq \mathcal{C}(Q1, To)$ of observable arrows, we can have the following.

Definition 2.34 Let $\mathcal{C}, Q, T, o, O$ be as above. We define $\lesssim$ to be the union, over all objects A, B, of relations $\lesssim_{A,B} \subseteq \mathcal{C}(QA, TB)^2$ defined by

$$f \lesssim_{A,B} g \iff \forall \rho \in \mathcal{C}(Q(TB^A), To).\ \Lambda^{Q,T}(f); \rho \in O \implies \Lambda^{Q,T}(g); \rho \in O,$$

where

$$\Lambda^{Q,T}(f) \triangleq Q1 \xrightarrow{\delta} Q^2 1 \xrightarrow{Q\Lambda^T(\zeta'\,;\,f)} Q(TB^A).$$

▲

As in the monadic setting, we have the following enrichment properties.

Proposition 2.35 *Let $\mathcal{C}, Q, T, o, O$ and $\lesssim$ be as above. For any $f, g : QA \longrightarrow TB$ and any arrow h, if $f \lesssim g$ then:*

if $h : QB \longrightarrow TB'$ *then* $\delta\,;Qf\,;\ell\,;Th\,;\mu \lesssim \delta\,;Qg\,;\ell\,;Th\,;\mu$

if $h : QA' \longrightarrow TA$ *then* $\delta\,;Qh\,;\ell\,;Tf\,;\mu \lesssim \delta\,;Qh\,;\ell\,;Tg\,;\mu$

if $h : QA \longrightarrow TC$ *then* $\langle f, h\rangle\,;\psi \lesssim \langle g, h\rangle\,;\psi$ *and* $\langle h, f\rangle\,;\psi \lesssim \langle h, g\rangle\,;\psi$

if $A = A_1 \times A_2$ *then* $\Lambda^T_{QA_1,A_2,B}(\zeta'\,;f)\,;\eta \lesssim \Lambda^T_{QA_1,A_2,B}(\zeta'\,;g)\,;\eta$.

Proof: The claims follow from the following equations,

- $\delta\,;Q\Lambda^T(\zeta'\,;\delta\,;Qf\,;\ell\,;Th\,;\mu) = \delta\,;Q\Lambda^T(\zeta'\,;f)\,;\delta\,;Q\Lambda^T(\zeta'\,;Q\mathsf{ev}^T\,;\ell\,;Th\,;\mu)$
- $\delta\,;Q\Lambda^T(\zeta'\,;\delta\,;Qh\,;\ell\,;Tf\,;\mu) = \delta\,;Q\Lambda^T(\zeta'\,;f)\,;\delta\,;Q\Lambda^T(\zeta'\,;\ \bar{\zeta}\ ;\mathtt{id} \times h\,;\tau\,;T\mathsf{ev}^T\,;\mu)$
- $\delta\,;Q\Lambda^T(\zeta'\,;\langle f, h\rangle\,;\psi) = \delta\,;Q\Lambda^T(\zeta'\,;f)\,;\delta\,;Q\Lambda^T(\zeta'\,;Q\langle\mathsf{ev}^T, \pi_2\rangle\,;\ \bar{\zeta}\ ;\mathtt{id} \times h\,;\psi)$
- $\delta\,;Q\Lambda^T(\zeta'\,;\Lambda^T(\zeta'\,;f)\,;\eta) = \delta\,;Q\Lambda^T(\zeta'\,;f)\,;Q\Lambda^T(\Lambda^T(\mathsf{ev}^T)\,;\eta)$

which are true due to naturality of Λ^T. ■

Chapter 3

Nominal Games

Nominal games were introduced in [AGM$^+$04, Lai04] as a basis for the fully abstract modelling of nominal computation. They constitute a reformulation of ordinary games in nominal sets,[1] thus allowing for names (atoms) to appear in plays as atomic moves and therefore for strategies to involve (equivariant) name-reasoning.

In this thesis we follow the presentation of [AGM$^+$04] (the *AGMOS approach*), rectifying also discrepancies arising in [AGM$^+$04] from the incompatibility of unordered state with determinacy of strategies. Two further guidelines for (our) nominal games are the following.

- Use of moves with local state attached, a notion that had been used before by Ong for the semantics of Idealized Algol [Ong02].
- Use of call-by-value discipline, as advanced by Honda and Yoshida [HY99] for the semantics of call-by-value PCF.

These stem from the fact that the languages we examine are stateful subsets of ML [MTM97]. With regard to local state, our approach coincides with the AGMOS approach in that moves inside a play are attached with the full list of names available at the computation step they represent. This is advantageous in that it is simple and allows for better control over plays and strategies, witnessed e.g. by the concise proof of adequacy in the next chapter. Moreover, the approach is easily customisable to nominal languages with a variety of effects: once the denotational framework for names has been set, further nominal effects can be modelled by use of monads. Here this is exemplified through general references (chapter 4) and exceptions (chapter 5).

On the other hand, full access to local names and the use of monads for effects allow strategies to make too many distinctions at the intentional level and therefore our full-abstraction results rely on quotienting. Stricter approaches to local state followed in [Lai08, MT09], for languages with ground store, factor out such distinctions and lead to fully abstract models without quotienting. Those models make use of a local state that includes only those names that have been *used* and are still *available*, for appropriate notions of name-use and name-availability. Naturally, the added strictness comes with a cost of added complicacy in the manipulation of strategies and, in fact, the methods are no longer generic: different languages have different notions of name-use, name-availability and local state. In particular, the approach is not applicable to general references — at least not directly. See section 3.5 for further discussion.

The chapter is structured as follows. In section 3.1 we introduce the basic notions of nominal games, that is, nominal arenas, plays and strategies. We work on play- and strategy-composition and obtain the category $\mathcal{G}$ of nominal arenas and nominal strategies. In section 3.2 we focus on *innocence*, and produce the subcategory $\mathcal{V}$ of innocent strategies. In section 3.3 we further restrict ourselves to *total* strategies and obtain the category $\mathcal{V}_t$, which

[1] Although nominal sets are not explicitly mentioned in [Lai04], they are in the journal version of the paper [Lai08].

we show to have products, distributive coproducts and partial exponentials. In section 3.4 we construct a monad for fresh-name creation and a family of comonads for initial state on $\mathcal{V}_{\mathrm{t}}$. In the final section we discuss Laird's presentation of nominal games [Lai04, Lai08].

3.1 The basic category $\mathcal{G}$ of nominal games

We start from the basic category of nominal games $\mathcal{G}$, containing nominal arenas and nominal strategies. $\mathcal{G}$ will be further refined in the next sections so as to incorporate the notions of innocence and totality.

3.1.1 Nominal arenas and strategies

The basis for all constructions to follow is the category **Nom** of nominal sets. We proceed to arenas.

Definition 3.1 A ***nominal arena*** $A \triangleq (M_A, I_A, \vdash_A, \lambda_A)$ is given by:

- a strong nominal set M_A of *moves*,
- a nominal subset $I_A \subseteq M_A$ of *initial moves*,
- a nominal *justification relation* $\vdash_A \subseteq M_A \times (M_A \setminus I_A)$,
- a nominal *labelling function* $\lambda_A : M_A \to \{O, P\} \times \{A, Q\}$,
  which labels moves as *Opponent* or *Player moves*, and as *Answers* or *Questions*.

An arena A is subject to the following conditions.

(f) For each $m \in M_A$, there exists unique $k \geq 0$ such that $I_A \ni m_1 \vdash_A \cdots \vdash_A m_k \vdash_A m$, for some m_l's in M_A. k is called the *level* of m.

(l1) Initial moves are P-Answers.

(l2) If $m_1, m_2 \in M_A$ are at consecutive levels then λ_A assigns them complementary OP-labels.

(l3) Answers may only justify Questions. ▲

Note that, although the nominal arenas of [AGM+04] are defined by use of a set of weaker conditions than those above, the actual arenas *used* there fall within the above definition.

Note that initial moves have level 0. We let level-1 moves form the set J_A; since $\vdash_A$ is a nominal relation, J_A is a nominal subset of M_A (and so are $\bar{I}_A, \bar{J}_A$ below). Moves in M_A are denoted by m_A and variants, initial moves by i_A and variants, and level-1 moves by j_A and variants. By $\bar{I}_A$ we denote $M_A \setminus I_A$, and by $\bar{J}_A$ the set $M_A \setminus J_A$. We also write $\bar{\lambda}_A$ for the OP-complement of λ_A.

We move on to *prearenas*, which are the 'boards' on which nominal games are played.

Definition 3.2 A ***prearena*** is defined exactly as an arena, with the only exception of condition (l1): in a prearena initial moves are O-Questions.
Given arenas A and B, construct the prearena $A \to B$ by:

$$\begin{aligned} M_{A\to B} &\triangleq M_A + M_B \\ I_{A\to B} &\triangleq I_A \\ \lambda_{A\to B} &\triangleq [(i_A \mapsto OQ\ ,\ m_A \mapsto \bar{\lambda}_A(m_A))\ ,\ \lambda_B] \\ \vdash_{A\to B} &\triangleq \{(i_A, i_B)\} \cup \{(m, n) \mid m \vdash_{A,B} n\} \end{aligned}$$

▲

It is useful to think of the (pre)arena A as a vertex-labelled directed graph with vertex-set M_A and edge-set $\vdash_A$ with the labels on vertices given by λ_A (and satisfying (l1-3)). It follows from (f) that the graph so defined is *levelled*: its vertices can be partitioned into disjoint sets L0, L1, L2,... such that the edges may only travel from level i to level $i+1$ and only level-0 vertices have no incoming edges (and therefore (pre)arenas are directed acyclic). Accordingly, we will be depicting arenas by levelled graphs or triangles (e.g. figure 3.1).

The simplest arena is $0 \triangleq (\varnothing, \varnothing, \varnothing, \varnothing)$. Other flat arenas are 1 (*unit arena*), $\mathbb{N}$ (*arena of naturals*), and $\mathbb{A}^{\vec{a}}$ (*arena of $\vec{a}$-names*), for any $\vec{a} \in \mathbb{A}^{\#}$, which we define by:

$$M_1 = I_1 \triangleq \{*\}\,, \qquad M_{\mathbb{N}} = I_{\mathbb{N}} \triangleq \mathbb{N}\,, \qquad M_{\mathbb{A}^{\vec{a}}} = I_{\mathbb{A}^{\vec{a}}} \triangleq \mathbb{A}^{\vec{a}}\,. \tag{3.1}$$

Note that for $\vec{a}$ empty we get $\mathbb{A}^{\epsilon} = 1$, and that we write $\mathbb{A}_i$ for $\mathbb{A}^a$ with $a \in \mathbb{A}_i$.

More involved are the following constructions.

Definition 3.3 For nominal arenas A, B, define the arenas $A \otimes B$, $A_\perp$, $A \multimap B$, $A \Rightarrow B$ and $A + B$ as follows.

$$\begin{aligned}
M_{A\otimes B} &\triangleq I_A \times I_B + \bar{I}_A + \bar{I}_B && (A \otimes B)\\
I_{A\otimes B} &\triangleq I_A \times I_B\\
\lambda_{A\otimes B} &\triangleq [((i_A, i_B) \mapsto PA)\,, \lambda_A \upharpoonright \bar{I}_A\,, \lambda_B \upharpoonright \bar{I}_B]\\
\vdash_{A\otimes B} &\triangleq \{((i_A, i_B), m) \mid i_A \vdash_A m \vee i_B \vdash_B m\} \cup (\vdash_A \upharpoonright \bar{I}_A{}^2) \cup (\vdash_B \upharpoonright \bar{I}_B{}^2)
\end{aligned}$$

$$\begin{aligned}
M_{A_\perp} &\triangleq \{*_1\} + \{*_2\} + M_A && (A_\perp)\\
I_{A_\perp} &\triangleq \{*_1\}\\
\lambda_{A_\perp} &\triangleq [(*_1 \mapsto PA)\,, (*_2 \mapsto OQ)\,, \lambda_A]\\
\vdash_{A_\perp} &\triangleq \{(*_1, *_2), (*_2, i_A)\} \cup (\vdash_A \upharpoonright M_A{}^2)
\end{aligned}$$

$$\begin{aligned}
M_{A\multimap B} &\triangleq I_B + I_A \times J_B + \bar{I}_A + \bar{I}_B \cap \bar{J}_B && (A \multimap B)\\
I_{A\multimap B} &\triangleq I_B\\
\lambda_{A\multimap B} &\triangleq [(i_B \mapsto PA)\,, ((i_A, j_B) \mapsto OQ)\,, \bar{\lambda}_A \upharpoonright \bar{I}_A\,, \lambda_B \upharpoonright (\bar{I}_B \cap \bar{J}_B)]\\
\vdash_{A\multimap B} &\triangleq \{(i_B, (i_A, j_B)) \mid i_B \vdash_B j_B\} \cup \{((i_A, j_B), m) \mid (i_A \vdash_A m \vee j_B \vdash_B m)\}\\
&\quad \cup (\vdash_A \upharpoonright \bar{I}_A{}^2) \cup (\vdash_B \upharpoonright (\bar{I}_B \cap \bar{J}_B)^2)
\end{aligned}$$

$$A \Rightarrow B \triangleq A \multimap B_\perp \qquad (A \Rightarrow B)$$

$$\begin{aligned}
M_{A+B} &\triangleq M_A + M_B && (A + B)\\
I_{A+B} &\triangleq I_A + I_B\\
\lambda_{A+B} &\triangleq [\lambda_A\,, \lambda_B]\\
\vdash_{A+B} &\triangleq \vdash_A \cup \vdash_B
\end{aligned}$$

The constructions are sketched in figure 3.1. ▲

In the constructions above it is assumed that all moves which are not hereditarily justified by initial moves are discarded — and therefore the resulting arenas satisfy the (f) condition. Hence, for example, for any A, B,

$$J_B = \varnothing \implies A \multimap B = B\,.$$

Moreover, we usually identify arenas with graph-isomorphic structures; for example, for any A, B,

$$0 + A = A + 0 = A\,, \quad 1 \multimap A = A\,.$$

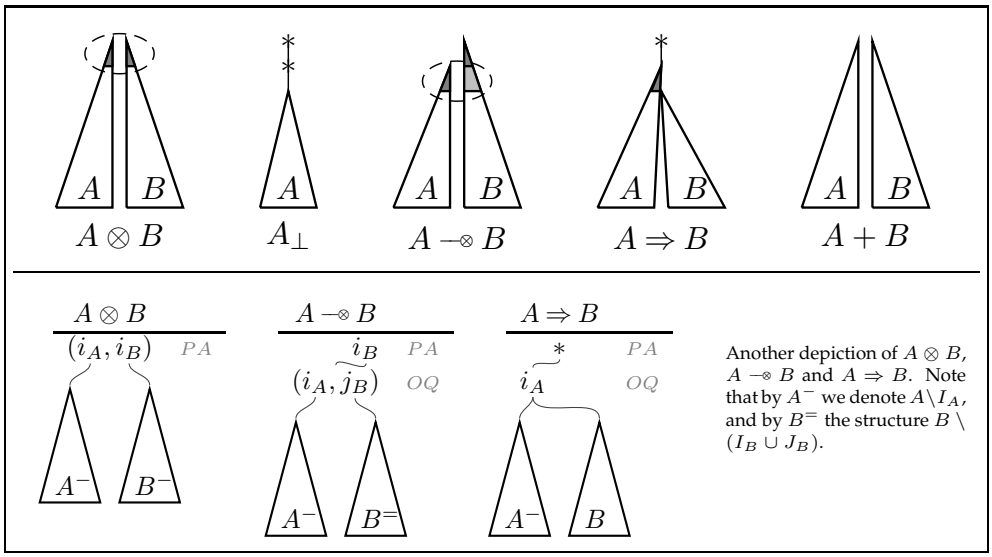


Figure 3.1: Basic arena constructions.

Using the latter convention, $A \Rightarrow B$ of the previous definition corresponds to $A \Rightarrow B$ of [HY99, AGM$^+$04]; concretely, it is given by:

$$\begin{aligned} M_{A\Rightarrow B} &\triangleq \{*\} + I_A + \bar{I}_A + M_B && (A \Rightarrow B)\\ I_{A\Rightarrow B} &\triangleq \{*\}\\ \lambda_{A\Rightarrow B} &\triangleq [(* \mapsto PA), (i_A \mapsto OQ), \bar{\lambda}_A \restriction \bar{I}_A, \lambda_B]\\ \vdash_{A\Rightarrow B} &\triangleq \{(*, i_A)\} \cup \{(i_A, m) \mid i_A \vdash_A m \vee m \in I_B\} \cup (\vdash_A \restriction \bar{I}_A{}^2) \cup (\vdash_B \restriction M_B{}^2)\,. \end{aligned}$$

Of the above constructors all look familiar apart from $\multimap$. The latter can be seen as a function-space constructor merging the contravariant part of its RHS with its LHS. For example, for any A,B,C, we have

$$A \multimap \mathbb{N} = \mathbb{N} \quad \text{and} \quad A \multimap (B \Rightarrow C) = (A \otimes B) \Rightarrow C\,.$$

In the first equality we see that $\mathbb{N}$, which appears on the RHS of $\multimap$, has no contravariant part and hence A is redundant. In the second equality B, which is the contravariant part of $B \Rightarrow C$, is merged with A. This construction will be of great use when considering a monadic semantics for store.

Before proceeding to plays and the essence of nominal games, let us introduce some useful notation for sequences (and lists).

Notation 3.4 (Sequences) A sequence s will be usually denoted by $xy\ldots$, where $x, y, \ldots$ are the elements of s. For sequences s, t,

- $s \leq t$ denotes that s is a prefix of t, and then $t = s\,(t \setminus s)$,
- s^- denotes s with its last element removed,
- if $s = s_1 \ldots s_n$ then s_1 is the first element of s and s_n the last. Also,
  - n is the length of s, and is denoted by $|s|$,
  - $s.i$ denotes s_i and $s.{-}i$ denotes s_{n+1-i}, that is, the i-th element from the tail of s (for example, $s.{-}1$ is s_n),
  - $s_{\leq s_i}$ denotes $s_1 \ldots s_i$, and so does $s_{<s_{i+1}}$. ▲

We move on to describe how nominal games are played. Given a prearena A, plays of a game consist of sequences of moves from A. These moves are augmented with name-lists (elements of $\mathbb{A}^{\#}$) to the effect of capturing name-environments.

Definition 3.5 A ***move-with-names*** of a prearena A is a pair, written $m^{\vec{a}}$, where m is a move of A and $\vec{a}$ is a finite list of distinct names (*name-list*). ▲

If x is a move-with-names then its name-list is denoted by $\mathsf{nlist}(x)$ and its underlying move by $\underline{x}$; therefore,

$$x = \underline{x}^{\mathsf{nlist}(x)} . \tag{3.2}$$

The above notation is extended to sequences of moves-with-names, so that for such a sequence s we write $s = \underline{s}^{\mathsf{nlist}(s)}$, where $\mathsf{nlist}(s)$ is a list, of length $|s|$, of lists of names.

A ***justified sequence*** over a prearena A is a finite sequence s of OP-alternating moves such that, except for $s.1$ which is initial, every move $s.i$ has a ***justification pointer*** to some $s.j$ such that $j < i$ and $s.j \vdash_A s.i$; we say that $s.j$ ***(explicitly) justifies*** $s.i$. A move in s is an ***open question*** if it is a question and there is no answer inside s justified by it.

There are two standard technical conditions that one may want to apply to justified sequences: *well-bracketing* and *visibility*. We say that a justified sequence s is ***well-bracketed*** if each answer $s.i$ appearing in s is explicitly justified by the last open question in $s_{<s.i}$, called the ***pending question***. Seeing questions as opening brackets and answers as closing ones this condition indeed corresponds to well-bracketing. For visibility, we need to introduce the notions of ***Player-*** and ***Opponent-view***. For a justified sequence s, its P-view $\ulcorner s \urcorner$ and its O-view $\llcorner s \lrcorner$ are defined as follows.

$$\begin{array}{rcll}
\ulcorner \epsilon \urcorner & \triangleq & \epsilon & \\
\ulcorner sx \urcorner & \triangleq & \ulcorner s \urcorner\, x & \text{if } x \text{ a P-move} \\
\ulcorner x \urcorner & \triangleq & x & \text{if } x \text{ is initial} \\
\ulcorner sxs'y \urcorner & \triangleq & \ulcorner s \urcorner\, xy & \text{if } y \text{ an O-move} \\
 & & & \text{expl. justified by } x
\end{array}$$

$$\begin{array}{rcll}
\llcorner \epsilon \lrcorner & \triangleq & \epsilon & \\
\llcorner sx \lrcorner & \triangleq & \llcorner s \lrcorner\, x & \text{if } x \text{ an O-move} \\
\llcorner sxs'y \lrcorner & \triangleq & \llcorner s \lrcorner\, xy & \text{if } y \text{ a P-move} \\
 & & & \text{expl. justified by } x
\end{array}$$

The ***visibility condition*** states that any O-move x in s is justified by a P-move in $\llcorner s_{<x} \lrcorner$, and any P-move y in s is justified by an O-move in $\ulcorner s_{<y} \urcorner$. We can now define plays.

Definition 3.6 Let A be a prearena. A ***legal sequence*** on A is a sequence of moves-with-names s such that $\underline{s}$ is a justified sequence satisfying Visibility and Well-Bracketing.
A legal sequence s is a ***play*** if $s.1$ has empty name-list and s also satisfies the following Name Change Conditions.

(NC1) The name-list of a P-move x in s contains as a prefix the name-list of the move preceding it. It possibly contains some other names, all of which are fresh for $s_{<x}$.

(NC2) Any name in the support of a P-move x in s that is fresh for $s_{<x}$ is contained in the name-list of x.

(NC3) The name-list of a non-initial O-move in s is that of the move explicitly justifying it.

The set of plays on a prearena A is denoted by P_A. ▲

It is important to observe that plays have strong support, due to the tagging of moves with lists of names (instead of sets of names [AGM$^+$04]). Note also that plays are the ϵ-plays of [Tze07]. Now, some further notation.

Notation 3.7 (Name-introduction) A name a is introduced (by Player) in a play s, written $a \in \mathcal{L}(s)$, whenever there exist consecutive moves yx in s such that x is a P-move and $a \in \mathsf{nlist}(x) \setminus \mathsf{nlist}(y)$. ▲

From plays we move on to strategies. Recall the notion of name-abstraction we introduced in definition 2.7; for any nominal set X and any $x \in X$,

$$[x] = \{\pi \circ x \mid \pi \in \mathrm{PERM}(\mathbb{A})\}\,.$$

Definition 3.8 Let A be a prearena. A ***strategy*** σ on A is a non-empty set of equivalence classes $[s]$ of plays in A, satisfying:

• **Prefix closure:** If $[su] \in \sigma$ then $[s] \in \sigma$.

• **Contingency completeness:** If even-length $[s] \in \sigma$ and sx is a play then $[sx] \in \sigma$.

• **Determinacy:** If even-length $[s_1x_1], [s_2x_2] \in \sigma$ and $[s_1] = [s_2]$ then $[s_1x_1] = [s_2x_2]$.

We write $\sigma : A$ whenever σ is a strategy on A. ▲

By convention, the empty sequence ϵ is a play and hence, by prefix closure and contingency completeness, all strategies contain $[\epsilon]$ and $[i_A]$'s. Note that strategies always have empty support because their elements are equivariant support abstractions.

Some basic strategies are the following — note that we give definitions *modulo prefix closure* (and recall that $\vec{a}' \subseteq \vec{a}$ if $\mathtt{S}(\vec{a}') \subseteq \mathtt{S}(\vec{a})$).

Definition 3.9 For any $\vec{a}' \subseteq \vec{a} \in \mathbb{A}^{\#}$, $i, n \in \mathbb{N}$ and any arena B, define the following strategies.

- $\tilde{n} : 1 \longrightarrow \mathbb{N} \triangleq \{\,[*\,n]\,\}$,
- $!_B : B \longrightarrow 1 \triangleq \{\,[i_B\,*]\,\}$,
- $\frac{\vec{a}}{\vec{a}'} : \mathbb{A}^{\vec{a}} \longrightarrow \mathbb{A}^{\vec{a}'} \triangleq \{\,[\vec{a}\,\vec{a}']\,\}$,
- $\mathtt{eq}_i : \mathbb{A}_i \otimes \mathbb{A}_i \longrightarrow \mathbb{N} \triangleq \{\,[(a,a)\,0]\,\} \cup \{\,[(a,b)\,1] \mid a \# b\,\}$,
- $\mathtt{id}_B : B \longrightarrow B \triangleq \{\,[s\,x\,x] \mid |s| \text{ even } \wedge\, [s] \in \mathtt{id}_B \wedge s\,x \in P_{B\to B}\,\}$. ▲

Note that in general we do not include justification pointers in definitions/expressions of strategies (or plays), unless they cannot be easily determined.

3.1.2 Composition

We proceed to composition of plays and strategies. In ordinary games, plays are composed by "parallel composition plus hiding" (v. [AJ94]); in nominal games we need to take some extra care for fresh names.

Definition 3.10 Let $s \in P_{A\to B}$ and $t \in P_{B\to C}$. We say that:

- s and t are ***almost composable***, $s \smile t$, if $\underline{s} \restriction B = \underline{t} \restriction B$.
- s and t are ***composable***, $s \asymp t$, if $s \smile t$ and, for any $s' \leq s$, $t' \leq t$ with $s' \smile t'$:
  - **(C1)** If s' ends in a (Player) move in A introducing some name a then $a \# t'$; dually, if t' ends in a move in C introducing some name a then $a \# s'$.
  - **(C2)** If both s', t' end in B and s' ends in a move introducing some name a then $a \# t'^{-}$; dually, if t' ends in a move introducing some name a then $a \# s'^{-}$. ▲

The following lemma is taken verbatim from [HY99], adapted from [BDE97].

Lemma 3.11 (Zipper lemma) *If $s \in P_{A\to B}$ and $t \in P_{B\to C}$ with $s \smile t$ then either $\underline{s} \restriction B = \underline{t} = \epsilon$, or s ends in A and t in B, or s ends in B and t in C, or both s and t end in B.* ■

Note that in the sequel we will use some standard *switching condition* results (see e.g. [AJM00, HY99]) without further mention. Composable plays are composed as below. Note that justification pointers inside $s \parallel t$ follow precisely those in s, t. Moreover, we may tag a move m as $m_{(O)}$ (or $m_{(P)}$) to specify it is an O-move (a P-move).

Definition 3.12 (Play composition) Let $s \in P_{A\to B}$ and $t \in P_{B\to C}$ with $s \asymp t$. Their ***parallel interaction*** $s \parallel t$ and their ***mix*** $s \bullet t$, which returns the final name-list in $s \parallel t$, are defined by mutual recursion as below,

$$\begin{array}{rcl@{\qquad}rcl}
\epsilon \parallel \epsilon & \triangleq & \epsilon & \epsilon \bullet \epsilon & \triangleq & \epsilon \\
sm_A^{\vec{b}} \parallel t & \triangleq & (s \parallel t) m_A^{sm_A^{\vec{b}} \bullet t} & sm_{A(P)}^{\vec{b}} \bullet t & \triangleq & (s \bullet t), (\vec{b} \setminus \mathsf{nlist}(s.{-}1)) \\
 & & & sm_{A(O)}^{\vec{b}} \bullet t & \triangleq & \vec{b}' \\
sm_B^{\vec{b}} \parallel tm_B^{\vec{c}} & \triangleq & (s \parallel t) m_B^{sm_B^{\vec{b}} \bullet tm_B^{\vec{c}}} & sm_{B(P)}^{\vec{b}} \bullet tm_{B(O)}^{\vec{c}} & \triangleq & (s \bullet t), (\vec{b} \setminus \mathsf{nlist}(s.{-}1)) \\
 & & & sm_{B(O)}^{\vec{b}} \bullet tm_{B(P)}^{\vec{c}} & \triangleq & (s \bullet t), (\vec{c} \setminus \mathsf{nlist}(t.{-}1)) \\
s \parallel tm_C^{\vec{c}} & \triangleq & (s \parallel t) m_C^{s \bullet tm_C^{\vec{c}}} & s \bullet tm_{C(P)}^{\vec{c}} & \triangleq & (s \bullet t), (\vec{c} \setminus \mathsf{nlist}(t.{-}1)) \\
 & & & s \bullet tm_{C(O)}^{\vec{c}} & \triangleq & \vec{c}'
\end{array}$$

where $\vec{b}'$ is the name-list of $m_{A(O)}$'s justifier inside $s \parallel t$, and similarly for $\vec{c}'$.
The ***composite*** of s and t is

$$s ; t \triangleq (s \parallel t) \upharpoonright AC .$$

The set of ***interaction sequences*** of A, B, C is defined by:

$$\mathsf{ISeq}(A, B, C) \triangleq \{ s \parallel t \mid s \in P_{A\to B} \wedge t \in P_{B\to C} \wedge s \asymp t \} .$$

▲

Our aim now is to show that the composite of plays is still a play. The following lemma examines the behaviour of name-lists in interactions of plays. In particular, it shows that condition (NC1) is preserved and that there is no loss of names by composition: although certain moves may be hidden in composition, their fresh names are propagated inside the name-lists. Note below that a *generalised P-move* in an interaction sequence of A, B, C is either a P-move in AC or a move in B.

Lemma 3.13 *Let $s \asymp t$ with $s \in P_{A\to B}$ and $t \in P_{B\to C}$.*

(a) *If $s \parallel t$ ends in a generalised P-move $m^{\vec{b}}$ then $\vec{b}$ contains as a prefix the name-list of $(s \parallel t).{-}2$. It possibly contains some other names, all of which are fresh for $(s \parallel t)^-$.*

(b) *If $s ; t$ ends in a P-move $m^{\vec{b}}$ then $\vec{b}$ contains as a prefix the name-list of $(s ; t).{-}2$. It possibly contains some other names, all of which are fresh for $(s ; t)^-$.*

(c) *If $s \parallel t$ ends in a move $m^{\vec{b}}$ then $\vec{b}$ contains as a prefix the name-list of the move explicitly justifying $m^{\vec{b}}$.*

(d) *If $s = s' m^{\vec{b}}$ ends in A and t in B then $\vec{b} \preceq s \bullet t$,*
if $s = s' m^{\vec{b}}$ and $t = t' m^{\vec{c}}$ end in B then $\vec{b} \preceq s \bullet t$ and $\vec{c} \preceq s \bullet t$,
if s ends in B and $t = t' m^{\vec{c}}$ in C then $\vec{c} \preceq s \bullet t$.

(e) $\mathtt{S}(s) \cup \mathtt{S}(t) = \mathtt{S}(s \parallel t) = \mathtt{S}(s ; t) \cup \mathtt{S}(s \bullet t)$.

Proof: Part (a) follows from definitions of $s \parallel t$ and $s \asymp t$, and then part (b) easily follows. For (c) we do induction on $|s \parallel t|$; the base case is trivial. Moreover, if $s \parallel t$ ends in an O-move in AC then the claim trivially holds, by definition of play-composition. So assume that $s \parallel t$

ends in a P-move in AB, consider $\ulcorner (s \parallel t) \upharpoonright AB \urcorner$ and take two consecutive moves xy in it. If y is a P-move in AB then xy are consecutive in $(s \parallel t) \upharpoonright AB$ and, by switching condition, they are also consecutive in $s \parallel t$; hence, by part (b) we have that $\mathsf{nlist}(x) \leq \mathsf{nlist}(y)$. If y is an O-move in AB and particularly in A then $\mathsf{nlist}(x) = \mathsf{nlist}(y)$, as x justifies y. Otherwise, y is in B and justified by x, and, since $s \parallel t$ ends in a P-move in AB, we can apply the IH on $s_{\leq y} \parallel t_{\leq y}$ and obtain $\mathsf{nlist}(x) \leq \mathsf{nlist}(y)$. Therefore, in any subsequence of moves in $\ulcorner (s \parallel t) \upharpoonright AB \urcorner$ the name-list of the last move contains that of the initial move as a prefix. By visibility of s, the move z justifying $(s \parallel t).{-}1$ appears in $\ulcorner (s \parallel t) \upharpoonright AB \urcorner$, hence $\mathsf{nlist}(z) \leq \mathsf{nlist}((s \parallel t).{-}1)$, as required. The case of $s \parallel t$ ending in a P-move in BC is entirely symmetrical.
For (d) we do induction on $|s \parallel t|$. The base case is encompassed in t being empty, which is trivial. Now assume $s = s' m^{\vec{b}}$ ends in A and t ends in B. If m is an O-move then the claim follows from the IH applied to $s_{\leq x}$, where x is the justifier of $m^{\vec{b}}$ in s, and the corresponding subsequence of t, and part (c). If m is a P-move then

$$\vec{b} = \mathsf{nlist}(s'.{-}1), (\vec{b} \setminus \mathsf{nlist}(s'.{-}1)) \overset{IH}{\preceq} s' \bullet t, (\vec{b} \setminus \mathsf{nlist}(s'.{-}1)) = s \bullet t\,.$$

The case of $t = t' m^{\vec{c}}$ ending in C is proved similarly. Now, if $s = s' m^{\vec{b}}$ and $t = t' m^{\vec{c}}$ both end in B and m a P-move in AB then, reasoning exactly as above, we have that $\vec{b} \preceq s \bullet t$. If m is non-initial in B then $m^{s \bullet t}$ is justified by some $n^{\vec{d}}$ in $s \parallel t$, and then, by IH, $\vec{c} \preceq \vec{d}$, and, by (c), $\vec{d} \leq s \bullet t$, which imply $\vec{c} \preceq s \bullet t$. The case of m being a P-move in t is proved similarly.
Now, for (e) we note that the following straightforwardly hold

$$\mathtt{S}(s \parallel t) \subseteq \mathtt{S}(s) \cup \mathtt{S}(t)\,, \qquad \mathtt{S}(s\,;t) \cup \mathtt{S}(s \bullet t) \subseteq \mathtt{S}(s \parallel t)\,.$$

Moreover, by (d) we obtain $\mathtt{S}(s) \cup \mathtt{S}(t) \subseteq \mathtt{S}(s \parallel t)$. Finally, we show that $\mathtt{S}(s \parallel t) \subseteq \mathtt{S}(s\,;t) \cup \mathtt{S}(s \bullet t)$ by induction on $|s \parallel t|$. The base case is encompassed in t being empty, which is trivial. Otherwise, if $s = s' m^{\vec{b}}$ ends in A and t ends in B then

$$\mathtt{S}(s \parallel t) = \mathtt{S}(s' \parallel t) \cup \mathtt{S}(m^{s \bullet t}) \overset{IH}{\subseteq} \mathtt{S}(s'\,;t) \cup \mathtt{S}(s' \bullet t) \cup \mathtt{S}(m^{s \bullet t}) \overset{(*)}{\subseteq} \mathtt{S}(s'\,;t) \cup \mathtt{S}(s \bullet t) \cup \mathtt{S}(m^{s \bullet t}) = \mathtt{S}(s\,;t)$$

where $(*)$ holds because if m is a P-move then, by (a), $s' \bullet t \leq s \bullet t$, while if m is an O-move then $\mathtt{S}(s' \bullet t) \subseteq \mathtt{S}(s'\,;t)$. Similarly for the case of t ending in C. For the case of both $s = s' m^{\vec{b}}$ and $t = t' m^{\vec{c}}$ ending in B assume wlog that $m^{\vec{b}}$ is a P-move in s. Then

$$\mathtt{S}(m) \subseteq \mathtt{S}(m^{\vec{b}}) \subseteq \mathtt{S}(s') \cup \mathtt{S}(\vec{b}) \subseteq \mathtt{S}(s' \parallel t') \cup \mathtt{S}(s \bullet t)$$

and hence

$$\begin{aligned}\mathtt{S}(s \parallel t) &= \mathtt{S}(s' \parallel t') \cup \mathtt{S}(m) \cup \mathtt{S}(s \bullet t) \subseteq \mathtt{S}(s' \parallel t') \cup \mathtt{S}(s \bullet t) \overset{IH}{\subseteq} \mathtt{S}(s'\,;t') \cup \mathtt{S}(s' \bullet t') \cup \mathtt{S}(s \bullet t) \\ &\overset{(a)}{\subseteq} \mathtt{S}(s'\,;t') \cup \mathtt{S}(s \bullet t) = \mathtt{S}(s\,;t) \cup \mathtt{S}(s \bullet t)\end{aligned}$$

as required. ∎

We can now prove the following.

Proposition 3.14 (Plays compose) *If $s \in P_{A \to B}$ and $t \in P_{B \to C}$ with $s \asymp t$, then $s\,;t \in P_{A \to C}$.*

Proof: We skip visibility and well-bracketing, as these follow from ordinary CBV game analysis. It remains to show that the name change conditions hold for $s\,;t$. (NC3) clearly does by definition, while (NC1) is part (b) of previous lemma.
For (NC2), let $s\,;t$ end in some P-move $m^{s \bullet t}$ and suppose $a \in \mathtt{S}(m^{s \bullet t})$ and $a \# (s\,;t)^{-}$. Suppose wlog that $s = s' m^{\vec{b}}$, and so $(s\,;t)^{-} = s'\,;t$. Now, if $a \# s' \bullet t$ then, by part (e) of previous lemma, $a \# s', t$ and therefore $a \in \vec{b}$, by (NC2) of s. By part (d) then, $a \in s \bullet t$. Otherwise, $a \in s' \bullet t$ and hence, by part (a), $a \in s \bullet t$. ∎

We now proceed to composition of strategies. Note that we write $\sigma : A \longrightarrow B$ if σ is a strategy on the prearena $A \to B$.

Definition 3.15 (Strategy composition) For strategies $\sigma : A \to B$ and $\tau : B \to C$, their composition is defined as

$$\sigma ; \tau \triangleq \{ [s ; t] \mid [s] \in \sigma \wedge [t] \in \tau \wedge s \asymp t \} .$$

▲

Note that, for any sequence u, if $[u] \in \sigma ; \tau$ then $u = \pi \circ (s ; t) = (\pi \circ s) ; (\pi \circ t)$ for some $[s] \in \sigma, [t] \in \tau, s \asymp t$ and π. Therefore, we can always assume $u = s ; t$ with $[s] \in \sigma, [t] \in \tau$ and $s \asymp t$.

Our next aim is to show that composites of strategies are strategies themselves. We proceed by first giving two technical lemmata.

Lemma 3.16 *For plays $s_1 \asymp t_1$ and $s_2 \asymp t_2$, if $s_1 \parallel t_1 = s_2 \parallel t_2$ then $s_1 = s_2$ and $t_1 = t_2$. Hence, if $s_1 \parallel t_1 \leq s_2 \parallel t_2$ then $s_1 \leq s_2$ and $t_1 \leq t_2$.*

Proof: The first part by easy induction on $|s_1 \parallel t_1| = |s_2 \parallel t_2|$. The second part follows. ■

Lemma 3.17 *Let $\sigma : A \to B$ and $\tau : B \to C$ be strategies with $[s_1], [s_2] \in \sigma$ and $[t_1], [t_2] \in \tau$. If $|s_1 \parallel t_1| \leq |s_2 \parallel t_2|$ and $[s_1 ; t_1] = [s_2 ; t_2]$ then there exists some π such that $\pi \circ (s_1 \parallel t_1) \leq s_2 \parallel t_2$.*

Proof: By induction on $|s_1 \parallel t_1|$. The base case is encompassed in t_1 being empty. In this case, by switching condition and determinacy of σ, $[s_1] = [s_1 ; t_1] = [s_2 ; t_2]$ implies that $s_2 = s_2' s_2''$ with $[s_1] = [s_2']$. Hence, $\pi \circ s_1 = s_2' \leq s_2 \parallel t_2$, for some permutation π, as required.

Now assume that s_1, t_1 both end in B, say $s_1 = s_1' m_1^{\vec{b}_1}$ and $t_1 = t_1' m_1^{\vec{c}_1}$. Then, $[s_1 ; t_1] = [s_1' ; t_1'] = [s_2 ; t_2]$ so, by IH, there exists some π such that $\pi \circ s_1' = s_2'$ and $\pi \circ t_1' = t_2'$, with $s_2 = s_2' s_2''$ and $t_2 = t_2' t_2''$. Moreover, s_2'', t_2'' are in B and non-empty; let $s_2''.1 = m_2^{\vec{b}_2}$ and $t_2''.1 = m_2^{\vec{c}_2}$ and assume wlog that m_2 is a P-move in $B \to C$, so the same holds for m_1. Then, by prefix closure, $[t_2' m_2^{\vec{c}_2}] \in \tau$, and, as $[t_1'] = [t_2']$, we have $[t_1' m_1^{\vec{c}_1}] = [t_2' m_2^{\vec{c}_2}]$, so $\pi' \circ t_1' m_1^{\vec{c}_1} = t_2' m_2^{\vec{c}_2}$ for some π'. Now, by (C2) we have that $(\mathtt{S}(m_i^{\vec{c}_i}) \setminus \mathtt{S}(t_i')) \cap \mathtt{S}(s_i') = \varnothing$, therefore, by Strong Support Lemma, there exists some π'' such that $\pi'' \circ m_1^{\vec{c}_1} = m_2^{\vec{c}_2}$, $\pi'' \circ t_1' = t_2'$ and $\pi'' \circ s_1' = s_2'$. Moreover, $\pi'' \circ s_1' = s_2'$ and $\pi'' \circ m_1 = m_2$ imply that $\pi'' \circ (s_1' m_1^{\vec{b}_1}) = s_2' m_2^{\vec{b}_2}$. Hence, $\pi'' \circ (s_1 \parallel t_1) \leq s_2 \parallel t_2$, as required.

Now assume s_1 ends in A and t_1 in B, say $s_1 = s_1' m_1^{\vec{b}_1}$. Then, $[s_1 ; t_1] = [s_2 ; t_2]$ implies that $s_2 = s_2' m_2^{\vec{b}_2} s_2''$ and $t_2 = t_2' t_2''$ with $[s_1 ; t_1] = [s_2' m_2^{\vec{b}_2} ; t_2']$, m_2 in A and s_2'', t_2'' in B. Then, $[s_1' ; t_1] = [s_2' ; t_2']$, and, by IH, there exists a π such that $\pi \circ s_1' = s_{21}'$, $\pi \circ t_1 = t_{21}'$, $s_2' = s_{21}' s_{22}'$, $t_2' = t_{21}' t_{22}'$ and s_{22}', t_{22}' in B.

If m_1 is an O-move then, by switching condition, s_1' ends in A, and so does s_2'. Hence, $s_{22}' = t_{22}' = \epsilon$ and thus $\pi \circ s_1' = s_2'$, $\pi \circ t_1 = t_2'$. Now, from $[s_1 ; t_1] = [s_2' m_2^{\vec{b}_2} ; t_2']$ we have $\pi' \circ (s_1 ; t_1) = s_2' m_2^{\vec{b}_2} ; t_2'$, some π'. Taking $\pi'' = \pi^{-1} \circ \pi'$ we have that $\pi'' \circ (s_1' ; t_1) = s_1' ; t_1$ and therefore, by strong support, π'' fixes all elements in $\mathtt{S}(s_1' ; t_1) \overset{lm}{=} \overset{3.13}{} \mathtt{S}(s_1') \cup \mathtt{S}(t_1)$, thus $\pi'' \circ s_1' = s_1', \quad \therefore \quad \pi' \circ s_1' = \pi \circ s_1' = s_2'$, and similarly $\pi' \circ t_1 = t_2'$. Hence, $\pi' \circ (s_1 \parallel t_1) = \pi' \circ s_1 \parallel \pi' \circ t_1 = s_2' m_2^{\vec{b}_2} \parallel t_2' \leq s_2 \parallel t_2$.

If m_1 is a P-move then, by prefix-closure, $[s_1' m_1^{\vec{b}_1}], [s_{21}' (s_{22}' m_2^{\vec{b}_2}).1] \in \sigma$ and $[s_1'] = [s_{21}']$, thus, by determinacy of σ, $[s_1' m_1^{\vec{b}_1}] = [s_{21}' (s_{22}' m_2^{\vec{b}_2}).1]$ so $s_{22}' = t_{22}' = \epsilon$ and $\pi' \circ (s_1' m_1^{\vec{b}_1}) = s_2' m_2^{\vec{b}_2}$, some π'. Because of (C1), we can now apply the Strong Support Lemma and obtain a π'' such that $\pi'' \circ (s_1 \parallel t_1) = s_2' m_2^{\vec{b}_2} \parallel t_2' \leq s_2 \parallel t_2$.

The case of s_1 ending in B and t_1 in C is entirely symmetrical. ■

Proposition 3.18 (Strategies compose) *If $\sigma : A \to B$ and $\tau : B \to C$ are strategies then so is $\sigma ; \tau$.*

Proof: By definition and proposition 3.14, $\sigma ; \tau$ contains equivalence classes of plays. We need also check the following.

• Prefix-closure: Assume $[u m^{\vec{b}}] \in \sigma ; \tau$. Then, by prefix closure of σ, τ, there exist s, t not

both ending in B and such that $s\,;t = um^{\vec{b}}$, $[s] \in \sigma$, $[t] \in \tau$. Now, assume wlog that m is in C, so $t = t'm^{\vec{b}'}$ and $s\,;t = (s\,;t')m^{\vec{b}}$. By prefix-closure of τ, $[t'] \in \tau$, $\therefore\ [s\,;t'] \in \sigma\,;\tau$.

• Contingency completeness: Assume $[u] \in \sigma\,;\tau$ is even-length and $um^{\vec{b}}$ a play. Then $u = s\,;t$, some $[s] \in \sigma, [t] \in \tau$. Suppose wlog that m is in A. As it is an O-move, u is either empty or it ends in a P-move in A. The former case is trivial. In the latter case, taking $\vec{b}'$ to be the name-list of m's justifier inside s, $sm^{\vec{b}'}$ is a play: visibility and well-bracketing follow from ordinary CBV-game analysis,[2]while name change conditions clearly hold. Moreover, $[sm^{\vec{b}'}\,;t] = [um^{\vec{b}}] \in \sigma\,;\tau$, as required.

• Determinacy: Assume even-length $[u_1x_1], [u_2x_2] \in \sigma\,;\tau$ with $[u_1] = [u_2]$, say $u_ix_i = s_i\,;t_i$, $[s_i] \in \sigma$ and $[t_i] \in \tau$, $i = 1, 2$. By prefix-closure of σ, τ we may assume that s_i, t_i don't both end in B, for $i = 1, 2$.
If s_i end in A then $s_i = s_i'n_i^{\vec{b}_i}$ and $s_i\,;t_i = (s_i'\,;t_i)n_i^{\vec{b}_i'}$, $i = 1, 2$. Now, $[s_1'\,;t_1] = [u_1] = [u_2] = [s_2'\,;t_2]$, so, by lemma 3.17 and assuming wlog that $|s_1' \parallel t_1| \leq |s_2' \parallel t_2|$, we have $\pi \circ (s_1' \parallel t_1) \leq (s_2' \parallel t_2)$, $\therefore\ \pi \circ s_1' \leq s_2'$, say $s_2' = s_2''s_2'''$ with $s_2'' = \pi \circ s_1'$ and s_2'' in B. Then $[s_2''] = [s_1']$, $\therefore\ [s_2''(s_2'''n_2^{\vec{b}_2}).1] = [s_1'n_1^{\vec{b}_1}]$, by determinacy of σ, and hence $|s_2'''| = 0$, $s_2' = \pi \circ s_1'$ and $t_2 = \pi \circ t_1$. Moreover, $\pi' \circ s_1'n_1^{\vec{b}_1} = s_2'n_2^{\vec{b}_2}$, some permutation π'. Now we can apply the Strong Support Lemma, as (C1) implies $(\mathtt{S}(n_i^{\vec{b}_i}) \setminus \mathtt{S}(s_i')) \cap \mathtt{S}(t_i) = \varnothing$. Hence, there exists a permutation π'' such that $\pi'' \circ s_1 = s_2$ and $\pi'' \circ t_1 = t_2$, $\therefore\ [s_1\,;t_1] = [s_2\,;t_2]$, as required.
If s_i end in B and t_i in C, then work similarly as above. These are, in fact, the only cases we need to check. Because if, say, s_2, t_1 end in B, s_1 in A and t_2 in C then t_1, s_2 end in P-moves and $[s_1^-\,;t_1] = [s_2\,;t_2^-]$ implies that s_1^-, t_2^- end in O-moves in B. If, say, $|s_1^- \parallel t_1| \leq |s_2 \parallel t_2^-|$ then we have, by lemma 3.17, $\pi \circ s_1^- \leq s_2$, some permutation π. So if $\pi \circ s_1^- = s_2'$ and $s_2 = s_2's_2''$, determinacy of σ dictates that $s_2''.1$ be in A, $\lightning$ to $|s_1\,;t_1| = |s_2\,;t_2|$ and $s_2\,;t_2$ ending in C. ∎

In the following remark we examine the previous proof closer in order to identify where exactly strong support is needed. This analysis provides a view on the reasons for which the nominal games model of [AGM$^+$04] is flawed. In fact, we provide specific counterexamples for needed properties which fail in that model.

Remark 3.19 (The need for strong support) The nominal games presented here differ from those of [AGM$^+$04] crucially in one aspect: the requirement for strong support. In [AGM$^+$04] plays are weakly supported since local state is modelled by finite sets of names, so a move-with-names is a move attached with a finite set of names (hence, no strong support), and other definitions differ accordingly. The problem is that thus determinacy is not preserved by strategy composition: information separating freshly created names may be hidden by composition and hence a composite strategy may break determinacy by distinguishing between composite plays that are equivalent.

In particular, in the proof of determinacy above we first derived from $[s_1'\,;t_1] = [s_2'\,;t_2]$ that there exists some π so that $\pi \circ s_1' = s_2$ and $\pi \circ t_1 = t_2$, by appealing to lemma 3.17; in the proof of that lemma, the Strong Support Lemma needs to be used several times. In fact, the statement

$$|s_1' \parallel t_1| = |s_2' \parallel t_2| \ \wedge\ [s_1'\,;t_1] = [s_2'\,;t_2] \implies \exists \pi.\, \pi \circ s_1' = s_2' \wedge \pi \circ t_1 = t_2$$

does not hold in a weak support setting such as that of [AGM$^+$04]. For take some $i \in \omega$ and

[2]Visibility holds because $\lfloor \underline{s} \rfloor = \lfloor \underline{u} \rfloor$. For well-bracketing, if m is an Answer then its justifier, say n, is in A, and n is the pending Question of $\underline{u}$. Now, because $\underline{u} \upharpoonright A = \underline{s} \upharpoonright A$, if the pending-Q of $\underline{s}$ is in A then it is n. Otherwise, the pending-Q of $\underline{s}$ is some n' in B, and $\underline{s} = s_1ns_2n's_3$. Since $\underline{s}$ satisfies well-bracketing all Answers in s_3 are justified by Q's within s_3, and since n' is the pending-Q all Q's in s_3 are answered. Hence, s_3 is even-length and n' is a P-move. Moreover, all A's in $s_3 \upharpoonright A$ are justified within $s_3 \upharpoonright A$, so $s_3 \upharpoonright A$ is also even-length, $\lightning$ to the switching condition

consider the following AGMOS-strategies.

$$\begin{aligned} \sigma : 1 \longrightarrow \mathbb{A}_i &\triangleq \{\, [*\, a^{\{a,b\}}] \mid a \neq b \in \mathbb{A}_i \,\}\,, \\ \tau : \mathbb{A}_i \longrightarrow \mathbb{A}_i \Rightarrow \mathbb{A}_i &\triangleq \{\, [a\, *\, c\, a] \mid a, c \in \mathbb{A}_i \,\}\,. \end{aligned} \tag{3.19:A}$$

Then

$$[*\, a^{\{a,b\}}\,; a\, *\, b] = [*\, *^{\{a,b\}}\, b^{\{a,b\}}] = [*\, *^{\{a,b\}}\, a^{\{a,b\}}] = [*\, a^{\{a,b\}}\,; a\, *\, a]\,,$$

yet for no π do we have $\pi \circ (*\, a^{\{a,b\}}) = *\, a^{\{a,b\}}$ and $\pi \circ (a * b) = a * a$. As a result, determinacy fails for $\sigma\,;\tau$ since both $[*\, *^{\{a,b\}}\, b^{\{a,b\}} a^{\{a,b\}}], [*\, *^{\{a,b\}}\, a^{\{a,b\}} a^{\{a,b\}}] \in \sigma\,;\tau$.

Another point where we used the Strong Support Lemma in the proof of determinacy was in showing (the dual of):

$$\exists \pi, \pi'.\, \pi \circ (s_1, t_1') = (s_2, t_2') \wedge \pi' \circ t_1' n_1^{\vec{b}_1} = t_2' n_2^{\vec{b}_2} \implies \exists \pi''.\, \pi'' \circ (s_1, t_1' n_1^{\vec{b}_1}) = (s_2, t_2' n_2^{\vec{b}_2})$$

$$\text{i.e.} \quad [s_1, t_1'] = [s_2, t_2'] \wedge [t_1' n_1^{\vec{b}_1}] = [t_2' n_2^{\vec{b}_2}] \implies [s_1, t_1' n_1^{\vec{b}_1}] = [s_2, t_2' n_2^{\vec{b}_2}]\,.$$

The above statement does not hold for AGMOS-games. To show this, we need to introduce[3] the flat arena $\mathbb{A}_i \odot \mathbb{A}_i$ with $M_{\mathbb{A}_i \odot \mathbb{A}_i} \triangleq \mathcal{P}_2(\mathbb{A}_i)$ (the set of 2-element subsets of $\mathbb{A}_i$). This is not a legal arena in our setting, since its moves are not strongly supported, but it is in the AGMOS setting. Consider the following strategies.

$$\begin{aligned} \sigma : \mathbb{A}_i \otimes \mathbb{A}_i \longrightarrow \mathbb{A}_i \odot \mathbb{A}_i &\triangleq \{\, [(a,b)\, \{a,b\}] \mid a \neq b \in \mathbb{A}_i \,\} \\ \tau : \mathbb{A}_i \odot \mathbb{A}_i \longrightarrow \mathbb{A}_i &\triangleq \{\, [\{a,b\}\, a] \mid a \neq b \in \mathbb{A}_i \,\} \end{aligned} \tag{3.19:B}$$

We have that $[(a,b)\, \{a,b\}, \{a,b\}] = [(a,b)\, \{a,b\}, \{a,b\}]$ and $[\{a,b\}\, a] = [\{a,b\}\, b]$, yet

$$[(a,b)\, \{a,b\}, \{a,b\}\, a] \neq [(a,b)\, \{a,b\}, \{a,b\}\, b]\,.$$

In fact, determinacy is broken since $[(a,b)\, a], [(a,b)\, b] \in \sigma\,;\tau$.

Our final task in this section is to show that composition of strategies is associative. Note first that by lemma 3.13, part (a), if $s \asymp t$ then the name-list of $(s \parallel t).{-}1$ contains as a prefix that of $(s\,;t).{-}1$. This allows for the following definition.

Definition 3.20 Let $s \in P_{A \to B}, t \in P_{B \to C}$ with $s \asymp t$. If $s\,;t$ ends in a move $m^{\vec{b}}$, define

$$s \circ t \triangleq s \bullet t \setminus \vec{b}\,,$$

that is, $s \bullet t = \vec{b}, s \circ t$. ▲

Note in particular that if $s\,;t$ and $s \parallel t$ end in the same move then $s \circ t = \epsilon$. Now we extend parallel interaction to triples of plays.

Definition 3.21 Let $s \in P_{A \to B}$, $t \in P_{B \to C}$ and $u \in P_{C \to D}$ with $(s\,;t) \asymp u$ and $s \asymp (t\,;u)$.

[3] In the AGMOS setting, plays with non-empty initial local state are allowed. Hence, we could have used to the same effect the $\{a,b\}$-strategies:

$$\sigma : \mathbb{A}_i \otimes \mathbb{A}_i \longrightarrow 1 \triangleq \{\, [(a,b)^{\{a,b\}} *^{\{a,b\}}]_{\{a,b\}} \,\}\,, \quad \tau : 1 \longrightarrow \mathbb{A}_i \triangleq \{\, [*^{\{a,b\}} a^{\{a,b\}}]_{\{a,b\}} \,\}\,.$$

Define $s \parallel t \parallel u$ and $s \bullet t \bullet u$ as follows,

$$\begin{array}{rcl@{\qquad}rcl}
\epsilon \parallel \epsilon \parallel \epsilon & \triangleq & \epsilon & \epsilon \bullet \epsilon \bullet \epsilon & \triangleq & \epsilon \\
sm_A^{\vec{b}} \parallel t \parallel u & \triangleq & (s \parallel t \parallel u) m_A^{sm_A^{\vec{b}} \bullet t \bullet u} & sm_{A(P)}^{\vec{b}} \bullet t \bullet u & \triangleq & (s \bullet t \bullet u), (\vec{b} \setminus \mathsf{nlist}(s.{-}1)) \\
 & & & sm_{A(O)}^{\vec{b}} \bullet t \bullet u & \triangleq & \vec{b}' \\
sm_B^{\vec{b}} \parallel tm_B^{\vec{c}} \parallel u & \triangleq & (s \parallel t \parallel u) m_B^{sm_B^{\vec{b}} \bullet tm_B^{\vec{c}} \bullet u} & sm_{B(P)}^{\vec{b}} \bullet tm_{B(O)}^{\vec{c}} \bullet u & \triangleq & (s \bullet t \bullet u), (\vec{b} \setminus \mathsf{nlist}(s.{-}1)) \\
 & & & sm_{B(O)}^{\vec{b}} \bullet tm_{B(P)}^{\vec{c}} \bullet u & \triangleq & (s \bullet t \bullet u), (\vec{c} \setminus \mathsf{nlist}(t.{-}1)) \\
s \parallel tm_C^{\vec{c}} \parallel um_C^{\vec{d}} & \triangleq & (s \parallel t \parallel u) m_C^{s \bullet tm_C^{\vec{c}} \bullet um_C^{\vec{d}}} & s \bullet tm_{C(P)}^{\vec{c}} \bullet um_{C(O)}^{\vec{d}} & \triangleq & (s \bullet t \bullet u), (\vec{c} \setminus \mathsf{nlist}(t.{-}1)) \\
 & & & s \bullet tm_{C(O)}^{\vec{c}} \bullet um_{C(P)}^{\vec{d}} & \triangleq & (s \bullet t \bullet u), (\vec{d} \setminus \mathsf{nlist}(u.{-}1)) \\
s \parallel t \parallel um_D^{\vec{d}} & \triangleq & (s \parallel t \parallel u) m_D^{s \bullet t \bullet um_D^{\vec{d}}} & s \bullet t \bullet um_{D(P)}^{\vec{d}} & \triangleq & (s \bullet t \bullet u), (\vec{d} \setminus \mathsf{nlist}(u.{-}1)) \\
 & & & s \bullet t \bullet um_{D(O)}^{\vec{c}} & \triangleq & \vec{d}'
\end{array}$$

where $\vec{b}'$ is the name-list of $m_{A(O)}$'s justifier inside $s \parallel t \parallel u$, and similarly for $\vec{d}'$. ▲

Note that the conditions $s \asymp t$, $t \asymp u$, $(s\,;t) \asymp u$ and $s \asymp (t\,;u)$ in the above definition indeed imply that exactly one of the following is the case: s ends in A, or s, t end in B, or t, u end in C, or u ends in D. We can now show the following.

Lemma 3.22 *If $s_1 \in P_{A_1 \to A_2}$, $s_2 \in P_{A_2 \to A_3}$ and $s_3 \in P_{A_3 \to A_4}$ with $(s_1\,;s_2) \asymp s_3$ and $s_1 \asymp (s_2\,;s_3)$ then*

$$(s_1; s_2); s_3 = (s_1 \parallel s_2 \parallel s_3) \upharpoonright A_1 A_4 = s_1\,;(s_2\,;s_3)\,,$$

$$(s_1; s_2) \bullet s_3, s_1 \circ s_2 = s_1 \bullet s_2 \bullet s_3 = s_1 \bullet (s_2\,;s_3), s_2 \circ s_3\,.$$

Proof: By induction on $k = |s_1 \parallel s_2 \parallel s_3|$. The case of $k = 0$ is trivial; otherwise:

$\rightsquigarrow (s_1 m_{A_1}^{\vec{b}}\,;s_2)\,;s_3 = ((s_1\,;s_2)\,;s_3) m_{A_1}^{\vec{b}'} \overset{IH}{=} ((s_1 \parallel s_2 \parallel s_3) m_{A_1}^{\vec{b}'}) \upharpoonright A_1 A_4$, where $\vec{b}' = (s_1 m_{A_1}^{\vec{b}}\,;s_2) \bullet s_3$. Thus, it suffices to show that $(s_1 m_{A_1}^{\vec{b}}\,;s_2) \bullet s_3 = s_1 m_{A_1}^{\vec{b}} \bullet s_2 \bullet s_3$. Since $s_1 m_{A_1}^{\vec{b}} \circ s_2 = \epsilon$, that would also imply $s_1 m_{A_1}^{\vec{b}} \bullet s_2 \bullet s_3 = (s_1 m_{A_1}^{\vec{b}}\,;s_2) \bullet s_3\,, s_1 m_{A_1}^{\vec{b}} \circ s_2$. Now, if m_{A_1} an O-move then the assertion holds by definition. On the other hand, if m_{A_1} a P-move then $\vec{b}' = (s_1\,;s_2) \bullet s_3\,,(\vec{b}'' \setminus \mathsf{nlist}((s_1\,;s_2).{-}1))$ and $\vec{b}'' = s_1 \bullet s_2\,,(\vec{b} \setminus \mathsf{nlist}(s_1.{-}1))$, while $s_1 m_{A_1}^{\vec{b}} \bullet s_2 \bullet s_3 = (s_1 \bullet s_2 \bullet s_3)\,,(\vec{b} \setminus \mathsf{nlist}(s_1.{-}1))$. By IH, $s_1 \bullet s_2 \bullet s_3 = (s_1\,;s_2) \bullet s_3\,, s_1 \circ s_2$, thus

$$\begin{aligned}
\vec{b}' &= (s_1\,;s_2) \bullet s_3\,,(\vec{b}' \setminus \mathsf{nlist}((s_1\,;s_2).{-}1)) \\
&= (s_1\,;s_2) \bullet s_3\,,((s_1 \bullet s_2\,,(\vec{b} \setminus \mathsf{nlist}(s_1.{-}1))) \setminus \mathsf{nlist}((s_1\,;s_2).{-}1)) \\
&= (s_1\,;s_2) \bullet s_3\,,(s_1 \bullet s_2 \setminus \mathsf{nlist}((s_1\,;s_2).{-}1))\,,(\vec{b} \setminus \mathsf{nlist}(s_1.{-}1)) \\
&= (s_1\,;s_2) \bullet s_3\,, s_1 \circ s_2\,,(\vec{b} \setminus \mathsf{nlist}(s_1.{-}1)) = s_1 m_{A_1}^{\vec{b}} \bullet s_2 \bullet s_3\,.
\end{aligned}$$

Also, $s_1 m_{A_1}^{\vec{b}}\,;(s_2\,;s_3) = (s_1\,;(s_2\,;s_3)) m_{A_1}^{\vec{b}'} \overset{IH}{=} ((s_1 \parallel s_2 \parallel s_3) m_{A_1}^{\vec{b}'}) \upharpoonright A_1 A_4$, with $\vec{b}' = s_1 m_{A_1}^{\vec{b}} \bullet (s_2\,;s_3)$. Note that $s_2 \parallel s_3$ necessarily ends in a P-move in A_2 and therefore $s_2 \circ s_3 = \epsilon$. Now, it suffices to show that $s_1 m_{A_1}^{\vec{b}} \bullet (s_2\,;s_3) = s_1 m_{A_1}^{\vec{b}} \bullet s_2 \bullet s_3$; that would also imply $s_1 m_{A_1}^{\vec{b}} \bullet s_2 \bullet s_3 = s_1 m_{A_1}^{\vec{b}} \bullet (s_2\,;s_3)\,, s_2 \circ s_3$. If m_{A_1} an O-move then the assertion holds by definition. If a P-move then $\vec{b}' = s_1 \bullet (s_2\,;s_3)\,,(\vec{b} \setminus \mathsf{nlist}(s_1.{-}1))$, which is what required, because of the IH.

$\rightsquigarrow (s_1 m_{A_2}^{\vec{b}}\,;s_2 m_{A_2}^{\vec{c}})\,;s_3 = ((s_1\,;s_2)\,;s_3) \overset{IH}{=} (s_1 \parallel s_2 \parallel s_3) \upharpoonright A_1 A_4 \overset{IH}{=} (s_1\,;(s_2\,;s_3)) = (s_1 m_{A_2}^{\vec{b}}\,;(s_2 m_{A_2}^{\vec{c}}\,;s_3))$.

If m_{A_2} is a P-move in $A_1 \to A_2$ then $s_2 \parallel s_3$ ends in a P-move in A_2, so $s_2 \circ s_3 = \epsilon$, and

$$\begin{aligned}
s_1 m^{\vec{b}}_{A_2} \circ s_2 m^{\vec{c}}_{A_2} &= s_1 m^{\vec{b}}_{A_2} \bullet s_2 m^{\vec{c}}_{A_2} \setminus \mathsf{nlist}((s_1 m^{\vec{b}}_{A_2} \,;\, s_2 m^{\vec{c}}_{A_2}).\text{-}1) \\
&= (s_1 \bullet s_2 \,, (\vec{b} \setminus \mathsf{nlist}(s_1.\text{-}1))) \setminus \mathsf{nlist}((s_1 \,;\, s_2).\text{-}1) \\
&= (s_1 \bullet s_2 \setminus \mathsf{nlist}((s_1 \,;\, s_2).\text{-}1)) \,, (\vec{b} \setminus \mathsf{nlist}(s_1.\text{-}1)) \\
&= s_1 \circ s_2 \,, (\vec{b} \setminus \mathsf{nlist}((s_1 \,;\, s_2).\text{-}1)) \,, \\
(s_1 m^{\vec{b}}_{A_2} \,;\, s_2 m^{\vec{c}}_{A_2}) \bullet s_3 \,, (s_1 m^{\vec{b}}_{A_2} \circ s_2 m^{\vec{c}}_{A_2}) &= (s_1 \,;\, s_2) \bullet s_3 \,, s_1 \circ s_2 \,, (\vec{b} \setminus \mathsf{nlist}(s_1.\text{-}1)) \\
&\overset{IH}{=} s_1 \bullet s_2 \bullet s_3 \,, (\vec{b} \setminus \mathsf{nlist}(s_1.\text{-}1)) = s_1 m^{\vec{b}}_{A_2} \bullet s_2 m^{\vec{c}}_{A_2} \bullet s_3 \\
s_1 m^{\vec{b}}_{A_2} \bullet (s_2 m^{\vec{c}}_{A_2} \,;\, s_3) \,, s_2 m^{\vec{c}}_{A_2} \circ s_3 &= s_1 m^{\vec{b}}_{A_2} \bullet (s_2 \,;\, s_3) m^{\vec{c}}_{A_2} = s_1 \bullet (s_2 \,;\, s_3) \,, (\vec{b} \setminus \mathsf{nlist}(s_1.\text{-}1)) \\
&\underset{s_2 \circ s_3 = \epsilon}{\overset{IH}{=}} s_1 \bullet s_2 \bullet s_3 \,, (\vec{b} \setminus \mathsf{nlist}(s_1.\text{-}1)) = s_1 m^{\vec{b}}_{A_2} \bullet s_2 m^{\vec{c}}_{A_2} \bullet s_3 \,.
\end{aligned}$$

The case of m_{A_2} being a P-move in $A_2 \to A_3$ is entirely symmetrical.

↬ The other cases are shown similarly. ∎

The two conditions in the previous lemma are sometimes equivalent.

Lemma 3.23 *If $s_1 \in P_{A_1 \to A_2}$, $s_2 \in P_{A_2 \to A_3}$, $s_3 \in P_{A_3 \to A_4}$ and either s_1 ends in A_1 or s_3 in A_4 then*

$$(s_1 \,;\, s_2) \asymp s_3 \iff s_1 \asymp (s_2 \,;\, s_3)$$

Proof: We show only the left-to-right implication; the other is shown similarly. Note that we may use lemma 3.13 without further mention.
Assume $(s_1 \,;\, s_2) \asymp s_3$. It is easy to see that $s_1 \asymp s_2 \smile s_3$, so, using the assumption for $A_1 A_4$, not both s_1, s_2 end in A_2 nor both s_2, s_3 end in A_3. Now let $s_2' \leq s_2$ and $s_3' \leq s_3$ with $s_2' \smile s_3'$. If $s_2'.\text{-}1$ introduces a (and s_3' ends in A_3) then, if this introduction occurs in A_3 then $(s_1' \,;\, s_2').\text{-}1$ introduces a, for relevant $s_1' \leq s_1$, so $a \# s_3'^{-}$. If the introduction occurs in A_2 then there exist least prefixes $s_1'' \leq s_1$ and $s_2' \leq s_2'' \leq s_2$ such that $|s_1'' \,;\, s_2''| = |s_1' \,;\, s_2'| + 1$ and $(s_1'' \,;\, s_2'').\text{-}1$ introduces a. Hence, $a \# s_3'$. On the other hand, if $s_3'.\text{-}1$ introduces a and s_2' ends in A_3 then, taking relevant $s_1' \leq s_1$, either $a \# s_1' \,;\, s_2'$ or $a \# (s_1' \,;\, s_2')^{-}$, according to whether a being introduced in A_4 or in A_3, which implies $a \# s_2'$ or $a \# s_2'^{-}$. Hence, $s_2 \asymp s_3$.
It is not difficult to see that $s_1 \smile (s_2 \,;\, s_3)$. Now let $s_i' \leq s_i$, $i = 1, 2, 3$, with $s_1' \smile (s_2' \,;\, s_3')$ and s_2', s_3' not both ending in A_3, so $(s_1' \,;\, s_2') \smile s_3'$ and thus $(s_1' \,;\, s_2') \asymp s_3'$. Assume $s_1'.\text{-}1$ introduces a name a and $s_2' \,;\, s_3'$ ends in A_2. If the introduction occurs in A_1 then $(s_1' \,;\, s_2').\text{-}1$ also introduces a, so $a \# s_2', s_3'$, $\therefore \;\; a \# (s_2' \,;\, s_3')$. If it occurs in A_2 then there exist least prefixes $s_1' \leq s_1'' \leq s_1$ and $s_2' \leq s_2'' \leq s_2$ such that $|s_1'' \,;\, s_2''| = |s_1' \,;\, s_2'| + 1$ and $(s_1'' \,;\, s_2'').\text{-}1$ introduces a. Hence, $a \# s_3'$ and, as $a \# s_2'^{-}$, $a \# (s_2' \,;\, s_3')^{-}$. Now assume $(s_2' \,;\, s_3')\text{-}1$ introduces a and s_1' ends in A_2. If the introduction occurs in A_2 then a is introduced by $s_2''.\text{-}1$, some relevant $s_2'' \leq s_2'$, or by $s_3''.\text{-}1$, some relevant $s_3'' \leq s_3'$. In the former case, $s_1' \asymp s_2'$ implies $a \# s_1'^{-}$. In the latter, a is introduced in A_3 by $s_3''.\text{-}1$ and, taking $s_2'' \leq s_2'$ with $s_2'' \asymp s_3''$, we have $s_1'^{-} \asymp s_2''$ and $a \# (s_1'^{-} \,;\, s_2'')^{-}$, so $a \# s_1'^{-}$. If the introduction occurs in A_4 then we follow a similar reasoning. ∎

With the results we have gathered we obtain a category of nominal games.

Proposition 3.24 *For any $\sigma : A \longrightarrow B$, $\mathtt{id}_A \,;\, \sigma = \sigma = \sigma \,;\, \mathtt{id}_B$.*
Moreover, for any $\sigma_1 : A' \longrightarrow A$ and $\sigma_3 : B \longrightarrow B'$, $(\sigma_1 \,;\, \sigma) \,;\, \sigma_3 = \sigma_1 \,;\, (\sigma \,;\, \sigma_3)$.

Proof: The first part is straightforward. For the second part, take some $[u] \in (\sigma_1 \,;\, \sigma) \,;\, \sigma_3$. By prefix-closure we may assume that $u = s \,;\, s_3$ with s and s_3 not both ending in B, and that

$s = s_1 ; s_2$ with s_1 and s_2 not both ending in A, so

$$\begin{aligned} u = (s_1 ; s_2) ; s_3 \wedge [s_1] \in \sigma_1 \wedge [s_2] \in \sigma \wedge [s_3] \in \sigma_3 \wedge (s_1 ; s_2) \asymp s_3 \\ \therefore\ u = s_1 ;(s_2 ; s_3) \wedge [s_1] \in \sigma_1 \wedge [s_2] \in \sigma \wedge [s_3] \in \sigma_3 \wedge s_1 \asymp (s_2 ; s_3) \\ \therefore\ [u] \in \sigma_1 ;(\sigma ; \sigma_3) \end{aligned}$$

Thus $(\sigma_1 ; \sigma_2) ; \sigma_3 \subseteq \sigma_1 ;(\sigma_2 ; \sigma_3)$ and similarly the other inclusion. ■

Definition 3.25 (Category of nominal games) $\mathcal{G}$ is the category having nominal arenas as objects and nominal strategies as arrows. ▲

3.1.3 Arena and strategy orders in $\mathcal{G}$

$\mathcal{G}$ is the raw material from which several subcategories of nominal games will emerge. Still though there is structure in $\mathcal{G}$ which will be inherited to the refined subcategories we will consider later on. In particular, we will consider orderings for arenas and strategies, the latter enriching $\mathcal{G}$ over Cpo.[4]

Strategies are (nominal) sets and hence ordered by the subset relation.

Definition 3.26 (Strategy order) For any arenas A, B and each $\sigma, \tau \in \mathcal{G}(A, B)$ define:

$$\sigma \sqsubseteq \tau \iff^{\Delta} \sigma \subseteq \tau\,.$$

For each $\sqsubseteq$-increasing sequence $(\sigma_i)_{i\in\omega}$ take $\bigsqcup_i \sigma_i \triangleq \bigcup_i \sigma_i$. ▲

It is easy to see that each such a $\bigsqcup_i \sigma_i$ is indeed a strategy: prefix closure, contingency completeness and determinacy easily follow from the fact that the sequences we consider are $\sqsubseteq$-increasing. Hence, each $\mathcal{G}(A, B)$ is a cpo with least element the empty strategy (i.e. the strategy containing only $[\epsilon]$ and $[i_A]$'s). More than that, these cpo's enrich $\mathcal{G}$.

Proposition 3.27 *$\mathcal{G}$ is Cpo-enriched wrt $\sqsubseteq$.*

Proof: Enrichment amounts to showing the following straightforward assertions.

$$\begin{aligned} \sigma \sqsubseteq \sigma' \wedge \tau \sqsubseteq \tau' &\implies \sigma ; \tau \sqsubseteq \sigma' ; \tau' \\ (\sigma_i)_{i\in\omega} \text{ an } \omega\text{-chain} &\implies (\bigsqcup_{i\in\omega} \sigma_i) ; \tau \sqsubseteq \bigsqcup_{i\in\omega} (\sigma_i ; \tau) \\ (\tau_i)_{i\in\omega} \text{ an } \omega\text{-chain} &\implies \sigma ;(\bigsqcup_{i\in\omega} \tau_i) \sqsubseteq \bigsqcup_{i\in\omega} (\sigma ; \tau_i) \end{aligned}$$

■

On the other hand, arenas are tuples and hence also ordered by a 'subset relation'.

Definition 3.28 (Arena order) For any $A, B \in Ob(\mathcal{G})$ define

$$A \trianglelefteq B \iff M_A \subseteq M_B \wedge I_A \subseteq I_B \wedge \lambda_A \subseteq \lambda_B \wedge \vdash_A \subseteq \vdash_B,$$

and for any $\trianglelefteq$-increasing sequence $(A_i)_{i\in\omega}$ define

$$\bigsqcup_{i\in\omega} A_i \triangleq \bigcup_{i\in\omega} A_i\,.$$

If $A \trianglelefteq B$ then we can define an embedding-projection pair of arrows by setting

$$\mathtt{incl}_{A,B} : A \longrightarrow B \triangleq \{[s] \mid s \in P_{A\to B} \wedge ([s] \in \mathtt{id}_A \vee (odd(|s|) \wedge [s^-] \in \mathtt{id}_A))\},$$

$$\mathtt{proj}_{B,A} : B \longrightarrow A \triangleq \{[s] \mid s \in P_{B\to A} \wedge ([s] \in \mathtt{id}_A \vee (odd(|s|) \wedge [s^-] \in \mathtt{id}_A))\}.$$

▲

[4] By cpo we mean a partially ordered set with least element and least upper bounds for increasing ω-sequences. Cpo is the category of cpos and continuous functions.

It is straightforward to see that $\bigsqcup_{i\in\omega} A_i$ is well-defined, and that $\trianglelefteq$ forms a cpo on $Ob(\mathcal{G})$ with least element the empty arena 0. By $\mathtt{incl}_{A,B}$ and $\mathtt{proj}_{B,A}$ being an embedding-projection pair we mean that

$$\mathtt{incl}_{A,B}\,;\mathtt{proj}_{B,A} = \mathtt{id}_A \quad \wedge \quad \mathtt{proj}_{B,A}\,;\mathtt{incl}_{A,B} \sqsubseteq \mathtt{id}_B\,.$$

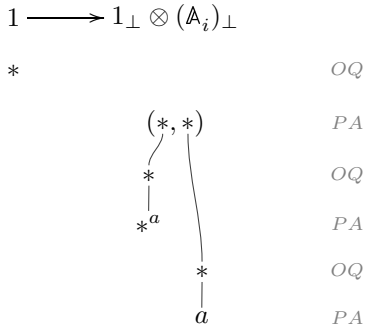
Note that in essence both $\mathtt{incl}_{A,B}$ and $\mathtt{proj}_{B,A}$ are equal to $\mathtt{id}_A$, the latter seen as a partially defined strategy on prearenas $A \to B$ and $B \to A$. Finally, it is easy to show the following.

$$A \trianglelefteq B \trianglelefteq C \implies \mathtt{incl}_{A,B}\,;\mathtt{incl}_{B,C} = \mathtt{incl}_{A,C} \tag{TRN}$$

3.2 Innocence

In game semantics for pure functional languages, the absence of computational effects corresponds to innocence in the strategies. Here, although our aim is to model languages with effects, our models will be constructed by use of innocent strategies: the effects will still be achieved, by using monads.

Innocence is the condition stipulating that strategies be completely determined by their behaviour on P-views. In our current setting the manipulation of P-views presents some difficulties since P-views of plays need not be plays themselves. For example, the P-view of the following play (where curved lines stand for justification pointers) is $*\,(*,*)\,*\,a$ and violates (NC2).



We rectify these problems by explicitly imposing innocence on plays too.

3.2.1 The subcategory $\mathcal{V}$

Definition 3.29 A play s is *innocent* if, for any $t \leq s$, $\ulcorner t \urcorner$ is a play. The set of innocent plays of A is denoted by P_A^{i}. ▲

The explicit condition of the above definition can be replaced by a more familiar-looking Name-Condition.

Proposition 3.30 *A legal sequence s is an innocent play iff $s.1$ has empty name-list and s satisfies* (NC1), (NC3) *and the following condition.*

(NC2′) *Any name in the support of a P-move x in s that is fresh for $\ulcorner s_{<x} \urcorner$ is contained in the name-list of x.*

Proof: If s is an innocent play then it satisfies (NC1,3). Moreover, if $a \in \mathsf{S}(x)$ and $a \# \ulcorner s_{<x} \urcorner = \ulcorner s_{\leq x} \urcorner_{<x}$ for some P-move x in s, then $\ulcorner s_{\leq x} \urcorner$ being a play implies that $a \in \mathsf{nlist}(x)$.
Conversely, if s satisfies (NC1,3) and (NC2′) then it clearly is a play. Take now some $t \leq s$; we need to show that $\ulcorner t \urcorner$ is a play. By ordinary game-semantics analysis, $\ulcorner t \urcorner$ is a legal sequence. Moreover, (NC3) is inherited from t. For (NC1), let x be a P-move in $\ulcorner t \urcorner$ and let y be the move preceding it. yx are consecutive in t and hence $\mathsf{nlist}(y) \leq \mathsf{nlist}(x)$. Moreover, if $a \in \mathsf{nlist}(x)$ and $a \# \mathsf{nlist}(y)$ then $a \# t_{<x}$ and thus $a \# \ulcorner t \urcorner_{<x}$. Finally, (NC2) for $\ulcorner t \urcorner$ is derived

from (NC2′) for t. ■

Summarising:

A legal sequence s is an ***innocent play*** if $s.1$ has empty name-list and s also satisfies the following Name Change Conditions:

(NC1) The name-list of a P-move x in s contains as a prefix the name-list of the move preceding it. It possibly contains some other names, all of which are fresh for $s_{<x}$.

(NC2′) Any name in the support of a P-move x in s that is fresh for $\ulcorner s_{<x} \urcorner$ is contained in the name-list of x.

(NC3) The name-list of a non-initial O-move in s is that of the P-move explicitly justifying it.

Figure 3.2: Definition of innocent play.

We can obtain the following characterisation of name-introduction in innocent plays.

Proposition 3.31 (Name-introduction) *Let s be an innocent play. A name a is introduced by Player in s iff there exists a P-move x in s such that $a \in \mathsf{S}(x)$ and $a \# \ulcorner s_{<x} \urcorner$.*

Proof: If a is introduced by a P-move x in s then $a \in \mathsf{nlist}(x)$ and $a \# \mathsf{nlist}(s_{<x}.{-}1)$, hence, by (NC1), $a \# s_{<x}$ so $a \# \ulcorner s_{<x} \urcorner$.
Conversely, if $a \in \mathsf{S}(x)$ and $a \# \ulcorner s_{<x} \urcorner$ then by (NC2′) we get $a \in \mathsf{nlist}(x)$, while $a \# \ulcorner s_{<x} \urcorner$ implies $a \# \mathsf{nlist}(s_{<x}.{-}1)$. ■

We proceed to show that innocent plays are closed under composition. First, we define P-views of interaction sequences. Recall that in an interaction sequence of A, B, C a move is a generalised P-move if it is either a P-move in AC or a move in B. The component of a generalised P-move x is AB if x represents a P-move in component AB, otherwise it is BC. Similar things apply for generalised O-moves.

Definition 3.32 Let $w \in \mathsf{ISeq}(A, B, C)$. Define its P-view $\ulcorner w \urcorner$ by recursion as follows.

$$
\begin{array}{rcll}
\ulcorner \epsilon \urcorner & \triangleq & \epsilon & \\
\ulcorner x \urcorner & \triangleq & x & \text{if } x \text{ an initial move in } A, \\
\ulcorner wx \urcorner & \triangleq & \ulcorner w \urcorner\, x & \text{if } x \text{ a generalised P-move}, \\
\ulcorner wxw'y \urcorner & \triangleq & \ulcorner w \urcorner\, xy & \text{if } y \text{ an O-move in } AC \text{ justified by } x.
\end{array}
$$

▲

We will need the following results, which are taken verbatim from [McC00].

Lemma 3.33

(a) *Let s be a legal sequence and y be a P-move. If $\ulcorner s \urcorner\, y$ is legal then sy is.*

(b) *Let $w \in \mathsf{ISeq}(A, B, C)$ and let x be a generalised O-move of w with component X. If x is not initial in X, write y for its justifier and y' for the move immediately before y. Then,*

(1) *If x is not initial in X then y' is a generalised O-move with component X.*

(2) *If x is not initial in X and appears in $\ulcorner w \urcorner$ then both y and y' appear in $\ulcorner w \urcorner$.*

(3) *If x appears in $\ulcorner w \urcorner$ then $\ulcorner \ulcorner w \urcorner_{\leq x} \upharpoonright X \urcorner = \ulcorner w_{\leq x} \upharpoonright X \urcorner$.* ■

We will also need the following lemmata. Note that if x is a move in s and $s \parallel t = (s_{<x} \parallel t')\tilde{x}w$ then we say that x *appears in* $s \parallel t$ *as* $\tilde{x}$.

Lemma 3.34 *If $s_1, s_2 \in P_{A\to B}$, $t_1, t_2 \in P_{B\to C}$ and $s_i \parallel t_i$ end in a generalised O-move x,*

(a) *if x has component AB then* $\ulcorner(s_1 \parallel t_1) \upharpoonright AB\urcorner = \ulcorner(s_2 \parallel t_2) \upharpoonright AB\urcorner \Longrightarrow \ulcorner s_1\urcorner = \ulcorner s_2\urcorner$,

(b) *if x has component BC then* $\ulcorner(s_1 \parallel t_1) \upharpoonright BC\urcorner = \ulcorner(s_2 \parallel t_2) \upharpoonright BC\urcorner \Longrightarrow \ulcorner t_1\urcorner = \ulcorner t_2\urcorner$.

Proof: We show (a) by induction on $|s_1| \geq 1$, and (b) is proved similarly. If $|s_1| = 1$ then $|s_2| = 1$ and, clearly, $s_1 = s_2$. If $s_1 = s_1' n^{\vec{b}_1} s_1'' m^{\vec{b}_1}$, with m an O-move in A justified by n, then $s_2 = s_2' n^{\vec{b}_2} s_2'' m^{\vec{b}_2}$. Moreover, $t_i = t_i' t_i''$, $i = 1, 2$, and $s_i \parallel t_i = (s_i' \parallel t_i') n^{\vec{b}_i'} u_i m^{\vec{b}_i'}$, with $\vec{b}_i' = (s_i' \bullet t_i')(\vec{b}_i \setminus \mathsf{nlist}(s_i'.{-}1))$, and hence $\ulcorner(s_1 \parallel t_1) \upharpoonright AB\urcorner = \ulcorner(s_2 \parallel t_2) \upharpoonright AB\urcorner$ implies that $\ulcorner(s_1' \parallel t_1') \upharpoonright AB\urcorner = \ulcorner(s_2' \parallel t_2') \upharpoonright AB\urcorner$, $s_1' \bullet t_1' = s_2' \bullet t_2'$ and $\vec{b}_1' = \vec{b}_2'$. By IH, $\ulcorner s_1'\urcorner = \ulcorner s_2'\urcorner$ and $\vec{b}_1 = \vec{b}_2$, as $\vec{b}_i = \mathsf{nlist}(s_i'.{-}1)(\vec{b}_i' \setminus \mathsf{nlist}(s_1' \bullet t_i'))$. Thus, $\ulcorner s_1\urcorner = \ulcorner s_2\urcorner$.
If $s_1 = s_1' n^{\vec{b}_1} s_1'' m^{\vec{b}_1}$, with m an O-move in B justified by n, then $s_2 = s_2' n^{\vec{b}_2} s_2'' m^{\vec{b}_2}$, and we work similarly to the previous case. ■

Lemma 3.35 *Let $s \in P_{A\to B}$, $t \in P_{B\to C}$ be innocent and $s \asymp t$. Then,*

(a) *there exist innocent $s' \in P_{A\to B}$, $t' \in P_{B\to C}$ with $s' \asymp t'$ such that $\ulcorner s \parallel t\urcorner = s' \parallel t'$,*

(b) $\mathtt{S}(\ulcorner s \parallel t\urcorner) = \mathtt{S}(\ulcorner s\,; t\urcorner) \cup \mathtt{S}(s \bullet t)$,

(c) *if $s \parallel t$ ends in a generalised O-move in AB and x appears in $\ulcorner s\urcorner$ then x appears in $\ulcorner s \parallel t\urcorner$ as some $\tilde{x}$; similarly for BC and t.*

Proof: For (a), we do induction on $|s \parallel t| \geq 1$. The base case is trivial. Otherwise, let $s \parallel t$ be ending in an O-move in AC, say wlog in a move x in A. Then $s = s_1 s_2 x$ and $t = t_1 t_2$, with x justified by $s_1.{-}1$, so $\ulcorner s \parallel t\urcorner = \ulcorner s_1 \parallel t_1\urcorner x'$. By IH, there exist s', t' such that $\ulcorner s_1 \parallel t_1\urcorner = s' \parallel t'$. Now, it is easy to see that $s'x$ is a play and that $\ulcorner s \parallel t\urcorner = s'x \parallel t'$.
If $s \parallel t$ ends in a P-move in AC, say wlog in a move x in A, then $s = s^- x$ and $s \parallel t = (s^- \parallel t)x'$. By IH, $\ulcorner s^- \parallel t\urcorner = s' \parallel t'$, some s', t'. Since $s^- \parallel t$ ends in a generalised O-move, we have that $\ulcorner(s^- \parallel t) \upharpoonright AB\urcorner = \ulcorner\urcorner s^- \parallel t\urcorner \upharpoonright AB\urcorner = \ulcorner(s' \parallel t') \upharpoonright AB\urcorner$, so $\ulcorner s^-\urcorner = \ulcorner s'\urcorner$. Hence, $\ulcorner s'\urcorner x$ is a play. We have that $s'x$ is a legal sequence and also that it satisfies (NC2'-3). For (NC1), if $b \in (\mathsf{nlist}(x) \setminus \mathsf{nlist}(s'.{-}1))$ then $b \in (\mathsf{nlist}(x) \setminus \mathsf{nlist}(s^-.{-}1))$, so $b \# s^-, t$, $\therefore\ b \# \ulcorner s^- \parallel t\urcorner = s' \parallel t'$, $\therefore\ b \# s'$. Hence, $\ulcorner s \parallel t\urcorner = s'x \parallel t$.
If $s \parallel t$ ends in a B-move then we combine the treatments of the two cases above.
For (b), we have that $\mathtt{S}(\ulcorner s \parallel t\urcorner) = \mathtt{S}(s' \parallel t') = \mathtt{S}(s'\,; t') \cup \mathtt{S}(s' \bullet t') = \mathtt{S}(\ulcorner s\,; t\urcorner) \cup \mathtt{S}(s \bullet t)$.
For (c), we have that $\ulcorner s \parallel t\urcorner = s' \parallel t'$ and x appears in s', as $\ulcorner(s \parallel t) \upharpoonright AB\urcorner = \ulcorner\urcorner s \parallel t\urcorner \upharpoonright AB\urcorner = \ulcorner(s' \parallel t') \upharpoonright AB\urcorner$, $\therefore\ \ulcorner s\urcorner = \ulcorner s'\urcorner$. ■

Now we can show the following.

Proposition 3.36 *If $s \in P_{A\to B}$, $t \in P_{B\to C}$ are innocent and $s \asymp t$ then $s\,; t$ is innocent.*

Proof: Suppose $s\,; t$ is not innocent and let u be its least prefix manifesting this, so $\ulcorner u\urcorner$ is not a play. Then $\ulcorner u\urcorner$ doesn't satisfy (NC2). By leastness, $u = u' m^{\vec{b}}$, m a P-move and there exists some a such that $a \in \mathtt{S}(m)$, $a \in \mathtt{S}(u')$ but $a \# \vec{b}, \ulcorner u'\urcorner$.
Suppose m is in A, so $s = s' m^{\vec{b}'} s''$, $t = t' t''$ and $u' = s'\,; t'$. $\vec{b}'$ is contained in $\vec{b}$, so $a \# \vec{b}'$ and hence $a \in \mathtt{S}(\ulcorner s'\urcorner)$, since s is innocent. Now, $s' \parallel t'$ O-ends in AB and $\vec{b}$ contains $s' \bullet t'$, so

$$\begin{aligned}\mathtt{S}(\ulcorner s'\urcorner) &\subseteq \mathtt{S}(\ulcorner(s' \parallel t') \upharpoonright AB\urcorner) = \mathtt{S}(\ulcorner\urcorner s' \parallel t'\urcorner \upharpoonright AB\urcorner) \subseteq \mathtt{S}(\ulcorner s' \parallel t'\urcorner \upharpoonright AB) \subseteq \mathtt{S}(\ulcorner s' \parallel t'\urcorner)\\ &= \mathtt{S}(\ulcorner s'\,; t'\urcorner) \cup \mathtt{S}(s' \bullet t') \subseteq \mathtt{S}(\ulcorner s'\,; t'\urcorner) \cup \mathtt{S}(\vec{b})\,.\end{aligned}$$

Thus, $a \in \mathtt{S}(\ulcorner s'\,; t'\urcorner)$ or $a \in \vec{b}$, $\lightning$. Similarly if m is in C. ■

We move on to innocent strategies.

Definition 3.37 A strategy σ is ***innocent*** if $[s] \in \sigma$ implies that s is innocent, and if even-length $[s_1 x_1] \in \sigma$ and odd-length $[s_2] \in \sigma$ have $[\ulcorner s_1 \urcorner] = [\ulcorner s_2 \urcorner]$ then there exists x_2 such that $[s_2 x_2] \in \sigma$ and $[\ulcorner s_1 x_1 \urcorner] = [\ulcorner s_2 x_2 \urcorner]$. ▲

Some nice properties of innocent strategies are the following.

Lemma 3.38 *Let σ be an innocent strategy.*

(1) *If $[s] \in \sigma$ then $[\ulcorner s \urcorner] \in \sigma$.*

(2) *If sy is an even-length innocent play and $[s], [\ulcorner sy \urcorner] \in \sigma$ then $[sy] \in \sigma$.*

(3) *If $\ulcorner sy \urcorner$ is even-length with $\mathsf{nlist}(y) = \mathsf{nlist}(s_{\cdot}{-}1)$ and $[s], [\ulcorner sy \urcorner] \in \sigma$ then $[sy] \in \sigma$.*

(4) *If s is an even-length innocent play and, for any even-length prefix s' of s, $[\ulcorner s' \urcorner] \in \sigma$ then $[s] \in \sigma$.*

Proof: For (1) we do induction on $|s|$. The base case is trivial. Now, if $s = s'y$ with y a P-move then $\ulcorner s \urcorner = \ulcorner s' \urcorner y$ and $[\ulcorner s' \urcorner] \in \sigma$ by prefix closure and IH. By innocence, there exists y' such that $[\ulcorner s' \urcorner y'] \in \sigma$ and $[\ulcorner s' \urcorner y'] = [\ulcorner sy \urcorner]$, so done. If $s = s_1 y s_2 x$ and x an O-move justified by y then $[\ulcorner s_1 y \urcorner] \in \sigma$ by prefix closure and IH, hence $[\ulcorner s_1 y \urcorner x] \in \sigma$ by contingency completeness.
For (2) note that by innocence we have $[sy'] \in \sigma$ for some y' such that $[\ulcorner sy \urcorner] = [\ulcorner sy' \urcorner]$. Then,

$$[\ulcorner s \urcorner, y] = [\ulcorner s \urcorner, y'] \wedge [\ulcorner s \urcorner, s] = [\ulcorner s \urcorner, s] \wedge (\mathtt{S}(y) \setminus \mathtt{S}(\ulcorner s \urcorner)) \cap \mathtt{S}(s) = (\mathtt{S}(y') \setminus \mathtt{S}(\ulcorner s \urcorner)) \cap \mathtt{S}(s) = \varnothing$$

Thus we can apply the strong support lemma and get $[sy] = [sy']$, as required.
For (3) it suffices to show that sy is an innocent play. As $s, \ulcorner s \urcorner y$ are plays, it suffices to show that sy satisfies the name conditions at y. (NC3) and (NC2') hold because $\ulcorner sy \urcorner$ a play. (NC1) also holds, as y is non-introducing.
For (4) we do induction on $|s|$. The base case is encompassed in $\ulcorner s \urcorner = s$, which is trivial. For the inductive step, let $s = s^- x$ with $\ulcorner s \urcorner \neq s$. By IH and contingency completeness we have $[s^-] \in \sigma$, and since $[\ulcorner s \urcorner] \in \sigma$, by (2), $[s] \in \sigma$. ■

We now want to show that innocent strategies are closed with respect to composition. We will need the following technical lemmata.

Lemma 3.39 *Let $\sigma : A \to B, \tau : B \to C$ be innocent strategies and let $[s_i] \in \sigma$, $[t_i] \in \tau$, $s_i = s_i' s_i''$, $t_i = t_i' t_i''$, $s_i \asymp t_i$, $\underline{s_i''} = \underline{t_i''}$ in B, $i = 1, 2$, and also $|s_1''| = |s_2''|$. Then*

(a) $[\ulcorner s_1' \parallel t_1' \urcorner] = [\ulcorner s_2' \parallel t_2' \urcorner] \Longrightarrow [\ulcorner s_1 \parallel t_1 \urcorner] = [\ulcorner s_2 \parallel t_2 \urcorner]$,

(b) $(\ulcorner s_1' \parallel t_1' \urcorner = \ulcorner s_2' \parallel t_2' \urcorner \wedge s_1 \bullet t_1 = s_2 \bullet t_2) \Longrightarrow \ulcorner s_1 \parallel t_1 \urcorner = \ulcorner s_2 \parallel t_2 \urcorner$.

Proof: (a) is proved by induction on $k = |s_i''| = |t_i''|$, using also lemma 3.34.
For (b), if $\ulcorner s_1' \parallel t_1' \urcorner = \ulcorner s_2' \parallel t_2' \urcorner$ then $\pi \circ \ulcorner s_1 \parallel t_1 \urcorner = \ulcorner s_2 \parallel t_2 \urcorner$ for some $\pi \circ \vec{a} = \vec{a}$, by (a). Then, π must be fixing $\ulcorner s_1' \parallel t_1' \urcorner$, and if $s_1 \bullet t_1 = s_2 \bullet t_2$ then π fixes also $s_1 \bullet t_1$. Now, using lemma 3.35,

$$\mathtt{S}(\ulcorner s_1 \parallel t_1 \urcorner) = \mathtt{S}(\ulcorner s_1 ; t_1 \urcorner) \cup \mathtt{S}(s_1 \bullet t_1) = \mathtt{S}(\ulcorner s_1' ; t_1' \urcorner) \cup \mathtt{S}(s_1 \bullet t_1) \subseteq \mathtt{S}(\ulcorner s_1' \parallel t_1' \urcorner) \cup \mathtt{S}(s_1 \bullet t_1).$$

Hence, $\mathtt{S}(\ulcorner s_1 \parallel t_1 \urcorner) = \mathtt{S}(\ulcorner s_1' \parallel t_1' \urcorner) \cup \mathtt{S}(s_1 \bullet t_1)$. Therefore, π fixes $\ulcorner s_1 \parallel t_1 \urcorner$, as required. ■

Lemma 3.40 *Let $\sigma : A \to B, \tau : B \to C$ be innocent strategies and let $[s_i] \in \sigma$, $[t_i] \in \tau$ and $s_i ; t_i$ be O-ending plays, $i = 1, 2$. Then*

$$\ulcorner s_1 ; t_1 \urcorner = \ulcorner s_2 ; t_2 \urcorner \Longrightarrow \ulcorner s_1 \parallel t_1 \urcorner = \ulcorner s_2 \parallel t_2 \urcorner .$$

Proof: By induction on $k = |s_1 \parallel t_1| \geq 1$. Let $s_i ; t_i$ be $(s_i' ; t_i')m^{\vec{b}}$. If $k = 1$ then ok.
Otherwise, m is not initial, so let $n^{\vec{b}}$ be the (common) move justifying $m^{\vec{b}}$ inside $s_i ; t_i$, and x be the move immediately preceding $n^{\vec{b}}$. Then, $(s_i' \parallel t_i')m^{\vec{b}} = (s_i \parallel t_i)_{\leq x} w_i n^{\vec{b}} w_i' m^{\vec{b}}$ with w_i in B, and, by IH and assumption, $\ulcorner (s_1 \parallel t_1)_{\leq x} \urcorner = \ulcorner (s_2 \parallel t_2)_{\leq x} \urcorner$.
So, we need only show $\ulcorner (s_1 \parallel t_1)_{\leq x} w_1 \urcorner = \ulcorner (s_2 \parallel t_2)_{\leq x} w_2 \urcorner$. Let's say $(s_i \parallel t_i)_{\leq x} w_i = u_i u_i' \parallel v_i v_i'$ with $(s_i \parallel t_i)_{\leq x} = u_i \parallel v_i$ and $\underline{u_i'} = \underline{v_i'}$, $i = 1, 2$.

If $|u_1'| \leq |u_2'|$, let $u_2' = u_{21}' u_{22}'$ and $v_2' = v_{21}' v_{22}'$ with $|u_1'| = |u_{21}'| = |v_{21}'| = |v_1'|$. As $[\ulcorner u_1 \parallel v_1 \urcorner] = [\ulcorner u_2 \parallel v_2 \urcorner]$, by previous lemma we get $[\ulcorner u_1 u_1' \parallel v_1 v_1' \urcorner] = [\ulcorner u_2 u_{21}' \parallel v_2 v_{21}' \urcorner]$. If, say, these O-end in AB then $[\ulcorner u_1 u_1' \urcorner] = [\ulcorner u_2 u_{21}' \urcorner]$, by lemma 3.34, so, because of innocence and prefix-closure of σ, $u_{22}'.1$ is in the same arena as n, which is not B (and similarly if they O-end in BC). Hence, $u_{22}' = v_{22}' = \epsilon$ and $[\ulcorner u_1 u_1' \parallel v_1 v_1' \urcorner] = [\ulcorner u_2 u_2' \parallel v_2 v_2' \urcorner]$, as required. ∎

Lemma 3.41 *Let $s \in P_{A \to B}$, $t \in P_{B \to C}$ with $s \asymp t$. If s is O-ending and $sx \in P_{A \to B}$, for some x in B, then there exists x' in B such that $tx' \in P_{B \to C}$ and $sx \smile tx'$. Moreover, if $\mathsf{S}(x) \subseteq \mathsf{S}(s)$ then $sx \asymp tx'$.*
Similarly if t is O-ending and $tx \in P_{B \to C}$.

Proof: We only show the sx case; the other one is shown similarly. Let $y = m^{\vec{b}}$ be x's justifier inside sx. Then y appears in t as some $y' = m^{\vec{b}'}$. If $x = n^{\vec{c}}$ then we claim that $tn^{\vec{b}'} \in P_{B \to C}$. Since $tn^{\vec{c}}$ clearly satisfies the name-conditions (NC1-3), it suffices to show that $tn^{\vec{c}}$ is legal, that is, it suffices to show that $\underline{tn}$ is legal. Now, $\underline{tn}$ is a justified sequence of moves, so we need only show it satisfies Visibility and Well-Bracketing.
As sx is legal, $\underline{sn}$ is legal and therefore, by results for ordinary game semantics (e.g. [HO00, prop. 4.4] or [HY99, prop. A.9]), $(\underline{s}.1)(\underline{sn} \upharpoonright B)$ is legal. Therefore, $\underline{y}$ appears in $\ulcorner \underline{s} \upharpoonright B \urcorner = \llcorner t \lrcorner$, and thus $\underline{y'}$ appears in $\llcorner t \lrcorner$ showing visibility. For well-bracketing, we need only consider the case of n being an answer. In this case, $\underline{y}$ is the pending question of $\underline{s} \upharpoonright B = \underline{t} \upharpoonright B$, and therefore $\underline{t}$ has a pending question. In fact, its pending question is $\underline{y'}$: had some question z in $\underline{t} \upharpoonright C$ been left unanswered, the last switch-move w from C to B would have necessarily been a P-question. But $\underline{t}_{\geq w}$ is odd-length, so there must either be a pending-Q in it — can't happen as z is the pending-Q — or an externally justified answer — which would violate well-bracketing. Thus, $\underline{tn}$ is legal, and hence tx' is a play.
Finally, if $\mathsf{S}(x) \subseteq \mathsf{S}(s)$ then C1 and C2 are also satisfied. ∎

The main lemma is the following, from which we prove that innocent strategies are closed under composition.

Lemma 3.42 *Let $\sigma : A \to B$, $\tau : B \to C$ be innocent strategies and $[s_i] \in \sigma$, $[t_i] \in \tau$ with $s_i \asymp t_i$, $i = 1, 2$, and $[\ulcorner s_1 \parallel t_1 \urcorner] = [\ulcorner s_2 \parallel t_2 \urcorner]$.*

- **(a)** *If $[s_1 s_1'] \in \sigma$, $[t_1 t_1'] \in \tau$ for sequences s_1', t_1' in B such that $s_1 s_1' \asymp t_1 t_1'$, then there exist sequences s_2', t_2' in B such that $s_2 s_2' \asymp t_2 t_2'$, $[s_2 s_2'] \in \sigma$, $[t_2 t_2'] \in \tau$ and $[\ulcorner s_1 s_1' \parallel t_1 t_1' \urcorner] = [\ulcorner s_2 s_2' \parallel t_2 t_2' \urcorner]$.*
- **(b)** *If $[s_1 m_1^{\vec{b}_1}] \in \sigma$ for some P-move m in A such that $s_1 m_1^{\vec{b}_1} \asymp t_1$ then there exists $m_2^{\vec{b}_2}$ such that $s_2 m_2^{\vec{b}_2} \asymp t_2$, $[s_2 m_2^{\vec{b}_2}] \in \sigma$ and $[\ulcorner s_1 m_1^{\vec{b}_1} \parallel t_1 \urcorner] = [\ulcorner s_2 m_2^{\vec{b}_2} \parallel t_2 \urcorner]$.*
- **(c)** *If $[t_1 n_1^{\vec{b}_1}] \in \tau$ for some P-move n in C such that $t_1 n_1^{\vec{b}_1} \asymp s_1$ then there exists $n_2^{\vec{b}_2}$ such that $t_2 n_2^{\vec{b}_2} \asymp s_2$, $[t_2 n_2^{\vec{b}_2}] \in \tau$ and $[\ulcorner s_1 \parallel t_1 n_1^{\vec{b}_1} \urcorner] = [\ulcorner s_2 \parallel t_2 n_2^{\vec{b}_2} \urcorner]$.*

Proof: We only show (a), by induction on $k = |s_1'| = |t_1'|$; (b) and (c) are shown as the induction step in (a). For $k = 0$ ok.
Now let $s_1' = s_1'' m_1^{\vec{b}_1}$ and $t_1' = t_1'' m_1^{\vec{c}_1}$ and assume, wlog, that m_1 a P-move in AB, so

$$[\ulcorner s_1 s_1' \parallel t_1 t_1' \urcorner] = [\ulcorner s_1 s_1'' \parallel t_1 t_1'' \urcorner m_1^{\vec{d}_1}] \text{ with } \vec{d}_1 = s_1 s_1'' \bullet t_1 t_1'', (\vec{b}_1 \setminus \mathsf{nlist}(s_1 s_1''.\text{-}1)) .$$

Then, by IH, there exist s_2'', t_2'' in B such that $[s_2 s_2''] \in \sigma$, $[t_2 t_2''] \in \tau$ and $[\ulcorner s_1 s_1'' \parallel t_1 t_1'' \urcorner] = [\ulcorner s_2 s_2'' \parallel t_2 t_2'' \urcorner]$.
Let's say $\pi \circ \ulcorner s_1 s_1'' \parallel t_1 t_1'' \urcorner = \ulcorner s_2 s_2'' \parallel t_2 t_2'' \urcorner$, some π, so, by lemma 3.34, $\pi \circ \ulcorner s_1 s_1'' \urcorner = \ulcorner s_2 s_2'' \urcorner$ and hence $[\ulcorner s_1 s_1'' \urcorner] = [\ulcorner s_2 s_2'' \urcorner]$. Since $[s_1 s_1'' m_1^{\vec{b}_1}] \in \sigma$ and σ innocent, there exists $m_2^{\vec{b}_2}$ such that $[s_2 s_2'' m_2^{\vec{b}_2}] \in \sigma$ and $[\ulcorner s_1 s_1'' m_1^{\vec{b}_1} \urcorner] = [\ulcorner s_2 s_2'' m_2^{\vec{b}_2} \urcorner]$.
In fact, we can choose $m_2^{\vec{b}_2}$ so that all its names that are fresh for $s_2 s_2''$ are also fresh for $t_2 t_2''$. Thus if $\vec{c}_2$ is the name-list of the justifier of m_2 in $t_2 t_2''$, determined by $m_2^{\vec{b}_2}$'s justifier in $s_2 s_2''$, then, by first part of lemma 3.41, $t_2 t_2'' m_2^{\vec{c}_2} \in P_{B \to C}$ and $s_2 s_2'' m_2^{\vec{b}_2} \asymp t_2 t_2'' m_2^{\vec{c}_2}$. As $m_2^{\vec{c}_2}$ is an O-move, $t_2 t_2'' m_2^{\vec{c}_2}$ is innocent, so $[t_2 t_2'' m_2^{\vec{c}_2}] \in \tau$.
We also have that $(\mathtt{S}(m_1^{\vec{b}_1}) \setminus \mathtt{S}(\ulcorner s_1 s_1'' \urcorner)) \cap \mathtt{S}(\ulcorner s_1 s_1'' \parallel t_1 t_1'' \urcorner) = \varnothing$, by NC1 and C2. Moreover, by choice of $\vec{b}_2$ we have $(\mathtt{S}(m_2^{\vec{b}_2}) \setminus \mathtt{S}(\ulcorner s_2 s_2'' \urcorner)) \cap \mathtt{S}(\ulcorner s_2 s_2'' \parallel t_2 t_2'' \urcorner) = \varnothing$.
Hence, we can apply lemma 2.11, so there exists π' such that $\pi' \circ m_1^{\vec{b}_1} = m_2^{\vec{b}_2}$, $\pi' \circ \ulcorner s_1 s_1'' \urcorner = \ulcorner s_2 s_2'' \urcorner$ and $\pi' \circ \ulcorner s_1 s_1'' \parallel t_1 t_1'' \urcorner = \ulcorner s_2 s_2'' \parallel t_2 t_2'' \urcorner$. Thus,

$$\begin{aligned}
\ulcorner s_2 s_2'' m_2^{\vec{b}_2} \parallel t_2 t_2'' m_2^{\vec{c}_2} \urcorner &= \ulcorner s_2 s_2'' \parallel t_2 t_2'' \urcorner \, m_2^{s_2 s_2'' \bullet t_2 t_2'', (\vec{b}_2 \setminus \mathsf{nlist}(s_2 s_2''.-1))} \\
&= (\pi' \circ \ulcorner s_1 s_1'' \parallel t_1 t_1'' \urcorner)(\pi' \circ m_1)^{\pi' \circ (s_1 s_1'' \bullet t_1 t_1''), \pi' \circ (\vec{b}_1 \setminus \mathsf{nlist}(s_1 s_1''.-1))} \\
&= \pi' \circ \ulcorner s_1 s_1'' \parallel t_1 t_1'' \urcorner \, m_1^{s_1 s_1'' \bullet t_1 t_1'', (\vec{b}_1 \setminus \mathsf{nlist}(s_1 s_1''.-1))} = \pi' \circ \ulcorner s_1 s_1'' m_1^{\vec{b}_1} \parallel t_1 t_1'' m_1^{\vec{c}_1} \urcorner
\end{aligned}$$

as required. ∎

Proposition 3.43 *If $\sigma : A \to B, \tau : B \to C$ are innocent strategies then so is $\sigma ; \tau$.*

Proof: $\sigma ; \tau$ is a strategy. Moreover, if $[s ; t] \in \sigma ; \tau$ for some $[s] \in \sigma$, $[t] \in \tau$, then, by proposition 3.36, $s ; t$ is innocent. Now let $[u_1 n_1^{\vec{c}_1}], [u_2] \in \sigma ; \tau$ be even- and odd-length respectively, with $[\ulcorner u_1 \urcorner] = [\ulcorner u_2 \urcorner]$. By definition and prefix closure, $u_2 = s_2 ; t_2$ and $u_1 n_1^{\vec{c}_1} = (s_1 s_1' ; t_1 t_1') n_1^{\vec{c}_1} = (s_1 ; t_1) n_1^{\vec{c}_1}$ with $[s_1 s_1'], [s_2] \in \sigma$, $[t_1 t_1'], [t_2] \in \tau$ and $s_i \parallel t_i$ O-ending in AC. Hence, by lemma 3.40, $[\ulcorner s_1 \parallel t_1 \urcorner] = [\ulcorner s_2 \parallel t_2 \urcorner]$.
By previous lemma then, there exist s_2', t_2' in B such that $[s_2 s_2'] \in \sigma$, $[t_2 t_2'] \in \tau$ and $[\ulcorner s_1' s_1 \parallel t_1 t_1'' \urcorner] = [\ulcorner s_2' s_2 \parallel t_2 t_2'' \urcorner]$.
Suppose now n_1 is in A and $u_1 n_1^{\vec{c}_1} = s_1' s_1 n_1^{\vec{c}_1} \parallel t_1 t_1''$. By previous lemma, there exists $n_2^{\vec{c}_2'}$ such that $[s_2 s_2' n_2^{\vec{c}_2'}] \in \sigma$ and $[\ulcorner s_1 s_1' n_1^{\vec{c}_1} \parallel t_1 t_1' \urcorner] = [\ulcorner s_2 s_2' n_2^{\vec{c}_2'} \parallel t_2 t_2' \urcorner] = [\ulcorner (s_2 s_2' \parallel t_2 t_2') n_2^{\vec{c}_2} \urcorner]$, $\therefore$ $[\ulcorner (s_1 ; t_1) n_1^{\vec{c}_1} \urcorner] = [\ulcorner (s_2 ; t_2) n_2^{\vec{c}_2} \urcorner]$ and $[(s_2 ; t_2) n_2^{\vec{c}_2}] \in \sigma ; \tau$, as required. Similarly if n_1 is in C. ∎

Hence, and since identities are innocent, we obtain a subcategory of innocent strategies.

Definition 3.44 $\mathcal{V}$ is the lluf subcategory of $\mathcal{G}$ of innocent (nominal) strategies. ▲

Henceforth, we will assume plays and strategies as being innocent unless stated otherwise. It is easy to see that $\mathcal{V}$ inherits Cpo-enrichment from $\mathcal{G}$ by the subset ordering $\sqsubseteq$ of definition 3.26.

Proposition 3.45 *$\mathcal{V}$ is Cpo-enriched by $\sqsubseteq$.* ∎

3.2.2 Viewfunctions

We argued previously that innocent strategies are specified by their behaviour on P-views. We formalise this argument by representing innocent strategies by *viewfunctions*.

Definition 3.46 Let A be a prearena. A ***viewfunction*** f on A is a set of equivalence classes of innocent plays of A which are even-length P-views, satisfying:

- **Even-prefix closure:** If $[s] \in f$ and t is an even-length prefix of s then $[t] \in f$.
- **Single-valuedness:** If $[s_1 x_1], [s_2 x_2] \in f$ and $[s_1] = [s_2]$ then $[s_1 x_1] = [s_2 x_2]$.

Let σ be an innocent strategy. Its viewfunction is given by:

$$\mathtt{viewf}(\sigma) \triangleq \{[s] \in \sigma \mid |s| \text{ even} \wedge \ulcorner s \urcorner = s\}\,.$$

Conversely, if f is a viewfunction on a prearena A then its strategy is given by:

$$\mathtt{strat}(f) \triangleq \bigcup\nolimits_n \mathtt{strat}_n(f)\,,$$

where $\mathtt{strat}_0(f) \triangleq \{[\epsilon]\}$ and

$$\begin{aligned}\mathtt{strat}_{2n+1}(f) &\triangleq \{[sx] \mid sx \in P_A^{\mathtt{i}} \wedge [s] \in \mathtt{strat}_{2n}(f)\}\,,\\ \mathtt{strat}_{2n+2}(f) &\triangleq \{[sy] \mid sy \in P_A^{\mathtt{i}} \wedge [s] \in \mathtt{strat}_{2n+1}(f) \wedge [\ulcorner sy \urcorner] \in f\}\,.\end{aligned}$$

▲

Note in the above definition that, for any even-length s, $[s] \in \mathtt{strat}(f)$ implies $[\ulcorner s \urcorner] \in f$. We first show that the conversion functions above are well-defined, and then that they are inverses.

Lemma 3.47 *For any innocent strategy σ, $\mathtt{viewf}(\sigma)$ is a viewfunction.*

Proof: Since σ is innocent, elements in $\mathtt{viewf}(\sigma)$ are by definition equivalence classes of innocent plays that are even-length P-views. Moreover, if $[s] \in \mathtt{viewf}(\sigma)$ and $t \leq s$ is even-length then $[t] \in \sigma$. But s being a P-view implies t is a P-view, so $[t] \in \mathtt{viewf}(\sigma)$. Finally, let $[s_1x_1], [s_2x_2] \in \mathtt{viewf}(\sigma)$ and $[s_1] = [s_2]$. By determinacy of σ, $[s_1x_1] = [s_2x_2]$, as required. ■

Lemma 3.48 *For any viewfunction f, $\mathtt{strat}(f)$ is an innocent strategy.*

Proof: By definition, $\mathtt{strat}(f)$ contains innocent plays and satisfies prefix-closure. For contingency completeness note that if $[s] \in \mathtt{strat}_{2n}(f)$ and $sx \in P_A$ then necessarily $sx \in P_A^{\mathtt{i}}$, as x an O-move, so $[sx] \in \mathtt{strat}_{2n+1}(f)$.
Now, for determinacy suppose that even-length $[sxy], [s'x'y'] \in \mathtt{strat}(f)$ and $[sx] = [s'x']$. Then, $[\ulcorner sx \urcorner y], [\ulcorner s'x' \urcorner y'] \in f$ and $[\ulcorner sx \urcorner] = [\ulcorner s'x' \urcorner]$, so by single-valuedness of f, $[\ulcorner sx \urcorner y] = [\ulcorner s'x' \urcorner y']$. But now the three previous equalities suffice for applying the strong support lemma and obtain $[sxy] = [s'x'y']$.
For innocence, let $[s_1x_1y_1], [s_2x_2] \in \mathtt{strat}(f)$ with $[\ulcorner s_1x_1 \urcorner] = [\ulcorner s_2x_2 \urcorner]$ being odd-length, say $\pi \circ \ulcorner s_1x_1 \urcorner = \ulcorner s_2x_2 \urcorner$. Let $y_2 \triangleq \pi \circ \pi' \circ y_1$, where π' simply swaps names introduced by y_1 in $s_1x_1y_1$ with completely fresh ones, i.e. fresh for s_1x_1, s_2x_2, y_1, π. Since $s_1x_1y_1$ is a legal sequence, so is $s_2x_2y_2$: visibility is obvious and well-bracketing follows from the fact that the (possible) pending question of s_2x_2 is the same as that of $\ulcorner s_2x_2 \urcorner$ (see e.g. [McC00, lemma 2.1]). Moreover, $s_2x_2y_2$ obviously satisfies NC3, and $\ulcorner s_2x_2y_2 \urcorner = \pi \circ (\ulcorner s_1x_1 \urcorner \pi' \circ y_1) = \pi \circ \pi' \circ \ulcorner s_1x_1y_1 \urcorner$ implies NC2′. For NC1, if $a \in \mathtt{S}(y_2)$ and $a \# \mathsf{nlist}(x_2)$ then $\pi^{-1}(a) \in \mathtt{S}(\pi' \circ y_1)$ and $\pi^{-1}(a) \# \mathsf{nlist}(x_1)$. But then $\pi^{-1}(a)$ is one of the completely fresh names, so $\pi^{-1}(a) = a$ and $a \# s_2x_2$. Hence, $s_2x_2y_2$ an innocent play. Since $[s_2x_2] \in \mathtt{strat}(f)$ and $[\ulcorner s_2x_2y_2 \urcorner] = [\ulcorner s_1x_1y_1 \urcorner] \in f$, $[s_2x_2y_2] \in \mathtt{strat}(f)$. ■

Proposition 3.49 *For any viewfunction f and innocent strategy σ,*

$$f = \mathtt{viewf}(\mathtt{strat}(f)) \wedge \sigma = \mathtt{strat}(\mathtt{viewf}(\sigma))\,.$$

Proof: For the first part, we show first $[s] \in f \implies [s] \in \mathtt{strat}(f)$. We do induction on even $|s|$. The base case is obvious. For the inductive step, if $[sxy] \in f$ then, by IH and prefix closure of f, we have that $[s] \in \mathtt{strat}(f)$, and thus $[sxy] \in \mathtt{strat}(f)$ as $[\ulcorner sxy \urcorner] = [sxy] \in f$. Clearly then $f \subseteq \mathtt{viewf}(\mathtt{strat}(f))$.
Conversely, we show that $[s] \in \mathtt{viewf}(\mathtt{strat}(f)) \implies [s] \in f$. Now, $[s] \in \mathtt{viewf}(\mathtt{strat}(f))$ implies that $[s] \in \mathtt{strat}(f)$ and $\ulcorner s \urcorner = s$, so $[s] = [\ulcorner s \urcorner] \in f$.
For the second part, we show that, for each n, $S_n \triangleq \mathtt{strat}_n(\mathtt{viewf}(\sigma)) \subseteq \sigma$, by induction

on n. The base case is obvious. Now, for odd $n+1$, if $[sx] \in S_{n+1}$ then $[s] \in S_n$, so $[s] \in \sigma$ by IH, and $[sx] \in \sigma$ by contingency completeness. For even $n+1$, if $[sy] \in S_{n+1}$ then $[s] \in S_n$ and $[\ulcorner sy \urcorner] \in \mathtt{viewf}(\sigma) \subseteq \sigma$, so by IH and lemma 3.38 we have $[sy] \in \sigma$.
Conversely, let $S \triangleq \mathtt{strat}(\mathtt{viewf}(\sigma))$. We show by induction on the length of plays in σ that $\sigma \subseteq S$. The base case is trivial. Now, if odd-length $[sx] \in \sigma$ then $[s] \in S$, because of prefix closure of σ and IH, and thus, clearly, $[sx] \in S$. If even-length $[sy] \in \sigma$ then, again, $[s] \in S$, and $[sy] \in S$ because $[\ulcorner sy \urcorner] \in \mathtt{viewf}(\sigma)$ by lemma 3.38. ∎

A direct consequence is the following.

Corollary 3.50 *For any viewfunctions f, g and innocent strategies σ, τ,*

1. $f \subseteq \mathtt{strat}(f)$,

2. $\sigma \subseteq \tau \iff \mathtt{viewf}(\sigma) \subseteq \mathtt{viewf}(\tau) \quad$ *and* $\quad f \subseteq g \iff \mathtt{strat}(f) \subseteq \mathtt{strat}(g)$,

3. $\mathtt{viewf}(\sigma) \subseteq \tau \wedge \mathtt{viewf}(\tau) \subseteq \sigma \implies \sigma = \tau$.

Moreover, $\sqsubseteq$ yields a cpo on viewfunctions, and `viewf`, `strat` *are continuous wrt $\sqsubseteq$.*

Proof: For 1 we have: $f = \mathtt{viewf}(\mathtt{strat}(f)) \subseteq \mathtt{strat}(f)$.
For 2, because of previous proposition, it suffices to show only the " $\implies$ " directions. For the first conjunct we simply use the definition of `viewf`. For the latter we can show that, for each $n \in \omega$, $f \subseteq g \implies \mathtt{strat}_n(f) \subseteq \mathtt{strat}_n(g)$, by straightforward induction on n.
For 3 note that $[s] \in \mathtt{viewf}(\sigma) \cap \tau \implies [s] \in \mathtt{viewf}(\tau)$, so $\mathtt{viewf}(\sigma) \subseteq \tau \wedge \mathtt{viewf}(\tau) \subseteq \sigma$ implies that $\mathtt{viewf}(\sigma) = \mathtt{viewf}(\tau)$, and hence $\sigma = \tau$.
The fact that viewfunctions form a cpo is straightforward. For continuity of `strat` and `viewf`, because of 2, we need only show that unions of ω-chains are preserved. So let $(\sigma_i)_{i\in\omega}$ be an increasing sequence of strategies and let $[s] \in \mathtt{viewf}(\bigcup_{i\in\omega} \sigma_i)$. Then $[s] \in \sigma_i$, some i, and hence $[s] \in \bigcup_{i\in\omega} \mathtt{viewf}(\sigma_i)$. Therefore, $\mathtt{viewf}(\bigcup_{i\in\omega} \sigma_i) \subseteq \bigcup_{i\in\omega} \mathtt{viewf}(\sigma_i)$ and, by monotonicity, $\mathtt{viewf}(\bigcup_{i\in\omega} \sigma_i) = \bigcup_{i\in\omega} \mathtt{viewf}(\sigma_i)$.
On the other hand, `viewf` being continuous and `strat` being its inverse imply that `strat` is also continuous. ∎

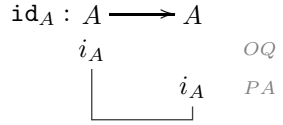
3.2.3 Diagrams of viewfunctions

We saw in the previous section that innocent strategies can be represented by their viewfunctions. A viewfunction is a set of (equivalence classes of) plays, so the formal way to express such a construction is explicitly as a set. For example, we have that

$$\mathtt{viewf}(\mathtt{id}_A) = \{\, [s\,m\,m] \mid [s] \in \mathtt{viewf}(\mathtt{id}_A) \wedge sm \in P_{A\to A} \wedge (m \in I_A \vee s.{-}1 \vdash_A m) \,\},$$

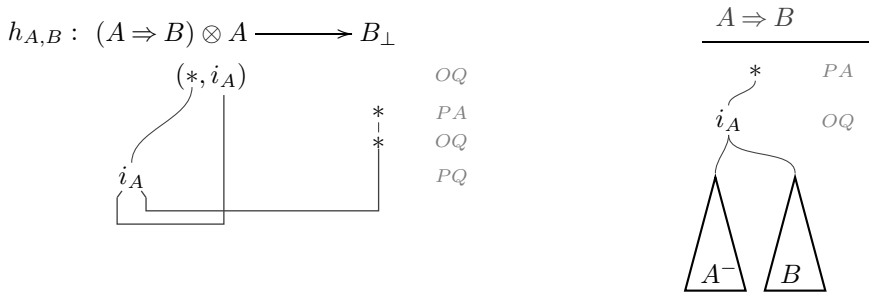
where the last m is justified by $s.{-}2$ and the penultimate one by $s.{-}1$ (in case $m \notin I_A$). The above behaviour is called ***copycat*** (v. [AJ94]) and is perhaps the most focal notion in game semantics.

A more convenient way to express viewfunctions is by means of diagrams. For example, for $\mathtt{id}_A$ we can have the following depiction.



The polygonal line in the above depiction stands for a ***copycat link***, meaning that the strategy copycats between the two i_A's. A more advanced example of this notation is the strategy

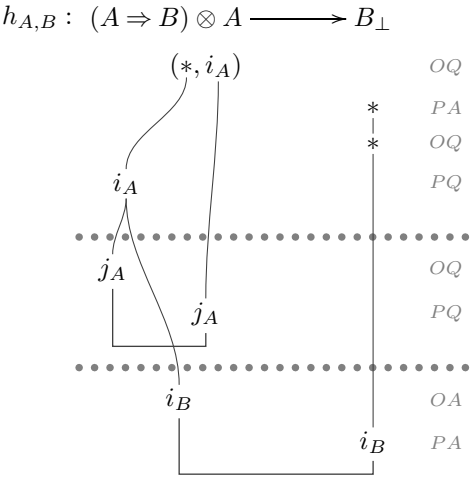
on the left below.



Note first that curved lines (and also the line connecting the two $*$'s) stand for justification pointers. Moreover, recall that the arena $A \Rightarrow B$ has the form given on the right above, so the leftmost i_A (l-i_A) in the diagram of $h_{A,B}$ has two child components, A^- and B. Then, the copycat links starting from the l-i_A have the following meaning. $h_{A,B}$ copycats between the A^--component of l-i_A and the other i_A, and copycats also between the B-component of l-i_A and the lower $*$. That is (modulo prefix-closure),

$$h_{A,B} \triangleq \mathtt{strat}\{\, [(*, i_A)\ *\ *i_A\, s] \mid [i_A\, i_A\, s] \in \mathtt{viewf}(\mathtt{id}_A) \vee [s] \in \mathtt{viewf}(\mathtt{id}_B)\,\}\,.$$

Another way to depict $h_{A,B}$ is by cases, with regard to Opponent's next move after l-i_A, as follows.



Finally, we will sometimes label copycat links by strategies (e.g. in the proof of proposition 3.61). Labelling a copycat link by a strategy σ means that the specified strategy plays like σ between the linked moves, instead of doing copycat. In this sense, ordinary copycat links can be seen as links labelled with identities.

3.3 Totality

A basic problem with the category $\mathcal{V}$, for our denotational purposes, is its lack of products. In order to obtain products we restrict ourselves to *total strategies*.

3.3.1 The subcategory $\mathcal{V}_{\mathsf{t}}$

We introduce the notion of total strategies, specifying those strategies which immediately answer initial questions without introducing fresh names. We extend this type of reasoning to level-1 moves, yielding several subclasses of innocent strategies. Note that an arena A is *pointed* if I_A is singleton.

Definition 3.51 ($\mathcal{V}_{\mathtt{t}}$, $\mathcal{V}_{\mathtt{tt}}$, $\mathcal{V}_{\mathtt{t}*}$, $\mathcal{V}_{\mathtt{tt}*}$) An innocent strategy $\sigma : A \longrightarrow B$ is ***total*** if for any $[i_A] \in \sigma$ there exists $[i_A\, i_B] \in \sigma$. A total strategy $\sigma : A \longrightarrow B$ is:

- ***l4*** if whenever $[s] \in \sigma$ and $\underline{s.{-}1} \in J_A$ then $|\ulcorner s \urcorner| = 4$,
- ***t4*** if for any $[i_A\, i_B\, j_B] \in \sigma$ there exists $[i_A\, i_B\, j_B\, j_A^{\vec{b}}] \in \sigma$,
- ***tl4*** if it is both t4 and l4,
- ***ttotal*** if it is tl4 and for any $[i_A\, i_B\, j_B] \in \sigma$ there exists $[i_A\, i_B\, j_B\, j_A] \in \sigma$.

A total strategy $\tau : C \otimes A \longrightarrow B$ is:

- ***l4**** if whenever $[s] \in \tau$ and $\underline{s.{-}1} \in J_A$ then $|\ulcorner s \urcorner| = 4$,
- ***t4**** if for any $[(i_C, i_A) i_B\, j_B] \in \tau$ there exists $[(i_C, i_A) i_B\, j_B\, j_A^{\vec{b}}] \in \tau$,
- ***tl4**** if it is both t4* and l4*,
- ***ttotal**** if it is tl4* and for any $[(i_C, i_A)\, i_B\, j_B] \in \sigma$ there exists $[(i_C, i_A)\, i_B\, j_B\, j_A] \in \sigma$.

We let $\mathcal{V}_{\mathtt{t}}$ be the lluf subcategory of $\mathcal{V}$ of total strategies, and $\mathcal{V}_{\mathtt{tt}}$ its lluf subcategory of ttotal strategies. $\mathcal{V}_{\mathtt{t}*}$ and $\mathcal{V}_{\mathtt{tt}*}$ are the full subcategories of $\mathcal{V}_{\mathtt{t}}$ and $\mathcal{V}_{\mathtt{tt}}$ respectively containing pointed arenas. ▲

These subclasses of strategies will be demystified in the sequel. For now, note that l4 stands for "linear in the 4th move", and t4 for "total in the 4th move".

We now proceed to examine properties of $\mathcal{V}_{\mathtt{t}}$. Eventually, we will see that it contains finite products and distributive coproducts, that it contains *some* exponentials, and that lifting promotes to a functor. Note that in the definitions to follow we will usually define strategies by means of their viewfunctions modulo even-prefix closure.

3.3.2 Lifting and product

We first upgrade the lifting and tensor arena-constructions of definition 3.3 to functors. Eventually, tensor will give us products. In the following definition recall $\mathcal{L}$ from notation 3.7 and note that we write $\mathcal{L}(m) \# m'$ for $\mathcal{L}(m) \cap \mathtt{S}(m') = \varnothing$.

Definition 3.52 Let $f : A \to A'$, $g : B \to B'$ be arrows in $\mathcal{V}_{\mathtt{t}}$. Define the strategies $f \otimes g : A \otimes B \to A' \otimes B'$ and $f_\perp : A_\perp \to A'_\perp$ as follows.

$$f \otimes g \triangleq \mathtt{strat}\{\, [(i_A, i_B)\,(i_{A'}, i_{B'})\, s] \mid ([i_A\, i_{A'}\, s] \in \mathtt{viewf}(f) \wedge [i_B\, i_{B'}] \in g \wedge \mathcal{L}(i_A\, i_{A'}\, s) \# i_B) \vee ([i_B\, i_{B'}\, s] \in \mathtt{viewf}(g) \wedge [i_A\, i_{A'}] \in f \wedge \mathcal{L}(i_B\, i_{B'}\, s) \# i_A) \,\}$$

$$f_\perp \triangleq \mathtt{strat}\{\, [* *' *' * s] \mid [s] \in \mathtt{viewf}(f) \,\}.$$

▲

Note that $f_\perp$ is always ttotal. Let us we give an informal description of the above constructions:

- $f_\perp : A_\perp \to A'_\perp$ initially plays a sequence of asterisks $[* *' *'*]$ and then continues playing like f.
- $f \otimes g : A \otimes B \to A' \otimes B'$ answers initial moves $[(i_A, i_B)]$ with f's answer to $[i_A]$ and g's answer to $[i_B]$. Then, according to whether Opponent plays in $J_{A'}$ or in $J_{B'}$, Player plays like f or like g respectively.

We proceed to show that the above yield functors. The following lemma is straightforward but arises quite often when we are dealing with definitions like that of $f \otimes g$.

Lemma 3.53 *Let $s_1 s$ and $s_2 s$ be legal sequences of moves-with-names on a prearena A which both satisfy NC3, and s_1, s_2 be P-ending plays such that $\mathtt{S}(s_2) \subseteq \mathtt{S}(s_1)$, $\mathcal{L}(s_2 s) \# s_1$ and s_1 appears in $\ulcorner s_1 s' \urcorner$, for any $s' \leq s$. Then, $s_1 s$ is a play iff $s_2 s$ is a play.*

Proof: The claim is trivial if $|s| = 0$, so assume $|s| > 0$. For the "if"-part, we need to show NC1-2′. So let $s' \leq s$ be P-ending sequence. If $a \in \mathsf{nlist}(s'.{-}1)$ and $a \# \mathsf{nlist}(s'.{-}2)$ then $a \# s_2 s'^{-}$ and $a \in \mathcal{L}(s_2 s)$, so $a \# s_1, s_2 s'^{-}$, hence $a \# s_1 s'^{-}$. If $a \in \mathtt{S}(s'.{-}1)$ and $a \# \ulcorner s_1 s'^{-} \urcorner$ then, since s_1 appears in $\ulcorner s_1 s' \urcorner$, $a \# \ulcorner s_2 s'^{-} \urcorner$ and hence $a \in \mathsf{nlist}(s'.{-}1)$.
For the "only if"-part, we again need to show NC1-2′. So let $s' \leq s$ be P-ending sequence. If $a \in \mathsf{nlist}(s'.{-}1)$ and $a \# \mathsf{nlist}(s'.{-}2)$ then $a \# s_1 s'^{-}$, so $a \# s_2 s'^{-}$. If $a \in \mathtt{S}(s'.{-}1)$ and $a \# \ulcorner s_2 s'^{-} \urcorner$ then $a \in \mathcal{L}(s_2 s)$ so $a \# \ulcorner s_2 s'^{-} \urcorner, s_1$, hence $a \# \ulcorner s_1 s'^{-} \urcorner$ and $a \in \mathsf{nlist}(s'.{-}1)$. ∎

Lemma 3.54 *For f, f', g and g' as above, the following are arrows in $\mathcal{V}_{\mathtt{t}}$.*

$$f \otimes g : A \otimes B \to A' \otimes B' \,, \quad f_\perp : A_\perp \to B_\perp \,.$$

Proof: For $f \otimes g$, it suffices to show that ϕ is a viewfunction:

$$\phi \triangleq \{\, [(i_A, i_B)\,(i_{A'}, i_{B'})\,s] \mid ([i_A\, i_{A'}\, s] \in \mathtt{viewf}(f) \wedge [i_B\, i_{B'}] \in g \wedge \mathcal{L}(i_A\, i_{A'}\, s) \# i_B) \\ \vee\, ([i_B\, i_{B'}\, s] \in \mathtt{viewf}(g) \wedge [i_A\, i_{A'}] \in f \wedge \mathcal{L}(i_B\, i_{B'}\, s) \# i_A) \,\} \,.$$

Elements in ϕ are plays: let $[t] \in \phi$ and suppose wlog that $t = (i_A, i_B)\,(i_{A'}, i_{B'})\,s$ with $[i_A\, i_{A'}\, s] \in \mathtt{viewf}(f)$, $[i_B\, i_{B'}] \in g$ and $\mathcal{L}(i_A\, i_{A'}\, s) \# i_B$. Then t is legal because $i_A\, i_{A'}\, s$ is. Moreover, NC3 trivially holds and therefore, by previous lemma, t is a play.
For even-prefix closure, let $t = (i_A, i_B)\,(i_{A'}, i_{B'})\,sxy$, $[t] \in \phi$, and suppose, wlog, that $[i_A\, i_{A'}\, sxy] \in \mathtt{viewf}(f)$, $[i_B\, i_{B'}] \in g$ and $\mathcal{L}(i_A\, i_{A'}\, s) \# i_B$. Then, $[i_A\, i_{A'}\, s] \in \mathtt{viewf}(f)$ and thus $[t^{--}] \in \phi$.
For single-valuedness, let $[t_1], [t_2] \in \phi$ and $[t_1^-] = [t_2^-]$, say modulo π_0 (i.e. $t_1^- = \pi_0 \circ t_2^-$). If $t_\kappa = (i_{A(\kappa)}, i_{B(\kappa)})\,(i_{A'(\kappa)}, i_{B'(\kappa)})$, for $\kappa = 1, 2$, then, by single-valuedness of $\mathtt{viewf}(f)$, $[i_{A(1)}] = [i_{A(2)}]$ implies $[i_{A(1)}\, i_{A'(1)}] = [i_{A(2)}\, i_{A'(2)}]$, and similarly $[i_{B(1)}\, i_{B'(1)}] = [i_{B(2)}\, i_{B'(2)}]$. Let's say the former equality is modulo π_1 and the latter is modulo π_2. We now have $i_{A(1)} = \pi_0 \circ i_{A(2)} = \pi_0 \circ \pi_1^{-1} \circ i_{A(1)}$, $\therefore \quad i_{A'(1)} = \pi_0 \circ \pi_1^{-1} \circ i_{A'(1)}$ because $\mathtt{S}(i_{A'(1)}) \subseteq \mathtt{S}(i_{A(1)})$, $\therefore \quad i_{A(1)}\, i_{A'(1)} = \pi_0 \circ i_{A(2)}\, i_{A'(2)}$, and similarly for the B-counterpart. Hence, $t_1 = \pi_0 \circ t_2$ as required. If $t_\kappa = [(i_{A(\kappa)}, i_{B(\kappa)})\,(i'_{A'(\kappa)}, i'_{B'(\kappa)})\,s_\kappa x_\kappa]$, for $\kappa = 1, 2$, then suppose wlog that $[i_{B(\kappa)}\, i_{B'(\kappa)}] \in g$ and $[i_{A(\kappa)}\, i_{A'(\kappa)}\, s_\kappa x_\kappa] \in \mathtt{viewf}(f)$, and $\mathcal{L}(i_{A(\kappa)}\, i_{A'(\kappa)}\, s_\kappa x_\kappa) \# i_{B(\kappa)}$. Then $[i_{A(1)}\, i_{A'(1)}\, s_1] = [i_{A(2)}\, i_{A'(2)}\, s_2]$, so $[i_{A(1)}\, i_{A'(1)}\, s_1 x_1] = [i_{A(2)}\, i_{A'(2)}\, s_2 x_2]$. Now using the strong support lemma we get $[t_1] = [t_2]$. This completes the proof of "ϕ is a viewfunction". The case of $f_\perp$ is similar (and simpler). ∎

Lemma 3.55 *Let A, A', B, B' be arenas and $i_{A,A',B,B'} \in I_{A,A',B,B'}$ respectively, and let s_A and s_B be justified sequences of moves-with-names from $\bar{I}_A \cup \bar{I}_{A'}$ and $\bar{I}_B \cup \bar{I}_{B'}$ respectively. If $f : A \longrightarrow A'$ and $g : B \longrightarrow B'$ in $\mathcal{V}_{\mathtt{t}}$, then:*

1. $[(i_A, i_B)\,(i_{A'}, i_{B'})\,s_A] \in f \otimes g \iff [i_A\, i_{A'}\, s_A] \in f \wedge [i_B\, i_{B'}] \in g \wedge \mathcal{L}(i_A\, i_{A'}\, s_A) \# i_B$,
2. $[(i_A, i_B)\,(i_{A'}, i_{B'})\,s_B] \in f \otimes g \iff [i_A\, i_{A'}] \in f \wedge [i_B\, i_{B'}\, s_B] \in g \wedge \mathcal{L}(i_B\, i_{B'}\, s_B) \# i_A$,
3. $[* * * * s_A] \in f_\perp \iff [s_A] \in f$.

Proof: For 1, let $s = (i_A, i_B)\,(i_{A'}, i_{B'})\,s_A$, $s_f = i_A\, i_{A'}\, s_A$ and $s_g = i_B\, i_{B'}$. For "$\Longrightarrow$", we do induction on $|s_A| \geq 0$; the base case is by definition. If $|s_A|$ is odd then, because of the IH and contingency completeness of f, it suffices to show that s_f is a play. But this follows easily from s and s_f^- being plays. If $|s_A| > 0$ is even then, by IH and lemma 3.38, it suffices to show that s_f a play and that $[\ulcorner s_f \urcorner] \in f$. Note that $[\ulcorner s \urcorner] \in f \otimes g$ implies $\ulcorner s_f \urcorner \in f$ and $\mathcal{L}(\ulcorner s_f \urcorner) \# i_B$, and therefore, using also the IH, $\mathcal{L}(s_f) \# i_B$. Moreover, s_f is legal and satisfies NC3 so we can use lemma 3.53.
For "$\Longleftarrow$", we do again induction on $|s_A| \geq 0$, and the base case is by definition. The case

of $|s_A|$ odd is shown using the IH and contingency completeness. If $|s_A| > 0$ is even then, by IH and lemma 3.38, it suffices to show that $\ulcorner s \urcorner \in f \otimes g$ and s a play. The former is straightforward and the latter follows from lemma 3.53.
This completes the proof of 1. 2 and 3 are proved similarly. ∎

Proposition 3.56 *The following are functors.*

$$\_ \otimes \_ : \mathcal{V}_{\mathtt{t}} \times \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{t}} \qquad (\_)_\perp : \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{tt}*}$$

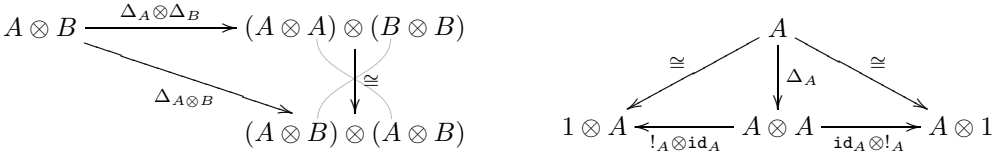
Proof: The above constructions have been shown to be well-defined on objects and arrows, so we need only show functoriality. For the tensor $\otimes$, it is not difficult to see that $\mathtt{id}_A \otimes \mathtt{id}_B = \mathtt{id}_{A\otimes B}$. We also need to show that, for any $A \xrightarrow{f} A' \xrightarrow{f'} A''$ and $B \xrightarrow{g} B' \xrightarrow{g'} B''$, $(f; f') \otimes (g; g') = (f \otimes g); (f' \otimes g')$.
Let $u = [(i_A, i_B)\,(i_{A''}, i_{B''})\,s] \in \mathtt{viewf}((f; f') \otimes (g; g'))$ and assume wlog that $[i_A\, i_{A''}\, s] \in f; f'$, $[i_B\, i_{B''}] \in g; g'$ and $\mathcal{L}(i_A\, i_{A''}\, s) \# i_B$. Let then $i_A\, i_{A''}\, s = i_A\, i_{A'}\, s'\,; i_{A'}\, i_{A''}\, s''$ and $i_B\, i_{B''} = i_B\, i_{B'}\,; i_{B'}\, i_{B''}$, with not both s', s'' ending in A' and $[i_A\, i_{A'}\, s'] \in f$, $[i_{A'}\, i_{A''}\, s''] \in f'$, $[i_B\, i_{B'}] \in g$ and $[i_{B'}\, i_{B''}] \in g'$. Note that $\mathcal{L}(i_A\, i_{A''}\, s) \# i_B$ implies $\mathcal{L}(i_A\, i_{A'}\, s') \# i_B$ and $\mathcal{L}(i_{A'}\, i_{A''}\, s'') \# i_B$, and from the latter $\mathcal{L}(i_{A'}\, i_{A''}\, s'') \# i_{B'}$. By the previous lemma we have that $[(i_A, i_B)\,(i_{A'}, i_{B'})\,s'] \in f \otimes g$ and $[(i_{A'}, i_{B'})\,(i_{A''}, i_{B''})\,s''] \in f' \otimes g'$, and hence $u = [(i_A, i_B)\,(i_{A'}, i_{B'})\,s'\,; (i_{A'}, i_{B'})\,(i_{A''}, i_{B''})\,s''] \in f \otimes g\,; f' \otimes g'$.
Conversely, if $u \in \mathtt{viewf}(f \otimes g\,; f' \otimes g')$ then $u = [(i_A, i_B)\,(i_{A''}, i_{B''})\,s]$, where s is (exclusively) in $A \to A''$ or in $B \to B''$, since only O can switch components and u is a P-view. Suppose wlog that the $A \to A''$ case holds. We then have that $u = [(i_A, i_B)\,(i_{A'}, i_{B'})\,s'\,; (i_{A'}, i_{B'})\,(i_{A''}, i_{B''})\,s'']$, with not both s', s'' ending in A'. Then, by previous lemma, $[i_A\, i_{A'}\, s'] \in f$, $[i_B\, i_{B'}] \in g$ and $\mathcal{L}(i_A\, i_{A'}\, s'') \# i_B$, and $[i_{A'}\, i_{A''}\, s''] \in f'$, $[i_{B'}\, i_{B''}] \in g'$ and $\mathcal{L}(i_{A'}\, i_{A''}\, s') \# i_{B'}$. But now $[i_A\, i_{A'}\, s'\,; i_{A'}\, i_{A''}\, s''] = [i_A\, i_{A''}\, s] \in f; f'$ and $[i_B\, i_{B''}] \in g; g'$. Now, $(i_A, i_B)\,(i_{A'}, i_{B'})\,s' \asymp (i_{A'}, i_{B'})\,(i_{A''}, i_{B''})\,s''$ implies that $\mathcal{L}((i_{A'}, i_{B'})\,(i_{A''}, i_{B''})\,s'') \# i_B$, i.e. $\mathcal{L}(i_{A'}\, i_{A''}\, s'') \# i_B$. Hence, $\mathcal{L}(i_A\, i_{A''}\, s) = (\mathcal{L}(i_A\, i_{A'}\, s') \cup \mathcal{L}(i_{A'}\, i_{A''}\, s'')) \# i_B$, and thus, by previous lemma, $u \in (f; f') \otimes (g; g')$.
Finally, the case of lifting $(\_)_\perp$ is much simpler, and is proved along the same lines. ∎

Thus, we have shown in full formality that tensor and lifting are functors. Note that in the sequel we will generally avoid to give proofs of simple facts about strategies at this level of detail.

We now show that $\otimes$ yields products, and hence that $\mathcal{V}_{\mathtt{t}}$ is cartesian.

Proposition 3.57 $\mathcal{V}_{\mathtt{t}}$ *is cartesian:* 1 *is a terminal object and* $\otimes$ *is a product constructor.*

Proof: Terminality of 1 is clear. Moreover, it is straightforward to see that $\otimes$ yields a symmetric monoidal structure on $\mathcal{V}_{\mathtt{t}}$, with its unit being 1 and its associativity, left-unit, right-unit and symmetry isomorphisms being the canonical ones. Hence, it suffices to show that there exists a natural coherent diagonal, that is, a natural transformation $\Delta : Id_{\mathcal{V}_{\mathtt{t}}} \longrightarrow \otimes \circ \langle Id_{\mathcal{V}_{\mathtt{t}}}, Id_{\mathcal{V}_{\mathtt{t}}} \rangle$ (where $\langle Id_{\mathcal{V}_{\mathtt{t}}}, Id_{\mathcal{V}_{\mathtt{t}}} \rangle$ is the diagonal functor on $\mathcal{V}_{\mathtt{t}}$) such that the following diagrams commute for any A, B in $\mathcal{V}_{\mathtt{t}}$.



But it is easy to see there is a canonical choice for Δ,

$$\Delta_A : A \longrightarrow A \otimes A \triangleq \mathtt{strat}\{\, [i_A\ (i_A, i_A)\, s] \mid [i_A\ i_A\ s] \in \mathtt{viewf}(\mathtt{id}_A)\,\},$$

which makes the above diagrams commute. Naturality follows from the single-threaded nature of strategies (v. [Har99]). ∎

Now that we have defined the diagonal Δ we can show the following (cf. [AJM00]). Recall tl4* strategies from definition 3.51.

Lemma 3.58 (Separation of Head Occurrence) *Let A be a pointed arena and $f : A \longrightarrow B$ be a t4 strategy. Then there exists a tl4* strategy $\tilde{f} : A \otimes A \longrightarrow B$ such that $f = \Delta ; \tilde{f}$.*

Proof: Let us tag the two copies of A in $A \otimes A$ as $A_{(1)}$ and $A_{(2)}$, and take

$$\tilde{f} \triangleq \mathtt{strat}\{[(i_A, i_A) i_B\, j_B\, j^{\vec{b}}_{A_{(2)}} s] \mid [i_A\, i_B\, j_B\, j^{\vec{b}}_{A_{(2)}} s] \,\tilde{\in}\, \mathtt{viewf}(f) \wedge \forall i.\, \underline{s.i} \notin J_{A_{(2)}}\},$$

where $\tilde{\in}$ is the composition of de-indexing from $M_{A_{(1)}}$ and $M_{A_{(2)}}$ to M_A with $\in$. Intuitively, $\tilde{f}$ plays the first J_A-move of f in $A_{(2)}$, and then mimics f until the next J_A-move of f, which is played in $A_{(1)}$. All subsequent J_A-moves are also played in $A_{(1)}$. Clearly, $\tilde{f}$ is tl4* and $f = \Delta ; \tilde{f}$. ∎

Products in $\mathcal{V}_{\mathtt{t}}$ are given concretely by triples $A \xleftarrow{\pi_1} A \otimes B \xrightarrow{\pi_2} B$, where

$$\pi_1 = \mathtt{strat}\{\, [(i_A, i_B)\, i_A\; s] \mid [i_A\; i_A\; s] \in \mathtt{viewf}(\mathtt{id}_A)\,\},$$

and π_2 similarly, while for each $A \xleftarrow{f} C \xrightarrow{g} B$ we have

$$\langle f, g\rangle : C \longrightarrow A \otimes B = \mathtt{strat}\{\, [i_C\; (i_A, i_B)\, s] \mid ([i_C\; i_A\; s] \in \mathtt{viewf}(f) \wedge [i_C\; i_B] \in \mathtt{viewf}(g)) \\ \vee ([i_C\; i_A] \in \mathtt{viewf}(f) \wedge [i_C\; i_B\; s] \in \mathtt{viewf}(g))\,\}.$$

Finally, we need to generalise the tensor product to a version applicable to countably many arguments. In arenas, the construction comprises of gluing countably many arenas together at their initial moves. The problem that arises then is that the product of infinitely many (initial) moves need not have finite support, breaking the arena specifications. Nevertheless, in case we are interested only in pointed arenas, this is easily bypassed: a pointed arena has a unique initial move, which is therefore equivariant, and the product of equivariant moves is of course also equivariant.

Proposition and Definition 3.59 (Big tensor) *For pointed arenas $\{A_i\}_{i\in\omega}$ define $\bigotimes_i A_i$ by:*

$$\begin{aligned} M_{\bigotimes_i A_i} &\triangleq \{*\} + \biguplus_i \bar{I}_{A_i} && (\bigotimes_i A_i)\\ I_{\bigotimes_i A_i} &\triangleq \{*\} \\ \lambda_{\bigotimes_i A_i} &\triangleq [(* \mapsto PA), [\lambda_{A_i} \upharpoonright \bar{I}_{A_i}{}^{i\in\omega}]] \\ \vdash_{\bigotimes_i A_i} &\triangleq \{(*, j_{A_i}) \mid i \in \omega\} \cup \bigcup_i (\vdash_{A_i} \upharpoonright \bar{I}_{A_i}{}^2)\,. \end{aligned}$$

For $\{f_i : A_i \to B_i\}_{i\in\omega}$ with A_i's and B_i's pointed define:

$$\bigotimes_i f_i \triangleq \mathtt{strat}\{[* * s] \mid \exists k.\, [i_{A_k}\, i_{B_k}\, s] \in \mathtt{viewf}(f_k)\}.$$

Then $\bigotimes_{\_} : \prod \mathcal{V}_{\mathtt{t}*} \longrightarrow \mathcal{V}_{\mathtt{t}*}$ is a functor. ∎

The proof is similar to that of the binary tensor. Note that we could proceed and show that the aforedefined tensor yields general products of pointed objects, but this will not be of use here and is therefore left to the reader as an exercise.

3.3.3 Partial exponentials

We have seen that the tensor constructor equips $\mathcal{V}_{\mathtt{t}}$ with products. We now show that the arrow constructor yields appropriate partial exponentials, which will be sufficient for our denotational tasks.

Let us introduce the following transformations on strategies.

Definition 3.60 For all arenas A, B, C with C pointed, define a bijection

$$\Lambda^B_{A,C} : \mathcal{V}_{\mathtt{t}}(A \otimes B, C) \xrightarrow{\cong} \mathcal{V}_{\mathtt{t}}(A, B \multimap C)$$

by taking, for each $h : A \otimes B \longrightarrow C$ and $g : A \longrightarrow B \multimap C$,[5]

$$\Lambda^B_{A,C}(h) : A \longrightarrow B \multimap C \triangleq \mathtt{strat}\{\, [i_A\ i_C\ (i_B, j_C)\, s] \mid [(i_A, i_B)\ i_C\ j_C\ s] \in \mathtt{viewf}(h)\,\},$$

$$\Lambda^{B\ -1}_{A,C}(g) : A \otimes B \longrightarrow C \triangleq \mathtt{strat}\{\, [(i_A, i_B)\ i_C\ j_C\ s] \mid [i_A\ i_C\ (i_B, j_C)\, s] \in \mathtt{viewf}(g)\,\}.$$

Moreover, take $\mathtt{ev}_{A,B} : (A \multimap B) \otimes A \longrightarrow B \triangleq \Lambda^{A\ -1}_{A \multimap B, B}(\mathtt{id}_{A \multimap B})$.
Finally, for each $(f, g) : (A, B) \longrightarrow (A', B')$, take

$$f \multimap g : A' \multimap B \longrightarrow A \multimap B' \triangleq \Lambda^{A'}_{A \multimap B, A' \multimap B'}(\mathtt{id} \otimes f\,;\mathtt{ev}\,;g).$$

▲

It is not difficult to see that Λ and Λ^{-1} are well-defined and mutual inverses. What is more, they supply us with exponentials.

Proposition 3.61 *$\mathcal{V}_{\mathtt{t}}$ has partial exponentials wrt to $\otimes$, in the following sense. For any object B, the functor $\_ \otimes B : \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{t}}$ has a partial right adjoint $B \multimap \_ : \mathcal{V}_{\mathtt{t}*} \longrightarrow \mathcal{V}_{\mathtt{t}}$, that is, for any object A and any pointed object C the bijection*

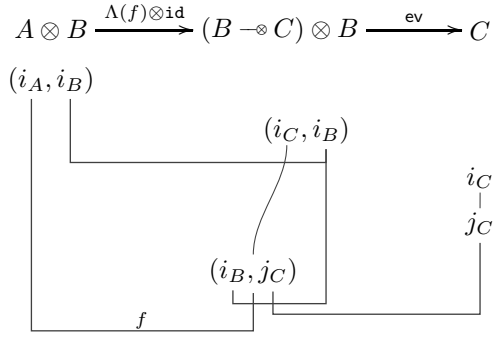
$$\Lambda^B_{A,C} : \mathcal{V}_{\mathtt{t}}(A \otimes B, C) \xrightarrow{\cong} \mathcal{V}_{\mathtt{t}}(A, B \multimap C)$$

is natural in A.

Proof: It suffices to show that, for any $f : A \otimes B \longrightarrow C$ and $g : A \longrightarrow B \multimap C$,

$$\Lambda(f) \otimes \mathtt{id}\,;\mathtt{ev} = f\,, \quad g \otimes \mathtt{id}\,;\mathtt{ev} = \Lambda^{-1}(g)\,.$$

The above equalities are straightforward. For example, the viewfunction of $\Lambda(f) \otimes \mathtt{id}\,;\mathtt{ev}$ is given by the following diagram,



which gives also the viewfunction of f. ■

A consequence of partial exponentiation is that $\multimap$ naturally upgrades to a functor:

$$\_ \multimap \_ : (\mathcal{V}_{\mathtt{t}})^{\mathrm{op}} \times \mathcal{V}_{\mathtt{t}*} \longrightarrow \mathcal{V}_{\mathtt{t}*}\,.$$

Now, in case g is ttotal, the strategy $f \multimap g : A' \multimap B \longrightarrow A \multimap B'$ is given concretely by $\mathtt{strat}(\phi)$, where

$$\phi = \{[i_B\, i_{B'}\, (i_A, j_{B'})\, (i_{A'}, j_B)\, s] \mid ([i_A\, i_{A'}\, s] \in \mathtt{viewf}(f) \wedge [i_B\, i_{B'}\, j_{B'}\, j_B] \in g \wedge \mathcal{L}(i_A\, i_{A'}\, s) \# i_B, j_{B'})$$
$$\vee\, ([i_B\, i_{B'}\, j_{B'}\, j_B\, s] \in \mathtt{viewf}(g) \wedge [i_A\, i_{A'}] \in f \wedge \mathcal{L}(i_B\, i_{B'}\, j_{B'}\, j_B\, s) \# i_A)\}.$$

[5] Note the reassignment of pointers that takes place implicitly in the definitions of Λ, Λ^{-1}, in order e.g. for $(i_A, i_B)\, i_C\, j_C\, s$ to be a play of $\mathtt{viewf}(h)$.

That is, $f \multimap g$ answers initial moves $[i_B]$ like g and then responds to $[i_B\, i_{B'}\, (i_A, j_{B'})]$ with f's answer to $[i_A]$ and g's response to $[i_B\, i_{B'}\, j_{B'}]$ (recall g ttotal). It then plays like f or like g, according to Opponent's next move. Note that ϕ is a viewfunction even if B, B' are not pointed.

A special case of ttotality in the second argument arises in the defined functor:

$$\_ \Rightarrow \_ : (\mathcal{V}_{\mathtt{t}})^{\mathrm{op}} \times \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{tt}*} \triangleq \_ \multimap (\_)_\perp \,. \tag{3.3}$$

Remark 3.62 In the work on CBV games of Honda & Yoshida [HY99] the following version of partial exponentiation is shown.

$$\mathcal{V}(A \otimes B, C) \cong \mathcal{V}_{\mathtt{t}}(A, B \Rightarrow C) \tag{3.4}$$

Interestingly, that version can be derived from ours (using also another bijection shown in [HY99]),

$$\mathcal{V}(A \otimes B, C) \cong \mathcal{V}_{\mathtt{t}}(A \otimes B, C_\perp) \cong \mathcal{V}_{\mathtt{t}}(A, B \multimap C_\perp) = \mathcal{V}_{\mathtt{t}}(A, B \Rightarrow C)\,.$$

But also vice versa, if C is pointed then $C \cong C_2 \Rightarrow C_1$, for some arenas C_1, C_2,[6] and

$$\mathcal{V}_{\mathtt{t}}(A \otimes B, C_2 \Rightarrow C_1) \overset{(3.4)}{\cong} \mathcal{V}(A \otimes B \otimes C_2, C_1) \overset{(3.4)}{\cong} \mathcal{V}_{\mathtt{t}}(A, (B \otimes C_2) \Rightarrow C_1) = \mathcal{V}_{\mathtt{t}}(A, B \multimap (C_2 \Rightarrow C_1))\,.$$

3.3.4 Coproducts

We show very briefly that $\mathcal{V}_{\mathtt{t}}$ has distributive coproducts. In fact, $\mathcal{V}_{\mathtt{t}}$ is an *extensive* category (which subsumes distributivity, v. [CLW93]), but this will not be of real use here.

Definition 3.63 For any arenas A, B, define the arrows:

$$\begin{aligned} \iota_1 : A \longrightarrow A + B &\triangleq \{[i_A\, i_A\, s] \mid [i_A\, i_A\, s] \in \mathtt{id}_A\}\,, \\ \iota_2 : B \longrightarrow A + B &\triangleq \{[i_B\, i_B\, s] \mid [i_B\, i_B\, s] \in \mathtt{id}_B\}\,. \end{aligned}$$

Moreover, for any $A \xrightarrow{f} C \xleftarrow{g} B$ take

$$[f, g] : A + B \longrightarrow C \triangleq \{[i_A\, i_C\, s] \mid [i_A\, i_C\, s] \in f\} \cup \{[i_B\, i_C\, s] \mid [i_B\, i_C\, s] \in g\}\,.$$

Finally, for each arena A, define the arrow $!_A : 0 \longrightarrow A \triangleq \{[\epsilon]\}$. ▲

Note that we use the same symbol, $!_A$, both for terminal and initial arrows; this usually does not cause confusion.

Proposition 3.64 *The structure defined above equips $\mathcal{V}_{\mathtt{t}}$ with finite coproducts. Moreover, for all arenas A, B, C, the following arrow is an iso.*

$$\mathtt{dst} \triangleq A \otimes B + A \otimes C \xrightarrow{[\mathtt{id}_A \otimes \iota_1,\, \mathtt{id}_A \otimes \iota_2]} A \otimes (B + C)$$

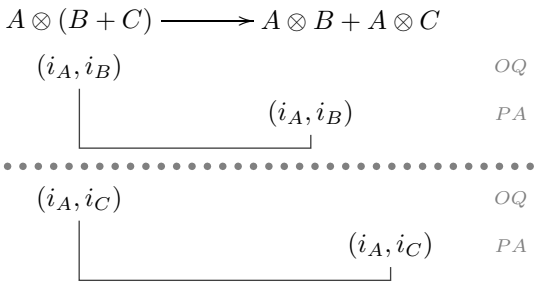
[6] In fact, for C to be expressed as $C_2 \Rightarrow C_1$ we need a stronger version of condition (f), definition 3.1, namely:

(f′) For each $m \in M_A$, there exists unique $k \geq 0$ and a unique sequence $x_1 \ldots x_n \in \{Q, A\}^*$ such that $I_A \ni m_1 \vdash_A \cdots \vdash_A m_k \vdash_A m$, for some m_l's in M_A with $\lambda_C^{QA}(m_l) = x_l$.

In such a case, C_1 and C_2 are given by taking $K_C^A \triangleq \{m \in M_C \mid \exists j_C.\, j_C \vdash_C m \wedge \lambda_C(m) = PA\}$ and

$$\begin{aligned} M_{C_1} &\triangleq K_C^A + \{m \in M_C \mid \exists k \in K_C^A.\, k \vdash_C \cdots \vdash_C m\} & M_{C_2} &\triangleq \bar{I}_C \setminus M_{C_1} \\ I_{C_1} &\triangleq K_C^A & I_{C_2} &\triangleq J_C \\ \vdash_{C_1} &\triangleq \vdash_C \restriction (M_{C_1} \times \bar{I}_{C_1}) & \vdash_{C_2} &\triangleq \vdash_C \restriction (M_{C_2} \times \bar{I}_{C_2}) \\ \lambda_{C_1} &\triangleq \lambda_C \restriction M_{C_1} & \lambda_{C_2} &\triangleq [i_{C_2} \mapsto PA, m \mapsto \bar{\lambda}_C(m)]\,. \end{aligned}$$

Proof: It is not difficult to see that the above yield finite coproducts. For distributivity, the following strategy is an inverse to `dst`.



■

3.3.5 Strategy and arena orders

Recall the orders defined for strategies ($\lesssim$) and arenas ($\trianglelefteq$) in section 3.1.3. These being subset orderings are automatically inherited by $\mathcal{V}_{\mathtt{t}}$. Moreover, they are very well-behaved in the following sense.

Proposition 3.65 *$\mathcal{V}_{\mathtt{t}}$ and $\mathcal{V}_{\mathtt{tt}}$ are PreCpo-enriched wrt $\sqsubseteq$.*[7] *Moreover, the following are locally continuous functors.*

$$(\_)_\perp : \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{tt}*}\,, \qquad (\_ \otimes \_) : \mathcal{V}_{\mathtt{t}} \times \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{t}}\,, \qquad (\bigotimes \_) : \prod \mathcal{V}_{\mathtt{t}*} \longrightarrow \mathcal{V}_{\mathtt{t}*}\,,$$
$$(\_ \multimap \_) : \mathcal{V}_{\mathtt{t}}^{\mathrm{op}} \times \mathcal{V}_{\mathtt{tt}*} \longrightarrow \mathcal{V}_{\mathtt{tt}*}\,, \quad (\_ \Rightarrow \_) : \mathcal{V}_{\mathtt{t}}^{\mathrm{op}} \times \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{tt}*}\,, \quad (\_ + \_) : \mathcal{V}_{\mathtt{t}} \times \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{t}}\,.$$

Proof: Enrichment follows from enrichment of $\mathcal{G}$; only the least element is lost, since it is not necessarily total. To show that the defined functors are locally continuous we make use of corollary 3.50. For example, given $\sigma \sqsubseteq \sigma'$ and $\tau \sqsubseteq \tau'$, in order to show that $\sigma \otimes \tau \sqsubseteq \sigma' \otimes \tau'$ it suffices to show that $\mathtt{viewf}(\sigma \otimes \tau) \subseteq \mathtt{viewf}(\sigma' \otimes \tau')$, which is straightforward from the definition of tensor. On the other hand, if $(\sigma_i)_{i\in\omega}$ is an ω-chain then, in order to show that $(\bigsqcup_i \sigma_i) \otimes \tau \sqsubseteq \bigsqcup_i (\sigma_i \otimes \tau)$, it suffices to show that $\mathtt{viewf}((\bigsqcup_i \sigma_i) \otimes \tau) \sqsubseteq \bigsqcup_i \mathtt{viewf}(\sigma_i \otimes \tau)$, which is straightforward. ■

The order on arenas in $\mathcal{V}_{\mathtt{t}}$ is the same as in $\mathcal{G}$, and therefore $Ob(\mathcal{V}_{\mathtt{t}})$ is a cpo with least element 0. Note though that a requirement needs to be added for projections to be total strategies.

Definition 3.66 For any $A, B \in Ob(\mathcal{V}_{\mathtt{t}})$ and $k \in \omega$ define

$$A \trianglelefteq_k B \iff A \trianglelefteq B \wedge (B \upharpoonright \{m \in M_B \mid level(m) < k\}) \trianglelefteq A$$

If $A \trianglelefteq_1 B$ then we can define a (total) projection arrow

$$\mathtt{proj}_{B,A} : B \longrightarrow A \triangleq \mathtt{strat}\{[s] \mid [s] \in \mathtt{viewf}(\mathtt{id}_A)\}$$

▲

This indexed version of the ordering relation allows us to stipulate totality and ttotality on projections and inclusions:

$$A \trianglelefteq_1 B \implies \mathtt{proj}_{B,A} \in \mathcal{V}_{\mathtt{tt}}(B, A)\,,$$
$$A \trianglelefteq_2 B \implies \mathtt{incl}_{A,B} \in \mathcal{V}_{\mathtt{tt}}(A, B)\,.$$

Moreover, we have the following.

[7] By precpo we mean a cpo which may not have a least element. PreCpo is the category of precpos and continuous functions.

Proposition 3.67 *All of the functors of proposition 3.65 are continuous wrt $\trianglelefteq$. Moreover,*

$$\begin{aligned}
A \trianglelefteq A' \wedge B \trianglelefteq B' &\implies \mathtt{incl}_{A,A'} \otimes \mathtt{incl}_{B,B'} = \mathtt{incl}_{A\otimes B, A'\otimes B'} \\
A \trianglelefteq_1 A' \wedge B \trianglelefteq_1 B' &\implies \mathtt{proj}_{A',A} \otimes \mathtt{proj}_{B',B} = \mathtt{proj}_{A'\otimes B', A\otimes B} \\
\forall i \in \omega.\, A_i \trianglelefteq A'_i &\implies \bigotimes_i \mathtt{incl}_{A_i,A'_i} = \mathtt{incl}_{\bigotimes_i A_i, \bigotimes_i A'_i} \\
\forall i \in \omega.\, A_i \trianglelefteq A'_i &\implies \bigotimes_i \mathtt{proj}_{A'_i,A_i} = \mathtt{proj}_{\bigotimes_i A'_i, \bigotimes_i A_i} \\
A \trianglelefteq_1 A' \wedge B \trianglelefteq B' &\implies \mathtt{proj}_{A',A} \Rightarrow \mathtt{incl}_{B,B'} = \mathtt{incl}_{A\Rightarrow B, A'\Rightarrow B'} \\
A \trianglelefteq A' \wedge B \trianglelefteq_1 B' &\implies \mathtt{incl}_{A,A'} \Rightarrow \mathtt{proj}_{B',B} = \mathtt{proj}_{A'\Rightarrow B', A\Rightarrow B} \\
A \trianglelefteq_1 A' \wedge B \trianglelefteq_2 B' &\implies \mathtt{proj}_{A',A} \multimap \mathtt{incl}_{B,B'} = \mathtt{incl}_{A\multimap B, A'\multimap B'} \\
A \trianglelefteq A' \wedge B \trianglelefteq_1 B' &\implies \mathtt{incl}_{A,A'} \multimap \mathtt{proj}_{B',B} = \mathtt{proj}_{A'\multimap B', A\multimap B} \\
A \trianglelefteq A' \wedge B \trianglelefteq B' &\implies \mathtt{incl}_{A,A'} + \mathtt{incl}_{B,B'} = \mathtt{incl}_{A+B, A'+B'} \\
A \trianglelefteq_1 A' \wedge B \trianglelefteq_1 B' &\implies \mathtt{proj}_{A',A} + \mathtt{proj}_{B',B} = \mathtt{proj}_{A'+B', A+B}
\end{aligned}$$

Proof: It is not difficult to show $\trianglelefteq$-continuity. Now, all the above clauses are in effect special cases of functoriality statements, since the underlying sets of inclusions and projections correspond to identity strategies. ■

3.4 A monad, and some comonads

We now proceed to construct a monad and a family of comonads on $\mathcal{V}_\mathtt{t}$ that will be of use in later chapters. Specifically, we will upgrade lifting to a monad and introduce a family of product comonads for initial state.

3.4.1 Lifting monad

It is a more-or-less standard result that the lifting functor induces a monad.

Definition 3.68 (Lifting monad) Define the natural transformations $\mathtt{up}, \mathtt{dn}, \mathtt{st}$ as follows.

$$\begin{aligned}
\mathtt{up}_A &: A \longrightarrow A_\perp = \mathtt{strat}\{\, [i_A *_1 *_2 i_A\, s] \mid [i_A\, i_A\, s] \in \mathtt{viewf}(\mathtt{id}_A) \,\} \\
\mathtt{dn}_A &: A_{\perp\perp} \longrightarrow A_\perp \triangleq \mathtt{strat}\{\, [*_1 *'_1 *'_2 *_2 *_3 *_4\, s] \mid [s] \in \mathtt{viewf}(\mathtt{id}_A) \,\} \\
\mathtt{st}_{A,B} &: A \otimes B_\perp \longrightarrow (A\otimes B)_\perp \triangleq \mathtt{strat}\{\, [(i_A, *_1) *'_1 *'_2 *_2\ i_B\,(i_A, i_B)\, s] \\
&\qquad\qquad \mid [(i_A, i_B)\,(i_A, i_B)\, s] \in \mathtt{viewf}(\mathtt{id}_{A\otimes B}) \,\}
\end{aligned}$$

(primed asterisks are used for arenas on the RHS, where necessary). ▲

Proposition 3.69 *The quadruple $((\_)_\perp, \mathtt{up}, \mathtt{dn}, \mathtt{st})$ is a strong monad on $\mathcal{V}_\mathtt{t}$. Moreover, it yields monadic exponentials by taking $(C_\perp)^B$ to be $B \Rightarrow C$, for each B, C.*

Proof: It is not difficult to see that $((\_)_\perp, \mathtt{up}, \mathtt{dn}, \mathtt{st})$ is a strong monad. Moreover, for each B, C we have that $B \Rightarrow C = B \multimap C_\perp$ is a $(\_)_\perp$-exponential, because of exponentiation properties of $\multimap$. ■

Although finding a canonical arrow from A to $A_\perp$ is elementary ($\mathtt{up}_A$), finding a canonical arrow in the inverse direction is not always possible. In some cases, e.g. $A = \mathbb{A}_i$, there is no such arrow at all, let alone canonical. An exception occurs when A is pointed.

Definition 3.70 For any pointed arena A define:

$$\mathtt{pu}_A : A_\perp \longrightarrow A \triangleq \mathtt{strat}\{\, [*\, i_A\, j_A\ *\ i_A\, j_A\, s] \mid [i_A\, i_A\, j_A\, j_A\, s] \in \mathtt{viewf}(\mathtt{id}_A) \,\}.$$

▲

Lemma 3.71 $\mathtt{pu}_A$ *yields a natural transformation* $\mathtt{pu} : (\_)_{\perp(\mathcal{V}_{\mathtt{tt}*})} \longrightarrow Id_{\mathcal{V}_{\mathtt{tt}*}}$. *Moreover, for any arenas* A, B *with* B *pointed,*

- $\mathtt{up}_A \,;\mathtt{pu}_A = \mathtt{id}_A$,
- $\mathtt{pu}_{A_\perp} = \mathtt{dn}_A$,
- $\mathtt{pu}_{A\multimap B} = \Lambda\Big((A \multimap B)_\perp \otimes A \xrightarrow{\mathtt{st}'} ((A \multimap B) \otimes A)_\perp \xrightarrow{\mathtt{ev}_\perp} B_\perp \xrightarrow{\mathtt{pu}_B} B\Big)$. ∎

3.4.2 Initial-state comonads

Our way of modelling terms-in-local-state will be by using initial state comonads, in the spirit of intensional program modelling of Brookes & Geva [BG92]. In our setting, the initial state can be any list $\vec{a}$ of distinct names; we define a comonad for each one of those lists.

Definition 3.72 (Initial-state comonads) For each $\vec{a} \in \mathbb{A}^\#$ define the triple $(Q^{\vec{a}}, \varepsilon, \delta)$ as follows.

$$Q^{\vec{a}} : \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{t}} \triangleq \mathbb{A}^{\vec{a}} \otimes \_ \,,$$
$$\varepsilon : Q^{\vec{a}} \longrightarrow Id_{\mathcal{V}_{\mathtt{t}}} \triangleq \{\varepsilon_A : \mathbb{A}^{\vec{a}} \otimes A \xrightarrow{\pi_2} A\} \,,$$
$$\delta : Q^{\vec{a}} \longrightarrow (Q^{\vec{a}})^2 \triangleq \{\delta_A : \mathbb{A}^{\vec{a}} \otimes A \xrightarrow{\Delta \otimes \mathtt{id}} \mathbb{A}^{\vec{a}} \otimes \mathbb{A}^{\vec{a}} \otimes A\} \,.$$

For each $\vec{a}' \subseteq \vec{a}$ define the natural transformation $\frac{\vec{a}}{\vec{a}'} : Q^{\vec{a}} \longrightarrow Q^{\vec{a}'}$ by taking,

$$(\tfrac{\vec{a}}{\vec{a}'})_A : \mathbb{A}^{\vec{a}} \otimes A \longrightarrow \mathbb{A}^{\vec{a}'} \otimes A \triangleq (\tfrac{\vec{a}}{\vec{a}'})_1 \otimes \mathtt{id}_A \,,$$
$$(\tfrac{\vec{a}}{\vec{a}'})_1 : \mathbb{A}^{\vec{a}} \otimes 1 \longrightarrow \mathbb{A}^{\vec{a}'} \otimes 1 \triangleq \{[(\vec{a}, *)\,(\vec{a}', *)]\} \,.$$ ▲

Note that Q^ϵ, the comonad for empty initial state, is the identity comonad. Note also that we have suppressed indices $\vec{a}$ from transformations ε, δ for notational economy.

From the results in the previous chapter, we know that each triple $(Q^{\vec{a}}, \varepsilon, \delta)$ forms a product comonad on $\mathcal{V}_{\mathtt{t}}$. Moreover, it is straightforward to show the following.

Proposition 3.73 (Chain rule) *For each* $\vec{a}' \subseteq \vec{a} \in \mathbb{A}^\#$, *the transformation* $\frac{\vec{a}}{\vec{a}'}$ *is a comonad morphism. Moreover,* $\frac{\vec{a}}{\epsilon} = \varepsilon : Q^{\vec{a}} \longrightarrow Id_{\mathcal{V}_{\mathtt{t}}}$, $\frac{\vec{a}}{\vec{a}} = \mathtt{id} : Q^{\vec{a}} \longrightarrow Q^{\vec{a}}$ *and, for each* $\vec{a}' \subseteq \vec{a}'' \subseteq \vec{a}$,

$$\frac{\vec{a}}{\vec{a}''} \,;\, \frac{\vec{a}''}{\vec{a}'} = \frac{\vec{a}}{\vec{a}'} \,.$$ ∎

Finally, for each name-type i, we have a name-test arrow:

$$\mathtt{eq}_i : \mathbb{A}_i \otimes \mathbb{A}_i \longrightarrow \mathbb{N} \triangleq \{\, [(a,a)\,0] \,\} \cup \{\, [(a,b)\,1] \mid a \# b \,\}$$

which clearly makes the following diagram commute.

$$\begin{array}{ccccc} Q^a 1 & \xrightarrow{\Delta} & \mathbb{A}_i \otimes \mathbb{A}_i & \xleftarrow{\langle \frac{ab}{a}, \frac{ab}{b} \rangle} & Q^{ab} 1 \\ \downarrow{\scriptstyle !} & & \downarrow{\scriptstyle \mathtt{eq}_i} & & \downarrow{\scriptstyle !} \\ 1 & \xrightarrow[\widetilde{0}]{} & \mathbb{N} & \xleftarrow[\widetilde{1}]{} & 1 \end{array} \qquad \text{(N1)}$$

Remark 3.74 (Why use comonads) We briefly discuss about our use of initial-state comonads for modelling local state, instead of following the more standard method of a local-state monad. We focus on the $s\nu$-calculus. An appropriate local-state monad for this calculus would be a monad T of the form

$$TX = \mathbb{A}^\# \Rightarrow \mathbb{A}^\# \otimes X$$

and would entail that each typed term $\vec{a} \mid \Gamma \vdash M : A$ be translated to a morphism $[\![M]\!] : [\![\Gamma]\!] \longrightarrow \mathbb{A}^{\#} \Rightarrow (\mathbb{A}^{\#} \otimes [\![A]\!])$, or, equivalently,

$$[\![M]\!] : \mathbb{A}^{\#} \otimes [\![\Gamma]\!] \longrightarrow (\mathbb{A}^{\#} \otimes [\![A]\!])_{\perp} .$$

However, the generic treatment of initial state in the above description is inadequate for our purposes. $\mathbb{A}^{\#}$ contains all possible initial states, and $[\![M]\!]$ would have to be specified for each single one of these — even for irrelevant ones.

What seems more fitting to our nominal framework is a translation to *specific initial states*, up to permutation, which concretely means

$$[\![M]\!] : \mathbb{A}^{\vec{a}} \otimes [\![\Gamma]\!] \longrightarrow (\mathbb{A}^{\#} \otimes [\![A]\!])_{\perp}$$

in our running example. But since in nominal games all information pertaining fresh names is embedded in moves (cf. [AGM+04, Ong02]), the appearance of $\mathbb{A}^{\#}$ in the RHS above is redundant. Hence, the desired interpretation boils down to

$$[\![M]\!] : \mathbb{A}^{\vec{a}} \otimes [\![\Gamma]\!] \longrightarrow [\![A]\!]_{\perp}$$

which is precisely the (underlying) interpretation pursued in the next chapters.

3.4.3 Fresh-name constructors

Combining the monad and comonads of the previous sections we can obtain a monadic-comonadic setting $(\mathcal{V}_{\mathtt{t}}, (\_)_{\perp}, Q)$, where by Q we denote the family $(Q^{\vec{a}})_{\vec{a} \in \mathbb{A}^{\#}}$. This setting, which in fact yields a sound model of the $s\nu$-calculus, will be used as the basis of our semantics of nominal computation in the sequel. As discussed in remark 3.74, nominal computation of type A, in name-environment $\vec{a}$ and variable-environment Γ, will be translated in the set of strategies

$$\{\, \sigma : Q^{\vec{a}}[\![\Gamma]\!] \longrightarrow [\![A]\!]_{\perp} \,\} .$$

The lifting functor, representing the monadic part of our semantical setting, will therefore incorporate the computational effect of fresh-name creation.

We describe in this section the game-semantical expression of fresh-name creation. Fresh names are created by means of natural transformations which transform a comonad $Q^{\vec{a}}$, say, to a monad-comonad composite $(Q^{\vec{a}a}\_)_{\perp}$.

Definition 3.75 Consider the setting $(\mathcal{V}_{\mathtt{t}}, (\_)_{\perp}, Q)$. We define, for each $\vec{a}a \in \mathbb{A}^{\#}$, natural transformations

$$\mathtt{new}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow (Q^{\vec{a}a}\_)_{\perp}$$

by:

$$\mathtt{new}^{\vec{a}a}_{A} \triangleq \mathbb{A}^{\vec{a}} \otimes A \xrightarrow{\mathtt{new}^{\vec{a}a}_{1} \otimes \mathtt{id}_{A}} (\mathbb{A}^{\vec{a}a})_{\perp} \otimes A \xrightarrow{\mathtt{st}'} (\mathbb{A}^{\vec{a}a} \otimes A)_{\perp} ,$$

$$\mathtt{new}^{\vec{a}a}_{1} : \mathbb{A}^{\vec{a}} \otimes 1 \longrightarrow (\mathbb{A}^{\vec{a}a} \otimes 1)_{\perp} \triangleq \mathtt{strat}\{[(\vec{a}, *) * *(\vec{a}a, *)^{a}]\} .$$

▲

We will usually omit superscripts in `new`, for economy. That `new` is a natural transformation is straightforward: for any $f : A \longrightarrow B$ we can form the following commutative diagram.

$$\begin{array}{ccccc}
\mathbb{A}^{\vec{a}} \otimes A & \xrightarrow{\mathtt{new}_1 \otimes \mathtt{id}} & (\mathbb{A}^{\vec{a}a})_{\perp} \otimes A & \xrightarrow{\mathtt{st}'} & (\mathbb{A}^{\vec{a}a} \otimes A)_{\perp} \\
\downarrow{\scriptstyle \mathtt{id} \otimes f} & & \downarrow{\scriptstyle \mathtt{id} \otimes f} & & \downarrow{\scriptstyle (\mathtt{id} \otimes f)_{\perp}} \\
\mathbb{A}^{\vec{a}} \otimes B & \xrightarrow[\mathtt{new}_1 \otimes \mathtt{id}]{} & (\mathbb{A}^{\vec{a}a})_{\perp} \otimes B & \xrightarrow[\mathtt{st}']{} & (\mathbb{A}^{\vec{a}a} \otimes B)_{\perp}
\end{array}$$

Moreover, $\mathtt{new}$ is ***strength-coherent***, in the following sense. For any arenas A, B it is easy to see that the following diagram commutes.

$$\begin{array}{ccc} A \otimes Q^{\vec{a}} B & \xrightarrow{\zeta} & Q^{\vec{a}}(A \otimes B) \\ \downarrow{\scriptstyle \mathtt{id}\otimes\mathtt{new}_B} & & \downarrow{\scriptstyle \mathtt{new}_{A\otimes B}} \\ A \otimes (Q^{\vec{a}a} B)_\perp & \xrightarrow[\mathtt{st}\,;\,\zeta_\perp]{} & (Q^{\vec{a}a}(A \otimes B))_\perp \end{array} \tag{3.5}$$

Finally, we can show the following.

Proposition 3.76 *For all $\vec{a}, \vec{a}'a \in \mathbb{A}^{\#}$ with $\vec{a} \subseteq \vec{a}'$ and any arena A, the following diagrams commute.*

$$\begin{array}{ccc} Q^{\vec{a}} A & \xrightarrow{\langle \mathtt{id}, \mathtt{new}_A \rangle} & Q^{\vec{a}} A \otimes (Q^{\vec{a}a} A)_\perp \\ \downarrow{\scriptstyle \mathtt{new}_A} & & \downarrow{\scriptstyle \mathtt{st}} \\ (Q^{\vec{a}a} A)_\perp & \xrightarrow[\langle \frac{\vec{a}a}{\vec{a}\,A}, \mathtt{id} \rangle_\perp]{} & (Q^{\vec{a}} A \otimes Q^{\vec{a}a} A)_\perp \end{array} \qquad \begin{array}{ccc} Q^{\vec{a}'} A & \xrightarrow{\mathtt{new}^{\vec{a}'a}} & (Q^{\vec{a}'a} A)_\perp \\ \downarrow{\scriptstyle \frac{\vec{a}'}{\vec{a}}} & & \downarrow{\scriptstyle (\frac{\vec{a}'a}{\vec{a}a})_\perp} \\ Q^{\vec{a}} A & \xrightarrow[\mathtt{new}^{\vec{a}a}]{} & (Q^{\vec{a}a} A)_\perp \end{array} \tag{N2}$$

Proof: It is easy to see that commutativity of the LHS diagram reduces to commutativity of

$$\begin{array}{ccc} \mathbb{A}^{\vec{a}} & \xrightarrow{\langle \mathtt{id}, \mathtt{new}_1 \rangle} & \mathbb{A}^{\vec{a}} \otimes (\mathbb{A}^{\vec{a}a})_\perp \\ \downarrow{\scriptstyle \mathtt{new}_1} & & \downarrow{\scriptstyle \mathtt{st}} \\ (\mathbb{A}^{\vec{a}a})_\perp & \xrightarrow[\langle \frac{\vec{a}a}{\vec{a}\,1}, \mathtt{id} \rangle_\perp]{} & (\mathbb{A}^{\vec{a}} \otimes \mathbb{A}^{\vec{a}a})_\perp \end{array}$$

which is straightforward. The RHS diagram is shown similarly. ∎

The fresh-name constructor allows us to define name-abstraction for strategies.

Definition 3.77 (Name-abstraction, $\langle\!\rangle$) For any $\sigma : Q^{\vec{a}a} B \longrightarrow C$, where C is pointed, define:

$$\langle\!\langle a \rangle\!\rangle \sigma \triangleq Q^{\vec{a}} B \xrightarrow{\mathtt{new}_B^{\vec{a}a}} (Q^{\vec{a}a} B)_\perp \xrightarrow{\sigma_\perp} C_\perp \xrightarrow{\mathtt{pu}_C} C\,.$$

▲

Name-abstraction can be given an explicit description as follows. For any sequence of moves-with-names s and any name $a \# \mathsf{nlist}(s)$, let s^a be s with a added in the head of all of its name-lists. Then, for σ as above, we can show that:

$$\mathtt{viewf}(\langle\!\langle a \rangle\!\rangle \sigma) = \{\, [(\vec{a}, i_B)\, i_C\, j_C\, m^{a\vec{b}}\, s^a] \mid [(\vec{a}a, i_B)\, i_C\, j_C\, m^{\vec{b}}\, s] \in \mathtt{viewf}(\sigma) \wedge a \# i_B, j_C \,\} \tag{3.6}$$

Thus, for example, for any $f, g : Q^{\vec{a}a} 1 \longrightarrow B_\perp$ we have:

$$f = g \iff \langle\!\langle a \rangle\!\rangle f = \langle\!\langle a \rangle\!\rangle g\,. \tag{3.7}$$

We end our discussion on fresh-name constructors with a technical lemma stating that name-abstraction and currying commute.

Lemma 3.78 *Let $f : Q^{\vec{a}a}(A \otimes B) \longrightarrow C$, with C a pointed arena. Then*

$$\langle\!\langle a \rangle\!\rangle \Lambda(\zeta\,;\, f) = \Lambda(\zeta'\,;\, \langle\!\langle a \rangle\!\rangle f) : Q^{\vec{a}} A \longrightarrow B \multimap C\,.$$

Proof: As follows.

$$\begin{aligned}
\langle a\rangle\,\Lambda(\zeta';f) &= \mathtt{new}_A^{\vec{a}a};(\Lambda(\zeta';f))_\perp;\mathtt{pu}_{B\multimap C} = \mathtt{new}_A^{\vec{a}a};(\Lambda(\zeta';f))_\perp;\Lambda(\mathtt{st}';\mathtt{ev}_\perp;\mathtt{pu}_C)\\
&= \Lambda(\mathtt{new}_A^{\vec{a}a}\otimes\mathtt{id}_B;(\Lambda(\zeta';f))_\perp\otimes\mathtt{id}_B;\mathtt{st}';\mathtt{ev}_\perp;\mathtt{pu}_C)\\
&= \Lambda(\mathtt{new}_A^{\vec{a}a}\otimes\mathtt{id}_B;\mathtt{st}';(\Lambda(\zeta';f)\otimes\mathtt{id}_B)_\perp;\mathtt{ev}_\perp;\mathtt{pu}_C)\\
&= \Lambda(\mathtt{new}_A^{\vec{a}a}\otimes\mathtt{id}_B;\mathtt{st}';(\zeta';f)_\perp;\mathtt{pu}_C)\\
&\overset{\text{(N2)}}{=} \Lambda(\zeta';\mathtt{new}_{A\otimes B}^{\vec{a}a};f_\perp;\mathtt{pu}_C) = \Lambda(\zeta';\langle a\rangle\,f)
\end{aligned}$$

■

Note that the above result does *not* imply that ν- and λ-abstractions commute in our semantics of nominal languages, i.e. that we obtain identifications of the form $[\![\nu a.\lambda x.M]\!] = [\![\lambda x.\nu a.M]\!]$. As we will see in the next chapters, λ-abstraction is not simply currying: because of monads, it corresponds to currying and composing with the monadic unit.

Generalised constructors The previous construction can be generalised as follows. For any $\vec{a}\subseteq\vec{a}'$ define

$$\binom{\vec{a}}{\vec{a}'} : Q^{\vec{a}} \longrightarrow (Q^{\vec{a}'}\_)_\perp \tag{3.8}$$

from

$$\binom{\vec{a}}{\vec{a}'}_1 : Q^{\vec{a}}1 \longrightarrow (Q^{\vec{a}'}1)_\perp \triangleq \{\,[(\vec{a},*)**(\vec{a}',*)^{\vec{a}'\smallsetminus\vec{a}}]\,\}, \tag{3.9}$$

where $\vec{a}'\setminus\vec{a}$ is $\vec{a}'$ with all names from $\vec{a}$ removed. Clearly, this too is strength-preserving and moreover makes the following diagram commute, for any A.

$$\begin{array}{ccc}
Q^{\vec{a}}A & \xrightarrow{\langle \mathtt{id},\binom{\vec{a}}{\vec{a}'}_A\rangle} & Q^{\vec{a}}A\otimes(Q^{\vec{a}'}A)_\perp\\
{\scriptstyle\binom{\vec{a}}{\vec{a}'}_A}\downarrow & & \downarrow{\scriptstyle\mathtt{st}}\\
(Q^{\vec{a}'}A)_\perp & \xrightarrow[\langle\frac{\vec{a}'}{\vec{a}}_A,\mathtt{id}\rangle_\perp]{} & (Q^{\vec{a}}A\otimes Q^{\vec{a}'}A)_\perp
\end{array} \tag{N2'}$$

We can now define a generalised name-abstraction constructor for strategies. For any $\sigma : Q^{\vec{a}'}B\longrightarrow C$ with C pointed,

$$\langle\vec{a}\,|\,\vec{a}'\rangle\,\sigma \triangleq Q^{\vec{a}}B\xrightarrow{\binom{\vec{a}}{\vec{a}'}_B}(Q^{\vec{a}'}B)_\perp\xrightarrow{\sigma_\perp}C_\perp\xrightarrow{\mathtt{pu}_C}C\,. \tag{3.10}$$

The above can be given an explicit description as follows.

$$\begin{aligned}
&\mathtt{viewf}(\,\langle\vec{a}\,|\,\vec{a}'\rangle\,\sigma) = \qquad\qquad (3.11)\\
&\quad\{\,[(\vec{a},i_B)\,i_C\,j_C\,m^{(\vec{a}'\setminus\vec{a})\vec{b}}\,s^{(\vec{a}'\setminus\vec{a})}] \mid [(\vec{a}',i_B)\,i_C\,j_C\,m^{\vec{b}}\,s]\in\mathtt{viewf}(\sigma)\wedge(\vec{a}'\setminus\vec{a})\,\#\,i_B,j_C\,\}
\end{aligned}$$

This shows that the constructor $\langle\_\,|\,\_\rangle$ indeed generalises $\langle\_\rangle$: taking $\langle\vec{a}'\rangle \triangleq \langle a_1\rangle\cdots\langle a_n\rangle$ we have $\langle\vec{a}'\rangle = \langle\vec{a}\,|\,\vec{a}\vec{a}'\rangle$. Finally, similarly to lemma 3.78 we can show that, for any $f : Q^{\vec{a}'}(A\otimes B)\longrightarrow C$ with C pointed,

$$\langle\vec{a}\,|\,\vec{a}'\rangle\,\Lambda(\zeta';f) = \Lambda(\zeta';\langle\vec{a}\,|\,\vec{a}'\rangle\,f) : Q^{\vec{a}}A\longrightarrow B\multimap C\,. \tag{3.12}$$

3.5 Nominal games à la Laird

As aforementioned, there have been two independent original presentations of nominal games, one due to Abramsky, Ghica, Murawski, Ong and Stark [AGM+04] and another one

due to Laird [Lai04, Lai08]. Although Laird's constructions are not explicitly based on nominal sets (natural numbers are used instead of atoms), they constitute nominal constructions nonetheless. In this section we highlight the main differences between our nominal games, which follow [AGM$^+$04], and those of [Lai04, Lai08].

Laird's presentation concerns the ν-calculus with pointers, i.e. with references to names. The main difference in his presentation is in the treatment of name-introduction. In particular, a name does not appear in a play at the point of evaluation of its ν-constructor, but rather at the point of its first *use*; let us refer to this condition as *name-frugality* (cf. [MT09]). An immediate result is that strategies are no longer innocent, as otherwise e.g. $\nu a.\lambda x.a$ and $\lambda x.\nu a.a$ would have the same denotation.[8] More importantly, name-frugality implies that strategies capture the examined nominal language more *accurately*: Opponent is not expected to guess names he is not supposed to know and thus, for example, the denotations of $\nu a.\mathtt{skip}$ and $\mathtt{skip}$ are identical. In our setting, Player is not frugal with his names and therefore the two terms above are identified only at the extensional level (i.e. after quotienting).[9]

The major difference between [Lai04] and [Lai08] lies in the modelling of (ground-type, name-storing) store. In [Lai04] the store is modelled by attaching to strategies a global, top-level (non-monadic), store-arena. Then, a good-store-discipline is imposed on strategies via extra conditions on strategy composition which enforce that hidden store-moves follow the standard read/write pattern. As a result (and in contrast to our model), the model relies heavily on quotienting by the intrinsic preorder in order for the store to work properly.

The added accuracy obtained by using frugality conditions is fully exploited in [Lai08], where a carefully formulated setting of moves-with-store[10] allows for an *explicit characterisation* result, that is, a semantic characterisation of operational equality at the intensional level. The contribution of using moves-with-store in that result is that thus the semantics is relieved from the (too revealing) internal workings of store: for example, terms like $(a := b)\,;\lambda x.!\,a\,;0$ and $(a := b)\,;\lambda x.0$ are equated semantically at the intensional level, in contrast to what happens in our model.[11] Note, though, that in a setting with higher-order store such that of $\nu\rho$, moves-with-store would not be as simple since stores would need to store higher-order values, that is, strategies.

Laird's approach is therefore advantageous in its use of name-frugality conditions, which allow for more accurate models. At the same time, though, frugality conditions are an extra burden in constructing a model: apart from the fact that they need to be dynamically preserved in play-composition by garbage collection, they presuppose an appropriately defined notion of *name-use*. In [Lai04, Lai08], a name is considered as used in a play if it is accessible through the store (in a reflexive transitive manner) from a name that has been *explicitly played*. This definition, however, does not directly apply to languages with different nominal effects (e.g. higher-order store). Moreover, frugality alone is not enough for languages like Reduced ML [Sta94] or the ν-calculus: a name may have been used in a play but may still be inaccessible to some participant (e.g. if it is outside his view [MT09]). On the other hand, our approach is advantageous in its simplicity and its applicability on a wide rage of nominal effects, but suffers from the accuracy issues discussed above.

[8] Non-innocence can be seen as beneficial in terms of simplicity of the model, since strategies then have one condition less. On the other hand, though, innocent strategies are specified by means of their viewfunctions, which makes their presentation simpler. Moreover, non-innocence diminishes the power of definability results, as finitary behaviours are less expressive in the absence of innocence.

[9] Note here, though, that the semantics being too explicit about the created names can prove beneficial: here we are able to give a particularly concise proof adequacy for $\nu\rho$ (see section 4.3.4 and compare e.g. with respective proof in [AHM98]) by exploiting precisely this extra information!

[10] Inter alia, frugality of names implies that sequences of moves-with-store have strong support even if stores are represented by sets!

[11] In our model they correspond to the strategies (see also section 4.3):

$$\sigma_1 \triangleq \{[(a,b) * \circledast(*,\circledast)(n,\circledast)\, a\, c\, 0]\}\,, \quad \sigma_2 \triangleq \{[(a,b) * \circledast(*,\circledast)(n,\circledast)\, 0]\}\,.$$

Thus, the inner-workings of the store revealed by σ_1 (i.e. the moves $a\,c$) differentiate it from σ_2. In fact, in our attempts to obtain an explicit characterisation result from our model, we found store-related inaccuracies to be the most stubborn ones.

Chapter 4

Nominal References

In this chapter we construct in nominal games a fully abstract semantics for a language with nominal general references called the *$\nu\rho$-calculus*. General references are references which can store not only values of ground type (integers, booleans, etc.) but also of higher-order type (procedures, higher-order functions) or references themselves. They constitute a very powerful and useful programming construct, allowing for the encoding of a wide range of computational effects and programming paradigms (e.g. object-oriented programming [AHM98, section 2.3] or aspect-oriented programming [SO07]). The denotational modelling of higher-order references is quite demanding since, on top of phenomena of dynamic update and interference, one has to cope with the inherent cyclicity of higher-order storage.

The $\nu\rho$-calculus is a functional language with dynamically allocated general references, reference-equality tests and "good variables", which faithfully reflects the practice of real programming languages such as ML [MTM97]. In particular, it extends the $s\nu$-calculus by using names for general references. In terms of the *What's new?* motto (cf. [PS93]), names can be

> *created with local scope, updated and dereferenced, tested for equality and passed around via function application, but that is all.*

The fully abstract model of $\nu\rho$ is the first such for a language with general references and good variables.

Fully abstract models for general references were given via game semantics in [AHM98] and via abstract categorical semantics (and games) in [Lai02]. Neither approach used names. The model of [AHM98] is based on the idea of relaxing strategy conditions in order to model computational effects. In particular, it models references as variables of a read/write product type and it uses strategies which violate visibility in order to use values assigned to references previously in a play. The synchronisation of references is managed by *cell strategies* which model fresh-reference creation. Because references are modelled by products, and in order to produce a fully abstract semantics, the examined language needs to include *bad variables*, which in turn yield unwanted behaviours affecting severely the expressivity of the language, and prohibit the use of equality tests for references.[1] On the other hand, the approach in [Lai02] bypasses the bad-variables problem by not including types for references (variables and references of the same type coincide). This contributes new intuitions on sequential categorical behaviour (*sequoidal category*), but we think that is somehow distanced from the common notion of reference in functional programming.

The full-abstraction problem has also been tackled via trace semantics in [Lai07]. The language examined is a version of that in [AHM98] without bad variables. The latter are not needed since the modelling of references is achieved by names pointing to a store (which

[1] By "bad variables" we mean read/write constructs of reference type which are not references. They are necessary for obtaining full-abstraction in [AHM98] since read/write-product semantical objects may not necessarily denote references.

is analogous to our approach). Of relevance is also the fully abstract trace model for a language with nominal threads and nominal objects presented in [JR02]. An important difference between trace models and game models is that the former are defined operationally (i.e. traces are computed by using the operational semantics), whereas game models are defined in a purely compositional manner. Nonetheless, trace models and game models have many similarities, deriving mainly from their sequential-interactive representation of computation, and in particular there are connections between [Lai07] and the work herein that should be further examined.

The chapter is structured as follows. In section 4.1 we introduce the $\nu\rho$-calculus and define its notion of observational equivalence, which yields the equational theory of the language. We then proceed to its denotational semantics by first formulating a fully abstract categorical semantics in section 4.2. The semantics is built in *$\nu\rho$-models*: these are categories equipped with a collection of comonads for initial state, and a monad for fresh-name creation and storage (cf. section 2.3.7). Finally, in section 4.3 we construct a concrete such model in nominal games. Working in the category $\mathcal{V}_{\mathtt{t}}$ (definition 3.51), we first obtain a $\nu\rho$-model by using the monadic-comonadic setting of section 3.4 and attaching to it a store monad (cf. section 2.3.3). For the latter we use a store arena ξ, which is obtained as the solution of a recursive Store Equation (SE). The model in $\mathcal{V}_{\mathtt{t}}$ is sound but not complete, because game strategies are allowed a 'liberal' use of the store. This is resolved by introducing *tidiness conditions* for strategies, by which we obtain a subcategory $\mathcal{T}$ of nominal games for which we show definability and full abstraction. Note that the whole approach can be straightforwardly adapted to ground-type references, thus giving e.g. a fully abstract model for Reduced ML [Sta94].

4.1 The $\nu\rho$-calculus

The syntax of the language which we now introduce is built inside the category **Nom** of nominal sets. Names are used for general references, so we assume that there is a set of names (atoms) $\mathbb{A}_A \in (\mathbb{A}_i)_{i\in\omega}$, for each type A in the language. Types include types for commands, naturals and references, product types and arrow types.

Definition 4.1 The $\nu\rho$-calculus is a functional calculus of nominal general references. Its types, terms and values are given as follows.

$$\mathrm{TY} \ni A, B ::= \mathbb{1} \mid \mathbb{N} \mid [A] \mid A \to B \mid A \times B$$

$\mathrm{TE} \ni M, N ::=$	$x \mid \lambda x.M \mid M\,N \mid \langle M, N\rangle \mid \mathtt{fst}\,M \mid \mathtt{snd}\,N$	λ-calculus
	$\mid \mathtt{skip} \mid n \mid \mathtt{pred}\,M \mid \mathtt{succ}\,N$	return/arithmetic
	$\mid \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2$	if_then_else
	$\mid a$	reference to type A ($a \in \mathbb{A}_A$)
	$\mid [M = N]$	name-equality test
	$\mid \nu a.M$	ν-abstraction
	$\mid M := N$	update
	$\mid\ !M$	dereferencing

$$\mathrm{VA} \ni V, W ::= n \mid \mathtt{skip} \mid a \mid x \mid \lambda x.M \mid \langle V, W\rangle$$

The typing system involves terms in environments $\vec{a} \mid \Gamma$, where $\vec{a}$ a list of (distinct) names and Γ a finite set of variable-type pairs. Typing rules are given in figure 4.1. ▲

As in the case of the strong ν-calculus previously, we note that TE and VA are strong nominal sets, and that terms are equated up to α-equivalence. The operational semantics of the calculus naturally involves computation in some ***store environment*** where created names

$$\frac{}{\vec{a} \mid \Gamma \vdash n : \mathbb{N}} \qquad \frac{}{\vec{a} \mid \Gamma, x{:}A \vdash x : A} \qquad \frac{}{\vec{a} \mid \Gamma \vdash \mathtt{skip} : \mathbb{1}}$$

$$\frac{\vec{a} \mid \Gamma \vdash M : A \times B}{\vec{a} \mid \Gamma \vdash \mathtt{fst}\, M : A} \qquad \frac{\vec{a} \mid \Gamma \vdash M : A \times B}{\vec{a} \mid \Gamma \vdash \mathtt{snd}\, M : B} \qquad \frac{\vec{a} \mid \Gamma \vdash M : A \qquad \vec{a} \mid \Gamma \vdash N : B}{\vec{a} \mid \Gamma \vdash \langle M, N \rangle : A \times B}$$

$$\frac{\vec{a} \mid \Gamma \vdash M : \mathbb{N}}{\vec{a} \mid \Gamma \vdash \mathtt{pred}\, M : \mathbb{N}} \qquad \frac{\vec{a} \mid \Gamma \vdash M : \mathbb{N}}{\vec{a} \mid \Gamma \vdash \mathtt{succ}\, M : \mathbb{N}} \qquad \frac{\vec{a} \mid \Gamma \vdash M : \mathbb{N} \qquad \vec{a} \mid \Gamma \vdash N_1, N_2 : A}{\vec{a} \mid \Gamma \vdash \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 : A}$$

$$\frac{\vec{a} \mid \Gamma, x{:}A \vdash M : B}{\vec{a} \mid \Gamma \vdash \lambda x.M : A \to B} \qquad \frac{\vec{a} \mid \Gamma \vdash M : A \to B \qquad \vec{a} \mid \Gamma \vdash N : A}{\vec{a} \mid \Gamma \vdash M\, N : B}$$

$$\frac{}{\vec{a} \mid \Gamma \vdash a : [A]}\ {}^{a \in \mathbb{A}_A}_{\wedge a \in \vec{a}} \qquad \frac{\vec{a}a \mid \Gamma \vdash M : B}{\vec{a} \mid \Gamma \vdash \nu a.M : B} \qquad \frac{\vec{a} \mid \Gamma \vdash M : [A] \qquad \vec{a} \mid \Gamma \vdash N : [A]}{\vec{a} \mid \Gamma \vdash [M = N] : \mathbb{N}}$$

$$\frac{\vec{a} \mid \Gamma \vdash M : [A]}{\vec{a} \mid \Gamma \vdash\ !M : A} \qquad \frac{\vec{a} \mid \Gamma \vdash M : [A] \qquad \vec{a} \mid \Gamma \vdash N : A}{\vec{a} \mid \Gamma \vdash M := N : \mathbb{1}}$$

Figure 4.1: The $\nu\rho$-calculus: typing rules.

have their values stored. Formally, we define store environments S to be lists of the form:

$$S ::= \epsilon \mid a, S \mid a :: V, S\,. \tag{4.1}$$

Observe that the store may include names that have been created but remain as yet unassigned a value. For each store environment S we define its domain to be the name-list given by:

$$\mathtt{dom}(\epsilon) \triangleq \epsilon\,, \quad \mathtt{dom}(a, S) \triangleq a, \mathtt{dom}(S)\,, \quad \mathtt{dom}(a :: V, S) \triangleq a, \mathtt{dom}(S)\,. \tag{4.2}$$

We only consider environments whose domains are lists of distinct names. We write $S \models_{\Gamma, A} M$, or simply $S \models M$, only if $\mathtt{dom}(S) \mid \Gamma \vdash M : A$ is valid (i.e. derivable).

Definition 4.2 The operational semantics is given in terms of a small-step reduction relation, the rules of which are given in figure 4.2. Evaluation contexts E are of the forms:

$$(\lambda x.N)\,\_\,,\ \_\,N\,,\ \mathtt{fst}\,\_\,,\ \mathtt{snd}\,\_\,,\ \langle \_, N \rangle\,,\ \langle V, \_ \rangle\,,\ \mathtt{if0}\,\_\,\mathtt{then}\,N_1\,\mathtt{else}\,N_2\,,$$
$$\mathtt{pred}\,\_\,,\ \mathtt{succ}\,\_\,,\ [\_ = N]\,,\ [a = \_]\,,\ !\_\,,\ \_ := N\,,\ a := \_\,.$$

▲

We can see that $\nu\rho$ is not strongly normalising with the following example. Recall the standard CBV encoding of sequencing:

$$M\,;N \triangleq (\lambda z.N)M \tag{4.3}$$

with $z \notin \mathtt{fv}(N)$.

Example 4.3 For each type A, take

$$\mathtt{stop}_A \triangleq \nu b.(b := \lambda x.(!\,b)\mathtt{skip})\,;(!\,b)\mathtt{skip}$$

with $b \in \mathbb{A}_{\mathbb{1} \to A}$. We can see that $\mathtt{stop}_A$ diverges, since:

$$\begin{aligned} \models \mathtt{stop}_A &\longrightarrow\!\!\!\!\!\rightarrow\ b :: \lambda x.(!\,b)\mathtt{skip} \models (!\,b)\mathtt{skip} \\ &\longrightarrow\ b :: \lambda x.(!\,b)\mathtt{skip} \models (\lambda x.(!\,b)\mathtt{skip})\mathtt{skip} \\ &\longrightarrow\ b :: \lambda x.(!\,b)\mathtt{skip} \models (!\,b)\mathtt{skip}\,. \end{aligned}$$

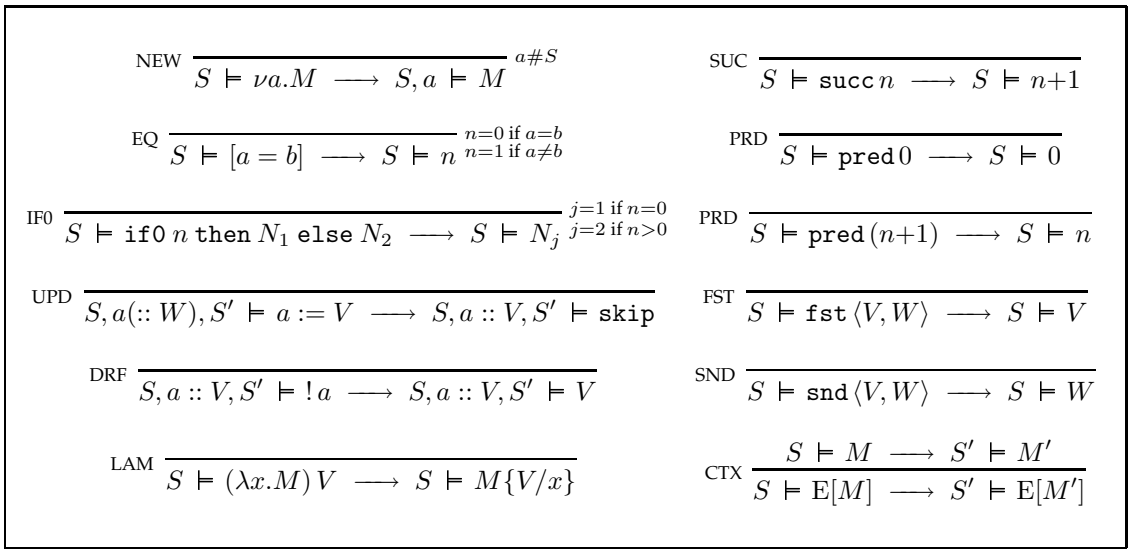


Figure 4.2: The $\nu\rho$-calculus: reduction rules.

Moreover, taking $a \in \mathbb{A}_A$ we have

$$\vDash \nu a.!\, a \longrightarrow a \vDash !\, a$$

and no further reductions are possible. In section 4.3.8 we will show that $\nu a.!\, a$ and $\mathtt{stop}_A$ are observationally equivalent (definition 4.7). Note, though, that the two terms correspond to different kinds of divergence, which are indistinguishable in $\nu\rho$: while $\nu a.!\, a$ stands for *deadlock*, $\mathtt{stop}_A$ stands for *livelock*. ■

The great expressive power of general references is seen in the fact that we can encode the **Y** combinator. The following example is adapted from [AHM98].

Example 4.4 Taking $a \in \mathbb{A}_{A\to A}$, define:

$$\mathbf{Y}_A \triangleq \lambda f.\nu a.(a := \lambda x.f(!\, a)x)\,;!\, a\,.$$

$\mathbf{Y}_A$ has type $((A \to A) \to A \to A) \to A \to A$ and, for any relevant term M and value V, we have:

$$\vDash (\lambda y.My)(\mathbf{Y}_A(\lambda y.My))V \longrightarrow\!\!\!\!\!\rightarrow\ a :: \lambda x.(\lambda y.My)(!\, a)x \vDash (\lambda y.My)(!\, a)V\,,$$

$$\begin{aligned} \vDash (\mathbf{Y}_A(\lambda y.My))V &\longrightarrow\!\!\!\!\!\rightarrow\ a :: \lambda x.(\lambda y.My)(!\, a)x \vDash (!\, a)V \\ &\longrightarrow\ a :: \lambda x.(\lambda y.My)(!\, a)x \vDash (\lambda x.(\lambda y.My)(!\, a)x)V \\ &\longrightarrow\ a :: \lambda x.(\lambda y.My)(!\, a)x \vDash (\lambda y.My)(!\, a)V\,. \end{aligned}$$

■

Contexts in $\nu\rho$ are generally more complicated than the evaluation contexts in the previous definition. Intuitively, contexts are "terms with a (single) hole", yet this clause leaves many details implicit. In particular, contexts transform not only the syntax of the term, but also its typing environment. We formalise this by use of *typed contexts*.

Definition 4.5 (Contexts) $\nu\rho$-contexts are defined as follows.

$$\begin{aligned} \mathrm{CT} \ni \mathrm{C} ::= {} & \_ \mid \lambda x.\mathrm{C} \mid \mathrm{C}\, N \mid M\, \mathrm{C} \mid [\mathrm{C} = N] \mid [M = \mathrm{C}] \mid \mathrm{C} := N \mid M := \mathrm{C} \mid\ !\,\mathrm{C} \\ & \mid \mathtt{if0}\ \mathrm{C}\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 \mid \mathtt{if0}\ M\ \mathtt{then}\ \mathrm{C}\ \mathtt{else}\ N_2 \mid \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ \mathrm{C} \\ & \mid \langle \mathrm{C}, N\rangle \mid \langle M, \mathrm{C}\rangle \mid \mathtt{fst}\,\mathrm{C} \mid \mathtt{snd}\,\mathrm{C} \mid \mathtt{pred}\,\mathrm{C} \mid \mathtt{succ}\,\mathrm{C} \mid \nu a.\mathrm{C} \end{aligned}$$

A ***basic context*** is a context which does not contain a subcontext of the form $\nu a.\mathrm{C}$.
Context types are of the form $(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',A')$, where $\vec{a}\,|\,\Gamma,\vec{a}'\,|\,\Gamma'$ are typing environments and $A,A'\in\mathrm{TY}$. Typing rules for contexts follow those for terms:

$$\frac{}{\_:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',A)}\;{}^{\vec{a}\subseteq\vec{a}'}_{\wedge\,\Gamma\subseteq\Gamma'} \qquad \frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',\mathbb{N}) \qquad \vec{a}'\mid\Gamma'\vdash N_1,N_2:B}{\mathtt{if0}\;\mathrm{C}\;\mathtt{then}\;N_1\;\mathtt{else}\;N_2:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B)}$$

$$\frac{\vec{a}'\mid\Gamma'\vdash M:\mathbb{N} \qquad \mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B) \qquad \vec{a}'\mid\Gamma'\vdash N_2:B}{\mathtt{if0}\;M\;\mathtt{then}\;\mathrm{C}\;\mathtt{else}\;N_2:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B)}$$

$$\frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B\times C)}{\mathtt{fst}\,\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B)} \qquad \frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B) \qquad \vec{a}'\mid\Gamma'\vdash N:C}{\langle\mathrm{C},N\rangle:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B\times C)}$$

$$\frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',\mathbb{N})}{\mathtt{pred}\,\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',\mathbb{N})} \qquad \frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B\to C) \qquad \vec{a}'\mid\Gamma'\vdash N:B}{\mathrm{C}\,N:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',C)}$$

$$\frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma'\uplus\{x{:}B\},C)}{\lambda x.\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B\to C)} \qquad \frac{\vec{a}'\mid\Gamma'\vdash M:B\to C \qquad \mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B)}{M\,\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',C)}$$

$$\frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}'a,\Gamma',B)}{\nu a.\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B)} \qquad \frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',[B]) \qquad \vec{a}'\mid\Gamma'\vdash N:[B]}{[\mathrm{C}=N]:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',\mathbb{N})}$$

$$\frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',[B])}{!\,\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',B)} \qquad \frac{\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',[B]) \qquad \vec{a}'\mid\Gamma'\vdash N:B}{\mathrm{C}:=N:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',\mathbb{1})}$$

(plus omitted counterparts). ▲

Holes in contexts are denoted by "$\_$" (elsewhere, they are usually denoted by "$[\_]$"). In a context of type $(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',A')$ the type of its hole is $(\vec{a},\Gamma,A)$, and $(\vec{a}',\Gamma',A')$ is the resulting type. The first typing rule above states that if $\_$ has type $(\vec{a},\Gamma,A)$ then the type of the context $\_$ may have more names or more variables. This allows us to consider typed terms $\vec{a}\mid\Gamma\vdash M:A$ in all compatible contexts. Finally, note that contexts are *not* equated up to α-equivalence.

We now proceed to context-instantiations. Note below that by $M^{+(\vec{a}'\setminus\vec{a})}$ we mean the typed term $\vec{a}'\mid\Gamma\vdash M:A$.

Definition 4.6 If $\vec{a}\mid\Gamma\vdash M:A$ a typed term and $\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\vec{a}',\Gamma',A')$ then we define the instantiation $\mathrm{C}[M]$ by induction as follows.

$$\begin{aligned}
_[M] &\triangleq M^{+(\vec{a}'\setminus\vec{a})}\\
(\nu a.\mathrm{C})[M] &\triangleq \nu a.\mathrm{C}[M]\\
(\mathrm{C}\,N)[M] &\triangleq \mathrm{C}[M]\,N\\
(\lambda x.\mathrm{C})[M] &\triangleq \lambda x.\mathrm{C}[M] \qquad \ldots
\end{aligned}$$

▲

The type of observables can be any base type; here we take it to be $\mathbb{N}$, as the latter is present in all the languages we examine. Around observable terms we build the notion of observational equivalence: two terms are equivalent if, whenever they are put inside a variable- and name-closing context of resulting type $\mathbb{N}$, usually called a ***program context***, they reduce to the same observable term.

Definition 4.7 For typed terms $\vec{a}\mid\Gamma\vdash M:A$ and $\vec{a}\mid\Gamma\vdash N:A$, define

$$\vec{a}\mid\Gamma\vdash M\lessapprox N \iff \forall\mathrm{C}.\,(\exists S'.\ \Vdash\mathrm{C}[M]\longrightarrow\!\!\!\!\!\rightarrow\ S'\Vdash 0)\implies(\exists S''.\ \Vdash\mathrm{C}[N]\longrightarrow\!\!\!\!\!\rightarrow\ S''\Vdash 0)$$

where $\mathrm{C}:(\vec{a},\Gamma,A)\mapsto(\epsilon,\varnothing,\mathbb{N})$. Moreover, $\approx\;\triangleq\;\lessapprox\cap\gtrapprox$. ▲

Usually we omit $\vec{a}$ and Γ and write simply $M \lessapprox N$.

Let us examine some examples in observational equivalence which are suggestive of the expressivity of the $\nu\rho$-calculus. Let us introduce the following abbreviation which compares two terms of type $\mathbb{N}$ as booleans. For any pair of terms $M, N : \mathbb{N}$ take

$$[M \Leftrightarrow N] \triangleq \texttt{if0}\ M\ \texttt{then}\ N\ \texttt{else}\ (\texttt{if0}\ N\ \texttt{then}\ 1\ \texttt{else}\ 0)\,.$$

Then, taking

$$\begin{aligned} M_1 &\triangleq \lambda f.\,0 : ([\mathbb{1}] \to \mathbb{N}) \to \mathbb{N} \\ M_2 &\triangleq \lambda f.\,\nu a.\nu b.[fa \Leftrightarrow fb] : ([\mathbb{1}] \to \mathbb{N}) \to \mathbb{N} \\ M_3 &\triangleq \lambda f.\,\nu a.[fa \Leftrightarrow fa] : ([\mathbb{1}] \to \mathbb{N}) \to \mathbb{N} \\ M_4 &\triangleq \lambda f.\,\texttt{stop} : (\mathbb{1} \to \mathbb{1}) \to \mathbb{1} \\ M_5 &\triangleq \lambda f.\,f\,\texttt{skip}\,;\texttt{stop} : (\mathbb{1} \to \mathbb{1}) \to \mathbb{1} \end{aligned} \tag{4.4}$$

we have the following equivalences and inequivalences.

$$M_1 \not\approx M_2 \tag{4.5}$$

$$M_2 \not\approx M_3 \tag{4.6}$$

$$M_4 \approx M_5 \tag{4.7}$$

(4.7) will be established by semantical means is section 4.3.8. For (4.5) we can use a context that is sensitive to the fact that f has been applied to an argument. This can be easily achieved by supplying an f that updates the store, as e.g. in

$$\mathrm{C} \triangleq \nu c.c := 2\,;\_(\lambda x.c := \texttt{pred}\,!\,c\,;!\,c)\,.$$

However, the intention behind the comparison of the two terms was to establish whether f could distinguish between the two fresh names, or it would return the same result in both cases; in this sense, the choice of M_1 is not adequate in a calculus with side-effects. More to the point is the comparison between M_2 an M_3. It turns out that $\nu\rho$ can distinguish between them, e.g. by taking

$$\mathrm{C} \triangleq \nu c.\nu d.c := d\,;\_(\lambda x.\texttt{if0}\ [x = !\,c]\ \texttt{then}\ 0\ \texttt{else}\ c := x\,;1)$$

where $c \in \mathbb{A}_{[1]}$ and $d \in \mathbb{A}_1$. We can see that the context can *remember* the fresh name a after the first time it encounters it, so in particular it can distinguish it from the fresh b.

Regarding the comparison to the $s\nu$-calculus, we note that both (4.5) and (4.6) are equivalences in $s\nu$ (and (4.7) is irrelevant because of termination). The former can be shown using logical relations [Sta94, Chapter 4][2] while the latter is established for the $\nu\varepsilon$-calculus in section 5.2.6.

4.2 Semantics

We now examine sufficient conditions for a fully abstract semantics of $\nu\rho$ in an abstract categorical setting. Our aim is to construct fully abstract models in an appropriate categorical setting, pinpointing the parts of structure needed for such a task. In section 4.3 we will apply this knowledge in constructing a concrete such model in nominal games.

Translating each term M into an object $[\![M]\!]$ and assuming a preorder "$\lesssim$" in the semantics, full-abstraction amounts to the assertion:

$$M \lessapprox N \iff [\![M]\!] \lesssim [\![N]\!]\,. \tag{FA}$$

Notice that this formulation does not coincide with the full-abstraction specification given previously in the introduction, i.e. with

$$M \approx N \iff [\![M]\!] = [\![N]\!]\,. \tag{4.8}$$

[2] Note that $M_1 \approx M_2$ is not the "hard" equivalence proven in [Sta94, BK08]: $\lambda f.\,0 \approx \nu a.\nu b.\lambda f.\,[fa \Leftrightarrow fb]$.

Nevertheless, once we achieve (FA) we can construct an *extensional model*, via a quotienting construction, for which (4.8) holds. Being a quotiented structure, the extensional model does not have an explicit, simple description, and for this reason we prefer working with the intensional model (i.e. the unquotiented one). Of course, an intensional model satisfying (4.8) would be preferred but this cannot be achieved in our nominal games. Therefore, our categorical models will be guided by the (FA) formulation.

4.2.1 Soundness

We proceed to present categorical models for the $\nu\rho$-calculus. The approach we take is monadic and comonadic, over a computational monad T and a family of local-state comonads $Q = (Q^{\vec{a}})_{\vec{a}\in\mathbb{A}^\#}$, so that the morphism related to each $\vec{a} \mid \Gamma \vdash M : A$ is of the form $[\![M]\!] : Q^{\vec{a}}[\![\Gamma]\!] \longrightarrow T[\![A]\!]$. Computation in $\nu\rho$ is store-update and fresh-name creation, so T is a store monad, while (initial) local state is given by product comonads.

Definition 4.8 A $\nu\rho$-model $\mathcal{M}$ is a structure $(\mathcal{M}, T, Q)$ such that:

I. $\mathcal{M}$ is a category with finite products, with 1 being the terminal object and $A \times B$ the product of A and B.

II. T is a strong monad (T, η, μ, τ) with exponentials.

III. $\mathcal{M}$ contains a natural numbers object $\mathbb{N}$ equipped with successor/predecessor arrows and $\tilde{n} : 1 \longrightarrow \mathbb{N}$, each $n \in \mathbb{N}$. Moreover, for each object A, there is an appropriate arrow for zero-equality tests $\mathtt{cnd}_A : \mathbb{N} \times TA \times TA \longrightarrow TA$.

IV. Q is a family of product comonads $(Q^{\vec{a}}, \varepsilon, \delta, \zeta)_{\vec{a}\in\mathbb{A}^\#}$ on $\mathcal{M}$ such that:

(a) the basis of Q^ϵ is 1, and $Q^{\vec{a}} = Q^{\vec{a}'}$ whenever $[\vec{a}] = [\vec{a}']$,

(b) if $\vec{a}' \subseteq \vec{a}$ then there exists a comonad morphism $\frac{\vec{a}}{\vec{a}'} : Q^{\vec{a}} \longrightarrow Q^{\vec{a}'}$ such that $\frac{\vec{a}}{\epsilon} = \varepsilon$, $\frac{\vec{a}}{\vec{a}} = \mathtt{id}$ and, whenever $\vec{a}' \subseteq \vec{a}'' \subseteq \vec{a}$,

$$\frac{\vec{a}}{\vec{a}''} ; \frac{\vec{a}''}{\vec{a}'} = \frac{\vec{a}}{\vec{a}'}$$

(c) for each $\vec{a}a \in \mathbb{A}^\#$ there exists a natural transformation $\mathtt{nu}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow TQ^{\vec{a}a}$ which is strength-coherent and, for each $A \in Ob(\mathcal{M})$ and $\vec{a}a \subseteq \vec{a}'a$, the following diagrams commute.

$$\begin{array}{ccc} Q^{\vec{a}}A & \xrightarrow{\langle \mathtt{id}, \mathtt{nu}_A\rangle} & Q^{\vec{a}}A \times TQ^{\vec{a}a}A \\ {\scriptstyle \mathtt{nu}_A}\downarrow & & \downarrow{\scriptstyle \tau} \\ TQ^{\vec{a}a}A & \xrightarrow[T\langle \frac{\vec{a}a}{\vec{a}}, \mathtt{id}\rangle]{} & T(Q^{\vec{a}}A \times Q^{\vec{a}a}A) \end{array} \qquad \begin{array}{ccc} Q^{\vec{a}'}A & \xrightarrow{\mathtt{nu}_A^{\vec{a}'a}} & TQ^{\vec{a}'a}A \\ {\scriptstyle \frac{\vec{a}'}{\vec{a}}}\downarrow & & \downarrow{\scriptstyle T\frac{\vec{a}'a}{\vec{a}a}} \\ Q^{\vec{a}}A & \xrightarrow[\mathtt{nu}_A^{\vec{a}a}]{} & TQ^{\vec{a}a}A \end{array} \qquad \text{(N2)}$$

V. Setting $\mathbb{A}_A \triangleq Q^a 1$, for each $a \in \mathbb{A}_A$, there is a name-equality arrow $\mathtt{eq}_A : \mathbb{A}_A \times \mathbb{A}_A \longrightarrow \mathbb{N}$ in $\mathcal{M}$ such that, for any distinct $a, b \in \mathbb{A}_A$, the following diagram commutes.

$$\begin{array}{ccccc} Q^a 1 & \xrightarrow{\Delta} & \mathbb{A}_A \times \mathbb{A}_A & \xleftarrow{\langle \frac{ab}{a}, \frac{ab}{b}\rangle} & Q^{ab}1 \\ {\scriptstyle !}\downarrow & & \downarrow{\scriptstyle \mathtt{eq}_A} & & \downarrow{\scriptstyle !} \\ 1 & \xrightarrow[\tilde{0}]{} & \mathbb{N} & \xleftarrow[\tilde{1}]{} & 1 \end{array} \qquad \text{(N1)}$$

VI. Setting $[\![\mathbb{1}]\!] \triangleq 1$, $[\![\mathbb{N}]\!] \triangleq \mathbb{N}$, $[\![[A]]\!] \triangleq \mathbb{A}_A$, $[\![A \to B]\!] \triangleq T[\![B]\!]^{[\![A]\!]}$, $[\![A \times B]\!] \triangleq [\![A]\!] \times [\![B]\!]$, $\mathcal{M}$ contains, for each $A \in \mathrm{TY}$, arrows

$$\mathtt{drf}_A : \mathbb{A}_A \longrightarrow T[\![A]\!] \quad \text{and} \quad \mathtt{upd}_A : \mathbb{A}_A \times [\![A]\!] \longrightarrow T1$$

such that the following diagrams commute,

$$\mathbb{A}_A \times [\![A]\!] \xrightarrow{\langle \mathtt{id}, \mathtt{upd}_A \rangle ; \tau ; \cong} T(\mathbb{A}_A \times [\![A]\!]) \underset{T\pi_2}{\overset{T(\pi_1 ; \mathtt{drf}_A) ; \mu}{\rightrightarrows}} T[\![A]\!]$$

$$\mathbb{A}_A \times [\![A]\!] \times [\![A]\!] \xrightarrow{\langle \mathtt{id}\times\pi_1 ; \mathtt{upd}_A, \mathtt{id}\times\pi_2 ; \mathtt{upd}_A \rangle} T1 \times T1 \underset{\pi_2}{\overset{\psi ; \cong}{\rightrightarrows}} T1 \tag{NR}$$

$$Q^{ab}1 \times [\![A]\!] \times [\![B]\!] \xrightarrow{\langle \frac{ab}{a}\times\pi_1 ; \mathtt{upd}_A, \frac{ab}{b}\times\pi_2 ; \mathtt{upd}_B \rangle} T1 \times T1 \underset{\psi' ; \cong}{\overset{\psi ; \cong}{\rightrightarrows}} T1$$

and, moreover,

$$(\mathtt{nu}_A^{\vec{a}a} \times \mathtt{upd}_B) ; \psi = (\mathtt{nu}_A^{\vec{a}a} \times \mathtt{upd}_B) ; \psi' . \tag{SNR}$$

i.e. updates and fresh names are independent effects. ▲

Strength-coherence for $\mathtt{nu}$ means that, for any pair of objects A, B, the following diagram commutes (note that we systematically avoid writing superscripts of $\mathtt{nu}$).

$$\begin{array}{ccc} A \times Q^{\vec{a}}B & \xrightarrow{\zeta} & Q^{\vec{a}}(A \times B) \\ \downarrow{\scriptstyle \mathtt{id}\times\mathtt{nu}_B} & & \downarrow{\scriptstyle \mathtt{nu}_{A\times B}} \\ A \times TQ^{\vec{a}a}B & \xrightarrow[\tau ; T\zeta]{} & TQ^{\vec{a}a}(A \times B) \end{array}$$

The above essentially states that, for each object A, $\mathtt{nu}_A$ can be expressed as:

$$Q^{\vec{a}}A \xrightarrow{\cong} Q^{\vec{a}}1 \times A \xrightarrow{\mathtt{nu}_1 \times \mathtt{id}} TQ^{\vec{a}a}1 \times A \xrightarrow{\tau'} T(Q^{\vec{a}a}1 \times A) \xrightarrow{\cong} TQ^{\vec{a}a}A$$

It is evident that the role reserved for $\mathtt{nu}$ in our semantics is *fresh-name creation*. Accordingly, $\mathtt{nu}$ gives rise to a categorical name-abstraction operation: for any arrow $f : Q^{\vec{a}a}A \longrightarrow TB$ in $\mathcal{M}$, we define

$$\langle a \rangle f \triangleq Q^{\vec{a}}A \xrightarrow{\mathtt{nu}_A} TQ^{\vec{a}a}A \xrightarrow{Tf} T^2B \xrightarrow{\mu} TB . \tag{4.9}$$

The (NR) diagrams give the basic equations for dereferencings and updates (cf. conditions on categorical models of Reduced ML [Sta94, section 5.8] and commutative diagrams of [PP02, definition 1]). The first diagram stipulates that by dereferencing an updated reference we get the value of the update. The second diagram ensures that the value of a reference is that of the last update: doing two consecutive updates to the same reference is the same as doing only the last one. The last diagram states that updates of distinct references are independent effects.

Let us now proceed with the semantics of $\nu\rho$ in $\nu\rho$-models.

Definition 4.9 Let $(\mathcal{M}, T, Q)$ be a $\nu\rho$-model. Recall the type translation:

$$[\![\mathbb{1}]\!] \triangleq 1 , \quad [\![\mathbb{N}]\!] \triangleq \mathbb{N} , \quad [\![[A]]\!] \triangleq \mathbb{A}_A , \quad [\![A \to B]\!] \triangleq T[\![B]\!]^{[\![A]\!]} , \quad [\![A \times B]\!] \triangleq [\![A]\!] \times [\![B]\!] .$$

A typed term is $\vec{a} \mid \Gamma \vdash M : A$ translated to an arrow

$$[\![M]\!]_{\vec{a} \mid \Gamma} : Q^{\vec{a}}[\![\Gamma]\!] \longrightarrow T[\![A]\!]$$

in $\mathcal{M}$, which we write simply as $[\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow TA$, as in figure 4.3. ▲

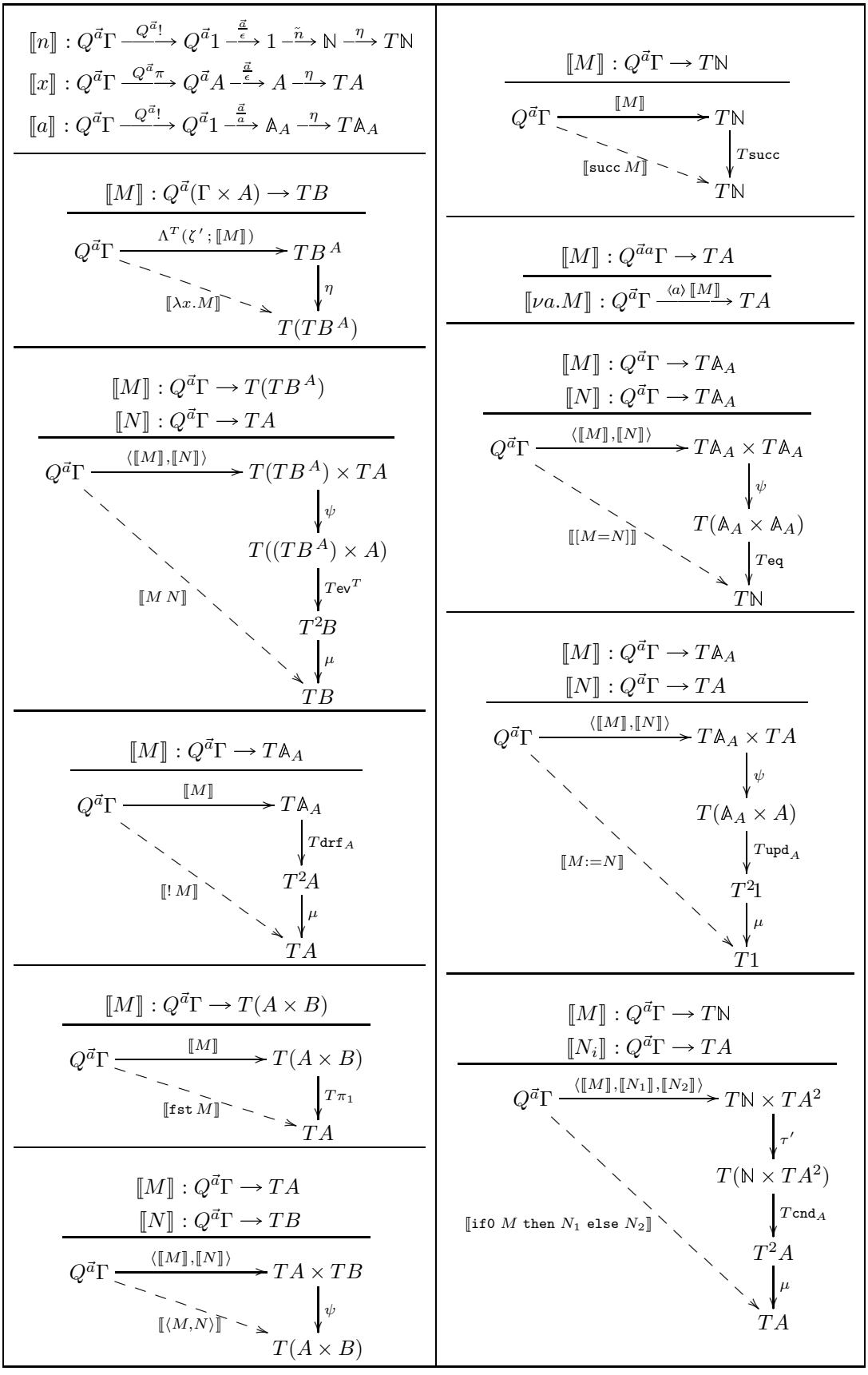


Figure 4.3: The semantic translation of $\nu\rho$.

Note that the translation of values follows a common pattern: for any $\vec{a} \mid \Gamma \vdash V : A$ we have $[\![V]\!] = |V|\,;\eta$, where

$$\begin{array}{rcl@{\qquad}rcl}
|x| & \triangleq & Q^{\vec{a}}\Gamma \xrightarrow{Q^{\vec{a}}\pi} Q^{\vec{a}}A \xrightarrow{\frac{\vec{a}}{\epsilon}} A & |n| & \triangleq & Q^{\vec{a}}\Gamma \xrightarrow{Q^{\vec{a}}!} Q^{\vec{a}}1 \xrightarrow{\frac{\vec{a}}{\epsilon}} 1 \xrightarrow{\tilde{n}} \mathbb{N} \\
|a| & \triangleq & Q^{\vec{a}}\Gamma \xrightarrow{Q^{\vec{a}}!} Q^{\vec{a}}1 \xrightarrow{\frac{\vec{a}}{a}} \mathbb{A}_A & |\mathtt{skip}| & \triangleq & Q^{\vec{a}}\Gamma \xrightarrow{Q^{\vec{a}}!} Q^{\vec{a}}1 \xrightarrow{\frac{\vec{a}}{\epsilon}} 1 \\
|\lambda x.M| & \triangleq & Q^{\vec{a}}\Gamma \xrightarrow{\Lambda^T(\zeta\,';\,[\![M]\!])} TB^A & |\langle V, W\rangle| & \triangleq & Q^{\vec{a}}\Gamma \xrightarrow{\langle |V|,\,|W|\rangle} A \times B\,.
\end{array} \tag{4.10}$$

Some first lemmas we can show are the following. The proofs are not difficult, and are deferred to the appendix.

Lemma 4.10 *For any $\vec{a} \mid \Gamma \vdash M : A$ and $\vec{a} \subseteq \vec{a}'$,*

$$[\![M]\!]_{\vec{a}'|\Gamma} = Q^{\vec{a}'}\Gamma \xrightarrow{\frac{\vec{a}'}{\vec{a}}} Q^{\vec{a}}\Gamma \xrightarrow{[\![M]\!]_{\vec{a}|\Gamma}} TA\,.$$

Moreover, if $\Gamma = x_1 : B_1, ..., x_n : B_n$ and $\vec{a} \mid \Gamma \vdash M : A$ and $\vec{a} \mid \Gamma \vdash V_i : B_i$ are derivable, then

$$[\![M\{\vec{V}/\vec{x}\}]\!] = Q^{\vec{a}}\Gamma \xrightarrow{\langle \mathtt{id},\,|V_1|,\,...,\,|V_n|\rangle} Q^{\vec{a}}\Gamma \times \Gamma \xrightarrow{\zeta\,';\,Q^{\vec{a}}\pi_2} Q^{\vec{a}}\Gamma \xrightarrow{[\![M]\!]} TA\,.$$

■

Lemma 4.11 *For any relevant f, g,*

$$\langle a \rangle \left(Q^{\vec{a}a}A \xrightarrow{\langle f,\,\frac{\vec{a}a}{\vec{a}}\,;\,g\rangle} TB \times TC \xrightarrow{\psi} T(B \times C) \right) = Q^{\vec{a}}A \xrightarrow{\langle \langle a\rangle f,\,g\rangle} TB \times TC \xrightarrow{\psi} T(B \times C)\,,$$

$$\langle a \rangle \left(Q^{\vec{a}a}A \xrightarrow{f} TB \xrightarrow{Tg} T^2C \xrightarrow{\mu} TC \right) = Q^{\vec{a}}A \xrightarrow{\langle a\rangle f} TB \xrightarrow{Tg} T^2C \xrightarrow{\mu} TC\,.$$

■

Lemma 4.12 *Let $\vec{a} \mid \Gamma \vdash M : A$ and $\vec{a} \mid \Gamma \vdash \mathrm{E}[M] : B$ be derivable, with E being an evaluation context. Then $[\![\mathrm{E}[M]]\!]$ is equal to:*

$$Q^{\vec{a}}\Gamma \xrightarrow{\langle \mathtt{id},\,[\![M]\!]\rangle} Q^{\vec{a}}\Gamma \times TA \xrightarrow{\tau} T(Q^{\vec{a}}\Gamma \times A) \xrightarrow{T\zeta\,'} TQ^{\vec{a}}(\Gamma \times A) \xrightarrow{T[\![\mathrm{E}[x]]\!]} T^2B \xrightarrow{\mu} TB\,.$$

■

We write $S \vDash M \xrightarrow{\mathrm{r}} S' \vDash M'$ with $\mathrm{r} \in \{$LAM,NEW,IF0,...,PRD,UPD,DRF$\}$ if the last non-CTX rule in the related derivation is r. Also, to any store S, we relate a term $\bar{S}$ of type $\mathbb{1}$ by setting:

$$\bar{\epsilon} \triangleq \mathtt{skip}, \quad \overline{a, S} \triangleq \bar{S}, \quad \overline{a :: V, S} \triangleq (a := V\,;\bar{S})\,.$$

We can show the following.

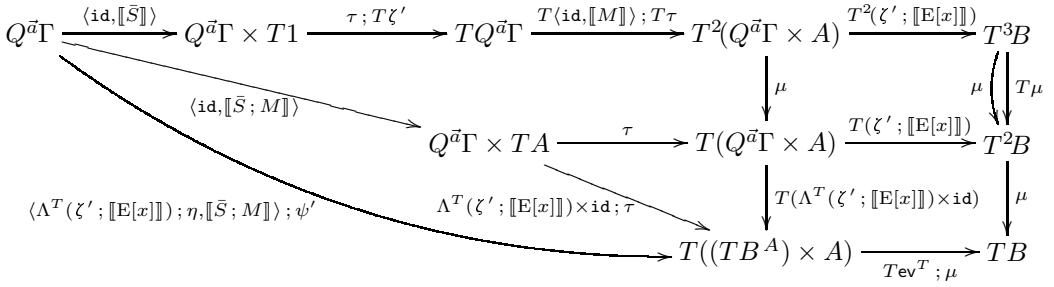
Proposition 4.13 (Correctness) *For any typed term $\vec{a} \mid \Gamma \vdash M : A$, and S with $\mathtt{dom}(S) = \vec{a}$, and* r *as above,*

1. *if* $\mathrm{r} \notin \{\mathrm{NEW}, \mathrm{UPD}, \mathrm{DRF}\}$ *then* $S \vDash M \xrightarrow{\mathrm{r}} S \vDash M' \implies [\![M]\!] = [\![M']\!]$,
2. *if* $\mathrm{r} \in \{\mathrm{UPD}, \mathrm{DRF}\}$ *then* $S \vDash M \xrightarrow{\mathrm{r}} S' \vDash M' \implies [\![\bar{S}\,;M]\!] = [\![\bar{S}'\,;M']\!]$,
3. $S \vDash M \xrightarrow{\mathrm{NEW}} S, a \vDash M' \implies [\![\bar{S}\,;M]\!] = \langle a \rangle [\![\bar{S}\,;M']\!]$.

Therefore, $S \vDash M \longrightarrow\!\!\!\!\!\rightarrow S' \vDash M' \implies [\![\bar{S}\,;M]\!] = \langle \vec{a}' \rangle [\![\bar{S}'\,;M']\!]$, *with* $\mathtt{dom}(S') = \vec{a}\vec{a}'$.

Proof: The last assertion follows easily from 1-3. For 1-3 we do induction on the size of the reduction's derivation. The base case follows from the specifications of definition 4.8 and

the penultimate lemma. For the inductive step we have that, for any S, M, E, the following diagram commutes.



By the previous lemma, the upper path is equal to $\langle \mathtt{id}, [\![\bar{S}]\!] \rangle\,;\tau\,;T\,\bar{\zeta}'\,;T[\![\mathrm{E}[M]]\!]\,;\mu$ and therefore to $[\![\bar{S}\,;\mathrm{E}[M]]\!]$. Hence, we can immediately show the inductive steps of 1-2. For 3, assuming $S \vDash \mathrm{E}[M] \xrightarrow{\text{NEW}} S, a \vDash \mathrm{E}[M']$ and $[\![\bar{S}\,;M]\!] = \langle a \rangle [\![\bar{S}\,;M']\!]$, we have, using also lemmas 4.10 and 4.11,

$$\begin{aligned}
\langle a \rangle [\![\bar{S}\,;\mathrm{E}[M']]\!] &= \langle a \rangle(\langle \Lambda^T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\eta, [\![\bar{S}\,;M']\!] \rangle\,;\psi'\,;T\mathsf{ev}^T\,;\mu) \\
&= \langle a \rangle(\langle \Lambda^T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\eta, [\![\bar{S}\,;M']\!] \rangle\,;\psi')\,;T\mathsf{ev}^T\,;\mu \\
&= \langle \Lambda^T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\eta, \langle a \rangle [\![\bar{S}\,;M']\!] \rangle\,;\psi'\,;T\mathsf{ev}^T\,;\mu \\
&= \langle \Lambda^T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\eta, [\![\bar{S}\,;M]\!] \rangle\,;\psi'\,;T\mathsf{ev}^T\,;\mu = [\![\bar{S}\,;\mathrm{E}[M]]\!]\,.
\end{aligned}$$

■

Our next target is to show soundness of the translation. Having proved correctness we only need computational adequacy, which we add explicitly as a specification to our models.

Definition 4.14 Let $\mathcal{M}$ be a $\nu\rho$-model and $[\![\_]\!]$ the respective translation of $\nu\rho$. $\mathcal{M}$ is ***adequate*** if, for any typed term $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$,

$$\exists S, \vec{b}.\, [\![M]\!] = \langle \vec{b} \rangle [\![\bar{S}\,;0]\!] \implies \exists S'.\, \vec{a} \vDash M \longrightarrow\!\!\!\!\!\rightarrow S' \vDash 0\,.$$

▲

Assume now our running $\mathcal{M}$ is an adequate $\nu\rho$-model.

Proposition 4.15 (Equational Soundness) *For terms* $\vec{a} \mid \Gamma \vdash M, N : A$,

$$[\![M]\!] = [\![N]\!] \implies M \lessapprox\!\gtrsim N\,.$$

Proof: Assume $[\![M]\!] = [\![N]\!]$ and $\vDash \mathrm{C}[M] \longrightarrow\!\!\!\!\!\rightarrow S' \vDash 0$. Then, by correctness, $[\![\mathrm{C}[M]]\!] = \langle \vec{a}' \rangle [\![\bar{S}'\,;0]\!]$, where $\vec{a}' = \mathtt{dom}(S')$. But $[\![M]\!] = [\![N]\!]$ implies $[\![\mathrm{C}[M]]\!] = [\![\mathrm{C}[N]]\!]$. Hence, by adequacy, there exists S'' such that $\vDash \mathrm{C}[N] \longrightarrow\!\!\!\!\!\rightarrow S'' \vDash 0$. ■

4.2.2 Completeness

The semantics needs to be equipped with a preorder to match the observational approximation preorder as in (FA). The chosen preorder is the intrinsic preorder with regard to some collection of observable arrows in the biKleisli monadic-comonadic setting (cf. definition 2.34). In particular, since we have a collection of monad-comonad pairs, we also need a collection of sets of observable arrows. Note that the observability conditions stipulated below are quite syntactic.

Definition 4.16 An adequate $\nu\rho$-model $\mathcal{M} = (\mathcal{M}, T, Q)$ is ***observational*** if, for all $\vec{a}$:

- there exists $O^{\vec{a}} \subseteq \mathcal{M}(Q^{\vec{a}}1, T\mathbb{N})$ such that, for all $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$,

$$[\![M]\!] \in O^{\vec{a}} \iff \exists S, \vec{b}.\, [\![M]\!] = \langle \vec{b} \rangle [\![\bar{S}\,;0]\!]\,,$$

- the induced intrinsic preorder $\lesssim \triangleq (\lesssim^{\vec{a}})_{\vec{a}\in\mathbb{A}^\#}$, defined on arrows in $\mathcal{M}(Q^{\vec{a}}A, TB)$ by

$$f \lesssim^{\vec{a}} g \iff^{\triangle} \forall \rho : Q^{\vec{a}}(TB^{\,A}) \longrightarrow T\mathbb{N}.\, (\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{\vec{a}}(g)\,;\rho \in O^{\vec{a}})\,,$$

  with $\Lambda^{\vec{a}}(f) \triangleq \Lambda^{Q^{\vec{a}},T}(f)$, satisfies, for all relevant $a, \vec{a}', f, f'$,

$$f \lesssim^{\vec{a}a} f' \implies \langle a\rangle f \lesssim^{\vec{a}} \langle a\rangle f' \quad \wedge \quad f \lesssim^{\vec{a}} f' \implies \tfrac{\vec{a}'}{\vec{a}}\,;f \lesssim^{\vec{a}'} \tfrac{\vec{a}'}{\vec{a}}\,;f'\,.$$

We write $\mathcal{M}$ as $(\mathcal{M}, T, Q, O)$. ▲

Recurring to our definition of $\Lambda^{Q^{\vec{a}},T}$ from chapter 2, we have that $\Lambda^{\vec{a}}(f)$ is the arrow:

$$Q^{\vec{a}}1 \xrightarrow{\delta} Q^{\vec{a}}Q^{\vec{a}}1 \xrightarrow{Q^{\vec{a}}\Lambda^T(\zeta'\,;f)} Q^{\vec{a}}(TB^{\,A})\,. \tag{4.11}$$

Hence, $O^{\vec{a}}$ contains those arrows that have a specific *observable behaviour* in the model, and the semantic preorder is built around this notion. In particular, terms that yield 0 have observable behaviour.

In order to make good use of the semantic preorder we need it to be a congruence with regard to the semantic translation. This is formalised as follows.

Definition 4.17 Let $\mathcal{M}$ be a $\nu\rho$-model and let $[\![\_]\!]$ be its semantic translation. For any $\vec{a}$, a preorder

$$R^{\vec{a}} \subseteq \bigcup_{A,B\in Ob(\mathcal{M})} \mathcal{M}(Q^{\vec{a}}A, TB)^2$$

is called a ***congruence*** if, for all basic contexts $\mathrm{C} : (\vec{a}, \Gamma, A) \mapsto (\vec{a}, \Gamma', A')$ and all terms $\vec{a} \mid \Gamma \vdash M, N : A$,

$$[\![M]\!]\ R^{\vec{a}}\ [\![N]\!] \implies [\![\mathrm{C}[M]]\!]\ R^{\vec{a}}\ [\![\mathrm{C}[N]]\!]\,.$$

A family $R = (R^{\vec{a}})_{\vec{a}\in\mathbb{A}^\#}$ of congruences is itself a congruence if, for all terms $\vec{a} \mid \Gamma \vdash M, N : A$ and all contexts $\mathrm{C} : (\vec{a}, \Gamma, A) \mapsto (\vec{a}', \Gamma', A')$,

$$[\![M]\!]\ R^{\vec{a}}\ [\![N]\!] \implies [\![\mathrm{C}[M]]\!]\ R^{\vec{a}'}\ [\![\mathrm{C}[N]]\!]\,.$$

▲

Observing how the semantic interpretation is constructed, we can derive a set of sufficient conditions for congruences.

Lemma 4.18 *Let $\mathcal{M}$ be a $\nu\rho$-model and let $R^{\vec{a}} \subseteq \bigcup_{A,B} \mathcal{M}(Q^{\vec{a}}A, TB)^2$ be a preorder, for some $\vec{a}$. If, for all relevant f, f' and h, whenever $f\ R^{\vec{a}}\ f'$ holds then the following diagrams hold,*

$$\begin{array}{ccc} Q^{\vec{a}}(A\times 1) & \xrightarrow{Q^{\vec{a}}\pi_1} & Q^{\vec{a}}A \\ {\scriptstyle Q^{\vec{a}}\pi_1}\downarrow & R^{\vec{a}} & \downarrow{\scriptstyle f'} \\ Q^{\vec{a}}A & \xrightarrow[f]{} & TB \end{array} \qquad \begin{array}{ccc} Q^{\vec{a}}A & \xrightarrow{\Lambda^T(\zeta'\,;f')} & TB^{\,C} \\ {\scriptstyle \Lambda^T(\zeta'\,;f)}\downarrow & R^{\vec{a}} & \downarrow{\scriptstyle \eta} \\ TB^{\,C} & \xrightarrow[\eta]{} & T(TB^{\,C}) \end{array}$$

$$\begin{array}{ccc} Q^{\vec{a}}A & \xrightarrow{f'} & TB \\ {\scriptstyle f}\downarrow & R^{\vec{a}} & \downarrow{\scriptstyle Th\,;\mu} \\ TB & \xrightarrow[Th\,;\mu]{} & TC \end{array} \qquad \begin{array}{ccc} Q^{\vec{a}}A & \xrightarrow{\Delta} & Q^{\vec{a}}A \times Q^{\vec{a}}A \\ {\scriptstyle \Delta}\downarrow & R^{\vec{a}} & \downarrow{\scriptstyle f'\times \mathtt{id}\,;\tau'} \\ Q^{\vec{a}}A \times Q^{\vec{a}}A & \xrightarrow[f\times \mathtt{id}\,;\tau']{} & T(B\times Q^{\vec{a}}A) \end{array}$$

then $R^{\vec{a}}$ is a congruence.
Moreover, a family $R = (R^{\vec{a}})_{\vec{a}\in\mathbb{A}^\#}$ of congruences is a congruence if, for all relevant $f, f', \vec{a}, a, \vec{a}'$,

$$f\ R^{\vec{a}a}\ f' \implies \langle a\rangle f\ R^{\vec{a}}\ \langle a\rangle f' \quad \wedge \quad f\ R^{\vec{a}}\ f' \implies \tfrac{\vec{a}'}{\vec{a}}\,;f\ R^{\vec{a}'}\ \tfrac{\vec{a}'}{\vec{a}}\,;f'\,.$$

Proof: We first note that the diagrams imply that if $f\ R^{\vec{a}}\ f'$ and $g\ R^{\vec{a}}\ g'$ then:

- $f\,;\eta = (\ \Delta\,;(\Lambda^T(\zeta'\,;Q^{\vec{a}}\pi_1\,;f)\,;\eta)\times \mathtt{id}\,;\tau'\,;T(\mathtt{id}\times!\,;\mathtt{ev}^T\,;\eta)\,;\mu\ )\ R^{\vec{a}}\ (f'\,;\eta)$.
- $f\,;h = (f\,;\eta\,;Th\,;\mu)\ R^{\vec{a}}\ (f'\,;h)$.
- $\langle f,g\rangle\,;\psi = (\Delta\,;f\times \mathtt{id}\,;\tau'\,;T(\mathtt{id}\times g\,;\tau)\,;\mu)\ R^{\vec{a}}\ (\langle f',g\rangle\,;\psi)$.
- $\langle f',g\rangle\,;\psi = (\Delta\,;g\times \mathtt{id}\,;\tau'\,;T(\mathtt{id}\times f')\,;T\tau\,;T^2\langle\pi_2,\pi_1\rangle\,;\mu)\ R^{\vec{a}}\ (\langle f',g'\rangle\,;\psi)$.
- $\langle f,g\rangle\,;\tau = (\langle f\,;\eta,g\rangle\,;\psi)\ R^{\vec{a}}\ (\langle f',g'\rangle\,;\tau)$.
- $\langle f,g\rangle\,;\tau' = (\langle f,g\,;\eta\rangle\,;\psi)\ R^{\vec{a}}\ (\langle f',g'\rangle\,;\tau')$.

It then follows that $R^{\vec{a}}$ is a congruence, by induction on contexts. Finally, in order to show that R is a congruence it suffices to show that for all $\vec{a}\mid\Gamma\vdash M,N:A$, $[\![M]\!]\ R^{\vec{a}}\ [\![N]\!]$ implies:

- $[\![M]\!]_{\vec{a}'|\Gamma}\ R^{\vec{a}'}\ [\![N]\!]_{\vec{a}'|\Gamma}$, for any $\vec{a}\subseteq\vec{a}'$,
- $[\![\nu a.M]\!]\ R^{\vec{a}'}\ [\![\nu a.N]\!]$, whenever $\vec{a}=\vec{a}'a$.

But, using also lemma 4.10, we see that these conditions precisely correspond to the conditions for congruences. ■

It is now straightforward to show that the semantic preorder is a congruence.

Corollary 4.19 *Let $(\mathcal{M},T,Q^{\vec{a}},O)$ be an observational $\nu\rho$-model. Then, $(\lesssim^{\vec{a}})_{\vec{a}\in\mathbb{A}^\#}$ is a congruence.*

Proof: We need only show the four diagrams of the previous lemma, and these are easily obtained from the enrichment properties of the semantic preorder, proven in proposition 2.35. For example, for the first diagram and for general $h:A'\longrightarrow A$, we have:

$$Q^{\vec{a}}h\,;f=\delta\,;Q^{\vec{a}}\varepsilon\,;Q^{\vec{a}}h\,;Q^{\vec{a}}\eta\,;\ell\,;Tf\,;\mu$$

which falls within the first claim of that proposition. ■

Assume now that we translate $\nu\rho$ in an observational $\nu\rho$-model. Then, one direction of (FA) follows immediately form the definition.

Lemma 4.20 (Inequational Soundness) *For terms $\vec{a}\mid\Gamma\vdash M,N:A$,*

$$[\![M]\!]\lesssim[\![N]\!]\implies M\lessapprox N\,.$$

Proof: Assume $[\![M]\!]\lesssim^{\vec{a}}[\![N]\!]$ and $\vDash \mathrm{C}[M]\longrightarrow\!\!\!\!\!\rightarrow\ S'\vDash 0$, so $[\![\mathrm{C}[M]]\!]=\langle\!\langle\vec{a}'\rangle\!\rangle\,[\![\bar{S}'\,;0]\!]$ with $\vec{a}'=\mathtt{dom}(S')$. $[\![M]\!]\lesssim^{\vec{a}}[\![N]\!]$ implies $[\![\mathrm{C}[M]]\!]\lesssim[\![\mathrm{C}[N]]\!]$, and hence $[\![\mathrm{C}[N]]\!]\in O^\epsilon$. Thus, by observationality and adequacy, there exists S'' such that $\vDash\mathrm{C}[N]\longrightarrow\!\!\!\!\!\rightarrow\ S''\vDash 0$. ■

In order to achieve completeness, and hence full-abstraction, we need our semantic translation to satisfy some definability requirement with regard to the intrinsic preorder.

Definition 4.21 Let $(\mathcal{M},T,Q,O)$ be an observational $\nu\rho$-model and let $[\![\_]\!]$ be the semantic translation of $\nu\rho$ to $\mathcal{M}$. $\mathcal{M}$ satisfies ***ip-definability*** if, for all $\vec{a},A,B$, there exists $D^{\vec{a}}_{A,B}\subseteq\mathcal{M}(Q^{\vec{a}}[\![A]\!],T[\![B]\!])$ such that:

- for each $f\in D^{\vec{a}}_{A,B}$ there exists term M such that $[\![M]\!]=f$,
- for each $f,g\in\mathcal{M}(Q^{\vec{a}}[\![A]\!],T[\![B]\!])$,

$$f\lesssim^{\vec{a}}g\iff\forall\rho\in D^{\vec{a}}_{A\to B,\mathbb{N}}\,.\,(\Lambda^{\vec{a}}(f)\,;\rho\in O^{\vec{a}}\implies\Lambda^{\vec{a}}(g)\,;\rho\in O^{\vec{a}})\,.$$

We write $\mathcal{M}$ as $(\mathcal{M},T,Q,O,D)$. ▲

For such a model $\mathcal{M}$ we achieve full abstraction.

Proposition 4.22 (FA) *For terms* $\vec{a} \mid \Gamma \vdash M, N : A$,

$$[\![M]\!] \lesssim [\![N]\!] \iff M \lessapprox N\,.$$

Proof: Soundness is by previous lemma. For completeness ($\Longleftarrow$), we do induction on the size of Γ.
For the base case suppose $\vec{a} \mid \varnothing \vdash M \lessapprox N$ and take any $\rho \in D_{\mathbb{1}\to A,\mathbb{N}}$ such that $\Lambda^{\vec{a}}([\![M]\!])\,;\rho \in O^{\vec{a}}$. Let $\rho = [\![\vec{a} \mid y : \mathbb{1} \to A \vdash L : \mathbb{N}]\!]$, some L, so $\Lambda^{\vec{a}}([\![M]\!])\,;\rho$ is

$$\Lambda^{\vec{a}}([\![M]\!])\,;[\![L]\!] = \delta\,;Q^{\vec{a}}|\lambda z.M|\,;[\![L]\!] = [\![(\lambda y.L)(\lambda z.M)]\!]$$

for some $z:\mathbb{1}$. The latter being in $O^{\vec{a}}$ implies that it equals $\langle\!\langle \vec{b} \rangle\!\rangle [\![\bar{S}\,;0]\!]$, some S. Now, $M \lessapprox N$ implies $(\lambda y.L)(\lambda z.M) \lessapprox (\lambda y.L)(\lambda z.N)$, hence $\nu\vec{b}.(\bar{S}\,;0) \lessapprox (\lambda y.L)(\lambda z.N)$, by soundness. But this implies that $\vec{a} \vDash (\lambda y.L)(\lambda z.N) \twoheadrightarrow S' \vDash 0$, so $[\![(\lambda y.L)(\lambda z.N)]\!] \in O^{\vec{a}}$, by correctness. Hence, $\Lambda^{\vec{a}}([\![N]\!])\,;\rho \in O^{\vec{a}}$, so $[\![M]\!] \lesssim^{\vec{a}} [\![N]\!]$, by ip-definability.
For the inductive step, if $\Gamma = x{:}B, \Gamma'$ then

$$\begin{aligned}\vec{a} \mid \Gamma \vdash M \lessapprox N &\implies \vec{a} \mid \Gamma' \vdash \lambda x.M \lessapprox \lambda x.N \overset{IH}{\implies} [\![\lambda x.M]\!] \lesssim^{\vec{a}} [\![\lambda x.N]\!] \\ &\implies [\![M]\!] = [\![(\lambda x.M)x]\!] \overset{(*)}{\lesssim^{\vec{a}}} [\![(\lambda x.N)x]\!] = [\![N]\!]\end{aligned}$$

where $(*)$ follows from corollary 4.19. ∎

4.3 The nominal games model

We embark on the adventure of modelling $\nu\rho$ in a category of nominal arenas and strategies. Our starting point is the category $\mathcal{V}_{\mathsf{t}}$ of nominal arenas and total strategies (definition 3.51). Recall that $\mathcal{V}_{\mathsf{t}}$ is constructed within the category **Nom** of nominal sets so, for each type A, we have an arena $\mathbb{A}_A$ for references to type A. Explicitly, $\mathbb{A}_A$ is the flat arena $\mathbb{A}^a$, with $a \in \mathbb{A}_A$, defined in (3.1).

The semantics is monadic in a ***store monad*** built around a store arena ξ, and comonadic in an initial state comonad. The store monad is defined on top of the lifting monad (see definition 3.68) by use of the side-effect constructor described in section 2.3.3, that is,

$$TA \triangleq \xi \multimap (A \otimes \xi)_\perp \qquad (i.e.\ TA = \xi \Rightarrow A \otimes \xi).$$

Now, ξ contains the values assigned to each name (reference), and thus it is of the form

$$\bigotimes\nolimits_{A\in \mathrm{TY}} (\mathbb{A}_A \Rightarrow [\![A]\!])$$

where $[\![A]\!]$ is the translation of each type A. Thus, a recursive definition of the type-translation is not possible because of the following cyclicity.

$$\begin{aligned}[\![A \to B]\!] &= [\![A]\!] \multimap (\xi \Rightarrow [\![B]\!] \otimes \xi) \\ \xi &= \bigotimes\nolimits_{A} (\mathbb{A}_A \Rightarrow [\![A]\!])\end{aligned} \tag{SE}$$

Rather, both ξ and the type-translation have to be *computed* as the least solution to the above domain equation. By the way, observe that $[\![A \to B]\!] = [\![A]\!] \otimes \xi \Rightarrow [\![B]\!] \otimes \xi$.

4.3.1 Solving the Store Equation

The full form of the store equation (SE) is the following.

$$[\![\mathbb{1}]\!] = 1\,, \qquad [\![\mathbb{N}]\!] = \mathbb{N}\,, \qquad [\![[A]]\!] = \mathbb{A}_A\,, \qquad [\![A \times B]\!] = [\![A]\!] \otimes [\![B]\!]\,,$$
$$[\![A \to B]\!] = [\![A]\!] \multimap (\xi \Rightarrow [\![B]\!] \otimes \xi)\,, \qquad \xi = \bigotimes\nolimits_{A} (\mathbb{A}_A \Rightarrow [\![A]\!])\,.$$

This can be solved either as a fixpoint equation in the cpo of nominal arenas or as a domain equation in the PreCpo-enriched category $\mathcal{V}_{\mathtt{t}}$. We follow the latter approach, which provides the most general notion of canonical solution (and which incorporates the solution in the cpo of nominal arenas, analogously to [McC00]). It uses the categorical constructions of [SP82, Fre90] for solving recursive domain equations, as adapted to games in [McC00].

Definition 4.23 Define the category

$$\mathcal{C} \triangleq \mathcal{V}_{\mathtt{t}} \times \prod_{A \in \mathrm{TY}} \mathcal{V}_{\mathtt{t}}$$

with objects D of the form $(D_\xi, {D_A}^{A \in \mathrm{TY}})$ and arrows f of the form $(f_\xi, {f_A}^{A \in \mathrm{TY}})$.
Now take $F : \mathcal{C}^{\mathrm{op}} \times \mathcal{C} \to \mathcal{C}$ to be defined on objects by:

$$F(D, E) \triangleq (\xi_{D,E}, {[\![A]\!]_{D,E}}^{A \in \mathrm{TY}}),$$

where

$$[\![\mathbb{1}]\!]_{D,E} \triangleq 1 \qquad [\![\mathbb{N}]\!]_{D,E} \triangleq \mathbb{N}$$

$$[\![A \times B]\!]_{D,E} \triangleq [\![A]\!]_{D,E} \otimes [\![B]\!]_{D,E} \qquad [\![A \to B]\!]_{D,E} \triangleq D_A \multimap (\xi_{E,D} \Rightarrow E_B \otimes \xi_{D,E})$$

$$[\![[A]]\!]_{D,E} \triangleq \mathbb{A}_A \qquad \xi_{D,E} \triangleq \bigotimes_{A \in \mathrm{TY}} (\mathbb{A}_A \Rightarrow E_A)$$

and similarly for arrows, with $F(f, g) \triangleq (\xi_{f,g}, {[\![A]\!]_{f,g}}^{A \in \mathrm{TY}})$. ▲

Now (SE) has been reduced to:

$$D = F(D, D) \tag{SE*}$$

where F is a locally continuous functor wrt the strategy ordering (proposition 3.65), and continuous wrt the arena ordering (proposition 3.67). The solution to (SE*) is given via a *local bilimit* construction to the following ω-chain in $\mathcal{C}$.[3]

Definition 4.24 In $\mathcal{C}$ form the sequence $(D_i)_{i \in \omega}$ taking D_0 as below and $D_{i+1} \triangleq F(D_i, D_i)$.

$$D_{0,\mathbb{1}} \triangleq 1 \qquad D_{0,\mathbb{N}} \triangleq \mathbb{N} \qquad D_{0,[A]} \triangleq \mathbb{A}_A$$

$$D_{0,A \to B} \triangleq 1 \qquad D_{0,A \times B} \triangleq D_{0,A} \otimes D_{0,B} \qquad D_{0,\xi} \triangleq \bigotimes_A (\mathbb{A}_A \Rightarrow 0)$$

Moreover, define arrows $e_i : D_i \longrightarrow D_{i+1}$ and $e_i^R : D_{i+1} \longrightarrow D_i$ by:

$$e_0 \triangleq \mathtt{incl}_{D_0, D_1}, \quad e_0^R \triangleq \mathtt{proj}_{D_1, D_0}, \qquad e_{i+1} \triangleq F(e_i^R, e_i), \quad e_{i+1}^R \triangleq F(e_i, e_i^R).$$

▲

The above inclusion and projection arrows are defined componentwise. In fact, there is a hidden lemma in the definition which allows us to define the projection arrow, namely that $D_0 \trianglelefteq_1 D_1$ (which means $D_{0,\xi} \trianglelefteq_1 D_{1,\xi}$ and $D_{0,A} \trianglelefteq_1 D_{1,A}$ for all A).

Thus, we have formed the ω-chain Δ:

$$D_0 \xrightarrow{e_0} D_1 \xrightarrow{e_1} D_2 \xrightarrow{e_2} D_3 \xrightarrow{e_3} \cdots \tag{Δ}$$

We now show that Δ is a $\trianglelefteq$-increasing sequence of objects and embeddings, and proceed to the main result.

[3] Recall that we call an arrow $e : A \longrightarrow B$ an ***embedding*** if there exists $e^R : B \longrightarrow A$ such that

$$e ; e^R = \mathtt{id}_A \ \wedge \ e^R ; e \sqsubseteq \mathtt{id}_B .$$

Given an ω-chain $\Delta = (D_i, e_i)_{i \in \omega}$ of objects and embeddings, a ***cone*** for Δ is an object D together with a family $(\eta_i : D_i \longrightarrow D)_{i \in \omega}$ of embeddings such that, for all $i \in \omega$, $\eta_i = e_i ; \eta_{i+1}$. Such a cone is a ***local bilimit*** for Δ if, for all $i \in \omega$,

$$\eta_i^R ; \eta_i \sqsubseteq \eta_{i+1}^R ; \eta_{i+1} \ \wedge \ \bigsqcup_{i \in \omega} (\eta_i^R ; \eta_i) = \mathtt{id}_D .$$

Lemma 4.25 *For $(e_i, e_i^R)_{i\in\omega}$ as above and any $i \in \omega$,*

$$e_i = \mathtt{incl}_{D_i,D_{i+1}} \quad \wedge \quad e_i^R = \mathtt{proj}_{D_{i+1},D_i}\,.$$

Proof: Doing induction on i we can show that $D_i \trianglelefteq_1 D_{i+1}$, all $i \in \omega$, and that the above equalities hold. The base case is true by definition; the inductive step follows from proposition 3.67. ∎

Theorem 4.26 *We obtain a local bilimit $(D^*, \eta_i{}^{i\in\omega})$ for Δ by taking:*

$$D^* \triangleq \bigsqcup_i D_i\,, \qquad \eta_i \triangleq \mathtt{incl}_{D_i,D^*} \quad \textit{(each } i \in \omega\textit{)}.$$

Hence, $\mathtt{id}_{D^*} : F(D^*, D^*) \longrightarrow D^*$ is a minimal invariant for F.

Proof: First, note that $D_0 \trianglelefteq_1 D_i$, for all $i \in \omega$, implies that all D_i's share the same initial moves, and hence $D_i \trianglelefteq_1 D^*$. Thus, for each $i \in \omega$, we can define $\eta_i^R \triangleq \mathtt{proj}_{D^*,D_i}$, and hence each η_i is an embedding. We now need to show the following.

1. $(D^*, \eta_i{}^{i\in\omega})$ is a cone for Δ,
2. for all $i \in \omega$, $\eta_i^R\,;\eta_i \sqsubseteq \eta_{i+1}^R\,;\eta_{i+1}$,
3. $\bigsqcup_{i\in\omega}(\eta_i^R\,;\eta_i) = \mathtt{id}_{D^*}$.

For 1, we nts that, for any i, $\mathtt{incl}_{D_1,D^*} = \mathtt{incl}_{D_i,D_{i+1}}\,;\mathtt{incl}_{D_{i+1},D^*}$, which follows from (TRN). For 2 we essentially nts that $\mathtt{id}_{D_i} \subseteq \mathtt{id}_{D_{i+1}}$, and for 3 that $\bigcup_i \mathtt{id}_{D_i} = \mathtt{id}_{D^*}$; these are both straightforward.
From the local bilimit $(D^*, \eta_i{}^{i\in\omega})$ we obtain a minimal invariant $\alpha : F(D^*, D^*) \longrightarrow D^*$ by taking (see e.g. [Abr07]):

$$\alpha \triangleq \bigsqcup_i \alpha_i\,, \quad \alpha_i \triangleq F(\eta_i, \eta_i^R)\,;\eta_{i+1} \overset{\text{prop. 3.67}}{=} \mathtt{proj}_{F(D^*,D^*),D_{i+1}}\,;\mathtt{incl}_{D_{i+1},D^*}\,.$$

Moreover, $D^* = F(D^*, D^*)$ by the Tarski-Knaster theorem, and therefore $\alpha_i = \eta_{i+1}^R\,;\eta_{i+1}$, which implies $\alpha = \mathtt{id}_{D^*}$. ∎

Thus, D^* is the canonical solution to $D = F(D, D)$, and in particular it solves:

$$D_{A\to B} = D_A \multimap (D_\xi \Rightarrow D_B \otimes D_\xi)$$
$$D_\xi = \bigotimes\nolimits_A (\mathbb{A}_A \Rightarrow D_A)\,.$$

Definition 4.27 (ξ, $\circledast$ and $[\![A]\!]$) Let D^* be as in the previous theorem. Define the store arena ξ and, for each type A, the translation $[\![A]\!]$ of A by:

$$\xi \triangleq D_\xi^*\,, \qquad [\![A]\!] \triangleq D_A^*\,.$$

▲

The arena ξ and the translation of compound types are given explicitly in the following figure. ξ is depicted by means of unfolding it to $\bigotimes_A(\mathbb{A}_A \Rightarrow [\![A]\!])$: it consists of an initial move $\circledast$ which justifies each name-question $a \in \mathbb{A}_A$, all types A, with the answer to the latter being the denotation of A (and modelling the stored value of a). Note that we reserve the symbol "$\circledast$" for the initial move of ξ. $\circledast$-moves in type-translations can be seen as *opening a new store*.

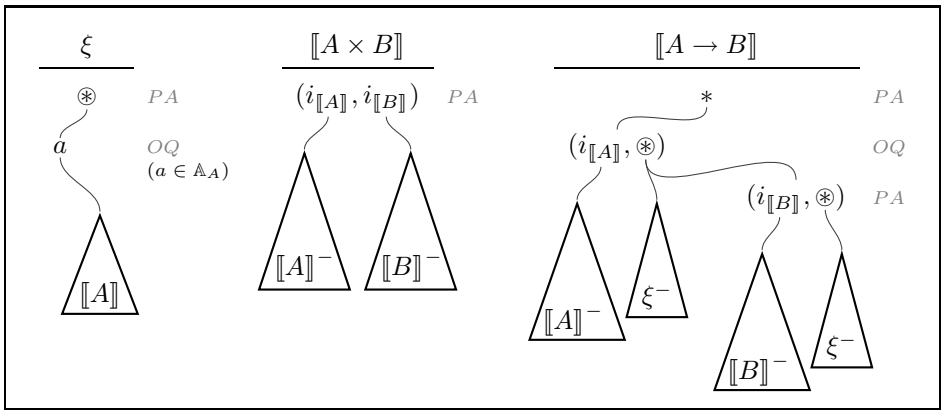


Figure 4.4: The store arena ξ and the translation of $\nu\rho$-types.

4.3.2 The store monad T

In section 2.3.3 we described the general construction of a monad of ξ'-side-effects starting from a given monad T'. Applying the construction to the lifting monad and the store arena ξ, as below, we obtain a ***store monad*** (T, η, μ, τ) on $\mathcal{V}_{\mathtt{t}}$.

$$\begin{aligned}
&T : \mathcal{C} \longrightarrow \mathcal{C} \triangleq \xi \Rightarrow (_ \otimes \xi)\\
&\eta_A : A \longrightarrow TA \triangleq \Lambda\left(A \otimes \xi \xrightarrow{\mathtt{up}} (A \otimes \xi)_\perp\right)\\
&\mu_A : T^2A \longrightarrow TA \triangleq \Lambda\left(T^2A \otimes \xi \xrightarrow{\mathtt{ev}} (TA \otimes D)_\perp \xrightarrow{\mathtt{ev}_\perp} (A \otimes \xi)_{\perp\perp} \xrightarrow{\mathtt{dn}} (A \otimes \xi)_\perp\right)\\
&\tau_{A,B} : A \otimes TB \longrightarrow T(A \otimes B) \triangleq \Lambda\left(A \otimes TB \otimes \xi \xrightarrow{\mathtt{id}\otimes\mathtt{ev}} A \otimes (B \otimes \xi)_\perp \xrightarrow{\mathtt{st}} (A \otimes B \otimes \xi)_\perp\right)
\end{aligned} \tag{4.12}$$

A concrete description of the store monad is given in figure 4.5. The diagram of TA gives a depiction of the arena as a levelled tree. On the other hand, the diagrams of strategies depict their viewfunctions, as described in section 3.2.3. For the particular case of $\circledast$-moves which appear as second moves in TA's, let us recall the convention we are following. Looking at the diagram for TA (figure 4.5), we see that $\circledast$ justifies a copy of ξ^- (left) and a copy of $A \otimes \xi$ (right). Thus, a copycat link connecting to the lower-left of a $\circledast$ expresses a copycat concerning the ξ^- justified by $\circledast$ (e.g. the link between the first two $\circledast$-moves in the diagram for μ_A), and similarly for copycat links connecting to the lower-right of a $\circledast$. Thus, for example, μ_A is given by:

$$\begin{aligned}
\mu_A = \mathtt{strat}(\ &\{\ [* * \circledast \circledast s] \mid [\circledast \circledast s] \in \mathtt{viewf}(\mathtt{id}_\xi)\ \}\\
&\cup \{\ [* * \circledast \circledast (*, \circledast') \circledast' s] \mid [\circledast' \circledast' s] \in \mathtt{viewf}(\mathtt{id}_\xi) \vee [s] \in \mathtt{viewf}(\mathtt{id}_{A\otimes\xi})\ \}\ ).
\end{aligned}$$

By proposition 2.23, and because lifting is a strong monad with exponentials (proposition 3.69), T is a strong monad with exponentials. Moreover, for each arena A we can define an arrow

$$\alpha_A \triangleq A_\perp \xrightarrow{(\eta_A)_\perp} (TA)_\perp \xrightarrow{\mathtt{pu}_{TA}} TA\,. \tag{4.13}$$

The transformation $\mathtt{pu}$ was introduced in section 3.4. From that section, recall also the fact that, for any pointed B, $\mathtt{pu}_{A\multimap B}$ is $\Lambda(\mathtt{st}'\,;\mathtt{ev}_\perp\,;\mathtt{pu}_B)$, and thus, taking also into account that $\mathtt{pu}_{C_\perp} = \mathtt{dn}_C$,

$$\mathtt{pu}_{TA} = \Lambda\left((TA)_\perp \otimes \xi \xrightarrow{\mathtt{st}'} (TA \otimes \xi)_\perp \xrightarrow{\mathtt{ev}_\perp} (A \otimes \xi)_{\perp\perp} \xrightarrow{\mathtt{dn}} (A \otimes \xi)_\perp\right). \tag{4.14}$$

By proposition 2.25 we then have that $\alpha : (\_)_\perp \longrightarrow T$ is a monad morphism, and that

$$\alpha_A = \Lambda(\mathtt{st}'_{A,\xi})\,. \tag{4.15}$$

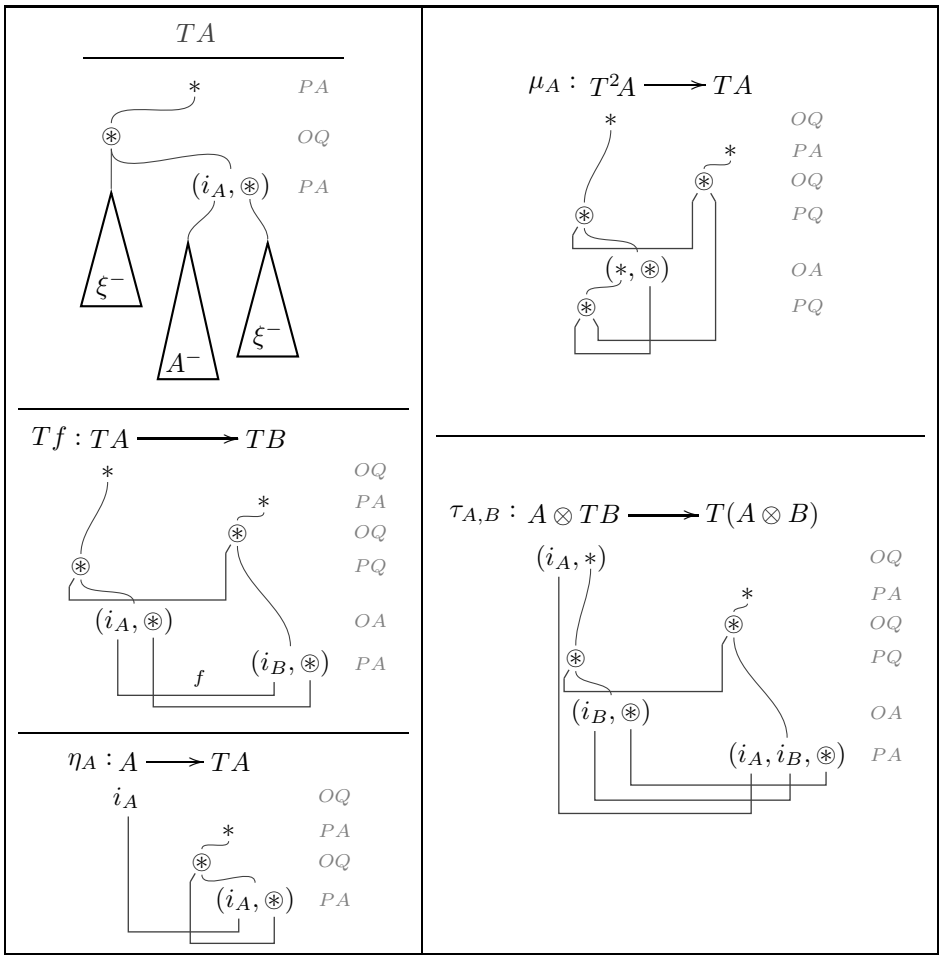


Figure 4.5: The store monad (T, η, μ, τ) for $\nu\rho$.

4.3.3 Obtaining the $\nu\rho$-model

Let us recapitulate the structure we have constructed thus far to the effect of obtaining a $\nu\rho$-model in $\mathcal{V}_{\mathtt{t}}$. Our numbering below follows that of definition 4.8.

I. $\mathcal{V}_{\mathtt{t}}$ is a category with finite products (proposition 3.57).

II. The store monad T is a strong monad with exponentials.

III. $\mathcal{V}_{\mathtt{t}}$ contains adequate structure for numerals.

IV. There is a family $(Q^{\vec{a}}, \varepsilon, \delta, \zeta)_{\vec{a} \in \mathbb{A}^\#}$ of product comonads, with each $Q^{\vec{a}}$ having basis $\mathbb{A}^{\vec{a}}$ (see section 3.4.2), which fulfills specifications (a,b). There are also fresh-name constructors,

$$\mathtt{new}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow (Q^{\vec{a}a})_\perp \,,$$

given in section 3.4.3, which satisfy (N2).

V. There are name-equality arrows, $\mathtt{eq}_A$ for each type A, making the (N1) diagram commute (section 3.4.2).

From `new` we can obtain a fresh-name transformation for the store monad.

Definition 4.28 For each $\vec{a}a \in \mathbb{A}^\#$, define a natural transformation $\mathtt{nu}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow TQ^{\vec{a}a}$ by:

$$\mathtt{nu}^{\vec{a}a}_A \triangleq Q^{\vec{a}}A \xrightarrow{\mathtt{new}_A} (Q^{\vec{a}a}A)_\perp \xrightarrow{\alpha_{Q^{\vec{a}a}A}} TQ^{\vec{a}a}A \,.$$

Moreover, for each $f : Q^{\vec{a}a}A \longrightarrow TB$ take

$$\langle a \rangle f \triangleq Q^{\vec{a}}A \xrightarrow{\mathtt{nu}_A} TQ^{\vec{a}a}A \xrightarrow{Tf} T^2B \xrightarrow{\mu_B} TB\,.$$

▲

Note that, for each $f : Q^{\vec{a}a}A \longrightarrow TB$,

$$\langle a \rangle f = \mathtt{nu}_A\,;Tf\,;\mu_B = \mathtt{new}_A\,;\alpha_{Q^{\vec{a}a}A}\,;Tf\,;\mu_B = \mathtt{new}_A\,;f_\perp\,;\alpha_{TB}\,;\mu_B \overset{(*)}{=} \mathtt{new}_A\,;f_\perp\,;\mathtt{pu}_{TB}$$

where $(*)$ is a consequence of (4.14, 4.15) and the definition of μ. Hence, our definition of name-abstraction here coincides with that of section 3.4.3 (definition 3.77).

Moreover, noting that for each sequence of moves s and each $a \# \mathsf{nlist}(s)$ we write s^a for s with a added in the head of all of its name lists, each arrow $\mathtt{nu}_A^{\vec{a}a}$ is explicitly given by:

$$\mathtt{nu}_A^{\vec{a}a} = \mathtt{strat}\{[(\vec{a}, i_A) * \circledast (\vec{a}a, i_A, \circledast)^a s^a] \mid a \# i_A \wedge ([i_A i_A s] \in \mathtt{viewf}(\mathtt{id}_A) \vee [\circledast \circledast s] \in \mathtt{viewf}(\mathtt{id}_\xi))\}$$

and diagrammatically as follows.

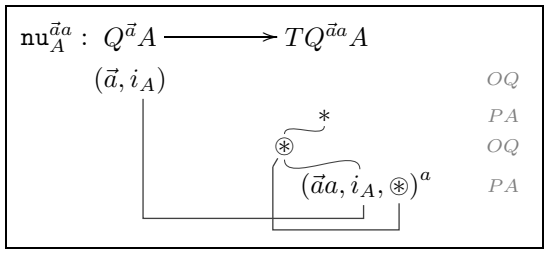


Figure 4.6: The fresh-name natural transformation for $\nu\rho$.

Using the fact that α is a monad morphism it is straightforward to obtain the following.

Proposition 4.29 *The* nu *transformation satisfies the* (N2) *diagrams of definition 4.8.* ■

What we are only missing for a $\nu\rho$-model is update and dereferencing maps. These are specified as follows.

Definition 4.30 For any type A we define the following arrows in $\mathcal{V}_t$,

$$\mathtt{drf}_A \triangleq \mathtt{strat}\{[a * \circledast\, a\, i_{[\![A]\!]}\, (i_{[\![A]\!]}, \circledast)\, s] \mid [\circledast\ \circledast\ s] \in \mathtt{viewf}(\mathtt{id}_\xi) \vee [i_{[\![A]\!]}\, i_{[\![A]\!]}\, s] \in \mathtt{viewf}(\mathtt{id}_{[\![A]\!]})\}\,,$$

$$\begin{aligned}\mathtt{upd}_A \triangleq \mathtt{strat}(&\{[(a, i_{[\![A]\!]}) * \circledast\, (*, \circledast)\, b\, b\, s] \mid [\circledast\ \circledast\ b\, b\, s] \in \mathtt{viewf}(\mathtt{id}_\xi) \wedge b \# a\} \\ &\cup \{[(a, i_{[\![A]\!]}) * \circledast\, (*, \circledast)\, a\, i_{[\![A]\!]}\, s] \mid [i_{[\![A]\!]}\, i_{[\![A]\!]}\, s] \in \mathtt{viewf}(\mathtt{id}_{[\![A]\!]})\})\,,\end{aligned}$$

depicted also in figure 4.7. ▲

These strategies work as follows. $\mathtt{upd}_A$ responds with the answer $(*, \circledast)$ to the initial sequence $(a, i_{[\![A]\!]}) * \circledast$ and then:

- for any name $b \# a$ that is asked by O to $(*, \circledast)$ (which is a store-opening move), it copies b under the store $\circledast$ (opened by O) and establishes a copycat link between the two b's;
- if O asks a to $(*, \circledast)$, it answers $i_{[\![A]\!]}$ and establishes a copycat link between the two $i_{[\![A]\!]}$'s.

On the other hand, $\mathtt{drf}_A$ does not immediately answer to the initial sequence $a * \circledast$ but rather asks (the value of) a to $\circledast$. Upon receiving O's answer $i_{[\![A]\!]}$, it answers $(i_{[\![A]\!]}, \circledast)$ and establishes two copycat links.

We can show by direct computation that updates and dereferencings work as required, i.e. make the (NR) diagrams commute. Moreover, these effects are independent from fresh-name creation, i.e. the (SNR) equation holds.

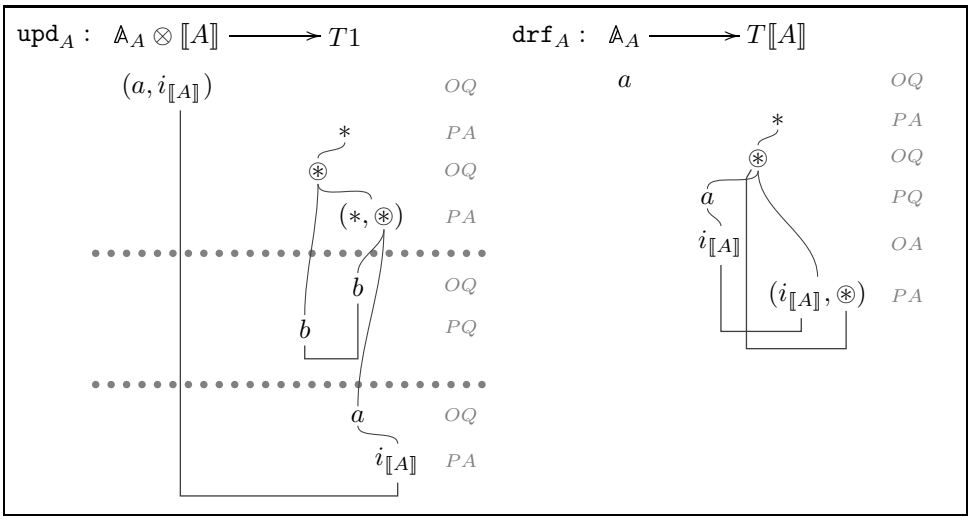


Figure 4.7: Update and dereferencing arrows in $\mathcal{V}_{\mathtt{t}}$.

Proposition 4.31 *The* (NR) *and* (SNR) *diagrams of definition 4.8 commute.* *Appendix* ■

We have therefore established the following.

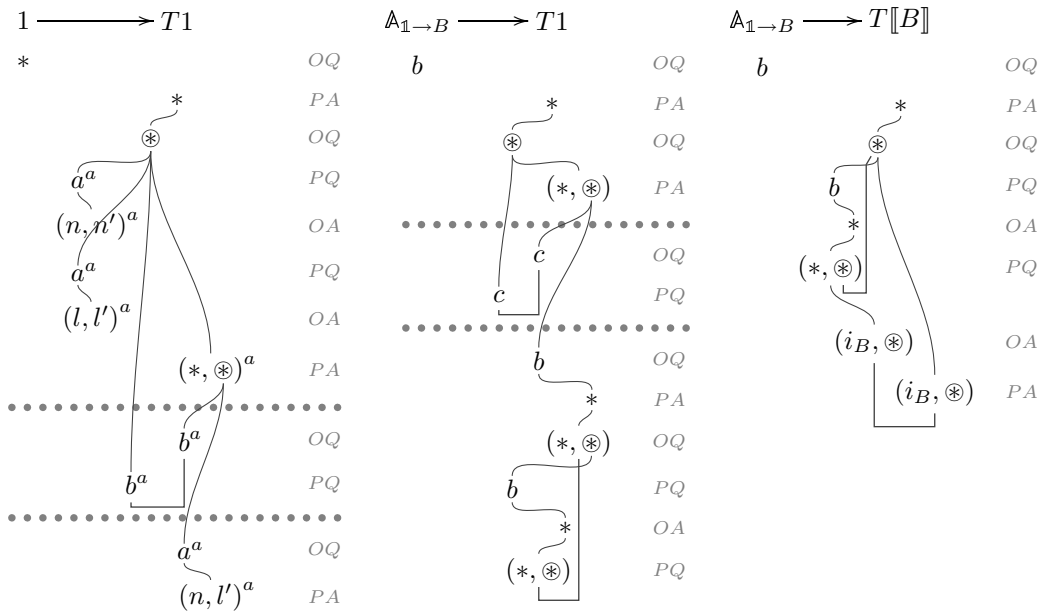
Theorem 4.32 $(\mathcal{V}_{\mathtt{t}}, T, Q)$ *is a* $\nu\rho$*-model.* ■

We close this section with some examples of translations of $\nu\rho$-terms in $\mathcal{V}_{\mathtt{t}}$ and a discussion on how the store-effect is achieved in our innocent setting.

Example 4.33 Consider the typed terms:

$$\epsilon \mid \varnothing \vdash \nu a.a := \langle \mathtt{fst}\,!\,a, \mathtt{snd}\,!\,a \rangle, \quad b \mid \varnothing \vdash b := \lambda x.(!\,b)\mathtt{skip}, \quad b \mid \varnothing \vdash (!\,b)\mathtt{skip}$$

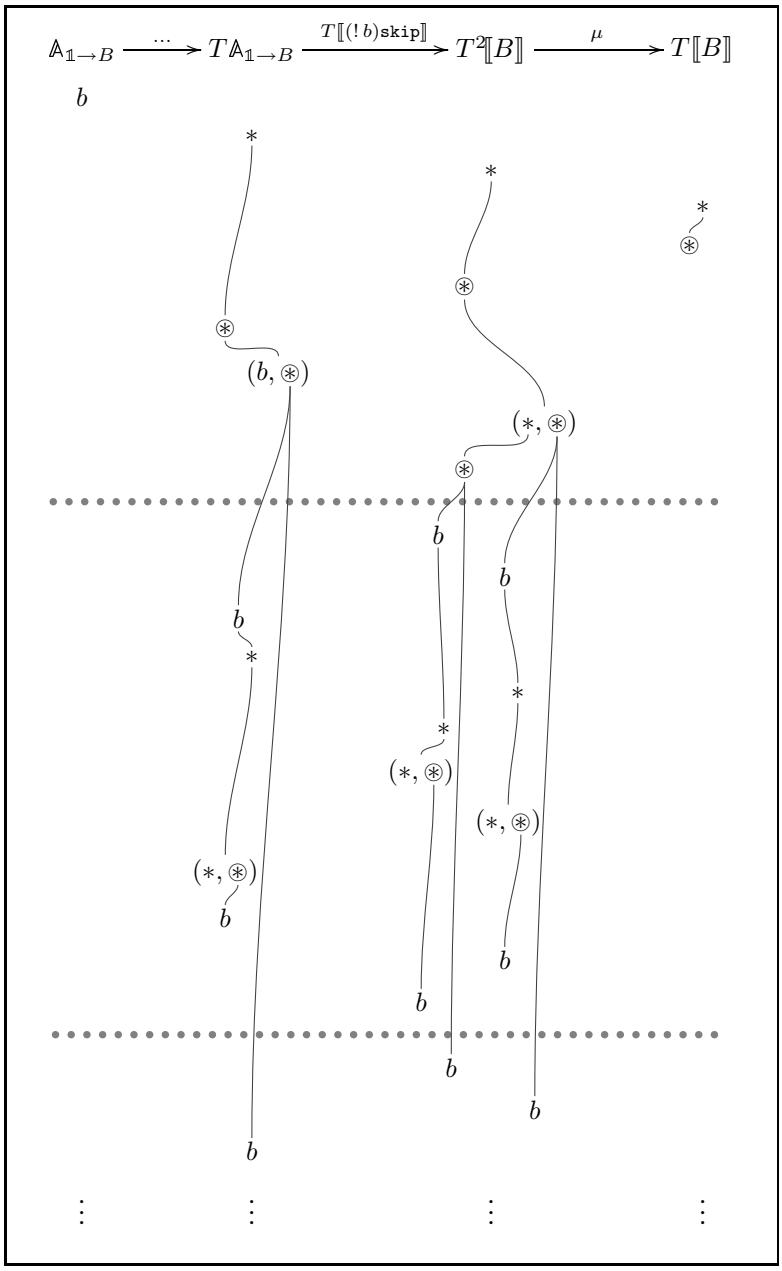
with $a \in \mathbb{A}_{\mathbb{N}\times\mathbb{N}}$ and $b \in \mathbb{A}_{\mathbb{1}\to B}$. Their translations in $\mathcal{V}_{\mathtt{t}}$ are as follows.



From the latter two we can compute

$$[\![b := \lambda x.(!\,b)\mathtt{skip}\,;(!\,b)\mathtt{skip}]\!] = \langle \mathtt{id}, [\![b := \lambda x.(!\,b)\mathtt{skip}]\!] \rangle\,;\tau\,;\cong\,;T[\![(!\,b)\mathtt{skip}]\!]\,;\mu$$

as follows.



We observe that the P-view of the interaction that each of the three component strategies sees after the three b's of the second dotted line is exactly the same as that seen after the three b's of the first dotted line. Hence, the part of the interaction between the two dotted lines is repeated ad infinitum (*infinite chattering*). Thus,

$$[\![b \mid \varnothing \vdash b := \lambda x.(!\,b)\mathtt{skip};(!\,b)\mathtt{skip}]\!] = \{\, [b * \circledast] \,\}$$

and therefore

$$[\![\mathtt{stop}_B]\!] : 1 \longrightarrow T[\![B]\!] = \{\, [* * \circledast] \,\}.$$

On the other hand, for $a \in \mathbb{A}_B$, $[\![\nu a.\,!\,a]\!] : 1 \longrightarrow T[\![B]\!]$ is given as follows.

$$\begin{aligned}
[\![\nu a.\,!\,a]\!] &= \langle a\rangle\, [\![a \mid \varnothing \vdash\ !\,a]\!] = \langle a\rangle\, \mathtt{drf}_B \\
&= \langle a\rangle(\mathtt{strat}\{\, [a * \circledast\, a\, i_B\, (i_B,\circledast)\, s] \mid [(\circledast, i_B)\,(\circledast, i_B)\, s] \in \mathtt{viewf}(\mathtt{id}_{\xi\otimes[\![B]\!]}) \,\}) \\
&= \mathtt{strat}\{\, [* * \circledast\, a^a i_B^a (i_B,\circledast)^a s^a] \mid [(\circledast, i_B)\,(\circledast, i_B)\, s] \in \mathtt{viewf}(\mathtt{id}_{\xi\otimes[\![B]\!]}) \,\}
\end{aligned}$$

■

Remark 4.34 (Innocent store) The approach to the modelling of store which we have presented differs fundamentally from previous such approaches in game semantics. Those approaches, be they for basic or higher-order store [AM97, AHM98], are based on the following methodology. References are modelled by read/write product types, and fresh-reference creation is modelled by a "cell" strategy which creates the fresh reference and imposes a good read/write discipline on it. In order for a cell to be able to return the last stored value, innocence has to be broken since each read-request hides previous write-requests from the P-view. Higher-order cells have to also break visibility in order to establish copycat links between read- and write-requests.

Here instead we have only used innocent strategies and a monad on a store ξ. Because of the monad, an arena $[\![A]\!]$ contains several copies of ξ, and therefore several stores are opened inside a play. The read/write discipline is then kept in an *interactive* way: when a participant asks (the value of) a name a at the last (relevant) store,[4] the other participant either answers with a value or asks himself a at the penultimate store, and so on until one of the participants answers or the first store in the play is reached (e.g. see figure 4.8). At each step, a participant answers the question a only if he updated the value of a before opening the current store (of that step, i.e. the last store in the participant's view) — note that this behaviour does *not* break innocence. If no such update was made by the participant then he simply passes a to the previous store and establishes a copycat link between the two a's. These links ensure that when an answer is eventually obtained then it will be copycatted all the way to answer the original question a. Thus, we innocently obtain a read/write discipline: at each question a, the last update of a is returned.

P – What's the value of a?
O – I don't know, you tell me: what's the value of a?
P – I don't know, you tell me: what's the value of a?
⋮
O – I don't know, you tell me: what's the value of a?
P – I know it, it is v.
⋮
O – I know it, it is v.
P – I know it, it is v.
O – I know it, it is v.

Figure 4.8: *A dialogue in innocent store.*

4.3.4 Adequacy

We proceed to show that $\mathcal{V}_{\mathsf{t}}$ is adequate (v. definition 4.14). First we characterise non-reducing terms as follows.

Lemma 4.35 *Let $\vec{a} \mid \varnothing \vdash M : A$ be a typed term. M is a value iff there exists a store S such that $S \Vdash M$ has no reducts and $[(\vec{a}, *) * \circledast (i_A, \circledast)^{\vec{b}}] \in [\![\bar{S}\,; M]\!]$, for some $i_A, \vec{b}$.*

Proof: The "only if"-part is straightforward. For the "if"-part assume that M is a non-value and take any S such that $S \Vdash M$ has no reducts. We show by induction on M that there exist no $i_A, \vec{b}$ such that $[(\vec{a}, *) * \circledast (i_A, \circledast)^{\vec{b}}] \in [\![\bar{S}\,; M]\!]$. The base case follows trivially from M not being a value. Now, for the inductive step, the specifications of $S \Vdash M$ (and M) imply that either $M \equiv !\,a$ with a not having a value in S, or $M \equiv \mathrm{E}[K]$ with E an evaluation context and K a non-value typed as $\vec{a} \mid \varnothing \vdash K : B$ and such that $S \Vdash K$ non-reducing.
In case of $M \equiv !\,a$, we have that $[(\vec{a}, *) * \circledast a] \in [\![\bar{S}\,; M]\!]$, which proves the claim because of

[4] i.e. at the last store-opening move played by the other participant.

determinacy. On the other hand, if $M \equiv \mathrm{E}[K]$ then, as in proof of proposition 4.13, we have that

$$[\![\bar{S}\,;M]\!] = \langle \Lambda(\zeta'\,;[\![\mathrm{E}[x]]\!]), [\![\bar{S}\,;K]\!]\rangle\,;\tau\,;T\mathsf{ev}\,;\mu = \langle \mathtt{id}, [\![\bar{S}\,;K]\!]\rangle\,;\tau\,;T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\mu$$

By IH, there are no $i_B, \vec{c}$ such that $[(\vec{a},*) * \circledast\, (i_B,\circledast)^{\vec{c}}] \in [\![\bar{S}\,;K]\!]$, which implies that there are no $i_A, \vec{b}$ such that $[(\vec{a},*) * \circledast\, (i_A,\circledast)^{\vec{b}}] \in [\![\bar{S}\,;M]\!]$. ∎

Because of the previous result, in order to show adequacy it will suffice to show that, whenever $[\![M]\!] = \langle\!\langle\vec{b}\rangle\!\rangle\, [\![\bar{S}\,;0]\!]$, there is no infinite reduction sequence starting from $\vec{a} \vDash M$. We will carry out the following reasoning.

- Firstly, since the calculus without DRF reductions is strongly normalising — this is inherited from strong normalisation of the $s\nu$-calculus — it suffices to show there is no reduction sequence starting from $\vec{a} \vDash M$ and containing infinitely many DRF reduction steps.
- In fact, the problem can be further reduced to showing that, whenever $[(\vec{a},*)*\circledast\,(0,\circledast)^{\vec{b}}] \in [\![M]\!]$, there is no reduction sequence starting from $\vec{a} \vDash M$ and containing infinitely many NEW reduction steps. But the latter clearly holds, since M cannot create more than $|\vec{b}|$ fresh names in that case, because of correctness.
  The reduction to this simpler problem is achieved as follows. For each term M, we construct a term M' by adding immediately before each dereferencing in M a fresh-name construction. The result is that, whenever there is a sequence with infinitely many DRF's starting from $S \vDash M$, there is a sequence with infinitely many NEW's starting from $S \vDash M'$. The reduction is completed by finally showing that, whenever we have $[(\vec{a},*) * \circledast\,(0,\circledast)^{\vec{b}}] \in [\![M]\!]$, we also have $[(\vec{a},*) * \circledast\,(0,\circledast)^{\vec{b}'}] \in [\![M']\!]$.

The crucial step in the proof is the reduction to "the simpler problem", and particularly showing the connection between $[\![M]\!]$ and $[\![M']\!]$ described above. The latter can be carried out by using the intrinsic preorder on strategies and showing O-adequacy (lemma 4.62): adequacy can then be *derived* from O-adequacy. Nevertheless, a direct proof is always useful (and more fun). We present such a proof in the remainder of this section.

In the following discussion we will be using κ for a $\nu\rho$-type. We start with a definition.

Definition 4.36 Let $\kappa \in \mathrm{TY}$. For any term M, we define:

$$\kappa \# M \iff \forall a \in \mathbb{A}_\kappa.\, a \# M\,.$$

For $\kappa \# M$, we define $(M)^{\circledast}$ inductively by:

$$(x)^{\circledast} \triangleq x\,, \quad (a)^{\circledast} \triangleq a\,, \quad \ldots \quad (\lambda x.N)^{\circledast} \triangleq \lambda x.(N)^{\circledast}\,, \quad (M\,N)^{\circledast} \triangleq (M)^{\circledast}(N)^{\circledast}\,, \quad \ldots$$

and $(!\,N)^{\circledast} \triangleq \nu c.!(N)^{\circledast}$, with $c \in \mathbb{A}_\kappa$. ▲

Note that $a \# M$ means that a does not appear, neither free nor bound, inside M. Our next target is to show that if $\kappa \# M$ then $[\![(M)^{\circledast}]\!]$ *does not use* names of type κ. What we mean by "non-usage" of names is defined formally below (recall the $\not\exists$-quantifier from page 17 and the notation $s = \underline{s}^{\mathsf{nlist}(s)}$ from page 43).

Definition 4.37 Let σ be a strategy. We say that the type κ *is essentially fresh for* σ, $\kappa \overset{\mathsf{ess}}{\#} \sigma$ if, for any $a \in \mathbb{A}_\kappa$,

$$\forall s.\, [s] \in \sigma \implies \not\exists b.\, [\underline{s}^{(a\ b)\,\circ\,\mathsf{nlist}(s)}] \in \sigma$$

▲

For economy, we let

$$(a\ b)^{\circ} s \triangleq \underline{s}^{(a\ b)\,\circ\,\mathsf{nlist}(s)}\,.$$

What the previous definition is essentially saying is that $\kappa \overset{\text{ess}}{\#} \sigma$ iff σ does *not really use* names of type κ, even though it may be introducing them.

There is an alternative definition of essential freshness for strategies, which involves only viewfunctions.

Proposition 4.38 *For any strategy σ, $\kappa \overset{\text{ess}}{\#} \sigma$ iff, for any $a \in \mathbb{A}_\kappa$,*

$$\forall s.\, [s] \in \mathtt{viewf}(\sigma) \implies \mathrel{\text{⩘}} b.\, [(a\ b)^\circ s] \in \sigma\,.$$

Proof: Suppose the condition holds and let us pick such an a and some $[s] \in \sigma$; we need to show that $\mathrel{\text{⩘}} b.\, [(a\ b)^\circ s] \in \sigma$. We do induction on $|s|$, taking as base case $|s| = 0$ which is trivial. For the inductive step, let $s = s^- x$. If x an O-move then $\mathrel{\text{⩘}} b.\, [(a\ b)^\circ s] \in \sigma$ by contingency completeness and the IH. On the other hand, if x a P-move and $\ulcorner s \urcorner \neq s$ then, for any fresh b, $[(a\ b)^\circ (s^-)], [\ulcorner (a\ b)^\circ s^- \urcorner] \in \sigma$ by IH and the hypothesis. By lemma 3.38, it suffices to show that $(a\ b)^\circ s$ is a play. For the latter, we need only check the Name Conditions. (NC3) and (NC2′) clearly hold, since $(a\ b)^\circ s^-, \ulcorner (a\ b)^\circ s^- \urcorner$ are plays. For (NC1), if $\mathsf{nlist}((a\ b)^\circ x)$ contains some new name c and $c \neq b$, then $c \# s^-$ since s is a play, so $c \# (a\ b)^\circ s^-$. If $c = b$ then $a \# s^-$ since s a play, $\therefore\ c \# (a\ b)^\circ s^-$ as b was chosen fresh. ∎

We can now show the following enrichment properties for essential freshness.

Lemma 4.39 *For any relevant σ, τ, if $\kappa \overset{\text{ess}}{\#} \sigma, \tau$ then:*

- $\kappa \overset{\text{ess}}{\#} \sigma\,;\tau$,
- $\kappa \overset{\text{ess}}{\#} \sigma_\perp$,
- $\kappa \overset{\text{ess}}{\#} \Lambda(\sigma)$,
- $\kappa \overset{\text{ess}}{\#} \langle \sigma, \tau \rangle$.

Proof: For the first claim we have that $[s\,;t] \in \sigma\,;\tau$ implies that, for any fresh b, $[(a\ b)^\circ s] \in \sigma$ and $[(a\ b)^\circ t] \in \tau$. Clearly, $(a\ b)^\circ s \smile (a\ b)^\circ t$. Moreover, the fresh-name conditions (C1-2) still hold: the only thing that could go wrong would be b being introduced at some point in $(a\ b)^\circ s$, say, without being fresh at the respective point in $\underline{t}$, which can't happen as b is chosen fresh.
The other claims follow easily from the previous proposition, and the definitions of these constructions on viewfunctions. ∎

Corollary 4.40 *For any term M,*

- $\kappa \# M \implies \kappa \overset{\text{ess}}{\#} [\![M]\!]$,
- $\kappa \# M \implies \kappa \overset{\text{ess}}{\#} [\![(M)^{\circledast}]\!]$.

Proof: Both claims are proven by induction on M. For the first claim, we simply use the fact that $\kappa \overset{\text{ess}}{\#} \sigma$ for any strategy σ that does not introduce any names of type κ.
For the second claim, we also use the previous lemma, yet the case of $M \equiv !\,N$ needs some extra attention. By IH we have that $\kappa \overset{\text{ess}}{\#} [\![(N)^{\circledast}]\!]$, while we know that

$$[\![(M)^{\circledast}]\!] = \mathtt{nu}_\Gamma\,;T\tfrac{\vec{a}c}{\vec{a}}\,;T[\![!(N)^{\circledast}]\!]\,;\mu$$

with $c \in \mathbb{A}_\kappa$ and M typed in environment $\vec{a}\,|\,\Gamma$. By previous lemma we have that $\kappa \overset{\text{ess}}{\#} T[\![!(N)^{\circledast}]\!]\,;\mu$, and hence it suffices to show that $\kappa \overset{\text{ess}}{\#} \mathtt{nu}\,;T\frac{\vec{a}c}{\vec{a}}$. We have that:

$$\mathtt{viewf}(\mathtt{nu}\,;T\tfrac{\vec{a}c}{\vec{a}}) = \{[(\vec{a}, i_\Gamma) * \circledast\, (\vec{a}, i_\Gamma, \circledast)^c\, s^c] \mid [i_\Gamma\, i_\Gamma\, s] \in \mathtt{viewf}(\mathtt{id}_{[\![\Gamma]\!]}) \vee [\circledast\, \circledast\, s] \in \mathtt{viewf}(\mathtt{id}_\xi)\}$$

From the above we observe that, for any $a \in \mathbb{A}_\kappa$, any $[s] \in \mathtt{viewf}(\mathtt{nu}\,; T\frac{\vec{a}c}{\vec{a}})$ and any fresh b, $[(a\ b)^\circ s] \in \mathtt{nu}\,; T\frac{\vec{a}c}{\vec{a}}$, as required. ∎

A consequence of essential freshness is that if $\kappa \overset{\text{ess}}{\#} \sigma$ then we can delete all κ-name introductions from σ and still have a strategy.

Definition 4.41 For any play s, define $s^{\smallsetminus\kappa}$ to be s with all names of type κ removed from its name lists. Take then, for each strategy σ,

$$\sigma^{\smallsetminus\kappa} \triangleq \{[s^{\smallsetminus\kappa}] \mid [s] \in \sigma\}\,.$$

▲

Lemma 4.42 *For any strategy σ, if $\kappa \overset{\text{ess}}{\#} \sigma$ then:*

1. *if $[s] \in \sigma$ then $s^{\smallsetminus\kappa}$ is a play,*
2. *if $[s_1], [s_2] \in \sigma$, $(\mathtt{S}(s_i) \setminus \mathtt{S}(s_i{}^{\smallsetminus\kappa})) \cap \mathtt{S}(\underline{s_i}) = \varnothing$ and $[s_1{}^{\smallsetminus\kappa}] = [s_2{}^{\smallsetminus\kappa}]$ then $[s_1] = [s_2]$,*
3. *$\sigma^{\smallsetminus\kappa}$ is a strategy.*

Proof: For 1, we only need to check the Name Conditions still hold, and in particular only (NC2′), as the other two trivially do. So let x be a P-move in $s^{\smallsetminus\kappa}$ and let $a \in \mathtt{S}(\underline{x})$ such that $a \# \ulcorner s^{\smallsetminus\kappa}_{<x} \urcorner$. If $a \notin \mathbb{A}_\kappa$ then $a \in \mathsf{nlist}(x)$; the case $a \in \mathbb{A}_\kappa$ is not possible since then we would have $[(a\ b)^\circ s] \in \sigma$ breaking (NC2′), any fresh b.
For 2, we do induction on $|s_1| = |s_2|$; the base case is trivial. For the inductive step, if $|s_1|$ is even then, by IH, $[s_1^-] = [s_2^-]$ and hence, by determinacy, $[s_1] = [s_2]$. Finally, if $s_i = s_i^- x_i$ with x_i an O-move then, by IH, $[s_1^-] = [s_2^-]$. Now using the condition on supports and the strong support lemma we obtain $[s_1^-\,\underline{x_1}] = [s_2^-\,\underline{x_2}]$ and thus, by (NC3), $[s_1] = [s_2]$.
We now show $\sigma^{\smallsetminus\kappa}$ is a strategy. Prefix closure and contingency completeness are obvious. For determinacy, take even-length $[s_1x_1], [s_2x_2] \in \sigma^{\smallsetminus\kappa}$ with $s_ix_i = (s_i'x_i')^{\smallsetminus\kappa}$ and $[s_i'x_i'] \in \sigma$, and assume $[s_1] = [s_2]$. As $\kappa\#\sigma$, we can choose $s_i'x_i'$ in such a way that the names in $\mathtt{S}(s_i'x_i') \setminus \mathtt{S}(s_ix_i)$ are fresh for $\underline{s_i'x_i'}$, and hence, by 2, $[s_1'] = [s_2']$. Then, by determinacy of σ, $[s_1'x_1'] = [s_2'x_2']$, which implies $[s_1x_1] = [s_2x_2]$. Finally, for innocence, let $[s_1x_1], [s_2] \in \sigma^{\smallsetminus\kappa}$ with $s_1x_1 = (s_1'x_1')^{\smallsetminus\kappa}$, $s_2 = s_2'^{\smallsetminus\kappa}$ and $[s_1'x_1'], [s_2'] \in \sigma$, and assume $[\ulcorner s_1 \urcorner] = [\ulcorner s_2 \urcorner]$. Then, as before, we may assume $[\ulcorner s_1' \urcorner] = [\ulcorner s_2' \urcorner]$ and therefore $[s_2'x_2'] \in \sigma$, for some $[\ulcorner s_2'x_2' \urcorner] = [\ulcorner s_1'x_1' \urcorner]$. We then have $[s_2x_2'^{\smallsetminus\kappa}] \in \sigma^{\smallsetminus\kappa}$ and $[\ulcorner s_1x_1 \urcorner] = [\ulcorner s_2x_2'^{\smallsetminus\kappa} \urcorner]$. ∎

We need a last lemma before proving adequacy.

Lemma 4.43 *For any term M, if $\kappa \overset{\text{ess}}{\#} M$ then $[\![M]\!] \subseteq [\![(M)^{\circledast}]\!]^{\smallsetminus\kappa}$.*

Proof: We do induction on M. The base case is encompassed in the case of $M \equiv (M)^{\circledast}$, in which case $s^{\smallsetminus\kappa} = s$ for any $[s] \in [\![M]\!]$ and the claim trivially holds. For the inductive step, as $[\![(M)^{\circledast}]\!]^{\smallsetminus\kappa}$ is a strategy by previous lemma, it suffices to show that

$$\mathtt{viewf}([\![M]\!]) \subseteq \{[s^{\smallsetminus\kappa}] \mid [s] \in [\![(M)^{\circledast}]\!]\}$$

We show some characteristic cases:

- $M \equiv \lambda x.N$. Let $[s_1\,; s_2] \in \mathtt{viewf}([\![M]\!])$, where $[\![M]\!] = \Lambda(\zeta'\,; [\![N]\!])\,; \eta$, and $[s_1] \in \Lambda(\zeta'\,; [\![N]\!])$, $[s_2] \in \eta$. Since $s_2{}^{\smallsetminus\kappa} = s_2$, it suffices to show that $[s_1] \in \Lambda(\zeta'\,; [\![(N)^{\circledast}]\!])^{\smallsetminus\kappa}$, and because the latter is a strategy it suffices to consider the case of s_1 being an even-length P-view (lemma 3.38). Then, by definition of Λ, s_1 is obtained from some $[s_1'] \in \mathtt{viewf}([\![N]\!])$ after a reordering of moves. By IH, $s_1' = t_1'^{\smallsetminus\kappa}$ for some $[t_1'] \in [\![(N)^{\circledast}]\!]$ and hence $s_1 = t_1{}^{\smallsetminus\kappa}$, with t_1 being the play in $\Lambda(\zeta'\,; [\![(N)^{\circledast}]\!])$ obtained by t_1'.

- $M \equiv N_1 N_2$. Let $[s_1\,;s_2] \in [\![M]\!] = \langle [\![N_1]\!], [\![N_2]\!]\rangle\,;\psi\,;T\mathsf{ev}\,;\mu$, and $[s_1] \in \langle [\![N_1]\!], [\![N_2]\!]\rangle$, $[s_2] \in \psi\,;T\mathsf{ev}\,;\mu$. As before, it suffices to show that $[s_1] \in \langle [\![N_1]\!], [\![N_2]\!]\rangle^{\backsim_\kappa}$, in the particular case of s_1 being an even-length P-view. We have that, wlog, s_1 is obtained from some $[s_1'] \in \mathtt{viewf}([\![N_1]\!])$ with the addition of $[\![N_2]\!]$'s initial answer (which is non-introducing). By IH, $s_1' = t_1'^{\backsim_\kappa}$ for $[t_1'] \in [\![(N_1)^{\circledast}]\!]$ and hence we obtain a $[t_1] \in [\![\langle (N_1)^{\circledast}, (N_2)^{\circledast}\rangle]\!]$ such that $s_1 = t_1{}^{\backsim_\kappa}$.

- $M \equiv !\,N$. In this case, assuming M is typed as $\vec{a} \mid \Gamma \vdash M : A$,

$$[\![M]\!] = [\![N]\!]\,;T\mathtt{drf}\,;\mu \quad \text{and} \quad [\![(M)^{\circledast}]\!] = \mathtt{nu}\,;T\tfrac{\vec{a}a}{\vec{a}}\,;T[\![!(N)^{\circledast}]\!]\,;\mu$$

  with $a \in \mathbb{A}_\kappa$. Let now $[s] \in \mathtt{viewf}([\![M]\!])$, so $s = s_1\,;s_2$ with $[s_1] \in [\![N]\!]$ and $[s_2] \in T\mathtt{drf}\,;\mu$. By IH, $s_1 = u_1{}^{\backsim_\kappa}$ for $[u_1] \in [\![(N)^{\circledast}]\!]$ with $u_1 \asymp s_2$. Hence, $s = t^{\backsim_\kappa}$ with $[t] = [u_1\,;s_2] \in [\![!(N)^{\circledast}]\!]$.
  Now, $[\![!(N)^{\circledast}]\!] = \eta\,;T[\![!(N)^{\circledast}]\!]\,;\mu$, hence $t = t_1\,;t_2$ with $[t_1] \in \eta$ and $[t_2] \in T[\![!(N)^{\circledast}]\!]\,;\mu$. t being a P-view implies that

$$t_1 = (\vec{a}, i_\Gamma) * \circledast\,(\vec{a}, i_\Gamma, \circledast)\,v_1$$

  with v_1 not containing any O-moves justified by the initial $(\vec{a}, i_\Gamma) * \circledast$. Hence, we can see that

$$t_1' \triangleq (\vec{a}, i_\Gamma) * \circledast\,(\vec{a}, i_\Gamma, \circledast)^a\,v_1^a$$

  is a play in $\mathtt{nu}\,;T\tfrac{\vec{a}a}{\vec{a}}$, for any fresh $a \in \mathbb{A}_\kappa$. Therefore, $[t_1'\,;t_2] \in \mathtt{nu}\,;T\tfrac{\vec{a}a}{\vec{a}}\,;T[\![!(N)^{\circledast}]\!]\,;\mu$, and hence $[s] \in [\![(M)^{\circledast}]\!]^{\backsim_\kappa}$. ∎

We have now gathered all the ingredients for proving the following.

Proposition 4.44 (Adequacy) *$(\mathcal{V}_{\mathtt{t}}, T, Q)$ is adequate: for any typed term $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$, if there exists some S such that $[\![M]\!] = \langle\!\langle \vec{b} \rangle\!\rangle\,[\![\bar{S}\,;0]\!]$ then there exists S' such that $\vec{a} \Vdash M \;\longrightarrow\!\!\!\!\!\rightarrow\; S' \vdash 0$.*

Proof: By lemma 4.35 it suffices to show that, for any such M, there is a non-reducing sequent $S' \vdash N$ such that $\vec{a} \Vdash M \;\longrightarrow\!\!\!\!\!\rightarrow\; S' \vdash N$, as then N would be a closed value of type $\mathbb{N}$ such that $[\![\bar{S}'\,;N]\!] \in O$ — and therefore $N \equiv 0$. But then it suffices to show that there is no infinite reduction sequence starting from $\vec{a} \Vdash M$ and containing infinitely many DRF reduction steps: leaving DRF's aside we are left with a $s\nu$-calculus with a non-recursive effect, which is strongly normalising for closed terms (cf. theorem 2.15).
So let $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$ be a typed term such that $[\![M]\!] = \langle\!\langle \vec{b} \rangle\!\rangle\,[\![\bar{S}\,;0]\!]$, for some S, and assume that $\vec{a} \Vdash M$ diverges using infinitely many DRF reduction steps. Then, $\vec{a} \Vdash (M)^{\circledast}$ diverges using infinitely many NEW reduction steps. Now, we have that $[(\vec{a}, *) * \circledast\,(0, \circledast)^{\vec{b}}] \in [\![M]\!]$ and hence, by previous lemma, there exists some $\vec{b}' \succeq \vec{b}$ such that $[(\vec{a}, *) * \circledast\,(0, \circledast)^{\vec{b}'}] \in [\![(M)^{\circledast}]\!]$. However, $\vec{a} \Vdash (M)^{\circledast}$ can reduce to some $S' \Vdash M'$ using $|\vec{b}'| + 1$ NEW reduction steps, so $[\![(M)^{\circledast}]\!] = \langle\!\langle \vec{c} \rangle\!\rangle\,[\![\bar{S}'\,;M']\!]$ with $|\vec{c}| = |\vec{b}'| + 1$, $\lightning$. ∎

Hence, $(\mathcal{V}_{\mathtt{t}}, T, Q)$ is a sound model for $\nu\rho$ and thus, for all terms M, N,

$$[\![M]\!] = [\![N]\!] \implies M \lessapprox N\,.$$

4.3.5 Tidy strategies

Leaving adequacy behind, the route for obtaining a fully abstract model of $\nu\rho$ proceeds to *definability*. That is, we aim for a model in which elements with *finite descriptions* correspond to translations of $\nu\rho$-terms.

However, $\mathcal{V}_{\mathtt{t}}$ does not satisfy such a requirement: it includes (finitary) store-related behaviours that are disallowed in the operational semantics of $\nu\rho$. In fact, our strategies treat

the store ξ like any other arena, while in $\nu\rho$ the treatment of store follows some basic guidelines. For example, if a store S is updated to S' then the original store S is not accessible any more (*irreversibility*). In strategies we do not have such a condition: in a play there may be several ξ's opened, yet there is no discipline on which of these are accessible to Player whenever he makes a move. Another condition involves the fact that a store either 'knows' the value of a name or it doesn't know it. Hence, when a name is asked, the store either returns its value or it deadlocks: there is no third option. In a play, however, when Opponent asks the value of some name, Player is free to evade answering and play somewhere else!

To disallow such behaviours we will constrain total strategies with further conditions, defining thus what we call *tidy strategies*. But first, let us specify store-related moves inside type-translating nominal arenas.

Definition 4.45 Consider $\mathcal{V}_{\nu\rho}$, the full subcategory of $\mathcal{V}_{\mathfrak{t}}$ with objects given by:

$$Ob(\mathcal{V}_{\nu\rho}) \ni A, B ::= 1 \mid \mathbb{N} \mid \mathbb{A}^{\vec{a}} \mid A \otimes B \mid A \multimap TB\,.$$

For each such arena A we define its set of ***store-Handles***, H_A, as follows.

$$\begin{aligned} H_1 = H_{\mathbb{N}} = H_{\mathbb{A}^{\vec{a}}} &\triangleq \varnothing\,,\\ H_{A\otimes B} &\triangleq H_A \cup H_B\,,\\ H_{A\multimap TB} &\triangleq \{(i_A, \circledast_A), (i_B, \circledast_B)\} \cup H_A \cup H_B \cup H_{\xi_A} \cup H_{\xi_B}\,, \quad \text{with } H_\xi \triangleq \bigcup\nolimits_C H_{[\![C]\!]}\,; \end{aligned}$$

where we write $A \multimap TB$ as $A \multimap (\xi_A \Rightarrow B \otimes \xi_B)$, and ξ as $\bigotimes_C(\mathbb{A}_C \Rightarrow [\![C]\!])$.
In an arena $A \in Ob(\mathcal{V}_{\nu\rho})$, a store-Handle justifies (all) questions of the form a, which we call ***store-Questions***. Answers to store-Questions are called ***store-Answers***. ▲

Note in particular that, for each type A, we have $[\![A]\!], Q^{\vec{a}}[\![A]\!], T[\![A]\!] \in Ob(\mathcal{V}_{\nu\rho})$, assuming that $T[\![A]\!]$ is equated with $1 \multimap T[\![A]\!]$. Note also there is a circularity in $H_{A\multimap TB}$ in the above definition. In fact, it suggests a definition by induction: we take $H_A \triangleq \bigcup_{i\in\omega} H^i_A$ and,

$$\begin{aligned} H^i_1 = H^i_{\mathbb{N}} = H^i_{\mathbb{A}^{\vec{a}}} = H^0_A &\triangleq \varnothing\,,\\ H^i_{A\otimes B} &\triangleq H^i_A \cup H^i_B\,,\\ H^{i+1}_{A\multimap TB} &\triangleq \{(i_A, \circledast_A), (i_B, \circledast_B)\} \cup H^i_A \cup H^i_B \cup H^{i+1}_{\xi_A} \cup H^{i+1}_{\xi_B}\,, \quad \text{with } H^{i+1}_\xi \triangleq \bigcup\nolimits_C H^i_{[\![C]\!]}\,. \end{aligned}$$

Intuitively, store-H's are store-opening moves, while store-Q's and store-A's are obtained from unfolding the store structure. Below we give examples of store-related moves in a simple arena.

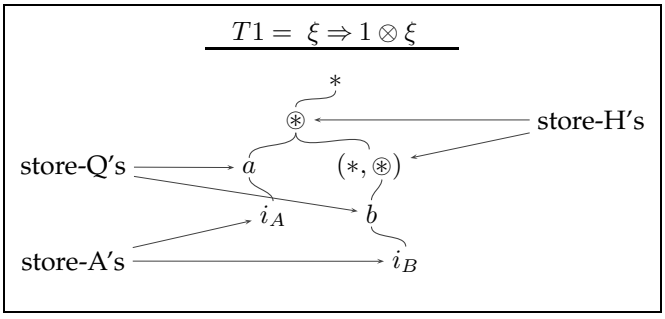


Figure 4.9: Store-H's -Q's -A's in arena $T1$.

From now on we work in $\mathcal{V}_{\nu\rho}$, unless stated otherwise. A first property we can show is that a move is exclusively either initial or a store-H -Q -A.

Proposition 4.46 *For any $A \in Ob(\mathcal{V}_{\nu\rho})$,*

$$M_A = I_A \uplus H_A \uplus \{m \in M_A \mid m \text{ a store-Q}\} \uplus \{m \in M_A \mid m \text{ a store-A}\}\,.$$

Proof: We show that any $m \in M_A$ belongs to exactly one of the above sets. We do induction on the level of m, $l(m)$, inside A and on the size of A, $|A|$, specified by the inductive definition of $Ob(\mathcal{V}_{\nu\rho})$. If m is initial then, by definition, it can't be a store-H. Neither can it be a store-Q or store-A, as these moves presuppose non-initiality.
Assume $l(m) > 0$. If A is base then trivial, while if $A = A_1 \otimes A_2$ then use the IH on $(l(m), |A|)$. Now, if $A = A_1 \multimap TA_2$ then let us write A as $A_1 \multimap (\xi_1 \Rightarrow A_2 \otimes \xi_2)$; we have the following cases.

- If $m = (i_{A_1}, \circledast_1) \in H_A$ then m a question and not a store-Q, as store-Q's are simply names.
- If $m = (i_{A_2}, \circledast_2) \in H_A$ then m an answer and not a store-A as its justifier is $(i_{A_1}, \circledast_1)$.
- If m is in A_1 or in A_2 then use the IH.
- If m is in ξ_1 then it is either some store-Q a to $(i_{A_1}, \circledast_1)$ (and hence not a store-H or store-A), or it is in some $[\![C]\!]$. In the latter case, if m initial in $[\![C]\!]$ then a store-A in $[\![A]\!]$ and therefore not a store-H, as m not a store-H in $[\![C]\!]$ by IH (on $l(m)$). If m is non-initial in $[\![C]\!]$ then use the IH and the fact that store-H's -Q's -A's of $[\![C]\!]$ are the same in $[\![A]\!]$.
- Similarly if m is in ξ_2. ■

The notion of store-Handles can be straightforwardly extended to prearenas.

Definition 4.47 Let $A, B \in Ob(\mathcal{V}_{\nu\rho})$. The set $H_{A\to B}$ of store-Handles in prearena $A \to B$ is $H_A \cup H_B$. Store-Q's and store-A's are defined accordingly. ▲

Using the previous proposition, we can see that, for any A and B, the set $M_{A\to B}$ can be decomposed as:

$$I_A \uplus I_B \uplus H_{A\to B} \uplus \{m \in M_{A\to B} \mid m \text{ a store-Q}\} \uplus \{m \in M_{A\to B} \mid m \text{ a store-A}\}\,.$$

We proceed to define tidy strategies. We endorse the following notational convention. Since stores ξ may occur in several places inside a (pre)arena we may use parenthesised indices to distinguish identical moves from different stores. For example, the same store-Question q may be occasionally denoted $q_{(O)}$ or $q_{(P)}$, the particular notation denoting the OP-polarity of the move. Moreover, by O-store-H's we mean store-H's played by Opponent, etc.

Definition 4.48 A total strategy σ is ***tidy*** if whenever odd-length $[s] \in \sigma$ then:

(TD1) If s ends in a store-Q q then $[sx] \in \sigma$, with x being either a store-A to q introducing no new names, or a copy of q. In particular, if $q = a^{\vec{a}}$ with $a \# \ulcorner s \urcorner^-$ then the latter case holds.

(TD2) If $[sq_{(P)}] \in \sigma$ with q a store-Q then $q_{(P)}$ is justified by last O-store-H in $\ulcorner s \urcorner$.

(TD3) If $\ulcorner s \urcorner = s' q_{(O)} q_{(P)} t\, y_{(O)}$ with q a store-Q then $[sy_{(P)}] \in \sigma$ with $y_{(P)}$ justified by $\ulcorner s \urcorner .{-}3$. ▲

(TD1) states that, whenever Opponent asks the value of a name, Player either immediately answers with its value or it copycats the question to the previous store-H. The former case corresponds to Player having updated the given name lastly (i.e. between the previous O-store-H and the last one). The latter case corresponds to Player not having done so and hence asking its value to the previous store configuration, starting thus a copycat between the last and the previous store-H. Hence, the store is, in fact, composed by layers of stores — one on top of the other — and only when a name has not been updated in the top layer is Player allowed to search for it in layers underneath. We can say that this is the nominal games equivalent of a *memory cell* (cf. remark 4.34). (TD3) further guarantees the above-described behaviour. It states that when Player starts a store-copycat then he must copycat the store-A and all following moves he receives, unless Opponent chooses to play elsewhere.

(TD2) guarantees the multi-layer discipline in the store: Player can see one store at each time, namely the last played by Opponent in the P-view.

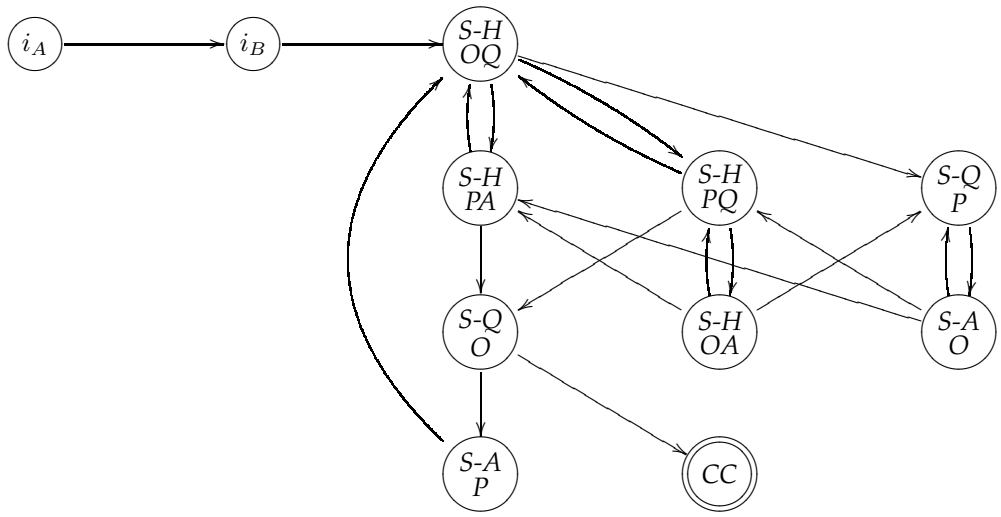
The following straightforward result shows that (TD3), as stated, provides the intended copycat behaviour.

Proposition 4.49 *Let σ be a tidy strategy. If $[s'q_{(O)}q_{(P)}t] \in \sigma$ is an even-length P-view and q is a store-Q then $q_{(O)}q_{(P)}t$ is a copycat.*

Proof: We do induction on $|t|$. The base case is straightforward. For the inductive step, let $t = t'xz$. Then, by prefix closure, $[s'q_{(O)}q_{(P)}t'x] \in \sigma$, this latter a P-view. By IH, $q_{(O)}q_{(P)}t'$ is a copycat. Moreover, by (TD3), $[s'q_{(O)}q_{(P)}t'xx] \in \sigma$ with last x justified by $(q_{(O)}q_{(P)}t'x).\text{-}3$, thus $s'q_{(O)}q_{(P)}t'xx$ a copycat. Now, by determinacy, $[s'q_{(O)}q_{(P)}t'xx] = [s'q_{(O)}q_{(P)}t'xz]$, so there exists π such that $\pi \circ x = x \wedge \pi \circ x = z$, $\therefore\ x = z$, as required. ■

A *good store discipline* would guarantee that store-Handles OP-alternate in a play. This indeed happens in P-views played by tidy strategies. In fact, such P-views have canonical decompositions, as we show below.

Proposition 4.50 (Tidy Discipline) *Let $\sigma : A \longrightarrow B$ be a tidy strategy and $[s] \in \sigma$ with $\ulcorner s \urcorner = s$. Then, s is decomposed as in the following diagram.*



(by CC we mean the state that, when reached by a sequence $s = \ulcorner s \urcorner$, the rest of s is copycat.)

Proof: The first two transitions are clear. After them neither P nor O can play initial moves, so all remaining moves in s are store-H -Q -A's. Assume now O has just played a question x_0 which is a store-H and the play continues with moves $x_1x_2x_3\ldots$.
x_1 cannot be a store-A, as this would not be justified by x_0, breaching well-bracketing. If x_1 is a store-Q then x_2 must be a store-A, by P-view. If x_1 is an answer-store-H then x_2 is an OQ, while if x_1 a question-store-H then x_2 is either a store-Q or a store-H.
If x_2 is a store-Q then, by (TD1), x_3 either a store-A or a store-Q, the latter case meaning transition to the CC state. If x_2 is not a store-Q then x_3 can't be a store-A: if x_3 were a store-A justified by $q \neq x_2$ then, as q wouldn't have been immediately answered, $s_{\geq q}$ would be a copycat and therefore we would be in the CC state right after playing q.
Finally, if x_3 is a store-A then x_4 must be justified by it, so it must be a Q-store-H. ■

Corollary 4.51 (Good Store Discipline) *Let $[s] \in \sigma$ with σ tidy and $\ulcorner s \urcorner = s$. Then:*

- *The subsequence of s containing its store-H's is OP-alternating and O-starting.*
- *If $s.\text{-}1 = q$ is a P-store-Q then either q is justified by last store-H in s, or s is in copycat mode at q.* ■

Our next aim is to show that $\nu\rho$ is modelled inside the subcategory of $\mathcal{V}_{\nu\rho}$ with tidy strategies. We first need to show that tidy strategies indeed form a subcategory of $\mathcal{V}_{\nu\rho}$, and then that all the structure necessary for the $\nu\rho$-model is available in tidy strategies. The following proposition gives equivalent definitions of tidy strategies, which will be of use in the sequel.

Proposition 4.52 *Let σ be a strategy.*

1. *σ is tidy iff whenever odd-length $[s] \in \sigma$ then* (TD1,2,3′) *hold, where:*

   (TD3′) If $\ulcorner s \urcorner = s' q_{(O)} q_{(P)} t\, y_{(O)}$ with q a store-Q and $q_{(O)} q_{(P)} t$ a copycat then $[s y_{(P)}] \in \sigma$ with $y_{(P)}$ justified by $\ulcorner s \urcorner .{-}3$.

2. **(a)** *σ is tidy iff whenever odd-length $[s] \in \sigma$ with $\ulcorner s \urcorner = s$ then* (TD1,2,3) *hold,*

   (b) *σ is tidy iff whenever odd-length $[s] \in \sigma$ with $\ulcorner s \urcorner = s$ then* (TD1,2,3′) *hold.*

Proof: For 1, it suffices to show that whenever σ satisfies (TD1,2,3′) and $[s' q_{(O)} q_{(P)} t\, y] \in \sigma$ is a P-view and q a store-Q, then $q_{(O)} q_{(P)} t$ is a copycat. But this is shown exactly as proposition 4.49, replacing "by (TD3)" by "by (TD3′)".
For 2, we show only the first part, the other part is shown similarly. We need only show the "if" direction. So assume the RHS hypothesis and let odd-length $[s] \in \sigma$, so $[\ulcorner s \urcorner] \in \sigma$.
If s ends in a store-Q q, then so does $\ulcorner s \urcorner$, so $[\ulcorner s \urcorner\, x] \in \sigma$, with x being a store-A not introducing new names or a copy of q. But x non-introducing and $[s], [\ulcorner s \urcorner\, x] \in \sigma$ implies $[sx] \in \sigma$, by lemma 3.38. If, in particular, $q = a$ with $a \# \ulcorner s \urcorner^{-}$ then x is a copy of q.
If $[sq] \in \sigma$ with q a store-Q then $[\ulcorner s \urcorner\, q] \in \sigma$ so q justified by last O-store-H in $\ulcorner s \urcorner$.
If $\ulcorner s \urcorner = s' q_{(O)} q_{(P)} t y$ with q a store-Q then $[\ulcorner s \urcorner\, y] \in \sigma$ with last y justified by $\ulcorner s \urcorner .{-}3$. By lemma 3.38, $[sy] \in \sigma$, as required. ∎

We can now show that strategies which 'mostly do copycats' are tidy.

Corollary 4.53 *A strategy σ is tidy if, for any odd-length $[s] \in \sigma$ with $\ulcorner s \urcorner = s$ and $|s| \geq 5$:*

1. *If $\forall x.\, [sx] \notin \sigma$ then $s.{-}1$ is not a store-Q and there are no consecutive store-Q's $q_{(O)} q_{(P)}$ inside s.*

2. *If $[sx] \in \sigma$ and $x \neq s.{-}1$ then s doesn't contain any $q_{(O)} q_{(P)}$ and also:*

   (a) *if $s.{-}1$ a store-Q then x is an answer to $s.{-}1$ not introducing new names and $s.{-}1 = a^{\vec{a}}$ with $a \in \mathtt{S}(s^{-})$,*

   (b) *if x a store-Q then it is justified by last O-store-H in s.*

3. *If $[sx] \in \sigma$ and $x = s.{-}1$ then one of the following is the case:*

   (a) *s doesn't contain any $q_{(O)} q_{(P)}$, and if x a store-Q then justified by last O-store-H in s;*

   (b) *x is justified by $s.{-}3$.*

Proof: By proposition 4.52, it suffices to show that such an s satisfies (TD1,2,3). For (TD1), if $s.{-}1$ a store-Q then $|s| \geq 5$ and, by 1, $[sx] \in \sigma$, for some x. If x is not a copy of $s.{-}1$ then, by 2a, $s.{-}1$ is not a fresh name and x is a non-intro answer to $s.{-}1$, as required. For (TD3), if $s = s' q_{(O)} q_{(P)} t\, y_{(O)}$ then $|s| \geq 7$ and, by 1,2, $[sy] \in \sigma$ and, by 3, y is justified by $s.{-}3$. Finally, for (TD2), if $[sx] \in \sigma$ with x a store-Q then $|s| \geq 3$. If $|s| = 3$ then x necessarily justified by $s.{-}1$ and that is the last O-store-H in s, as $s.1, s.2$ are initial moves. If $|s| \geq 5$ then either cases 2b,3a apply, or $x = s.{-}1$ and x justified by $s.{-}3$. In the latter case, $s.{-}1$ is an O-store-Q justified by $s.{-}2$, hence $s.{-}3$ is the last O-store-H in s, as required. ∎

Thus, for example, identity arrows are tidy as they fall under case 3b. In fact, as we will show later, all important structure is tidy. Let us now proceed to closure of tidy strategies under composition.

Lemma 4.54 *Let $\sigma : A \longrightarrow B$ and $\tau : B \longrightarrow C$ be tidy strategies, and let $[s\,;t] \in \sigma\,;\tau$, $[s] \in \sigma$ and $[t] \in \tau$, with $\ulcorner s \parallel t \urcorner = s \parallel t$ ending in a generalised O-move in AB and x, an O-move, being the last store-H in $\ulcorner s \urcorner$. Let x appear in $s \parallel t$ as $\tilde{x}$. Then, $\tilde{x}$ is the last store-H in $s \parallel t$ and if x is in A then all moves after $\tilde{x}$ in $s \parallel t$ are in A. Similarly for BC and t.*

Proof: We show the (AB, s) case, the other case being entirely dual. Let $s = s_1 x s_2$ and let x appear in $s \parallel t$ as some $\tilde{x}$. If x is in A then we claim that s_2 is in A. Suppose otherwise, so $s = s_1 x s_{21} y s_{22}$ with s_{21} in A and y a P-move in B. Since x appears in $\ulcorner s \urcorner$, the whole of $s_{21}y$ appears in it, as it is in P-view mode already. Since x is last store-H in $\ulcorner s \urcorner$, $s_{21}y$ is store-H-less. If y a store-Q then it should be justified by last O-store-H in $\ulcorner s_{<y} \urcorner$, that is x, which is not possible as x is in A. Thus, y must be a store-A, say to some O-store-Q q in B. Now, since q wasn't immediately answered by P, tidiness dictates that $\ulcorner s \urcorner$ be a copycat from move q and on. But then the move following x in s must be a copy of x in B, $\lightning$. Hence, s_2 is in A and therefore it appears in $\ulcorner s \urcorner$, which implies that it is store-H-less. Thus, $\tilde{x}$ is last store-H in $s \parallel t$.

If x is in B then we do induction on $|s \parallel t|$. The base case is encompassed in the case of s_2 being empty, which is trivial. So let $s_2 = s_{21} y s_{22} z$ with y justifying z (since x appears in $\ulcorner s \urcorner$, z has to be justified in s_2). z is not a store-H and neither is it a store-Q, as then y would be a store-H after x in $\ulcorner s \urcorner$. Thus z a store-A and y a store-Q, the latter justified by last O-store-H in $\ulcorner s_{<y} \urcorner = \ulcorner s \urcorner_{<y}$, that is x, so y, z in B. Now, $s = s_1 x s_{21} y s_{22} z$ and $t = t_1 x' t_{21} y' t_{22} z'$; we claim that s_{21} and t_{21} are store-H-less. Indeed, $s_{<y} \parallel t_{<y'}$ ends in a generalised O-move in AB and x is still the last store-H in $\ulcorner s_{<y} \urcorner$, from which we have, by IH, that $\tilde{x}$ is the last store-H in $s_{<y} \parallel t_{<y'}$.

Thus, $s \parallel t = (s_1 \parallel t_1)\tilde{x} v \tilde{y} u \tilde{z}$ with v store-H-less. It suffices to show that u is also store-H-less. In fact, $u = \underbrace{\tilde{y} \ldots \tilde{y}}_{n}\,\underbrace{\tilde{z} \ldots \tilde{z}}_{n}$ for some $n \geq 0$. Indeed, by tidiness of τ, $(t_{22} z').1$ is either an answer to y', whence $t_{22} = u = \epsilon$, or a copy of it under the last O-store-H in $\ulcorner t_{\leq y'} \urcorner$. If the latter is in B then σ reacts analogously, and so on, so there is initially a sequence $\tilde{y} \ldots \tilde{y}$ in u, played in B. As u finite, at some point σ (or τ) either answers y (y') or copycats it in A (in C). In the latter case, O immediately answers, as s (t) is in P-view mode in A (in C). Hence, in either cases there is an answer that is copycatted to all open $\tilde{y}$ in u, yielding thus the required pattern. Therefore, u is store-H-less. ∎

Lemma 4.55 *Let $\sigma : A \longrightarrow B$ and $\tau : B \longrightarrow C$ be tidy strategies, and let $[s\,;t] \in \sigma\,;\tau$, $[s] \in \sigma$ and $[t] \in \tau$, with $\ulcorner s \parallel t \urcorner = s \parallel t$ ending in a generalised O-move. If there exists $i \geq 1$ and store-Q's $\tilde{q}_1, ..., \tilde{q}_i$ with $\tilde{q} = \tilde{q}_j$, all $1 \leq j \leq i$, and $\tilde{q}_1, ..., \tilde{q}_{i-1}$ in B and $\tilde{q}_i$ in AC and $[(s \parallel t)\tilde{q}_1 ... \tilde{q}_i] \in \sigma \parallel \tau$, then $\tilde{q}_i$ is justified by the last O-store-H in $s\,;t$.*

Proof: By induction on $|s \parallel t|$. The base case is encompassed in the case of $s\,;t$ containing at most one O-store-H, which is trivial. Now let wlog $(s \parallel t)\tilde{q}_1 ... \tilde{q}_i = (sq_1 ... q_i) \parallel (tq'_1 ... q'_{i-1})$ with $[sq_1 ... q_i] \in \sigma$ and $[tq'_1 ... q'_{i-1}] \in \tau$, and let each q_j be justified by x_j and each q'_j by x'_j. Moreover, by hypothesis, $\underline{x_j} = \underline{x'_j}$, for $1 \leq j \leq i-1$, and therefore each such pair x_j, x'_j appears in $s \parallel t$ as some $\tilde{x}_j$, the latter justifying $\tilde{q}_j$ in $s \parallel t$.

Now, assume wlog that $s \parallel t$ ends in AB. Then, by tidiness of σ and τ we have that, for each $j \geq 1$,

$$q_{2j+1} = q_{2j} \quad , \quad q'_{2j} = q'_{2j-1} \quad , \quad \underline{q_j} = \underline{q'_j}$$

For each $j \geq 1$, q_{2j+1} is a P-move of σ justified by some store-H, say x_{2j+1}. By tidiness of σ, x_{2j+1} is the last O-store-H in $\ulcorner s_{<q_{2j+1}} \urcorner = \ulcorner s_{\leq q_{2j}} \urcorner$, and therefore x_{2j+1} is the last store-H in $\ulcorner s_{<x_{2j}} \urcorner$. Then, by previous lemma, $\tilde{x}_{2j+1}$ is the last store-H in $s_{<x_{2j}} \parallel t_{<x'_{2j}} = (s \parallel t)_{<\tilde{x}_{2j}}$. Similarly, $\tilde{x}_{2j}$ is the last store-H in $(s \parallel t)_{<\tilde{x}_{2j-1}}$. Hence, the store-H subsequence of $(s \parallel t)_{\leq \tilde{x}_1}$ ends in $\tilde{x}_i ... \tilde{x}_1$.

Now, by tidiness of σ, x_1 is the last O-store-H in $\ulcorner s \urcorner$. If x_1 is also the last store-H in $\ulcorner s \urcorner$ then, by previous lemma, $\tilde{x}_1$ is the last store-H in $s \parallel t$, hence $\tilde{x}_i$ is the last store-H in $s\,;t$. Otherwise, by corollary 4.51, q_1 is a copy of $s.{-}1 = q_0$. If q_0 is in A then its justifier is $s.{-}2 = x_0$ and,

because of CC-mode, the store-H subsequence of $s \parallel t$ ends in $\tilde{x}_i ... \tilde{x}_1 \tilde{x}_0$, so $\tilde{x}_i$ is the last O-store-H in $s;t$. If q_0 is in B then we can use the IH on $s^- \parallel t^-$ and $\tilde{q}_0, \tilde{q}_1, ..., \tilde{q}_i$, and obtain that $\tilde{x_i}$ is the last O-store-H in $s^-;t^- = s;t$. ∎

Proposition 4.56 *If $\sigma : A \longrightarrow B$ and $\tau : B \longrightarrow C$ are tidy strategies then so is $\sigma;\tau$.*

Proof: Take odd-length $[s;t] \in \sigma;\tau$ with not both s and t ending in B, $\ulcorner s \parallel t \urcorner = s \parallel t$ and $|s;t|$ odd. We need to show that $s;t$ satisfies (TD1-3). As (TD2) is a direct consequence of the previous lemma, we nos the other two conditions. Assume wlog that $s;t$ ends in A.
For (TD1), assume $s;t$ ends in a store-Q $\tilde{q}$. Then s ends in some q, which is justified by the P-store-H $s.\text{–}2 = x$ (also in A). q is either answered or copied by σ; in particular, if $\tilde{q} = a^{\vec{a}}$ with $a \# \ulcorner s;t \urcorner^- = s^-;t$ then $a \# s^-, t$, so σ copies q. If σ answers q with z then z doesn't introduce new names, so $[(s;t)\tilde{z}] \in \sigma;\tau$ with $\mathsf{nlist}(\tilde{z}) = \mathsf{nlist}(\tilde{q})$ and $\underline{\tilde{z}} = \underline{z}$, as required.
Otherwise, let σ copy q as q_1, say, under last O-store-H in $\ulcorner s \urcorner$, say x_1. If x_1 is in B then, by lemma 3.41, $sq_1 \asymp tq_1'$, with q_1, q_1' in B and q_1' being $\underline{q_1}$ with name-list that of its justifier, say x_1', where $\underline{x_1} = \underline{x_1'}$. Now $[tq_1'] \in \tau$ and it ends in a store-Q, so τ either answers it or copies it under last O-store-H in $\ulcorner tq_1' \urcorner$. In particular, if $q = a^{\vec{a}}$ with $a \# \ulcorner s;t \urcorner$ then, as above, $a \# t$ and τ copies q_1'. This same reasoning can be applied consecutively, with copycats attaching store-Q's to store-H's appearing each time earlier in s and t. As the latter are finite and initial store-H's are third moves in s and t, at some point either σ plays q_i in A or answers it in B, or τ plays q_i' in C or answers it in B. If an answer occurs then it doesn't introduce new names (by tidiness), so it is copycatted back to q closing all open q_j's and q_j''s. Otherwise, we need only show that, for each j, $\tilde{q}_j = \tilde{q}$, which we do by induction on j: $\tilde{q}_1 = \underline{q}^{s \bullet t, \epsilon}$ and $\tilde{q}_{j+1} = \underline{q}^{(s_{\leq q_j}) \bullet (t_{\leq q_j'}), \epsilon} = \tilde{q}_j \overset{IH}{=} \tilde{q}$. This proves (TD1).
For (TD3), assume $s;t = u\tilde{q}_{(O)}\tilde{q}_{(P)}v\tilde{y}$ with $\tilde{q}_{(O)}\tilde{q}_{(P)}v$ a copycat. Then, either both $\tilde{q}_{(O)}, \tilde{q}_{(P)}$ are in A, or one is in A and the other in C. Let's assume $\tilde{q}_{(O)}$ in A and $\tilde{q}_{(P)}$ in C — the other cases are shown similarly. Then, $\tilde{q}_{(O)}$ her(editarily)-justifies $\tilde{y}$, and let $s.\text{–}1 = y$ be justified by some x in s. Now, as above, $\tilde{q}_{(O)}\tilde{q}_{(P)}$ is witnessed by some $\tilde{q}_{(O)}\tilde{q}_1 \ldots \tilde{q}_i\tilde{q}_{(P)}$ in $s \parallel t$, with odd $i \geq 1$ and all $\tilde{q}_j$'s in B. We show by induction on $1 \leq k \leq i$ that there exist $x_1, ..., x_k, x_1', ..., x_k', y_1, ..., y_k, y_1', ..., y_k'$ in B such that $(sy_1 \ldots y_k \parallel ty_1' \ldots y_k') \in \sigma \parallel \tau$ and, for each relevant $j \geq 1$,

$$\underline{y_j} = \underline{y_j'} = \underline{y} \quad , \quad y_1 = y \quad , \quad y_{2j} = y_{2j+1} \quad , \quad y_{2j-1}' = y_{2j}' \quad , \quad \underline{x_j} = \underline{x_j'}$$

with q_j her-justifying x_j in s and x_j justifying y_j (and q_j' her-justifying x_j' in t and x_j' justifying y_j'), and $\tilde{x}_{j+1}, \tilde{x}_j$ consecutive in $s \parallel t$, and $\tilde{x}_1, \tilde{x}$ also consecutive.
For $k = 1$, let $s = s_1 q_{(O)} q_1 s_2 y$. Now, $\tilde{q}_{(O)}$ her-justifying $\tilde{y}$ implies that $q_{(O)}$ her-justifies y, hence it appears in $\ulcorner s \urcorner$. Thus $\ulcorner s \urcorner = s_1' q_{(O)} q_1 s_2' y$, so, by (original definition of) tidiness, $[sy_1] \in \sigma$ with $y_1 = y$ justified by $x_1 = \ulcorner s \urcorner.\text{–}3 = s.\text{–}3$. By lemma 3.41, $[ty_1'] \in \tau$ with $\underline{y_1'} = \underline{y_1}$. By proposition 4.49, $q_{(O)} q_1 s_2'$ is a copycat, so q_1 her-justifies x_1 and therefore x_1, y_1 in B. Finally, $x = \ulcorner s \urcorner.\text{–}2 = s.\text{–}2$ is a P-move so $\tilde{x}_1, \tilde{x}$ are consecutive in $s \parallel t$.
For even $k > 1$ we have, by IH, that $(sy_1 \ldots y_{k-1} \parallel ty_1' \ldots y_{k-1}') \in \sigma \parallel \tau$ with y_{k-1}' an O-move her-justified by q_{k-1}', an O-move. Then, q_{k-1}' appears in $\ulcorner ty_1' ... y_{k-1}' \urcorner$, so $\ulcorner ty_1' ... y_{k-1}' \urcorner = t_1 q_{k-1}' q_k' t_2 y_{k-1}'$, thus (by tidiness) $[ty_1' ... y_{k-1}' y_k'] \in \tau$ with $y_k' = y_{k-1}'$ justified by $x_k' = \ulcorner ty_1' ... y_{k-1}' \urcorner.\text{–}3$. Now, $q_{k-1}' q_k' t_2$ is a copycat so q_k' her-justifies x_k'. Moreover, x_k', x_{k-1}' are consecutive in $\ulcorner t \urcorner$, so, as x_{k-1}' a P-move, they are consecutive in t, and therefore $\tilde{x}_k, \tilde{x}_{k-1}$ consecutive in $s \parallel t$. Finally, by lemma 3.41, $[sy_1 \ldots y_{k-1} y_k] \in \sigma$ with $\underline{y_k} = \underline{y_k'}$. The case of k odd is entirely dual.
Now, working as above, we can show that there exist x_{i+1}', y_{i+1}' in C such that $[ty_1' ... y_i' y_{i+1}'] \in \tau$ and y_{i+1}' justified by x_{i+1}', x_{i+1}' her. justified by $q_{(P)}$, etc. Then $[(s;t)\tilde{y}_{i+1}] \in \sigma;\tau$ with $\tilde{x}_{i+1}, \tilde{x}_i, ..., \tilde{x}_1, \tilde{x}$ consecutive in $s \parallel t$, so $\tilde{x}_{i+1} = (s;t).\text{–}3$. Finally, as above, $\tilde{y}_{i+1} = \tilde{y}_j = \tilde{y}$, all j, as required. ∎

Hence, we can define our category of nominal arenas and tidy strategies.

Definition 4.57 $\mathcal{T}$ is the lluf subcategory of $\mathcal{V}_{\nu\rho}$ of tidy strategies. ▲

We now check that all structure required for a sound $\nu\rho$-model pass from $\mathcal{V}_{\mathtt{t}}$ to $\mathcal{T}$.

Proposition 4.58 ($\mathcal{T}$ an adequate model) *$\mathcal{T}$ forms an adequate $\nu\rho$-model by inheriting the necessary structure from $\mathcal{V}_{\mathtt{t}}$:*

I. *Projections and terminal arrows are tidy, and arrow pairing preserves tidiness.*

II. *η_A, τ_A, μ_A are tidy, and if h is tidy then so is Th. Moreover, Λ^T preserves and reflects tidiness.*

III. *Successor, predecessor, numeral and conditional arrows are tidy.*

IV. *$\varepsilon_A, \delta_A, \zeta_A$ are tidy, and if h is tidy then so is $Q^{\vec{a}}h$, for any $\vec{a}$. Moreover, $\left(\frac{\vec{a}}{\vec{a}'}\right)_A$ and $\mathtt{nu}_A$ are tidy.*

V. *Name-equality arrows are tidy.*

VI. *$\mathtt{upd}_A, \mathtt{drf}_A$ are tidy.*

Proof: Items III and V involve strategies with plays of length less than 3, hence tidy. The same holds for terminal arrows in I and $\left(\frac{\vec{a}}{\vec{a}'}\right)_1$ in IV. From corollary 4.53 we have that projections in I; η_A, τ_A, μ_A in III; $\mathtt{nu}_A$ in IV; and $\mathtt{upd}_A, \mathtt{drf}_A$ in VI are all tidy.
Now let $f : A \to B$, $g : A \to C$ be tidy strategies. Then,

$$\begin{aligned}\mathtt{viewf}(\langle f, g\rangle) = \{\, [i_A\,(i_B, i_C)\,s] \mid ([i_A\,i_B\,s] \in \mathtt{viewf}(f) \wedge [i_A\,i_C\,] \in g) \\ \vee\, ([i_A\,i_C\,s] \in \mathtt{viewf}(g) \wedge [i_A\,i_B\,] \in f)\,\}\end{aligned}$$

So let odd-length $[s] \in \langle f, g\rangle$ with $\ulcorner s \urcorner = s$ and $|s| \geq 5$, say wlog $s = i_A\,(i_B, i_C)\,s'\,x$ with $[i_A\,i_B\,s'] \in \mathtt{viewf}(f)$ and $[i_A\,i_C] \in g$. It is not difficult to see then that $[s_f] \in f$, for $s_f = i_A\,i_B\,s'\,x$. If x is a store-Q then $[s_f\,y] \in \mathtt{viewf}(f)$ for a relevant y, and so $[s\,y] \in \mathtt{viewf}(\langle f, g\rangle)$. We need only check the case of $x = a^{\vec{a}}$ with $a \# s^-$, which implies $a \# s_f^-$ and hence y a copy of x, as required. Now, if $[s\,y] \in \mathtt{viewf}(\langle f, g\rangle)$ with y a store-Q then $[s_f\,y] \in \mathtt{viewf}(f)$, so y is justified by last O-store-H in s_f, and hence y justified by last O-store-H in s. Finally, if $s = i_A\,(i_B, i_C)\,s''q_{(O)}\,q_{(P)}\,t\,x_{(O)}$ then $s_f = s = i_A\,i_B\,s''q_{(O)}\,q_{(P)}\,t\,x_{(P)}$, and therefore $[s_f\,x] \in \mathtt{viewf}(f)$ with $x_{(P)}$ justified by $s_f.\text{-}3$ and $[s\,x_{(P)}] \in \mathtt{viewf}(\langle f, g\rangle)$, as required. Hence, $\langle f, g\rangle$ tidy.
In along the same lines we can prove that T preserves tidiness, and that Λ^T preserves and reflects tidiness. Moreover, tidiness of product-related constructs implies tidiness of structural arrows in the comonads.
Finally, adequacy is clearly inherited from $\mathcal{V}_{\mathtt{t}}$. ■

Henceforth, by strategies we shall mean tidy strategies, unless stated otherwise.

4.3.6 Observationality

Strategy equality is *too fine grained* to capture contextual equivalence in a complete manner. For example, even simple contextual equivalences like

$$\mathtt{skip} \approxeq \nu a.\mathtt{skip}$$

are not preserved by the semantical translation, since strategies include in their name-lists all introduced names, even useless ones. For similar reasons, equivalences like

$$\nu a.\nu b.M \approxeq \nu b.\nu a.M$$

are not valid semantically. It is not only because of the treatment of name-creation that the semantics is not complete. The 'explicit' way in which the store works distinguishes equivalences like

$$a := 1\,;\lambda x.\,!\,a\,;2 \;\approxeq\; a := 1\,;\lambda x.2\,.$$

Thus, there are many ways in which our semantics is too expressive for our language. We therefore proceed to apply a quotienting by the intrinsic preorder and prove full-abstraction in the extensional model.

Following the steps described in section 4.2.2, in this section we introduce the intrinsic preorder on $\mathcal{T}$ and show that the resulting model is observational. Full-abstraction is then shown in the following section.

Definition 4.59 Expand $\mathcal{T}$ to $(\mathcal{T}, T, Q, O)$ by setting, for each $\vec{a} \in \mathbb{A}^{\#}$,

$$O^{\vec{a}} \triangleq \{f \in \mathcal{T}(Q^{\vec{a}}1, T\mathbb{N}) \mid \exists \vec{b}.\, [(\vec{a}, *) * \circledast (0, \circledast)^{\vec{b}}] \in f\}\,.$$

Then, for each $f, g \in \mathcal{T}(Q^{\vec{a}}A, TB)$, $f \lesssim^{\vec{a}} g$ if

$$\forall \rho : Q^{\vec{a}}(A \multimap TB) \longrightarrow T\mathbb{N}.\ (\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{\vec{a}}(g)\,;\rho \in O^{\vec{a}})\,. \qquad \blacktriangle$$

Thus, the observability predicate O is a family $(O^{\vec{a}})_{\vec{a}\in\mathbb{A}^{\#}}$, and the intrinsic preorder $\lesssim$ is a family $(\lesssim^{\vec{a}})_{\vec{a}\in\mathbb{A}^{\#}}$. Recall that by $\Lambda^{\vec{a}}(f)$ we mean $\Lambda^{Q^{\vec{a}},T}(f)$, that is,

$$\Lambda^{\vec{a}}(f) = Q^{\vec{a}}1 \xrightarrow{\delta} Q^{\vec{a}}Q^{\vec{a}}1 \xrightarrow{Q^{\vec{a}}\Lambda(\zeta'\,;\,f)} Q^{\vec{a}}(A \multimap TB)\,.$$

In particular, if $f \sqsubseteq g$ then $\Lambda^{\vec{a}}(f) \sqsubseteq \Lambda^{\vec{a}}(g)$ and therefore $\Lambda^{\vec{a}}(f)\,;\rho \sqsubseteq \Lambda^{\vec{a}}(g)\,;\rho$, which implies:

$$f \sqsubseteq g \implies f \lesssim^{\vec{a}} g \tag{4.16}$$

The intrinsic preorder is defined by use of *test arrows* ρ, which stand for possible program contexts. As the following result shows, not all such tests are necessary.

Lemma 4.60 (tl4 tests suffice) *Let $f, g \in \mathcal{T}(Q^{\vec{a}}1, B)$ with B pointed. The following are equivalent.*[5]

I. $\forall \rho : Q^{\vec{a}}B \longrightarrow T\mathbb{N}.\ \delta\,;Q^{\vec{a}}f\,;\rho \in O^{\vec{a}} \implies \delta\,;Q^{\vec{a}}g\,;\rho \in O^{\vec{a}}\,.$

II. $\forall \rho : Q^{\vec{a}}B \longrightarrow T\mathbb{N}.\ \rho \textit{ is tl4} \implies (\delta\,;Q^{\vec{a}}f\,;\rho \in O^{\vec{a}} \implies \delta\,;Q^{\vec{a}}g\,;\rho \in O^{\vec{a}})\,.$

Hence, for each $\vec{a}$ and $f, g \in \mathcal{T}(Q^{\vec{a}}A, TB)$, $f \lesssim^{\vec{a}} g$ iff

$$\forall \rho : Q^{\vec{a}}(A \multimap TB) \longrightarrow T\mathbb{N}.\ \rho \textit{ is tl4} \implies (\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{\vec{a}}(g)\,;\rho \in O^{\vec{a}})\,.$$

Proof: I $\Rightarrow$ II is trivial. Now assume II holds and let $\rho : Q^{\vec{a}}B \longrightarrow T\mathbb{N}$ be any strategy such that $\delta\,;Q^{\vec{a}}f\,;\rho \in O^{\vec{a}}$. Then, there exist $[s] \in \delta\,;Q^{\vec{a}}f$ and $[t] \in \rho$ such that $[s\,;t] = [(\vec{a}, *) * \circledast (0, \circledast)^{\vec{b}}] \in (\delta\,;Q^{\vec{a}}f)\,;\rho$. We show by induction on the number of J_B-moves appearing in $s \parallel t$ that $\delta\,;Q^{\vec{a}}g\,;\rho \in O^{\vec{a}}$.
If no such moves appear then $t = (\vec{a}, i_B) * \circledast (0, \circledast)^{\vec{b}}$, so done. If $n+1$ such moves appear then ρ is necessarily t4, as B is pointed, so by lemma 3.58 there exists tl4* strategy $\tilde{\rho}$ such that $\rho = \Delta\,;\tilde{\rho}$. It is not difficult to see that ρ being tidy implies that $\tilde{\rho}$ is tidy. Moreover, $\delta\,;Q^{\vec{a}}f\,;\rho = \delta\,;Q^{\vec{a}}f\,;\Delta\,;\tilde{\rho} = \delta\,;Q^{\vec{a}}f\,;\langle \mathtt{id}, Q^{\vec{a}}!\,;\delta\,;Q^{\vec{a}}f\rangle\,;\tilde{\rho} = \delta\,;Q^{\vec{a}}f\,;\rho'$, with ρ' being $\langle \mathtt{id}, Q^{\vec{a}}!\,;\delta\,;Q^{\vec{a}}f\rangle\,;\tilde{\rho}$. Now, by definition of $\tilde{\rho}$, $[(\vec{a}, *) * \circledast (0, \circledast)^{\vec{b}}] = [s'\,;t'] \in \delta\,;Q^{\vec{a}}f\,;\rho'$ with $s' \parallel t'$ containing n J_B-moves so, by IH, $\delta\,;Q^{\vec{a}}g\,;\rho' \in O^{\vec{a}}$. But $\delta\,;Q^{\vec{a}}g\,;\rho' = \delta\,;Q^{\vec{a}}g\,;\langle \mathtt{id}, Q^{\vec{a}}!\,;\delta\,;Q^{\vec{a}}f\rangle\,;\tilde{\rho} = \delta\,;Q^{\vec{a}}f\,;\langle Q^{\vec{a}}!\,;\delta\,;Q^{\vec{a}}g, \mathtt{id}\rangle\,;\tilde{\rho} = \delta\,;Q^{\vec{a}}f\,;\rho''$, with ρ'' being $\langle Q^{\vec{a}}!\,;\delta\,;Q^{\vec{a}}g, \mathtt{id}\rangle\,;\tilde{\rho}$. But ρ'' is tl4, thus, by hypothesis, $O^{\vec{a}} \ni \delta\,;Q^{\vec{a}}g\,;\rho'' = \delta\,;Q^{\vec{a}}g\,;\rho$, as required. ■

We can now prove the second half of observationality.

[5] Recall, from definition 3.51, that a total strategy $\sigma : A \longrightarrow B$ is:

- *l4* if whenever $[s] \in \sigma$ and $\underline{s.{-}1} \in J_A$ then $|\ulcorner s \urcorner| = 4$,
- *t4* if for any $[i_A\, i_B\, j_B] \in \sigma$ there exists $[i_A\, i_B\, j_B\, j_A^{\vec{b}}] \in \sigma$,
- *tl4* if it is both t4 and l4,
- *ttotal* if it is tl4 and for any $[i_A\, i_B\, j_B] \in \sigma$ there exists $[i_A\, i_B\, j_B\, j_A] \in \sigma$.

Lemma 4.61 *For any $f : Q^{\vec{a}'}1 \longrightarrow B$ and any tl4 morphism $\rho : Q^{\vec{a}}B \longrightarrow T\mathbb{N}$, with B pointed and $\vec{a} \subseteq \vec{a}'$,*

$$\delta\,;Q^{\vec{a}}\langle\vec{a}\,|\,\vec{a}'\rangle f\,;\rho \in O^{\vec{a}} \iff \delta\,;Q^{\vec{a}'}f\,;\tfrac{\vec{a}'}{\vec{a}}\,;\rho \in O^{\vec{a}'}.$$

Moreover, for all relevant f, g and $\vec{a} \subseteq \vec{a}'' \subseteq \vec{a}'$,

$$\begin{aligned} f \lesssim^{\vec{a}'} g &\implies \langle\vec{a}\,|\,\vec{a}'\rangle f \lesssim^{\vec{a}} \langle\vec{a}\,|\,\vec{a}'\rangle g\,,\\ f \lesssim^{\vec{a}} g &\implies \tfrac{\vec{a}'}{\vec{a}}\,;f \lesssim^{\vec{a}'} \tfrac{\vec{a}'}{\vec{a}}\,;g\,,\\ \tfrac{\vec{a}'}{\vec{a}''}\,;f \lesssim^{\vec{a}'} g &\implies \langle\vec{a}\,|\,\vec{a}''\rangle f \lesssim^{\vec{a}} \langle\vec{a}\,|\,\vec{a}'\rangle g\,. \end{aligned}$$

In particular, $f \lesssim^{\vec{a}a} g \implies \langle a\rangle f \lesssim^{\vec{a}} \langle a\rangle g$.

Proof: For the first part, ρ being tl4 and B being pointed imply that there exists some $\vec{b} \# \vec{a}$ and a ttotal strategy ρ' such that $\rho = \langle\!\langle\vec{b}\rangle\!\rangle\,\rho'$. Now let $\delta\,;Q^{\vec{a}}\langle\vec{a}\,|\,\vec{a}'\rangle f\,;\rho \in O^{\vec{a}}$, so there exists $[s\,;t] = [(\vec{a},*) \ast \circledast\,(0,\circledast)^{\vec{b}(\vec{a}'\setminus\vec{a})\vec{c}}] \in (\delta\,;Q^{\vec{a}}\langle\vec{a}\,|\,\vec{a}'\rangle f)\,;\rho$, and let $s = (\vec{a},*)\,(\vec{a},i_B)\,j_B\,m^{(\vec{a}'\setminus\vec{a})\vec{c}}s'$ and $t = (\vec{a},i_B) \ast \circledast\,j_B^{\vec{b}}\,t'$. Letting $s'^{\smallsetminus\alpha}$ be $\underline{s'}^{\mathsf{nlist}(s')\setminus(\vec{a}'\setminus\vec{a})}$, we can see that $[(\vec{a}',*)\,i_B\,j_B\,m^{\vec{c}}s'^{\smallsetminus\alpha}] \in f$ and thus $[s''] \triangleq [(\vec{a}',*)\,(\vec{a},i_B)\,j_B\,m^{\vec{c}}s'^{\smallsetminus\alpha}] \in \delta\,;Q^{\vec{a}'}f\,;\frac{\vec{a}'}{\vec{a}}$. Hence, $[s''\,;t] = [(\vec{a}',*) \ast \circledast\,(0,\circledast)^{\vec{b}\vec{c}}] \in \delta\,;Q^{\vec{a}'}f\,;\frac{\vec{a}'}{\vec{a}}\,;\rho$, as required. The converse is shown similarly.
For the second part, suppose $f \lesssim^{\vec{a}'} g : Q^{\vec{a}'}A \longrightarrow TB$ and take any tl4 morphism $\rho : Q^{\vec{a}}(A \multimap TB) \longrightarrow T\mathbb{N}$. Then,

$$\begin{aligned} \Lambda^{\vec{a}}(\langle\vec{a}\,|\,\vec{a}'\rangle f)\,;\rho \in O^{\vec{a}} &\iff \delta\,;Q^{\vec{a}}\Lambda(\zeta'\,;\langle\vec{a}\,|\,\vec{a}'\rangle f)\,;\rho \in O^{\vec{a}} \overset{(3.12)}{\iff} \delta\,;Q^{\vec{a}}\langle\vec{a}\,|\,\vec{a}'\rangle(\Lambda(\zeta'\,;f))\,;\rho \in O^{\vec{a}}\\ &\iff \delta\,;Q^{\vec{a}'}\Lambda(\zeta'\,;f)\,;\tfrac{\vec{a}'}{\vec{a}}\,;\rho \in O^{\vec{a}'}\\ &\overset{f\lesssim^{\vec{a}'}g}{\Longrightarrow} \delta\,;Q^{\vec{a}'}\Lambda(\zeta'\,;g)\,;\tfrac{\vec{a}'}{\vec{a}}\,;\rho \in O^{\vec{a}'} \iff \Lambda^{\vec{a}}(\langle\vec{a}\,|\,\vec{a}'\rangle g)\,;\rho \in O^{\vec{a}}. \end{aligned}$$

For the next claim, it is easy to see that, for any $h : Q^{\vec{a}'}1 \longrightarrow T\mathbb{N}$, $h \in O^{\vec{a}'}$ iff $\binom{\vec{a}}{\vec{a}'}\,;h_\perp\,;\mathtt{pu} \in O^{\vec{a}}$. Hence, if $f \lesssim^{\vec{a}} g$ then we have:

$$\begin{aligned} \delta\,;Q^{\vec{a}'}\Lambda(\zeta'\,;\tfrac{\vec{a}'}{\vec{a}}\,;f)\,;\rho \in O^{\vec{a}'} &\iff \delta\,;Q^{\vec{a}'}\tfrac{\vec{a}'}{\vec{a}}\,;Q^{\vec{a}'}\Lambda(\zeta'\,;f)\,;\rho \in O^{\vec{a}'}\\ &\iff \tbinom{\vec{a}}{\vec{a}'}\,;(\delta\,;Q^{\vec{a}'}\tfrac{\vec{a}'}{\vec{a}}\,;Q^{\vec{a}'}\Lambda(\zeta'\,;f)\,;\rho)_\perp\,;\mathtt{pu} \in O^{\vec{a}}\\ &\overset{(N2')}{\iff} \delta\,;Q^{\vec{a}}\Lambda(\zeta'\,;f)\,;\tbinom{\vec{a}}{\vec{a}'}\,;\rho_\perp\,;\mathtt{pu} \in O^{\vec{a}}\\ &\overset{f\lesssim^{\vec{a}}g}{\Longrightarrow} \delta\,;Q^{\vec{a}}\Lambda(\zeta'\,;g)\,;\tbinom{\vec{a}}{\vec{a}'}\,;\rho_\perp\,;\mathtt{pu} \in O^{\vec{a}} \iff \delta\,;Q^{\vec{a}'}\Lambda(\zeta'\,;\tfrac{\vec{a}'}{\vec{a}}\,;g)\,;\rho \in O^{\vec{a}'}. \end{aligned}$$

For the last claim, if $\frac{\vec{a}'}{\vec{a}''}\,;f \lesssim^{\vec{a}'} g$ then $\langle\vec{a}\,|\,\vec{a}'\rangle(\frac{\vec{a}'}{\vec{a}''}\,;f) \lesssim^{\vec{a}} \langle\vec{a}\,|\,\vec{a}'\rangle g$, so it suffices to show $\langle\vec{a}\,|\,\vec{a}'\rangle(\frac{\vec{a}'}{\vec{a}''}\,;f) \backsimeq \langle\vec{a}\,|\,\vec{a}''\rangle f$. Now, observing that, for any $h : Q^{\vec{a}''}1 \longrightarrow T\mathbb{N}$, $h \in O^{\vec{a}''}$ iff $\frac{\vec{a}'}{\vec{a}''}\,;h \in O^{\vec{a}'}$, we have:

$$\begin{aligned} \delta\,;Q^{\vec{a}}\Lambda(\zeta'\,;\langle\vec{a}\,|\,\vec{a}'\rangle(\tfrac{\vec{a}'}{\vec{a}''}\,;f))\,;\rho \in O^{\vec{a}} &\overset{(3.12)}{\iff} \delta\,;Q^{\vec{a}}\langle\vec{a}\,|\,\vec{a}'\rangle\Lambda(\zeta'\,;\tfrac{\vec{a}'}{\vec{a}''}\,;f)\,;\rho \in O^{\vec{a}}\\ &\iff \delta\,;Q^{\vec{a}'}\Lambda(\zeta'\,;\tfrac{\vec{a}'}{\vec{a}''}\,;f)\,;\tfrac{\vec{a}'}{\vec{a}}\,;\rho \in O^{\vec{a}'}\\ &\iff \tfrac{\vec{a}'}{\vec{a}''}\,;\delta\,;Q^{\vec{a}''}\Lambda(\zeta'\,;f)\,;\tfrac{\vec{a}''}{\vec{a}}\,;\rho \in O^{\vec{a}'} \iff \delta\,;Q^{\vec{a}''}\Lambda(\zeta'\,;f)\,;\tfrac{\vec{a}''}{\vec{a}}\,;\rho \in O^{\vec{a}''}\\ &\iff \delta\,;Q^{\vec{a}}\langle\vec{a}\,|\,\vec{a}''\rangle\Lambda(\zeta'\,;f)\,;\rho \in O^{\vec{a}} \overset{(3.12)}{\iff} \delta\,;Q^{\vec{a}}\Lambda(\zeta'\,;\langle\vec{a}\,|\,\vec{a}''\rangle f)\,;\rho \in O^{\vec{a}}, \end{aligned}$$

as required. ∎

In order to prove that $\mathcal{T}$ is observational, we are only left to show that

$$[\![M]\!] \in O^{\vec{a}} \iff \exists\vec{b}, S.\,[\![M]\!] = \langle\!\langle\vec{b}\rangle\!\rangle\,[\![\bar{S}\,;0]\!]$$

for any $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$. The "$\Longleftarrow$" direction is trivial. For the converse, because of correctness, it suffices to show the following generalisation of adequacy.[6]

[6] At this point, notice that the proof of adequacy (proposition 4.44) is, in fact, a proof of O-adequacy. As we find both proofs interesting, we present them both regardless of the redundancy.

Lemma 4.62 (O-Adequacy) *Let $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$ be a typed term. If $[\![M]\!] \in O^{\epsilon}$ then there exists some S such that $\vec{a} \vDash M \longrightarrow\!\!\!\!\!\rightarrow S \vDash 0$.*

Proof: The idea behind the proof is the same as that of the proof of adequacy (v. section 4.3.4). It suffices to show that, for any such M, there is a non-reducing sequent $S \vDash N$ such that $\vec{a} \vDash M \longrightarrow\!\!\!\!\!\rightarrow S \vDash N$; therefore, because of Strong Normalisation in the $s\nu$-calculus, it suffices to show that there is no infinite reduction sequence starting from $\vec{a} \vDash M$ and containing infinitely many DRF reduction steps.
To show the latter we will use an operation on terms adding new-name constructors just before dereferencings. The operation yields, for each term M, a term $(M)^{\circ}$ the semantics of which is equivalent to that of M. On the other hand, $\vec{a} \vDash (M)^{\circ}$ cannot perform infinitely many DRF reduction steps without creating infinitely many new names. For each term M, define $(M)^{\circ}$ by induction as:

$$(a)^{\circ} \triangleq a\,, \quad (x)^{\circ} \triangleq x\,, \quad \ldots \quad (\lambda x.M)^{\circ} \triangleq \lambda x.(M)^{\circ}\,, \quad (M\,N)^{\circ} \triangleq (M)^{\circ}(N)^{\circ}\,, \quad \ldots$$

and $(!\,N)^{\circ} \triangleq \nu a.!(N)^{\circ}$, some $a \notin \mathtt{fn}(N)$.
We show that $[\![(M)^{\circ}]\!] \backsimeq [\![M]\!]$, by induction on M; the base cases are trivial. The induction step follows immediately from the IH and the fact that $\backsimeq$ is a congruence, in all cases except for M being $!\,N$. In the latter case we have that $[\![(M)^{\circ}]\!] = \langle a \rangle(\frac{\vec{a}a}{\vec{a}}\,; [\![!(N)^{\circ}]\!])$, while the IH implies that $[\![M]\!] \backsimeq [\![!(N)^{\circ}]\!]$. Hence, it sts that for each $f : Q^{\vec{a}}A \longrightarrow TB$ we have $f \backsimeq \langle a \rangle(\frac{\vec{a}a}{\vec{a}}\,; f)$. Indeed, for any relevant ρ which is tl4,

$$\begin{aligned}
\Lambda^{\vec{a}}(\langle a \rangle(\tfrac{\vec{a}a}{\vec{a}}\,; f))\,; \rho \in O^{\vec{a}} &\overset{\text{lem 4.61}}{\iff} \delta\,; Q^{\vec{a}a}\Lambda(\zeta'\,; \tfrac{\vec{a}a}{\vec{a}}\,; f)\,; \tfrac{\vec{a}a}{\vec{a}}\,; \rho \in O^{\vec{a}a} \\
&\iff \delta\,; Q^{\vec{a}a}\tfrac{\vec{a}a}{\vec{a}}\,; \tfrac{\vec{a}a}{\vec{a}}\,; Q^{\vec{a}}\Lambda(\zeta'\,; f)\,; \rho \in O^{\vec{a}a} \\
&\iff \tfrac{\vec{a}a}{\vec{a}}\,; \Lambda^{\vec{a}}(f)\,; \rho \in O^{\vec{a}a} \iff \Lambda^{\vec{a}}(f)\,; \rho \in O^{\vec{a}}\,.
\end{aligned}$$

Now, take any $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$ and assume $[\![M]\!] \in O^{\vec{a}}$, and that $\vec{a} \vDash M$ diverges using infinitely many DRF reduction steps. Then, $\vec{a} \vDash (M)^{\circ}$ diverges using infinitely many NEW reduction steps. However, since $[\![(M)^{\circ}]\!] \backsimeq [\![M]\!]$, we have $[\![(M)^{\circ}]\!] \in O^{\vec{a}}$ and therefore $[(\vec{a}, *) * \circledast\,(0, \circledast)^{\vec{b}}] \in [\![(M)^{\circ}]\!]$ for some $\vec{b}$. However, $\vec{a} \vDash (M)^{\circ}$ reduces to some $S \vDash M'$ using $|\vec{b}|+1$ NEW reduction steps, so $[\![(M)^{\circ}]\!] = \langle \vec{c} \rangle\, [\![\bar{S}\,; M']\!]$ with $|\vec{c}| = |\vec{b}|+1$, $\lightning$ to determinacy. ■

We have therefore shown observationality.

Proposition 4.63 (Observationality) *$(\mathcal{T}, T, Q, O)$ is observational.* ■

4.3.7 Definability and full-abstraction

We now proceed to show definability in our model $\mathcal{T}$, and through it ip-definability. According to the results of section 4.2.2, this will suffice for full abstraction.

We first make precise the notion of *finitary strategy*, that is, of strategy with finite description, by introducing truncation functions that remove inessential branches from a strategy's description.

Definition 4.64 Let $\sigma : A \longrightarrow B$ in $\mathcal{T}$ and let $[s] \in \mathtt{viewf}(\sigma)$ be of even length. Define $\mathtt{trunc}(s)$ and $\mathtt{trunc}'(s)$ by induction as follows.

$$\mathtt{trunc}(\epsilon) = \mathtt{trunc}'(\epsilon) \triangleq \epsilon$$

$$\mathtt{trunc}(x_{(O)}y_{(P)}s') \triangleq \begin{cases} \epsilon & \text{, if } x = y \text{ are store-Q's} \\ xy\,\mathtt{trunc}(s') & \text{, o.w.} \end{cases}$$

$$\mathtt{trunc}'(x_{(O)}y_{(P)}s') \triangleq \begin{cases} \epsilon & \text{, if } x = y \text{ are store-Q's} \\ \epsilon & \text{, if } x \text{ store-Q , } y \text{ a store-A and } s' = \epsilon \\ \epsilon & \text{, if } x \in I_A, y \in I_B \text{ and } s' = \epsilon \\ xy\,\mathtt{trunc}'(s') & \text{, o.w.} \end{cases}$$

Moreover, say σ is ***finitary*** if $\mathtt{trunc}(\sigma)$ is finite, where

$$\mathtt{trunc}(\sigma) \triangleq \{[\mathtt{trunc}(s)] \mid [s] \in \mathtt{viewf}(\sigma) \wedge |s| > 3\}\,.$$

Finally, for any $[t] \in \sigma$ define:

$$\sigma_{\leq t} \triangleq \mathtt{strat}\{[s] \in \mathtt{viewf}(\sigma) \mid \exists\, t' \leq t.\, \mathtt{trunc}'(s) = \ulcorner t' \urcorner\}\,.$$

▲

Hence, finitary are those strategies whose viewfunctions become finite if we delete all the store-copycats and all default initial answers — the latter dictated by totality. Moreover, the strategy $\sigma_{\leq t}$ is the strategy we are left with if we truncate $\mathtt{viewf}(\sigma)$ by removing all its branches of size greater than 3 that are not contained in t, except for the store-copycats which are left intact and for the store-A's branches which are truncated to the point of leaving solely the store-A, so that we retain tidiness.

Note that, in general, $\mathtt{trunc}'(s) \leq \mathtt{trunc}(s) \leq s$. We can then show the following.

Proposition 4.65 *If σ is a strategy and $[t] \in \sigma$ is even-length then $\sigma_{\leq t}$ is a finitary strategy with $[t] \in \sigma_{\leq t}$ and $\sigma_{\leq t} \sqsubseteq \sigma$.*

Proof: To show that $\sigma_{\leq t}$ is an innocent strategy we need to show that

$$f \triangleq \{[s] \in \mathtt{viewf}(\sigma) \mid \exists\, t' \leq t.\, \mathtt{trunc}'(s) = \ulcorner t' \urcorner\}$$

is a viewfunction. For even-prefix closure, if $s = s'xy$ and $[s] \in f$ then $[s] \in \mathtt{viewf}(\sigma)$ and $\mathtt{trunc}'(s) = \ulcorner t' \urcorner$, some $t' \leq t$. We have that $\mathtt{trunc}'(s') \leq \mathtt{trunc}'(s)$ so $\mathtt{trunc}'(s') = \ulcorner t'' \urcorner$, some $t'' \leq t' \leq t$, so $[s'] \in f$. Single-valuedness is clear, as $f \subseteq \mathtt{viewf}(\sigma)$. The latter shows also that $\sigma_{\leq t} \sqsubseteq \sigma$.
Totality is obvious. For tidiness, let odd-length $[s] \in \sigma_{\leq t}$ with $\ulcorner s \urcorner = s$. (TD2) clearly holds. For (TD1), if s ends in a store-Q then there exists $[sx] \in \sigma$ satisfying (TD1) and, as $\mathtt{trunc}'(s^-) = \mathtt{trunc}'(sx)$, we have $[sx] \in f$. The same reasoning resolves (TD3).
We now show $\mathtt{trunc}(f)$ is finite. Each $[s'] \in \mathtt{trunc}(f)$ is $[\mathtt{trunc}(s)]$ for some $[s] \in \mathtt{viewf}(\sigma)$ and $t' \leq t$ such that $|s| > 3$ and $\mathtt{trunc}'(s) = \ulcorner t' \urcorner$. The cases of $[\mathtt{trunc}(s)] = [\mathtt{trunc}'(s)]$ are finitely many, as t is of finite length. For the rest, $\mathtt{trunc}(s) = \mathtt{trunc}'(s)xy$, x a store-Q a with $a \in \mathtt{S}(\mathtt{trunc}'(s))$ and y a store-A. Hence, for each $t' \leq t$ there are not more than $|\mathtt{S}(\ulcorner t' \urcorner)|$-many elements added in $\mathtt{trunc}(f)$. Thus, $\sigma_{\leq t}$ is finitary.
Finally, for any even-length $t' \leq t$, we have that $\mathtt{trunc}'(\ulcorner t' \urcorner) \leq \ulcorner t' \urcorner$, so there exists some $t'' \leq t$ such that $\mathtt{trunc}'(\ulcorner t' \urcorner) = \ulcorner t'' \urcorner$ and hence $[\ulcorner t' \urcorner] \in \sigma_{\leq t}$. Then, by lemma 3.38, we have $[t] \in \sigma_{\leq t}$. ■

We proceed to show definability. The proof is facilitated by the following lemma. Note that for economy we define strategies by means of their viewfunctions modulo totality and even-prefix closure. Moreover, we write $\sigma \upharpoonright i$ for the (total) restriction of a strategy σ to an initial move i, and $s^{\smallsetminus \vec{b}}$ for s with $\vec{b}$ removed from all of its name-lists.

Lemma 4.66 (Decomposition Lemma) *Let $\sigma : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!]$ be a strategy. We can decompose σ as follows.*

1. *If there exists an $i_{A(0)}$ such that $\exists x_0.\, [(\vec{a}, i_{A(0)}) * \circledast x_0] \in \sigma$ then*

$$\sigma = Q^{\vec{a}}[\![A]\!] \xrightarrow{\langle [x \overset{\vec{a}}{=} i_{A(0)}], \langle \sigma_0, \sigma' \rangle \rangle} \mathbb{N} \otimes (T[\![B]\!])^2 \xrightarrow{\mathtt{cnd}} T[\![B]\!]$$

where:

$$\begin{aligned}
[x \overset{\vec{a}}{=} i_{A(0)}] : Q^{\vec{a}}[\![A]\!] \longrightarrow \mathbb{N} &\triangleq \{[(\vec{a}, i_{A(0)})\, 0]\} \cup \{[(\vec{a}, i_A)\, 1] \mid [(\vec{a}, i_A)] \neq [(\vec{a}, i_{A(0)})]\}\,,\\
\sigma_0 : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] &\triangleq \mathtt{strat}\{\, [(\vec{a}, i_{A(0)})\, s] \in \mathtt{viewf}(\sigma)\,\}\,,\\
\sigma' : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] &\triangleq \mathtt{strat}\{\, [(\vec{a}, i_A)\, s] \in \mathtt{viewf}(\sigma) \mid [(\vec{a}, i_A)] \neq [(\vec{a}, i_{A(0)})]\,\}\,.
\end{aligned}$$

2. *If there exists an* $i_{A(0)}$ *such that* $\forall i_A.\,(\exists x_0.\,[(\vec{a}, i_A) * \circledast\, x_0] \in \sigma) \iff [(\vec{a}, i_A)] = [(\vec{a}, i_{A(0)})]$, *then* $\sigma = \langle\!\langle \vec{b} \rangle\!\rangle\, \sigma_{\vec{b}}$, *where:*

$$\sigma_{\vec{b}} : Q^{\vec{a}\vec{b}}[\![A]\!] \longrightarrow T[\![B]\!] \triangleq \mathtt{strat}\{\, [(\vec{a}\vec{b}, i_{A(0)}) * \circledast\, m_0\, s^{\searrow \vec{b}}] \mid [(\vec{a}, i_{A(0)}) * \circledast\, m_0^{\vec{b}}\, s] \in \mathtt{viewf}(\sigma)\,\}\,.$$

3. *If there exist* $i_{A(0)}, m_0$ *such that* $\forall i_A, x.\,[(\vec{a}, i_A) * \circledast\, x] \in \sigma \iff [(\vec{a}, i_A)\, x] = [(\vec{a}, i_{A(0)})\, m_0]$, *then one of the following is the case.*

(a) $m_0 = a$, *a store-Q of type* C *under* $\circledast$, *in which case we have* $\sigma = \sigma' \upharpoonright (\vec{a}, i_{A(0)})$, *where:*

$$\sigma' \triangleq Q^{\vec{a}}[\![A]\!] \xrightarrow{\langle \mathtt{id},\, \phi\rangle} Q^{\vec{a}}[\![A]\!] \otimes T[\![C]\!] \xrightarrow{\tau\,;\, T\zeta'} TQ^{\vec{a}}([\![A]\!] \otimes [\![C]\!]) \xrightarrow{T\sigma_a} T^2[\![B]\!] \xrightarrow{\mu} T[\![B]\!]\,,$$

$$\sigma_a \triangleq \mathtt{strat}\{\, [(\vec{a}, i_{A(0)}, i_C) * \circledast\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, a\, i_C\, s] \in \mathtt{viewf}(\sigma)\,\}\,,$$

$$\phi : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![C]\!] \triangleq \begin{cases} Q^{\vec{a}}!\,;\, \frac{\vec{a}}{a}\,;\, \mathtt{drf}_C & \text{, if } a \in \mathtt{S}(\vec{a}) \\ Q^{\vec{a}}\pi_j\,;\, \frac{\vec{a}}{\epsilon}\,;\, \mathtt{drf}_C & \text{, if } a \# \vec{a}\,. \end{cases}$$

(b) $m_0 = j_A \vee m_0 = (i_B, \circledast)$, *a store-H, in which case if* $[(\vec{a}, i_{A(0)}) * \circledast\, m_0\, a\, i_C] \in \sigma$, *for some store-Q* a *and store-A* i_C, *then*

$$\sigma = Q^{\vec{a}}[\![A]\!] \xrightarrow{\langle \Delta,\, \sigma_a\rangle} Q^{\vec{a}}[\![A]\!] \otimes Q^{\vec{a}}[\![A]\!] \otimes T[\![C]\!] \xrightarrow{\tau;\, T(\mathtt{id}\otimes\phi;\, \tau);\, \mu} TQ^{\vec{a}}[\![A]\!] \xrightarrow{T\sigma'\,;\, \mu} T[\![B]\!]$$

where:

$$\sigma_a : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![C]\!] \triangleq \mathtt{strat}\{\, [(\vec{a}, i_{A(0)}) * \circledast\, (i_C, \circledast)\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, m_0\, a\, i_C\, s] \in \mathtt{viewf}(\sigma) \vee [\circledast\ \circledast\ s] \in \mathtt{viewf}(\mathtt{id}_\xi)\,\}\,,$$

$$\sigma' : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] \triangleq \mathtt{strat}(\, \{[(\vec{a}, i_{A(0)}) * \circledast\, m_0\, y\, s] \in \mathtt{viewf}(\sigma) \mid y \neq a\,\} \cup \{[(\vec{a}, i_{A(0)}) * \circledast\, m_0\, a\, s] \mid [\circledast\ \circledast\ a\, s] \in \mathtt{viewf}(\mathtt{id}_\xi)\}\,)\,,$$

$$\phi : Q^{\vec{a}}[\![A]\!] \otimes [\![C]\!] \longrightarrow T1 \triangleq \begin{cases} (Q^{\vec{a}}!\,;\, \frac{\vec{a}}{a}) \otimes \mathtt{id}_{[\![C]\!]}\,;\, \mathtt{upd}_C & \text{, if } a \in \mathtt{S}(\vec{a}) \\ (Q^{\vec{a}}\pi_j\,;\, \frac{\vec{a}}{\epsilon}) \otimes \mathtt{id}_{[\![C]\!]}\,;\, \mathtt{upd}_C & \text{, if } a \# \vec{a}\,. \end{cases}$$

In both cases above, we take $j = \min\{j \mid (i_{A(0)})_j = a\}$.

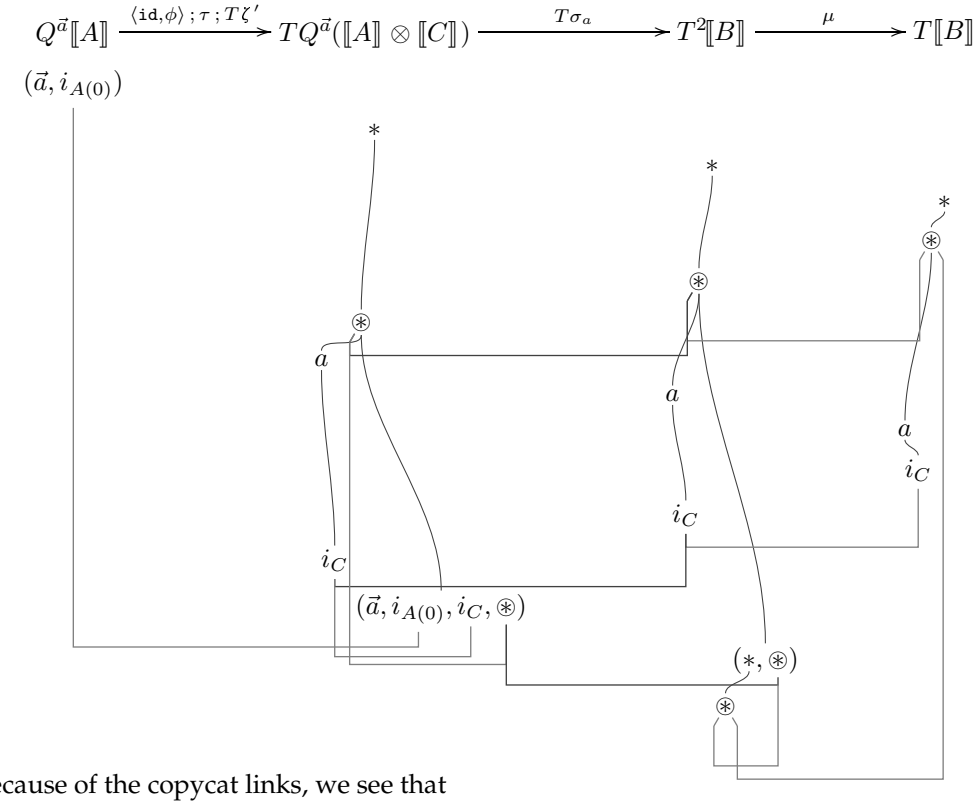
Proof: 1 is straightforward: we just partition σ into σ_0 and σ' and recover it by use of $[x \overset{\vec{a}}{=} i_{A(0)}]$ and `cnd`. For 2, we just use the definition of name-abstraction for strategies and the condition on σ.
For 3, it is clear that m_0 is either a store-Q a under $\circledast$, or a store-H j_A, or a store-H $(i_B, \circledast)$.

In case $m_0 = a$ with $a \in \mathbb{A}_C$, we define $\sigma_a : Q^{\vec{a}}([\![A]\!] \otimes [\![C]\!]) \longrightarrow T[\![B]\!] \triangleq \mathtt{strat}(f_a)$, where

$$f_a \triangleq \{\, [(\vec{a}, i_{A(0)}, i_C) * \circledast\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, a\, i_C\, s] \in \mathtt{viewf}(\sigma)\,\}\,.$$

To see that f_a is a viewfunction it suffices to show that its elements are plays, and for that it suffices to show that they are legal. Now, for any $[(\vec{a}, i_{A(0)}, i_C) * \circledast\, s] \in f_a$ with $[(\vec{a}, i_{A(0)}) * \circledast\, a\, i_C\, s] \in \mathtt{viewf}(\sigma)$, $(\vec{a}, i_{A(0)}, i_C) * \circledast\, s$ is a justified sequence and satisfies well-bracketing, as its open Q's outside s are the same as those in $(\vec{a}, i_{A(0)}) * \circledast\, a\, i_C\, s$, i.e. $\circledast$. Moreover, visibility is obvious. Hence, f_a is a viewfunction, and it inherits tidiness from σ. Moreover,

we have the following diagram.



Because of the copycat links, we see that

$$\begin{aligned}\mathtt{viewf}(\langle \mathtt{id}, \phi \rangle ; \tau ; T\zeta' ; T\sigma_a ; \mu) \upharpoonright (\vec{a}, i_{A(0)}) &= \{[(\vec{a}, i_{A(0)}) * \circledast\, a\, i_C\, s] \mid [(\vec{a}, i_{A(0)}, i_C) * \circledast\, s] \in \mathtt{viewf}(\sigma_a)\} \\ &= \mathtt{viewf}(\sigma)\,,\end{aligned}$$

as required. Note that the restriction to initial moves $[\vec{a}, i_{A(0)}]$ taken above is necessary in case ϕ contains a projection (in which case it may also answer other initial moves).

In case $m_0 = j_A$ (so m_0 a store-H) and $[(\vec{a}, i_{A(0)}) * \circledast\, m_0\, a\, i_C] \in \sigma$, we have that

$$\sigma = \mathtt{strat}(f_a \cup (f' \setminus f'_a))\,,$$

where f_a, f' are viewfunctions of type $Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!]$, so that f_a determines σ's behaviour if O plays a at the given point, and $f' \setminus f'_a$ determines σ's behaviour if O plays something else. That is,

$$\begin{aligned} f_a &\triangleq \{\, [(\vec{a}, i_{A(0)}) * \circledast\, j_A\, a\, i_C\, s] \in \mathtt{viewf}(\sigma) \,\} \\ f'_a &\triangleq \{\, [(\vec{a}, i_{A(0)}) * \circledast\, j_A\, a\, s] \mid [\circledast\ \circledast\ a\, s] \in \mathtt{viewf}(\mathtt{id}_\xi) \,\} \\ f' &\triangleq f'_a \cup \{\, [(\vec{a}, i_{A(0)}) * \circledast\, j_A\, y\, s] \in \mathtt{viewf}(\sigma) \mid y \neq a \,\}\,. \end{aligned}$$

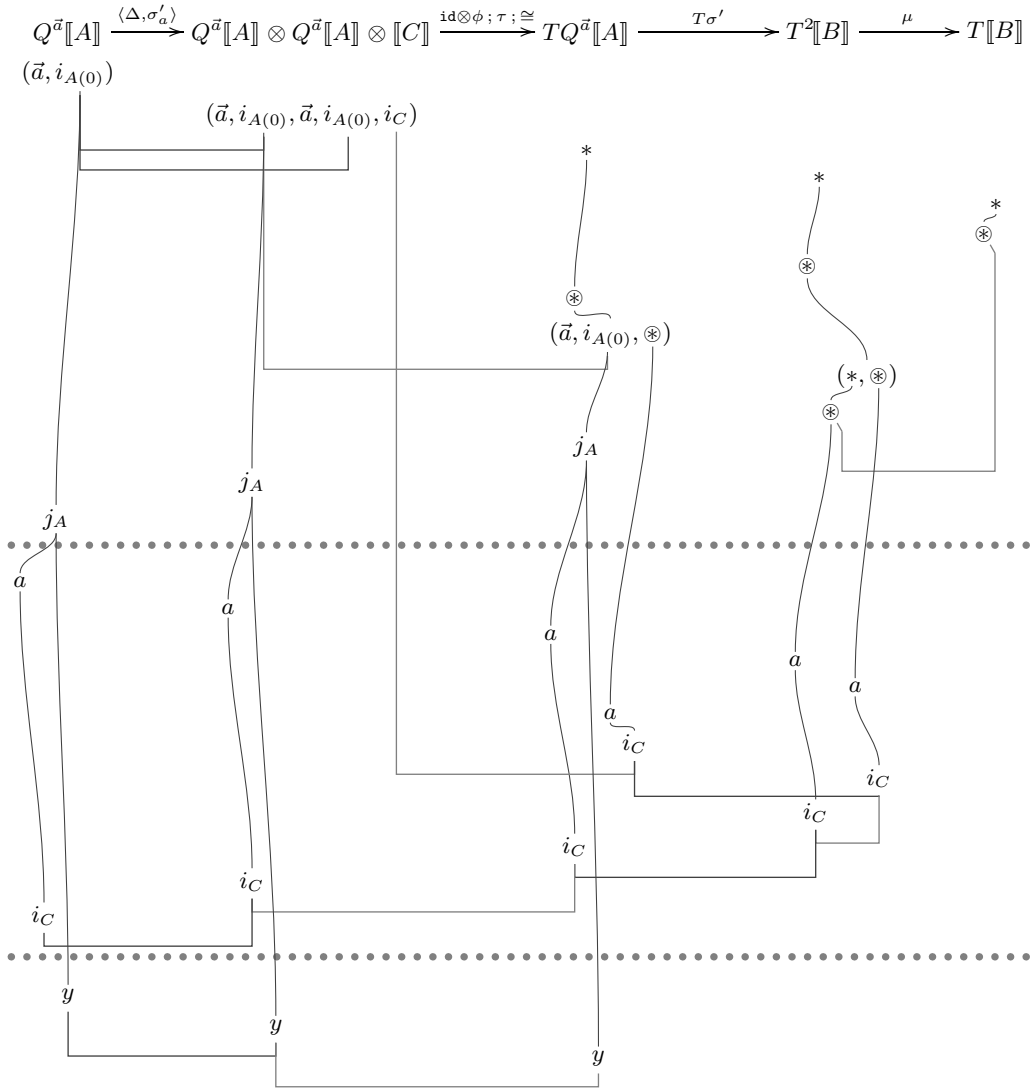
f' differs from $\mathtt{viewf}(\sigma)$ solely in the fact that it doesn't answer a but copycats it instead; it is a version of $\mathtt{viewf}(\sigma)$ which has forgotten the name-update of a. On the other hand, f_a contains exactly the information for this update. It is not difficult to see that f', f_a are indeed viewfunctions. We now define

$$\begin{aligned} f''_a : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![C]\!] &\triangleq \{\, [(\vec{a}, i_{A(0)}) * \circledast\, (i_C, \circledast)\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, j_A\, a\, i_C\, s] \in f_a \\ &\qquad\qquad \vee\, [\circledast\ \circledast\ s] \in \mathtt{viewf}(\mathtt{id}_\xi) \,\} \\ \sigma_a : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![C]\!] &\triangleq \mathtt{strat}(f''_a) \\ \sigma' : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] &\triangleq \mathtt{strat}(f') \\ \sigma'' : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] &\triangleq \langle \Delta, \sigma_a \rangle ; \tau ; T(\mathtt{id} \otimes \phi ; \tau) ; \mu ; \cong ; T\sigma' ; \mu\,. \end{aligned}$$

We can see that σ' is a tidy strategy. For σ_a, it suffices to show that f''_a is a viewfunction, since tidiness is straightforward. For that, we note that even-prefix closure and single-valuedness are clear, so it suffices to show that the elements of f''_a are plays.
So let $[(\vec{a}, i_{A(0)}) * \circledast (i_C, \circledast)\, s] \in f''_a$ with $[(\vec{a}, i_{A(0)}) * \circledast\, j_A\, a\, i_C\, s] \in \mathtt{viewf}(\sigma)$. We have that $(\vec{a}, i_{A(0)}) * \circledast (i_C, \circledast)\, s$ is a justified sequence, because s does not contain any moves justified by j_A or a. In the former case this holds because we have a P-view, and in the latter because a is a closed (answered) Q. Note also that there is no move in s justified by $\circledast$: such a move $(i_B, \circledast)$ would be an A ruining well-bracketing as j_A is an open Q, while a store-Q under $\circledast$ is disallowed by tidiness as $s.1$ is an O-store-H. Finally, well-bracketing and visibility are clear, while NC's follow from lemma 3.53.
We now proceed to show that $\sigma = \sigma''$. By the previous analysis on f''_a we have that $\sigma_a = \sigma'_a\,;\eta$ (modulo totality) where σ'_a is the possibly non-total strategy

$$\sigma'_a : Q^{\vec{a}}[\![A]\!] \longrightarrow [\![C]\!] \triangleq \mathtt{strat}\{\, [(\vec{a}, i_{A(0)})\, i_C\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, j_A\, a\, i_C] \in f_a \,\},$$

and hence $\sigma'' \upharpoonright (\vec{a}, i_{A(0)}) = \langle \Delta, \sigma'_a \rangle\,; \mathtt{id} \otimes \phi\,; \tau\,; \cong\,; T\sigma'\,; \mu$. We have the following diagram.



Following the copycat paths and observing that the response of σ'' to inputs different than

$[\vec{a}, i_{A(0)}]$ is merely the initial answer $*$ imposed by totality, we obtain:

$$\begin{aligned}\texttt{viewf}(\sigma'') &= \{\, [(\vec{a}, i_{A(0)}) * \circledast j_A\, a\, s], [(\vec{a}, i_{A(0)}) * \circledast j_A\, y\, s] \in \texttt{viewf}(\sigma'') \mid y \neq a \,\} \\ &= \{\, [(\vec{a}, i_{A(0)}) * \circledast j_A\, a\, i_C\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast (i_C, \circledast)\, s] \in f''_a \wedge s.1 \in J_{[\![C]\!]} \,\} \\ &\quad \cup \{\, [(\vec{a}, i_{A(0)}) * \circledast j_A\, y\, s] \in f' \mid y \neq a \,\} \\ &= f_a \cup (f' \setminus f'_a) = \texttt{viewf}(\sigma)\end{aligned}$$

as required.

In case $x = (i_B, \circledast)$ we work similarly as above. ∎

The proof of definability is a nominal version of standard definability results in game semantics. In fact, using the Decomposition Lemma we reduce the problem of definability of a finitary strategy σ to that of definability of a finitary strategy σ_0 of equal length, with σ_0 having no initial effects (i.e. fresh-name creation, name-update or name-dereferencing). On σ_0 we then apply almost verbatim the methodology of [HY99] — itself based on previous proofs of definability.

Theorem 4.67 (Definability) *Let A, B be types and $\sigma : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!]$ be finitary. Then σ is definable.*

Proof: We do induction on $(|\texttt{trunc}(\sigma)|, \|\sigma\|)$, where we let $\|\sigma\| \triangleq \max\{|\mathcal{L}(s)| \mid [s] \in \texttt{viewf}(\sigma)\}$, i.e. the maximum number of names introduced in any play of $\texttt{trunc}(\sigma)$. If $|\texttt{trunc}(\sigma)| = 0$ then $\sigma = [\![\texttt{stop}_B]\!]$; otherwise, there exist $x_0, i_{A(0)}$ such that $[(\vec{a}, i_{A(0)}) * \circledast x_0] \in \sigma$. By Decomposition Lemma,

$$\sigma = \langle [x \overset{\vec{a}}{=} i_{A(0)}], \langle \sigma_0, \sigma' \rangle \rangle ; \texttt{cnd}$$

with $|\texttt{trunc}(\sigma')| < |\texttt{trunc}(\sigma)|$ and $(0,0) < (|\texttt{trunc}(\sigma_0)|, \|\sigma_0\|) \leq (|\texttt{trunc}(\sigma)|, \|\sigma\|)$, so by IH there exists term M' such that $[\![M']\!] = \sigma'$. Hence, if there exist terms M_0, N_0 with $[\![M_0]\!] \upharpoonright (\vec{a}, i_{A(0)}) = \sigma_0$ and $[\![N_0]\!] = [x \overset{\vec{a}}{=} i_{A(0)}]; \eta$, then we can see that

$$\sigma = [\![\texttt{if0}\ N_0\ \texttt{then}\ M_0\ \texttt{else}\ M']\!] .$$

We first construct N_0. Assume that $A = A_1 \times A_2 \times \cdots \times A_n$ with A_i's non-products, and similarly $B = B_1 \times \cdots \times B_m$. Moreover, assume wlog that A is segmented in four parts: each of $A_1, ..., A_k$ is $\mathbb{N}$; each of $A_{k+1}, ..., A_{k+i}, ..., A_{k+k'}$ is $[A''_i]$; each of $A_{k+k'+1}, ..., A_{k+k'+i}, ..., A_{k+k'+k''}$ is $A'_i \to A''_i$; and the rest are all $\mathbb{1}$. Take $\vec{z}, \vec{z}', \vec{z}'', \vec{z}'''$ to be variable-lists of respective types. Define ϕ_0, ϕ'_0 by:

$\phi_0 \triangleq \kappa_1, ..., \kappa_k$, with $(\kappa_1, ..., \kappa_k)$ being the initial $\mathbb{N}$-segment of $i_{A(0)}$,

$$\phi'_0 \triangleq \kappa'_1, ..., \kappa'_{k'}, \text{ with each } \kappa'_i \triangleq \begin{cases} (i_{A(0)})_{k+i} & \text{, if } (i_{A(0)})_{k+i} \in \mathtt{S}(\vec{a}) \\ z'_j & \text{, if } (i_{A(0)})_{k+i} \# \vec{a} \\ & \quad \wedge\, j = \min\{j < i \mid (i_{A(0)})_{k+i} = (i_{A(0)})_{k+j}\} \\ \texttt{fresh}(i) & \text{, otherwise.} \end{cases}$$

$\texttt{fresh}(i)$ is a meta-constant denoting that Opponent has played a fresh name in A_{k+i}. If the same fresh name is played in several places inside $i_{A(0)}$ then we regard its leftmost occurrence as introducing it — this explains the second item in the cases-definition above. Now, define:

$$\begin{aligned} N_0 &\triangleq [\langle \vec{z}, \vec{z}' \rangle = \langle \phi_0, \phi'_0 \rangle], \quad \text{where} \\ [\langle \vec{z}, \vec{z}' \rangle = \langle \vec{\kappa}, \vec{\kappa}' \rangle] &\triangleq [z_1 = \kappa_1] \wedge \cdots \wedge [z_k = \kappa_k] \wedge [z'_1 = \kappa'_1] \wedge \cdots \wedge [z'_{k'} = \kappa'_{k'}], \\ [z' = \texttt{fresh}(i)] &\triangleq [z' \neq a_1] \wedge \cdots \wedge [z' \neq a_{|\vec{a}|}] \wedge [z' \neq z'_1] \wedge \cdots \wedge [z' \neq z'_{i-1}], \end{aligned}$$

with the logical connectives $\wedge$ and $\neg$ defined using $\texttt{if0}$'s, and $[z_i = \kappa_i]$ using $\texttt{pred}$'s, in the standard way. It is not difficult to show that indeed $[\![N_0]\!] \overset{\vec{a}}{=} [x = i_{A(0)}]; \eta$.

We proceed to find M_0. By second part of Decomposition Lemma, $\sigma_0 = \langle\!\langle \vec{b} \rangle\!\rangle \sigma_{\vec{b}}$ with $\vec{b} = \mathsf{nlist}(x_0)$, $|\mathtt{trunc}(\sigma_{\vec{b}})| = |\mathtt{trunc}(\sigma_0)|$ and $\|\sigma_{\vec{b}}\| = \|\sigma_0\| - |\vec{b}|$. If $|\vec{b}| > 0$ then, by IH, there exists term $M_{\vec{b}}$ such that $[\![M_{\vec{b}}]\!] = \sigma_{\vec{b}}$, so taking

$$M_0 \triangleq \nu\vec{b}.M_{\vec{b}}$$

we have $\sigma_0 = [\![M_0]\!]$.

Assume now $|\vec{b}| = 0$, so $x_0 = m_0$. σ_0 satisfies the hypotheses of the third part of the Decomposition Lemma. Hence, if $m_0 = a$, a store-Q of type C under $\circledast$, then

$$\sigma_0 = (\langle \mathtt{id}, \phi \rangle ; \tau ; T\zeta' ; T\sigma_a ; \mu) \upharpoonright (\vec{a}, i_{A(0)})$$

with $\mathtt{trunc}(\sigma_a) < \mathtt{trunc}(\sigma_0)$. Then, by IH, there exists $\vec{a} \mid \Gamma, y : C \vdash M_a : B$ such that $\sigma_a = [\![M_a]\!]$, and taking

$$M_0 \triangleq \begin{cases} (\lambda y.M_a)(!\, a) & \text{, if } a \in \mathtt{S}(\vec{a}) \\ (\lambda y.M_a)(!\, z'_j) & \text{, if } a \# \vec{a} \wedge j = \min\{j \mid a = (i_{A(0)})_{k+j}\} \end{cases}$$

we have $\sigma_0 = [\![M_0]\!] \upharpoonright (\vec{a}, i_{A(0)})$.

Otherwise, $m_0 = j_A \vee m_0 = (i_B, \circledast)$, a store-H. If there exists an $a \in \mathbb{A}_C$ such that σ_0 answers to $[i_{A(0)} * \circledast\, m_0\, a]$ then, by Decomposition Lemma,

$$\sigma_0 = \langle \Delta, \sigma_a \rangle ; \tau ; T(\mathtt{id} \otimes \phi ; \tau) ; \mu ; T\sigma' ; \mu$$

with $|\mathtt{trunc}(\sigma_a)|, |\mathtt{trunc}(\sigma')| < |\mathtt{trunc}(\sigma_0)|$. By IH, there exist $\vec{a} \mid \Gamma \vdash M_a : C$ and $\vec{a} \mid \Gamma \vdash M' : B$ such that $\sigma_a = [\![M_a]\!]$ and $\sigma' = [\![M']\!]$. Taking

$$M_0 \triangleq \begin{cases} (a := M_a); M' & \text{, if } a \in \mathtt{S}(\vec{a}) \\ (z'_j := M_a); M' & \text{, if } a \# \vec{a} \wedge j = \min\{j \mid a = (i_{A(0)})_{k+j}\} \end{cases}$$

we obtain $\sigma_0 = [\![M_0]\!]$. Note here that σ_a blocks initial moves $[\vec{a}, i_A] \neq [\vec{a}, i_{A(0)}]$ and hence we do not need the restriction.

We are left with the case of m_0 being as above and σ_0 not answering to any store-Q, which corresponds to the case of Player not updating any names before playing m_0.

If $m_0 = (i_B, \circledast)$ then we need to derive a value term $\langle V_1, ..., V_m \rangle$ (as $B = B_1 \times \cdots \times B_m$). For each p, if B_p is a base or reference type then we can choose a V_p canonically so that its denotation be i_{B_p} (the only interesting such case is this of i_{B_p} being a name $a \# \vec{a}$, where we take V_p to be z'_j, for $j = \min\{j \mid a = (i_{A(0)})_{k+j}\}$). Otherwise, $B_p = B'_p \to B''_p$ and from σ_0 we obtain the (tidy) viewfunction $f : Q^{\vec{a}}([\![A]\!] \otimes [\![B'_p]\!]) \longrightarrow T[\![B''_p]\!]$ by:

$$f \triangleq \{[(\vec{a}, i_{A(0)}, i_{B'_p}) * \circledast\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, (i_B, \circledast)\, (i_{B'_p}, \circledast)\, s] \in \mathtt{viewf}(\sigma_0)\}.$$

Note that, for any $[(\vec{a}, i_A) * \circledast\, (i_B, \circledast)\, (i_{B'_p}, \circledast)\, s] \in \mathtt{viewf}(\sigma_0)$, s cannot contain store-Q's justified by $\circledast$, as these would break (TD2). Hence, f fully describes σ_0 after $(i_{B'_p}, \circledast)$. By IH, there exists $\vec{a} \mid \Gamma, y : B'_p \vdash N : B''_p$ such that $[\![N]\!] = \mathtt{strat}(f)$; take then $V_p \triangleq \lambda y.N$. Hence, taking

$$M_0 \triangleq \langle V_1, ..., V_m \rangle$$

we obtain $\sigma_0 = [\![M_0]\!] \upharpoonright (\vec{a}, i_{A(0)})$.

If $m_0 = j_A$, played in some $A_{k+k'+i} = A'_i \to A''_i$, then $m_0 = (i_{A'_i}, \circledast)$. Assume that $A'_i = A'_{i,1} \times \cdots \times A'_{i,n_i}$ with $A'_{i,p}$'s being non-products. Now, O can either ask some name a (which would lead to a store-CC), or answer at A''_i, or play at some $A'_{i,p}$ of arrow type, say $A'_{i,p} = C_{i,p} \to C'_{i,p}$. Hence,

$$\mathtt{viewf}(\sigma_0) = f_A \cup \bigcup\nolimits_{p=1}^{n_i} f_p$$

where:

$$f_A \triangleq f_0 \cup \{\, [(\vec{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,(i_{A''_i}, \circledast)\, s] \in \mathtt{viewf}(\sigma_0) \,\}$$
$$f_p \triangleq f_0 \cup \{\, [(\vec{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,(i_{C_{i,p}}, \circledast)\, s] \in \mathtt{viewf}(\sigma_0) \,\}$$
$$f_0 \triangleq \{\, [(\vec{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\, s] \mid [\circledast\ \circledast\ s] \in \mathtt{viewf}(\mathtt{id}_\xi) \,\}$$

and where we assume $f_p \triangleq f_0$ if $A'_{i,p}$ is not an arrow type. It is not difficult to see that f_A, f_p are viewfunctions. Now, from f_A we obtain:

$$f'_A : Q^{\vec{a}}([\![A]\!] \otimes [\![A''_i]\!]) \longrightarrow T[\![B]\!] \triangleq \{\, [(\vec{a}, i_{A(0)}, i_{A''_i}) * \circledast s] \mid [(\vec{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,(i_{A''_i}, \circledast)\, s] \in f_A \,\}.$$

It is not difficult to see that f'_A is indeed a viewfunction (note that P cannot play a store-Q under $\circledast$ on the RHS once $(i_{A''_i}, \circledast)$ is played, by tidiness). By IH, there exists some $\vec{a} \mid \Gamma, y{:}A''_i \vdash M_A : B$ such that $[\![M_A]\!] = \mathtt{strat}(f'_A)$.
From each $f_p \neq f_0$ we obtain a viewfunction $f'_p : Q^{\vec{a}}([\![A]\!] \otimes [\![C_{i,p}]\!]) \longrightarrow T[\![C'_{i,p}]\!]$ by:

$$f'_p \triangleq \{[(\vec{a}, i_{A(0)}, i_{C_{i,p}}) * \circledast s] \mid [(\vec{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,(i_{C_{i,p}}, \circledast)\, s] \in f_p\}.$$

By IH, there exists some $\vec{a} \mid \Gamma, y' : C_{i,p} \vdash M_p : C'_{i,p}$ such that $[\![M_p]\!] = \mathtt{strat}(f'_p)$, so take $V_p \triangleq \lambda y'.M_p$. For each $A'_{i,p}$ of non-arrow type, the behaviour of σ_0 at $A'_{i,p}$ is fully described by $(i_{A'_i})_p$, so we choose V_p canonically as previously. $\langle V_1, ..., V_{n_i}\rangle$ is now of type A'_i and describes σ_0's behaviour in A'_i.
Now, taking

$$M_0 \triangleq (\lambda y.M_A)(z''_i\langle V_1, ..., V_{n_i}\rangle)$$

we obtain $\sigma_0 = [\![M_0]\!] \upharpoonright (\vec{a}, i_{A(0)})$. ∎

Finally, using the definability result and proposition 4.65 we can now show the following.

Corollary 4.68 *$\mathcal{T} = (\mathcal{T}, T, Q, O)$ satisfies ip-definability.*

Proof: For each $\vec{a}, A, B$, define $D^{\vec{a}}_{A,B} \triangleq \{f : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] \mid f \text{ is finitary}\}$. By definability, every $f \in D^{\vec{a}}_{A,B}$ is definable. We need also show:

$$(\forall \rho \in D^{\vec{a}}_{A\to B,\mathbb{N}}\,.\,\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{\vec{a}}(g)\,;\rho \in O^{\vec{a}}) \implies f \lesssim^{\vec{a}} g\,.$$

Assume the LHS assertion holds and let $\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}}$, some $\rho : Q^{\vec{a}}([\![A]\!] \multimap T[\![B]\!]) \longrightarrow T\mathbb{N}$. Then, let $[s\,;t] = [(\vec{a}, *) * \circledast (0, \circledast)^{\vec{b}}] \in \Lambda^{\vec{a}}(f)\,;\rho$, $[s] \in \Lambda^{\vec{a}}(f)$ and $[t] \in \rho$. By proposition 4.65, $[t] \in \rho_{\leq t}$, so $\Lambda^{\vec{a}}(f)\,;\rho_{\leq t} \in O^{\vec{a}}$. Moreover, $\rho_{\leq t} \in D^{\vec{a}}_{A\to B,\mathbb{N}}$, so $\Lambda^{\vec{a}}(g)\,;\rho_{\leq t} \in O^{\vec{a}}$, by hypothesis. Finally, $\rho_{\leq t} \sqsubseteq \rho$ implies $\Lambda^{\vec{a}}(g)\,;\rho_{\leq t} \sqsubseteq \Lambda^{\vec{a}}(g)\,;\rho$, hence the latter observable, so $f \lesssim^{\vec{a}} g$. ∎

Hence, we have shown full abstraction.

Theorem 4.69 *$\mathcal{T} = (\mathcal{T}, T, Q, O)$ is a fully abstract model of $\nu\rho$.* ∎

4.3.8 Equivalences established semantically

In this last section we prove several equivalences using the full-abstraction result. We first consider

$$\mathtt{stop}_B \cong \nu a.!\,a \tag{4.17}$$

with $a \in \mathbb{A}_B$, for any type B. By full-abstraction it suffices to show $[\![\mathtt{stop}_B]\!] \backsimeq [\![\nu a.!\,a]\!]$, and for the latter it suffices to show

$$[\![\nu a.!\,a]\!] \lesssim [\![\mathtt{stop}_B]\!]\,,$$

since the other direction is implied from the fact that $[\![\mathtt{stop}_B]\!] \sqsubseteq [\![\nu a.\,!\,a]\!]$. So take some tl4 strategy $\rho : T[\![B]\!] \longrightarrow T\mathbb{N}$ and consider any $[s\,;t] \in [\![\nu a.\,!\,a]\!]\,;\rho$ with $|s\,;t| > 3$. We know (example 4.33) that $[\![\nu a.\,!\,a]\!]$ is given by

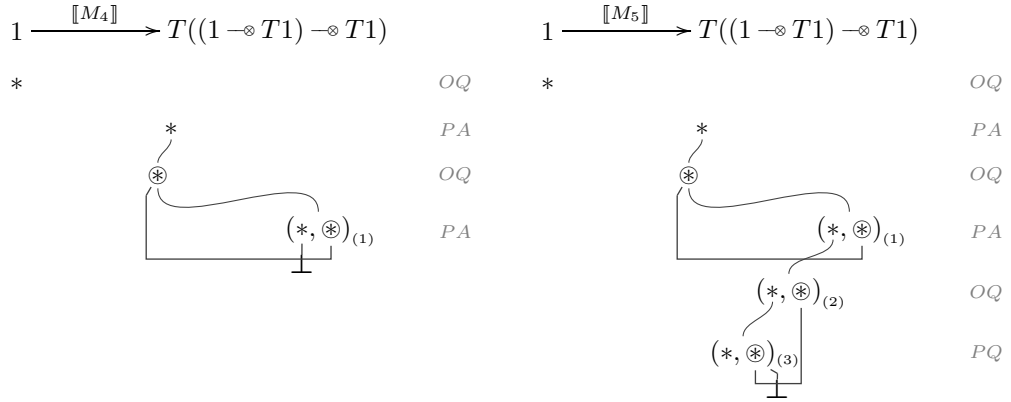
$$\mathtt{strat}\{\ [*\ *\circledast a^a\, i_B^a\,(i_B,\circledast)^a\, s^a] \mid [(\circledast,i_B)\,(\circledast,i_B)\,s] \in \mathtt{viewf}(\mathtt{id}_{\xi\otimes[\![B]\!]})\ \}\,,$$

so $s = **\circledast\, a^a s'$, while ρ being tl4 and $s \asymp t$ imply that $t = **\circledast\,\circledast^{\vec{b}}\, a^{\vec{b}}\, t'$, for some $\vec{b}\,\#\,a$. Since ρ is tidy and $a^{\vec{b}}$ is a fresh store-Q, $t'.1 = a^{\vec{b}}$ and therefore $s\,;t$ starts with the sequence $**\circledast\, a^{\vec{b}}$. Hence, $[\![\nu a.\,!\,a]\!]\,;\rho \notin O$ and therefore $[\![\nu a.\,!\,a]\!] \lesssim [\![\mathtt{stop}_B]\!]$.

We now show equivalence (4.7) of page 80. Recall that

$$M_4 \triangleq \lambda f.\,\mathtt{stop} : (\mathbb{1}\to\mathbb{1})\to\mathbb{1}\,,\quad M_5 \triangleq \lambda f.\,f\,\mathtt{skip}\,;\mathtt{stop} : (\mathbb{1}\to\mathbb{1})\to\mathbb{1}\,,$$

and that we need to show $M_4 \cong M_5$. By full-abstraction, it suffices to show $[\![M_4]\!] \backsimeq [\![M_5]\!]$, where the latter are given as follows.



Bottom links stand for deadlocks: if Opponent plays a move $(*,\circledast)_{(2)}$ under the last $*$ in $[\![M_4]\!]$ (providing thus the function f) then Player must play $[\![\mathtt{stop}]\!]$, i.e. remain idle. Similarly for $[\![M_5]\!]$: if Opponent gives an answer to $(*,\circledast)_{(3)}$ (providing thus the outcome of $f\,\mathtt{skip}$) then Player deadlocks the play.

Observe that $[\![M_4]\!] \sqsubseteq [\![M_5]\!]$, so we need only show $[\![M_5]\!] \lesssim [\![M_4]\!]$. Suppose $\rho : T((1\multimap T1)\multimap T1) \longrightarrow T\mathbb{N}$ is a tl4 tidy strategy such that $[**\circledast\,(0,\circledast)^{\vec{a}}] \in [\![M_5]\!]\,;\rho$ for some $\vec{a}$. Then, because of the form of $[\![M_5]\!]$, ρ can only play initial moves up to $(*,\circledast)_{(1)}$, then possibly ask some names to $(*,\circledast)_{(1)}$, and finally play $(0,\circledast)^{\vec{a}}$. Crucially, ρ cannot play $(*,\circledast)_{(2)}$ under $*$: this would introduce a question that could never be answered by $[\![M_5]\!]$, and therefore ρ would not be able to play $(0,\circledast)^{\vec{a}}$ without breaking well-bracketing. Hence, $[\![M_4]\!]$ and ρ can simulate the whole interaction and therefore $[**\circledast\,(0,\circledast)^{\vec{a}}] \in [\![M_4]\!]\,;\rho$.

Finally, we show the equivalences DROP and SWAP of [Sta94]. Assuming typed terms $\vec{a} \mid \Gamma \vdash M : A$ and $\vec{a}ab \mid \Gamma \vdash N : A$ (note that the latter implies $\vec{a}ba \mid \Gamma \vdash N : A$ is also a typed term, by lemma 2.17), these are formulated as follows.

$$\vec{a} \mid \Gamma \vdash \nu a.M \cong M \qquad \text{(DROP)}$$
$$\vec{a} \mid \Gamma \vdash \nu ab.N \cong \nu ba.N \qquad \text{(SWAP)}$$

Arguing semantically, and recalling lemma 4.10, it suffices to show that, for any $f : Q^{\vec{a}}\Gamma \longrightarrow B$ and any $g : Q^{\vec{a}ab} \longrightarrow B$ with B pointed,

$$f \backsimeq^{\vec{a}} \langle a\rangle(\tfrac{\vec{a}a}{\vec{a}}\,;f) \qquad \text{(DROP)}$$
$$\langle ab\rangle\, g \backsimeq^{\vec{a}} \langle ba\rangle(\tfrac{\vec{a}ba}{\vec{a}ab}\,;g)\,. \qquad \text{(SWAP)}$$

Observe now that both of the above follow from lemma 4.61.

Chapter 5

Nominal Exceptions

In this chapter we examine extensions of the $s\nu$-calculus in which names can be raised and handled as exceptions. Exceptions are a prevalent feature of programming languages for raising and handling eccentric program behaviour, and more generally for manipulating the flow of control. It is a key feature, for example, of ML, Java and C++. The raising of an exception forces a program to escape out of its context and to the nearest available exception-handler. Thus, exceptions provide a means of (an effect for) overriding nested behaviour of functional programs.

We start with a simple extension of the $s\nu$-calculus, the $\nu\varepsilon$-calculus, of which we briefly examine the syntax and abstract categorical semantics. The main focus, though, is on the $\nu\varepsilon\rho$-calculus, the extension of $\nu\rho$ with nominal exceptions. For $\nu\varepsilon\rho$ we carry out a similar analysis as with $\nu\rho$ in the previous chapter (factoring out material covered previously where possible), and thus construct a fully abstract semantics in nominal games. The construction combines elegantly the use of an exception monad with the nominal games setting where atoms are used (also) for exception names. We obtain a model of $\nu\varepsilon\rho$ in (innocent, well-bracketed) nominal games, which we then restrict to strategies with 'disciplined' exceptional behaviour (*x-tidy* strategies) to the effect of obtaining a fully abstract model.

A fully abstract model for a language with exceptions and ground-type references was constructed in [Lai01]. In rough terms, the model of [Lai01] allows for jumps in the precedence with which a program answers questions posed by the environment (i.e. it mildens the well-bracketing condition), thus translating in the semantical universe the override of nested behaviour. This yields a description of the exception effect that is both accurate and intuitive. However, the modelling of exceptions themselves is not nominal but rather based on the 'object-oriented' approach which encodes exceptions as products of raise/handle type. Therefore, "bad" constructors are included in the syntax, that is, the language examined includes *bad exceptions* (and also bad variables). Another point of difference between the two languages is that the one of [Lai01], being an extension of Idealized Algol [Rey81], is call-by-name with block-structured exceptions and references.

5.1 The $\nu\varepsilon$-calculus

Exceptions are a mechanism allowing program control to jump out of the current context and to the nearest handler. In the calculus we examine now, exceptions are terms of type $\mathbb{E}$. These can be raised and handled, and their closed values are given by names. The latter are taken from a set of atoms

$$\mathbb{A}_e \in (\mathbb{A}_i)_{i\in\omega}$$

and hence the calculus is an extension of $s\nu$ with names used for exceptions.

Definition 5.1 The $\nu\varepsilon$-calculus is a functional calculus of nominal exceptions. Its types,

terms and values are given as follows.

$$\begin{array}{lll}
\text{TY} \ni A, B ::= & \mathbb{N} \mid A \to B \mid A \times B \mid \mathbb{E} & \\
\text{TE} \ni M, N ::= & x \mid \lambda x.M \mid M\,N \mid \langle M, N\rangle \mid \mathtt{fst}\,M \mid \mathtt{snd}\,N & \lambda\text{-calculus} \\
& \mid n \mid \mathtt{pred}\,M \mid \mathtt{succ}\,N & \text{arithmetic} \\
& \mid \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 & \text{if_then_else} \\
& \mid a \mid \nu a.M \mid [M = N] & \nu\text{-calculus } (a \in \mathbb{A}_e) \\
& \mid \mathtt{raise}\ M & \text{raise exception} \\
& \mid \mathtt{try}\ N_1\ \mathtt{handle}\ M \Rrightarrow N_2 & \text{try/handle exception} \\
\text{VA} \ni V, W ::= & x \mid n \mid a \mid \lambda x.M \mid \langle V, W\rangle &
\end{array}$$

The typing system involves (as before) terms in environments $\vec{a} \mid \Gamma$; the main typing rules are the following.

$$\frac{}{\vec{a} \mid \Gamma \vdash a : \mathbb{E}}\ {\scriptstyle a \in \vec{a}} \qquad \frac{\vec{a}a \mid \Gamma \vdash M : B}{\vec{a} \mid \Gamma \vdash \nu a.M : B} \qquad \frac{\vec{a} \mid \Gamma \vdash M : \mathbb{E} \qquad \vec{a} \mid \Gamma \vdash N : \mathbb{E}}{\vec{a} \mid \Gamma \vdash [M = N] : \mathbb{N}}$$

$$\frac{\vec{a} \mid \Gamma \vdash M : \mathbb{E}}{\vec{a} \mid \Gamma \vdash \mathtt{raise}\ M : A} \qquad \frac{\vec{a} \mid \Gamma \vdash M : \mathbb{E} \qquad \vec{a} \mid \Gamma \vdash N_1, N_2 : A}{\vec{a} \mid \Gamma \vdash \mathtt{try}\ N_1\ \mathtt{handle}\ M \Rrightarrow N_2 : A}$$

▲

Observe that TE and VA are strong nominal sets. Regarding bound names and variables, the same definitions and conventions as in the case of the $s\nu$-calculus are in effect. Note that raised exceptions can have any type — even exception-type. Thus, for example, both the following terms can have type $\mathbb{E}$,

$$M_1 \triangleq a\,, \quad M_2 \triangleq \mathtt{raise}\,a\,,$$

but they are quite different: the former is a value of exception-type, while the latter is clearly non-value.

The operational semantics is defined by means of a small-step reduction relation, where terms reduce in name-list environments containing the names created thus far in the computation.

Definition 5.2 Reduction in the $\nu\varepsilon$-calculus involves $s\nu$-calculus reduction rules and rules for exceptions. The latter set of rules is given below.

$$\text{HL}\ \frac{}{\vec{a} \Vdash \mathtt{try}\ (\mathtt{raise}\ a)\ \mathtt{handle}\ a \Rrightarrow N \longrightarrow \vec{a} \Vdash N}$$

$$\text{VHL}\ \frac{}{\vec{a} \Vdash \mathtt{try}\ V\ \mathtt{handle}\ a \Rrightarrow N \longrightarrow \vec{a} \Vdash V}$$

$$\text{NHL}\ \frac{}{\vec{a} \Vdash \mathtt{try}\ (\mathtt{raise}\ b)\ \mathtt{handle}\ a \Rrightarrow N \longrightarrow \vec{a} \Vdash \mathtt{raise}\ b}\ {\scriptstyle a \neq b}$$

$$\text{XPN}\ \frac{}{\vec{a} \Vdash \mathrm{Z}[\mathtt{raise}\ a] \longrightarrow \vec{a} \Vdash \mathtt{raise}\ a}$$

$$\text{CTX}\ \frac{\vec{a} \Vdash M \longrightarrow \vec{a}' \Vdash M'}{\vec{a} \Vdash \mathrm{E}[M] \longrightarrow \vec{a}' \Vdash \mathrm{E}[M']}$$

Unhandled evaluation contexts Z are of the forms:

$$(\lambda x.N)\,\_\,,\ \_\,N\,,\ \langle \_, N\rangle\,,\ \langle V, \_\rangle\,,\ \mathtt{fst}\,\_\,,\ \mathtt{snd}\,\_\,,\ \mathtt{if0}\ \_\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2\,,$$
$$\mathtt{succ}\,\_\,,\ \mathtt{pred}\,\_\,,\ [\_ = N]\,,\ [a = \_]\,,\ \mathtt{raise}\,\_\,,\ \mathtt{try}\ N_1\ \mathtt{handle}\ \_ \Rrightarrow N_2\,.$$

General evaluation contexts E are of the forms:

$$\mathrm{Z}\,,\ \texttt{try}\ \_\ \texttt{handle}\ a \Rightarrow N\,.$$

▲

The $\nu\varepsilon$-calculus is an extension of the $s\nu$-calculus without recursive effects, and hence it is strongly normalising. Observational approximation is defined as usually: a term M observationally approximates a term N if, for any program context C, if $\mathrm{C}[M]$ reduces to 0 then so does $\mathrm{C}[N]$. Now, taking

$$\begin{aligned} M_1 &\triangleq \lambda f.\,0 : (\mathbb{E} \to \mathbb{N}) \to \mathbb{N} \\ M_2 &\triangleq \lambda f.\,\nu a.\nu b.[fa \Leftrightarrow fb] : (\mathbb{E} \to \mathbb{N}) \to \mathbb{N} \\ M_3 &\triangleq \lambda f.\,\nu a.[fa \Leftrightarrow fa] : (\mathbb{E} \to \mathbb{N}) \to \mathbb{N} \end{aligned} \tag{5.1}$$

we have the following equivalences/inequivalences in $\nu\varepsilon$.

$$M_1 \not\approx M_2 \tag{5.2}$$

$$M_2 \approx M_3 \tag{5.3}$$

M_1 and M_2, which are equivalent in $s\nu$, can be distinguished by a context that raises an exception as soon as f is used, e.g. the context

$$\mathrm{C} \triangleq \_(\lambda x.\,\texttt{raise}\,\nu a.a)\,.$$

On the other hand, the equivalence of M_2 and M_3 (which is invalid in $\nu\rho$) is given in section 5.2.6 semantically, after we introduce a fully abstract game semantics for the $\nu\varepsilon\rho$-calculus. We will also see that references and exceptions are more expressive than references alone.

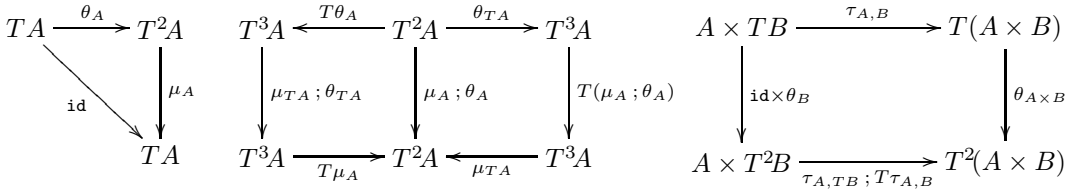
We move on to sound categorical semantics for the $\nu\varepsilon$-calculus. We follow the same recipe as in the case of the $\nu\rho$-calculus, that is, we work in a monadic-comonadic setting for names and on top of it we require structure for modelling exceptions. In the categorical semantics of $\nu\rho$ we used a single monad T for encapsulating both fresh-names and store. As we saw in the concrete game semantics, this was achieved by first constructing a fresh-names monad (lifting) and then, by use of a store-arena, deriving a store-monad T which embedded lifting. This methodology heavily uses the fact that the store-monad uses exponentials, and that lifting has such exponentials. Therefore, it is not relevant in the case of exceptions; the standard practice for exceptions is monad composition.

Since our model analysis is an extensional (macroscopical) one, we find more useful (and concise) the description of our monad T as separable into two components, rather than a compound monad over a distributive law ℓ (definition 2.26). We therefore introduce *precompound monads*.

5.1.1 Precompound monads

As discussed above, the computational monad T for the $\nu\varepsilon$-calculus contains a component for exceptions and another one for fresh names. The two-component nature of T does not need the full specifications of a compound monad for its description.

Definition 5.3 A strong monad (T, η, μ, τ) is ***precompound*** if there exists a natural transformation $\theta : T \longrightarrow T^2$ such that the following diagrams commute.



Moreover, each η_A is an inner- and outer-component arrow, where an arrow $f : A \longrightarrow TB$ is said to be:

- an *inner-component arrow* if $f\,;\theta_B = f\,;\eta_{TB}$,
- an *outer-component arrow* if $f\,;\theta_B = f\,;T\eta_B$.

We write T as (T, θ). ▲

In essence, θ is separating the two components in T, with each morphism

$$\theta_A : TA \longrightarrow T_{(o)}T_{(i)}A$$

sending the outer T-component of TA to $T_{(o)}$, and its inner T-component to $T_{(i)}$. From this viewpoint, outer-component arrows can be seen as involving computation in the outer component of T, and similarly for inner-component arrows.

Arrows in each component form distinct Kleisli categories: η-arrows are in both components and Kleisli-composition in the same component is a closed operation, as it is shown in the following diagrams.

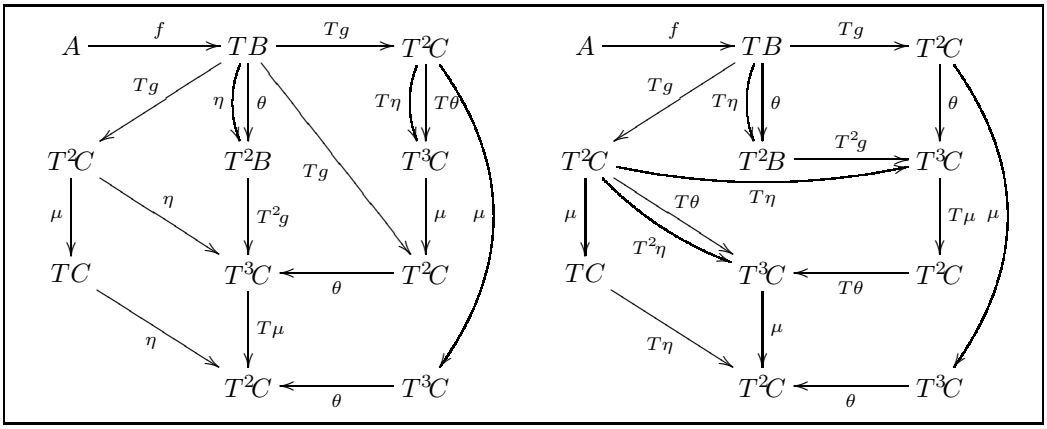


Figure 5.1: Kleisli-composition for inner- and outer-component arrows.

Every monad T is trivially precompound, by simply taking θ to be η (empty outer component) or $T\eta$ (empty inner component). More generally, compound monads are precompound.

Lemma 5.4 *Let T be a compound monad $(\ddot{T}\dot{T}, \ell)$. T is precompound with θ defined by:*

$$\theta_A : TA \longrightarrow T^2A \triangleq \ddot{T}\dot{\eta}_{\dot{T}A}\,;\ddot{T}\dot{T}\ddot{\eta}_{\dot{T}A}\,.$$

■

5.1.2 Sound categorical semantics

We now proceed to formulate an abstract categorical semantics for the $\nu\varepsilon$-calculus. The semantics is based on $\nu\varepsilon$-models, which resemble $\nu\rho$-models of the previous chapter.

Definition 5.5 A $\nu\varepsilon$-model $\mathcal{M}$ is a structure $(\mathcal{M}, T, Q)$ such that:

I. $\mathcal{M}$ is a category with finite products, with 1 being the terminal object and $A \times B$ the product of A and B.

II. T is a strong monad (T, η, μ, τ) with exponentials.

III. $\mathcal{M}$ contains a natural numbers object $\mathbb{N}$ equipped with successor/predecessor arrows and $\tilde{n} : 1 \longrightarrow \mathbb{N}$, each $n \in \mathbb{N}$. Moreover, for each object A, there is an appropriate arrow for zero-equality tests $\mathtt{cnd}_A : \mathbb{N} \times TA \times TA \longrightarrow TA$.

IV. Q is a family of product comonads $(Q^{\vec{a}}, \varepsilon, \delta, \zeta)_{\vec{a}\in\mathbb{A}^\#}$ on $\mathcal{M}$ such that:

(a) the basis of Q^ϵ is 1, and $Q^{\vec{a}} = Q^{\vec{a}'}$ whenever $[\vec{a}] = [\vec{a}']$,

(b) if $\vec{a}' \subseteq \vec{a}$ then there exists a comonad morphism $\frac{\vec{a}}{\vec{a}'} : Q^{\vec{a}} \longrightarrow Q^{\vec{a}'}$ such that $\frac{\vec{a}}{\epsilon} = \varepsilon$, $\frac{\vec{a}}{\vec{a}} = \mathtt{id}$ and, whenever $\vec{a}' \subseteq \vec{a}'' \subseteq \vec{a}$,

$$\frac{\vec{a}}{\vec{a}''} ; \frac{\vec{a}''}{\vec{a}'} = \frac{\vec{a}}{\vec{a}'}$$

(c) for each $\vec{a}a \in \mathbb{A}^\#$ there exists a strength-coherent (v. (3.5), page 72) natural transformation $\mathtt{nu}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow TQ^{\vec{a}a}$ such that, for each $A \in Ob(\mathcal{M})$ and $\vec{a}a \subseteq \vec{a}'a$, the following diagrams commute.

$$\begin{array}{ccc} Q^{\vec{a}}A & \xrightarrow{\langle \mathtt{id}, \mathtt{nu}_A\rangle} & Q^{\vec{a}}A \times TQ^{\vec{a}a}A \\ {\scriptstyle \mathtt{nu}_A}\downarrow & & \downarrow{\scriptstyle \tau} \\ TQ^{\vec{a}a}A & \xrightarrow[T\langle \frac{\vec{a}a}{\vec{a}}, \mathtt{id}\rangle]{} & T(Q^{\vec{a}}A \times Q^{\vec{a}a}A) \end{array} \qquad \begin{array}{ccc} Q^{\vec{a}'}A & \xrightarrow{\mathtt{nu}_A^{\vec{a}'a}} & TQ^{\vec{a}'a}A \\ {\scriptstyle \frac{\vec{a}'}{\vec{a}}}\downarrow & & \downarrow{\scriptstyle T\frac{\vec{a}'a}{\vec{a}a}} \\ Q^{\vec{a}}A & \xrightarrow[\mathtt{nu}_A^{\vec{a}a}]{} & TQ^{\vec{a}a}A \end{array} \qquad \text{(N2)}$$

V. Setting $\mathbb{A}_e \triangleq Q^a 1$, for $a \in \mathbb{A}_e$, there is a name-equality arrow $\mathtt{eq}_e : \mathbb{A}_e \times \mathbb{A}_e \longrightarrow \mathbb{N}$ in $\mathcal{M}$ such that, for any distinct $a, b \in \mathbb{A}_e$, the following diagram commutes.

$$\begin{array}{ccccc} Q^a 1 & \xrightarrow{\Delta} & \mathbb{A}_e \times \mathbb{A}_e & \xleftarrow{\langle \frac{ab}{a}, \frac{ab}{b}\rangle} & Q^{ab}1 \\ {\scriptstyle !}\downarrow & & \downarrow{\scriptstyle \mathtt{eq}_e} & & \downarrow{\scriptstyle !} \\ 1 & \xrightarrow[\widetilde{0}]{} & \mathbb{N} & \xleftarrow[\widetilde{1}]{} & 1 \end{array} \qquad \text{(N1)}$$

VI. $\mathcal{M}$ contains a natural transformation $\mathtt{inx} : K_{\mathbb{A}_e} \longrightarrow T$ for exception-raising, where $K_{\mathbb{A}_e}$ is the constant-$\mathbb{A}_e$ functor, such that the following diagrams commute.

$$\begin{array}{ccc} A \times \mathbb{A}_e & \xrightarrow{\mathtt{id}\times\mathtt{inx}_B} & A \times TB \\ {\scriptstyle \pi_2}\downarrow & & \downarrow{\scriptstyle \tau} \\ \mathbb{A}_e & \xrightarrow[\mathtt{inx}_{A\times B}]{} & T(A \times B) \end{array} \qquad \begin{array}{ccc} \mathbb{A}_e & \xrightarrow{\mathtt{inx}_{TB}} & T^2B \\ & {\scriptstyle \mathtt{inx}_B}\searrow & \downarrow{\scriptstyle \mu} \\ & & TB \end{array} \qquad \text{(NE1)}$$

Moreover, for each object A, an arrow $\mathtt{hdl}_A : \mathbb{A}_e \times TA \times TA \longrightarrow TA$ for exception-handling such that the following diagram commutes.

$$\begin{array}{ccccc} Q^{\dot{a}\dot{b}}1 \times TA & \xrightarrow{\langle \frac{\dot{a}\dot{b}}{\dot{a}}, \frac{\dot{a}\dot{b}}{\dot{b}}\rangle\times\mathtt{id}} & \mathbb{A}_e \times \mathbb{A}_e \times TA & \xleftarrow{\Delta\times\mathtt{id}} & \mathbb{A}_e \times TA \\ & & {\scriptstyle \mathtt{id}\times\mathtt{inx}_A\times\mathtt{id}}\downarrow & \swarrow{\scriptstyle \pi_2} & \\ {\scriptstyle \pi_1;\frac{\dot{a}\dot{b}}{\dot{b}}}\downarrow & & \mathbb{A}_e \times TA \times TA & \xleftarrow{\mathtt{id}\times\eta\times\mathtt{id}} & \mathbb{A}_e \times A \times TA \\ & & {\scriptstyle \mathtt{hdl}_A}\downarrow & \swarrow{\scriptstyle \pi_{12};\eta} & \\ \mathbb{A}_e & \xrightarrow[\mathtt{inx}_A]{} & TA & & \end{array} \qquad \text{(NE2)}$$

Finally, T is precompound, (T, θ), with $\mathtt{nu}$ being in the outer component and $\mathtt{inx}$ in the inner one.

▲

We observe that items I-IV appear verbatim in the definition of a $\nu\rho$-model, while V presents the same property applied to different objects of names. The reason is simple: these are the $s\nu$-calculus specifications of the model, and the $s\nu$-calculus is the common denominator of $\nu\varepsilon$ and $\nu\rho$. On the other hand, item VI gives the specifications for exceptions.

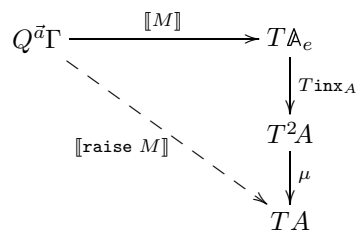
We now proceed with the modelling of the $\nu\varepsilon$-calculus in a $\nu\varepsilon$-model.

Definition 5.6 Let $(\mathcal{M}, T, Q)$ be a $\nu\varepsilon$-model. $\nu\varepsilon$-types are translated in $\mathcal{M}$ as follows.

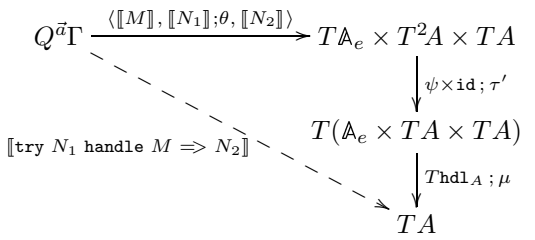
$$[\![\mathbb{N}]\!] \triangleq \mathbb{N}\,,\ \ [\![A \times B]\!] \triangleq [\![A]\!] \times [\![B]\!]\,,\ \ [\![A \to B]\!] \triangleq T[\![B]\!]^{[\![A]\!]}\,,\ \ [\![\mathbb{E}]\!] \triangleq \mathbb{A}_e\,.$$

A typed term $\vec{a} \mid \Gamma \vdash M : A$ is mapped to an arrow $[\![M]\!]_{\vec{a}\mid\Gamma} : Q^{\vec{a}}[\![\Gamma]\!] \longrightarrow T[\![A]\!]$ in $\mathcal{M}$, which we write simply as $[\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow TA$, by use of the following rules,

$$\frac{[\![M]\!] : Q^{\vec{a}}\Gamma \to T\mathbb{A}_e}{}$$



$$\frac{[\![M]\!] : Q^{\vec{a}}\Gamma \to T\mathbb{A}_e \qquad [\![N_i]\!] : Q^{\vec{a}}\Gamma \to TA}{}$$



and relevant rules of figure 4.3, p. 83. ▲

The fact that the translation of the $\nu\varepsilon$-calculus into a $\nu\varepsilon$-model follows closely that of $\nu\rho$ into a $\nu\rho$-model allows us to easily prove correctness of the translation for non-exceptional behaviour. The exceptional cases are then attacked as in the case of $\nu\varepsilon\rho$ in the next section.

Proposition 5.7 (Correctness ($\nu\varepsilon$)) *For any typed term $\vec{a} \mid \Gamma \vdash M : A$ and any $\mathrm{r} \neq \mathrm{NEW}$,*

1. $\vec{a} \vDash M \xrightarrow{\mathrm{r}} \vec{a} \vDash M' \implies [\![M]\!] = [\![M']\!]$,

2. $\vec{a} \vDash M \xrightarrow{\mathrm{NEW}} \vec{a}a \vDash M' \implies [\![M]\!] = [\![\nu a.M']\!]$.

Therefore, $\vec{a} \vDash M \longrightarrow\!\!\!\!\!\rightarrow\ \vec{a}\vec{a}' \vDash M' \implies [\![M]\!] = [\![\nu\vec{a}'.M']\!]$. ■

We close this section by giving adequacy specifications for $\nu\varepsilon$-models. We do not proceed to full-abstraction specifications, neither to concrete models in game semantics; these are seen in detail in the next language we examine, the $\nu\varepsilon\rho$-calculus.

Definition 5.8 Let $\mathcal{M}$ be a $\nu\varepsilon$-model and $[\![\_]\!]$ be the respective translation of the $\nu\varepsilon$-calculus. $\mathcal{M}$ is ***adequate*** if, for any pair of states $\vec{a}, \vec{a}'$, any $n \neq 0$ and any $a \in \mathbb{A}_e$,

$$[\![\nu\vec{a}.\mathtt{raise}\ a]\!] \neq [\![\nu\vec{a}'.0]\!] \neq [\![\nu\vec{a}.n]\!]\,.$$

▲

The above condition for adequacy is a simplified version of the respective condition for $\nu\rho$, given that the calculus we are now examining is strongly normalising. It can also be seen as an extended version of the *mono requirement* of computational models presented by Moggi [Mog89].

Assuming $\mathcal{M}$ is an adequate $\nu\varepsilon$-model we can show the following.

Proposition 5.9 (Equational Soundness ($\nu\varepsilon$))

$$[\![M]\!] = [\![N]\!] \implies M \lessapprox N$$

■

5.2 The *νερ*-calculus

We combine nominal general references and nominal exceptions to a new language, the *νερ*-calculus, extending both *νρ* and *νε*. Names in *νερ* are created with local scope, can be tested for equality and can be passed around via function application. Moreover, some names (*reference names*) can be dereferenced or updated, while others (*exception names*) can be raised or handled. The syntax is built in nominal sets by assuming a set of atoms $\mathbb{A}_e \in (\mathbb{A}_i)_{i\in\omega}$ for exception names and a set of atoms $\mathbb{A}_A \in (\mathbb{A}_i)_{i\in\omega}$ for reference names of type A, for each type A in the language.

Notation 5.10 As before, (general) names are denoted by a, b, c and variants. Note, though, that we use different notations for exception and reference names. In particular, we use $\dot{a}, \dot{b}, \dot{c}$ and variants for exception names; and $\ddot{a}, \ddot{b}, \ddot{c}$ and variants for reference names.

Definition 5.11 The *νερ*-calculus is a functional calculus of nominal references and exceptions. Its types, terms and values are given as follows.

$$\begin{array}{lll}
\mathrm{TY} \ni A, B ::= & \mathbb{1} \mid \mathbb{N} \mid A \times B \mid A \to B \mid [A] \mid \mathbb{E} & \\
\mathrm{TE} \ni M, N ::= & x \mid \lambda x.M \mid M\,N \mid \langle M, N\rangle \mid \mathtt{fst}\,M \mid \mathtt{snd}\,N & \lambda\text{-calculus} \\
& \mid \mathtt{skip} \mid n \mid \mathtt{pred}\,M \mid \mathtt{succ}\,N & \text{return/ arithmetic} \\
& \mid \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 & \text{if_then_else} \\
& \mid a \mid \nu a.M \mid [M = N] & \nu\text{-calculus} \\
& \mid \mathtt{raise}\,M \mid \mathtt{try}\ N_1\ \mathtt{handle}\ M \Rightarrow N_2 & \text{raise/ handle} \\
& \mid M := N \mid\ !\,M & \text{update/ dereferencing} \\
\mathrm{VA} \ni V, W ::= & x \mid \mathtt{skip} \mid n \mid a \mid \lambda x.M \mid \langle V, W\rangle &
\end{array}$$

The main typing rules are the following.

$$\frac{\vec{a}a \mid \Gamma \vdash M : B}{\vec{a} \mid \Gamma \vdash \nu a.M : B} \qquad \frac{\vec{a} \mid \Gamma \vdash M : A_\nu \qquad \vec{a} \mid \Gamma \vdash N : A_\nu}{\vec{a} \mid \Gamma \vdash [M = N] : \mathbb{N}}\ A_\nu \in \{\mathbb{E}\} \cup \{[A] \mid A \in \mathrm{TY}\}$$

$$\frac{}{\vec{a} \mid \Gamma \vdash \dot{a} : \mathbb{E}}\ {\scriptstyle \dot{a}\in\vec{a} \,\wedge\, \dot{a}\in\mathbb{A}_e} \qquad \frac{\vec{a} \mid \Gamma \vdash M : \mathbb{E}}{\vec{a} \mid \Gamma \vdash \mathtt{raise}\,M : A} \qquad \frac{\vec{a} \mid \Gamma \vdash M : \mathbb{E} \qquad \vec{a} \mid \Gamma \vdash N_1, N_2 : A}{\vec{a} \mid \Gamma \vdash \mathtt{try}\ N_1\ \mathtt{handle}\ M \Rightarrow N_2 : A}$$

$$\frac{}{\vec{a} \mid \Gamma \vdash \ddot{a} : [A]}\ {\scriptstyle \ddot{a}\in\vec{a} \,\wedge\, \ddot{a}\in\mathbb{A}_A} \qquad \frac{\vec{a} \mid \Gamma \vdash M : [A]}{\vec{a} \mid \Gamma \vdash\ !M : A} \qquad \frac{\vec{a} \mid \Gamma \vdash M : [A] \qquad \vec{a} \mid \Gamma \vdash N : A}{\vec{a} \mid \Gamma \vdash M := N : \mathbb{1}}$$

▲

The operational semantics is defined in ***mixed environments*** P containing information both about the (general) names created and about the values stored in those of them that denote references:

$$P ::= \epsilon \mid a, P \mid \ddot{a} :: V, P\,. \tag{5.4}$$

The ***domain*** of a mixed environment P is the list of names it enlists:

$$\mathtt{dom}(\epsilon) \triangleq \epsilon, \quad \mathtt{dom}(a, P) \triangleq a, \mathtt{dom}(P)\,, \quad \mathtt{dom}(\ddot{a} :: V, P) \triangleq \ddot{a}, \mathtt{dom}(P)\,, \tag{5.5}$$

and must be a list of distinct names.

Definition 5.12 Reduction in the *νερ*-calculus invokes *sν*-calculus reduction rules and rules

from $\nu\rho$ and $\nu\varepsilon$. The latter two sets are given below.

$$\text{DRF}\ \frac{}{P, \ddot{a} :: V, P' \vDash\, !\,\ddot{a} \longrightarrow P, \ddot{a} :: V, P' \vDash V}$$

$$\text{UPD}\ \frac{}{P, \ddot{a}(:: W), P' \vDash \ddot{a} := V \longrightarrow P, \ddot{a} :: V, P' \vDash \mathtt{skip}}$$

$$\text{HL}\ \frac{}{P \vDash \mathtt{try}\,(\mathtt{raise}\,\dot{a})\,\mathtt{handle}\,\dot{a} \Rightarrow N \longrightarrow P \vDash N}$$

$$\text{VHL}\ \frac{}{P \vDash \mathtt{try}\,V\,\mathtt{handle}\,\dot{a} \Rightarrow N \longrightarrow P \vDash V}$$

$$\text{NHL}\ \frac{}{P \vDash \mathtt{try}\,(\mathtt{raise}\,\dot{b})\,\mathtt{handle}\,\dot{a} \Rightarrow N \longrightarrow P \vDash \mathtt{raise}\,\dot{b}}\ \dot{a} \neq \dot{b}$$

$$\text{XPN}\ \frac{}{P \vDash \mathrm{Z}[\mathtt{raise}\,\dot{a}] \longrightarrow P \vDash \mathtt{raise}\,\dot{a}}$$

$$\text{CTX}\ \frac{P \vDash M \longrightarrow P' \vDash M'}{P \vDash \mathrm{E}[M] \longrightarrow P' \vDash \mathrm{E}[M']}$$

Unhandled evaluation contexts Z are of the forms:

$$(\lambda x.N)\,\_\,,\ \_\,N\,,\ \langle \_, N\rangle\,,\ \langle V, \_\rangle\,,\ \mathtt{fst}\,\_\,,\ \mathtt{snd}\,\_\,,\ \mathtt{if0}\,\_\,\mathtt{then}\,N_1\,\mathtt{else}\,N_2\,,\ \mathtt{succ}\,\_\,,\ \mathtt{pred}\,\_\,,$$
$$[\_ = N]\,,\ [a = \_]\,,\ \mathtt{raise}\,\_\,,\ \mathtt{try}\,N_1\,\mathtt{handle}\,\_ \Rightarrow N_2\,,\ !\,\_\,,\ \_ := N\,,\ \ddot{a} := \_\,.$$

General evaluation contexts E are of the forms:

$$\mathrm{Z}\,,\ \mathtt{try}\,\_\,\mathtt{handle}\,\dot{a} \Rightarrow N\,.$$

▲

Contexts in $\nu\varepsilon\rho$ extend $\nu\rho$-contexts in a straightforward manner, so contexts have types of the form $(\vec{a}, \Gamma, A) \mapsto (\vec{a}', \Gamma', A')$ (see definition 4.5). A ***program context*** is a name- and variable-closing context yielding $\mathbb{N}$, that is, a context of type $(\vec{a}, \Gamma, A) \mapsto (\epsilon, \varnothing, \mathbb{N})$. For typed terms $\vec{a} \mid \Gamma \vdash M : A$ and $\vec{a} \mid \Gamma \vdash N : A$, we say that M ***observationally approximates*** N, written $\vec{a} \mid \Gamma \vdash M \lessapprox N$ or simply $M \lessapprox N$, if, for any program context C,

$$(\,\exists P'.\ \vDash \mathrm{C}[M] \longrightarrow\!\!\!\!\!\rightarrow\ P' \vDash 0\,) \implies (\,\exists P''.\ \vDash \mathrm{C}[N] \longrightarrow\!\!\!\!\!\rightarrow\ P'' \vDash 0\,)\,.$$

Observational equivalence, $\cong$, is the symmetric closure of $\lessapprox$.

Let us examine briefly the expressivity of the $\nu\varepsilon\rho$-calculus in relation to the expressivity of the nominal calculi examined previously. Taking[1]

$$\begin{aligned}
M_1 &\triangleq \lambda f.\,0 : (\mathbb{E} \to \mathbb{N}) \to \mathbb{N} \\
M_2 &\triangleq \lambda f.\,\nu a.\nu b.[fa \Leftrightarrow fb] : (\mathbb{E} \to \mathbb{N}) \to \mathbb{N} \\
M_3 &\triangleq \lambda f.\,\nu a.[fa \Leftrightarrow fa] : (\mathbb{E} \to \mathbb{N}) \to \mathbb{N} \\
M_4 &\triangleq \lambda f.\,\mathtt{stop} : (\mathbb{1} \to \mathbb{1}) \to \mathbb{1} \\
M_5 &\triangleq \lambda f.\,f\,\mathtt{skip}\,;\mathtt{stop} : (\mathbb{1} \to \mathbb{1}) \to \mathbb{1}
\end{aligned} \tag{5.6}$$

we have the following inequivalences.

$$M_1 \ncong M_2 \tag{5.7}$$
$$M_2 \ncong M_3 \tag{5.8}$$
$$M_4 \ncong M_5 \tag{5.9}$$

[1]Recall $\mathtt{stop} \triangleq \nu b.(b := \lambda x.(!\,b)\mathtt{skip})\,;(!\,b)\mathtt{skip}$, $[M \Leftrightarrow N] \triangleq \mathtt{if0}\ M\ \mathtt{then}\ N\ \mathtt{else}\ (\mathtt{if0}\ N\ \mathtt{then}\ 1\ \mathtt{else}\ 0)$.

(5.7) and (5.8) are inherited from $\nu\rho$, while M_4 and M_5 (which are equivalent in $\nu\rho$) can be distinguished by use of exceptions, e.g. by the context

$$\mathrm{C} \triangleq \nu\dot{a}.\mathtt{try}\,(\_(\lambda y.\mathtt{raise}\,\dot{a}))\,;0\,\mathtt{handle}\,\dot{a} \Rightarrow 0\,.$$

The equivalences and inequivalences of the above terms in the calculi we examine in this thesis are summarised in figure 5.2, where A_ν is $\mathbb{E}$ in the case of $\nu\varepsilon$, and $[\mathbb{1}]$ in all other cases.

	$M_1 \cong M_2$	$M_2 \cong M_3$	$M_4 \cong M_5$	
$s\nu$	✓	✓	–	$M_1 \triangleq \lambda f.\,0 : (A_\nu \to \mathbb{N}) \to \mathbb{N}$
$\nu\rho$	✗	✗	✓	$M_2 \triangleq \lambda f.\,\nu a.\nu b.[fa \Leftrightarrow fb] : (A_\nu \to \mathbb{N}) \to \mathbb{N}$
$\nu\varepsilon$	✗	✓	–	$M_3 \triangleq \lambda f.\,\nu a.[fa \Leftrightarrow fa] : (A_\nu \to \mathbb{N}) \to \mathbb{N}$
$\nu\varepsilon\rho$	✗	✗	✗	$M_4 \triangleq \lambda f.\,\mathtt{stop} : (\mathbb{1} \to \mathbb{1}) \to \mathbb{1}$
				$M_5 \triangleq \lambda f.\,f\,\mathtt{skip}\,;\mathtt{stop} : (\mathbb{1} \to \mathbb{1}) \to \mathbb{1}$

Figure 5.2: Equivalences separating our nominal calculi.

5.2.1 Categorical semantics

We examine categorical semantics of $\nu\varepsilon\rho$ in the familiar monadic-comonadic setting. $\nu\varepsilon\rho$ extending $\nu\varepsilon$ and $\nu\rho$ means that a model of $\nu\varepsilon\rho$ 'incorporates' a model of $\nu\varepsilon$ and a model of $\nu\rho$, so the computational monad T we are after should incorporate a monad for exceptions and a monad for storage.

Compound monads are such a solution, but there is a subtlety here: we cannot achieve T by simply composing a monad $\dot{T}$ of $\nu\varepsilon$ with a monad $\ddot{T}$ of $\nu\rho$, and therefore neither can we compose a $\nu\varepsilon$-model with a $\nu\rho$-model. The reason for this complication is that storage in $\ddot{T}$ would be higher-order and therefore would need to include functions returning exceptions, a specification of $\dot{T}$.

We therefore consider a monad T with separate components for storage and exceptions which yields *itself* (rather than each of its components separately) both a $\nu\varepsilon$-model and a $\nu\rho$-model. As done previously in $\nu\varepsilon$, the compoundness of the monad is expressed extensionally via precompoundness.

Definition 5.13 A $\nu\varepsilon\rho$-model $\mathcal{M}$ is a structure $(\mathcal{M}, T, Q)$ such that:

A. $\mathcal{M}$ is both a $\nu\varepsilon$-model and a $\nu\rho$-model (with common structure for items I-IV of definitions 5.5, 4.8).

B. T is a precompound monad (T, θ), such that

- all arrows $\mathtt{inx}_A$ are inner-component,
- all arrows $\mathtt{upd}_A$ and $\mathtt{nu}_A$ are outer-component. ▲

We see that the precompound-monad analysis has paid off in conciseness in the above definition. Taking storage and fresh-names as outer-component and exceptions as inner follows the common practice when composing exceptions with other effects.

We carry on with the semantical translation of the $\nu\varepsilon\rho$-calculus in a $\nu\varepsilon\rho$-model.

Definition 5.14 Let $(\mathcal{M}, T, Q)$ be a $\nu\varepsilon\rho$-model. $\nu\varepsilon\rho$-types are translated in $\mathcal{M}$ as:

$$[\![\mathbb{1}]\!] \triangleq 1\,,\ [\![\mathbb{N}]\!] \triangleq \mathtt{N}\,,\ [\![A \times B]\!] \triangleq [\![A]\!] \times [\![B]\!]\,,\ [\![A \to B]\!] \triangleq T[\![B]\!]^{[\![A]\!]},\ [\![\mathbb{E}]\!] \triangleq \mathbb{A}_e\,,\ [\![[A]]\!] \triangleq \mathbb{A}_A\,.$$

A typed term $\vec{a} \mid \Gamma \vdash M : A$ is mapped to an arrow $[\![M]\!]_{\vec{a}\mid\Gamma} : Q^{\vec{a}}[\![\Gamma]\!] \to T[\![A]\!]$ in $\mathcal{M}$, which we write simply as $[\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow TA$, by use of the relevant rules of definition 5.6 and figure 4.3, p. 83. ▲

We reproduce most rules for the semantical translation in figure 5.3. The interesting part is the use of θ in the case of exception-handling. Its function is to separate the two components of TA yielded by $[\![N_1]\!]$, so that the inner-component is passed on to `hdl` and the outer-component is passed to the output of the computation. This allows us to disregard the outer-component of TA when applying `hdl`.

Correctness is now proved along the same lines as in $\nu\rho$. To any store P, we relate the term $\bar{P}$ of type $\mathbb{1}$ as:

$$\bar{\epsilon} \triangleq \texttt{skip}\,, \quad \overline{a, P} \triangleq \bar{P}\,, \quad \overline{\ddot{a} :: V, P} \triangleq (\ddot{a} := V\,;\bar{P})\,.$$

The following lemma is needed.

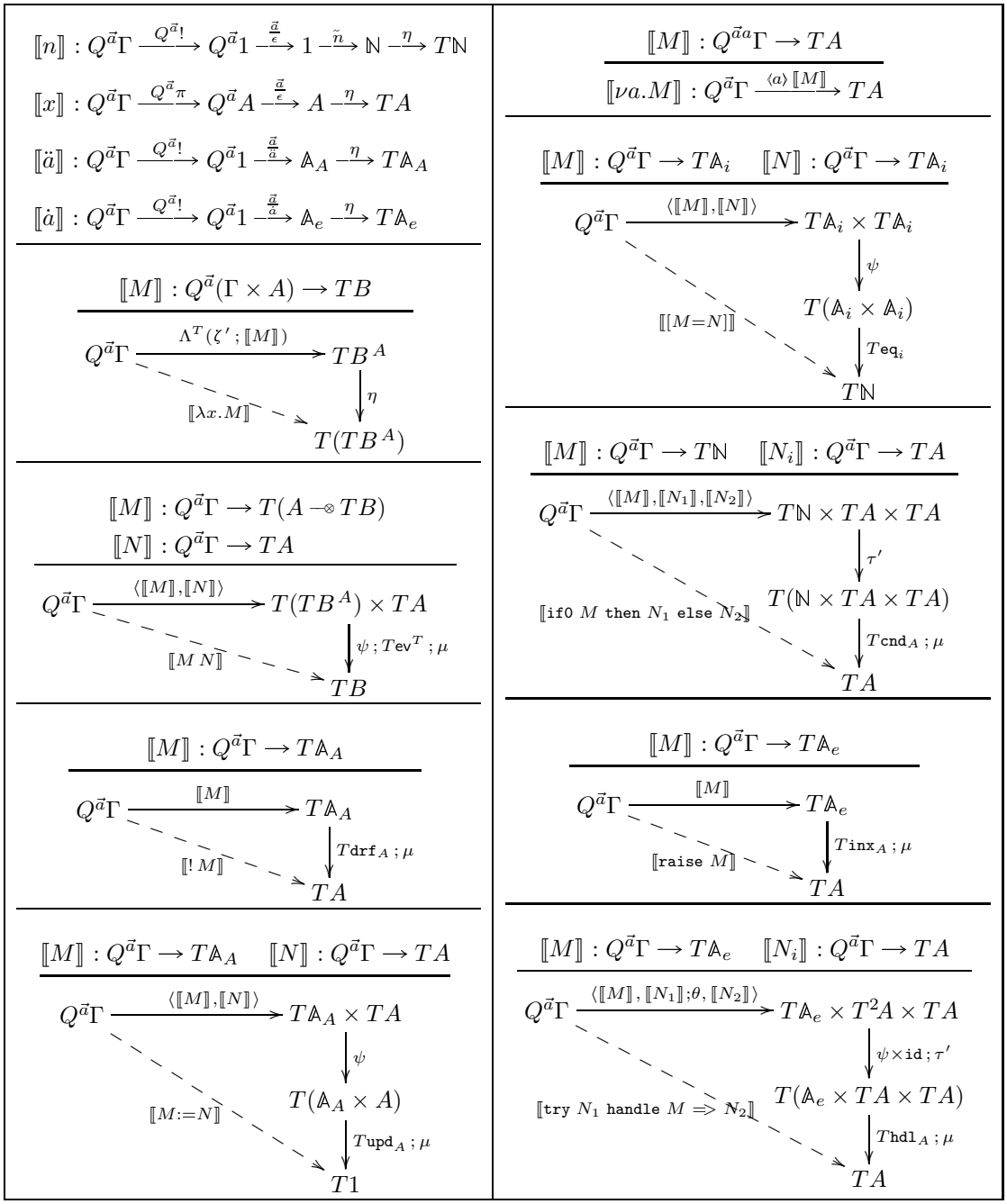


Figure 5.3: The semantic translation of $\nu\varepsilon\rho$-terms.

Lemma 5.15 *For any* $f : Q^{\vec{a}a}A \longrightarrow TB$,

$$\langle a \rangle (f\,;\theta_B) = \langle a \rangle f\,;\theta_B\,.$$

Proof: Noting that $\mathtt{nu}\,;Tf$ is outer-component, we have:

$$\langle a \rangle (f\,;\theta) = \mathtt{nu}\,;Tf\,;T\theta\,;\mu = \mathtt{nu}\,;Tf\,;T\eta\,;T\mu\,;T\theta\,;\mu = \mathtt{nu}\,;Tf\,;\theta\,;T\mu\,;T\theta\,;\mu = \mathtt{nu}\,;Tf\,;\mu\,;\theta$$

as required. ∎

Proposition 5.16 (Correctness) *For any typed term* $\vec{a} \mid \Gamma \vdash M : A$, *any* P *with* $\mathtt{dom}(P) = \vec{a}$ *and any transition rule* r,

1. *if* $\mathrm{r} \notin \{\mathrm{NEW}, \mathrm{UPD}, \mathrm{DRF}\}$ *then* $P \Vdash M \xrightarrow{\mathrm{r}} P \Vdash M' \implies [\![M]\!] = [\![M']\!]$,

2. *if* $\mathrm{r} \in \{\mathrm{UPD}, \mathrm{DRF}\}$ *then* $P \Vdash M \xrightarrow{\mathrm{r}} P' \Vdash M' \implies [\![\bar{P}\,;M]\!] = [\![\bar{P}'\,;M']\!]$,

3. $P \Vdash M \xrightarrow{\mathrm{NEW}} P, a \Vdash M' \implies [\![\bar{P}\,;M]\!] = \langle a \rangle [\![\bar{P}\,;M']\!]$.

Therefore, $P \Vdash M \longrightarrow\!\!\!\!\!\rightarrow P' \Vdash M' \implies [\![\bar{P}\,;M]\!] = [\![\nu\vec{a}'.(\bar{P}'\,;M')]\!]$, *with* $\mathtt{dom}(P') = \vec{a}\vec{a}'$.

Proof: For 1-3, we do induction on the derivation of the reduction. Because of proposition 4.13, we need only show the base case of 1 for exceptions, and the inductive step of 1-3 for the case of contexts involving exceptions. In fact, by similarity to other cases,[2] it suffices to consider only handled evaluation contexts at the inductive step.

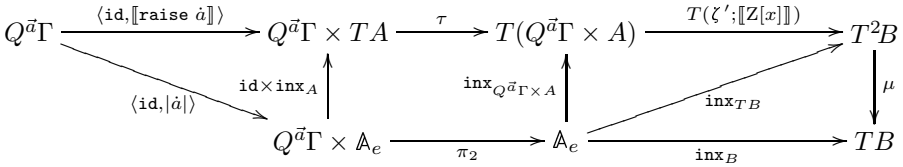
The base case follows from the specifications of $\nu\varepsilon\rho$-models. We consider only the most interesting case, that of XPN:

$$P \Vdash \mathrm{Z}[\mathtt{raise}\,\dot{a}] \longrightarrow P \Vdash \mathtt{raise}\,\dot{a}\,.$$

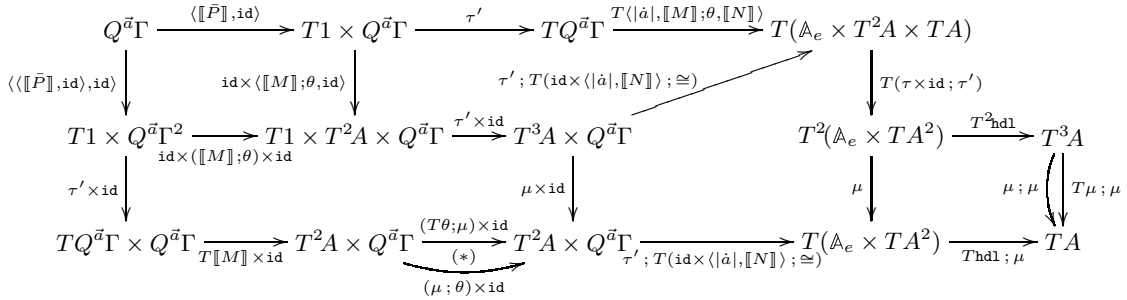
Similarly to lemma 4.12, we can show that, for any unhandled evaluation context Z,

$$[\![\mathrm{Z}[M]]\!] = \langle \mathtt{id}, [\![M]\!] \rangle\,;\tau\,;T\zeta'\,;T[\![\mathrm{Z}[x]]\!]\,;\mu\,, \tag{5.10}$$

and hence the following diagram commutes, as required.



For the inductive step, take $\mathrm{E} \triangleq \mathtt{try}\ \_\ \mathtt{handle}\ \dot{a} \Rightarrow N$. From the following diagram,



we have that $[\![\bar{P}\,;\mathrm{E}[M]]\!] = \langle [\![\bar{P}\,;M]\!]\,;\theta, \mathtt{id} \rangle\,;\tau'\,;T(\mathtt{id} \times \langle |\dot{a}|, [\![N]\!] \rangle\,; \cong)\,;T\mathtt{hdl}\,;\mu$, from which the inductive step for 1-3 follows, with the aid of lemmas 5.15, 4.11. Note that $(*)$ follows from the fact that $f \triangleq \langle [\![\bar{P}]\!], \mathtt{id} \rangle\,;\tau'\,;T[\![M]\!]$ is outer-component (which follows from $[\![\bar{P}]\!]$ being outer-component) by: $f\,;T\theta\,;\mu = f\,;T\eta\,;T\mu\,;T\theta\,;\mu = f\,;\theta\,;T\mu\,;T\theta\,;\mu = f\,;\mu\,;\theta$. ∎

From correctness, we can obtain soundness by adding a further specification on adequacy.

[2] In particular, $\mathtt{raise}\ \_$ is treated exactly like $!\,\_$, and $\mathtt{try}\ N_1\ \mathtt{handle}\ \_ \Rightarrow N_2$ like $\mathtt{if0}\ \_\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2$.

Definition 5.17 Let $\mathcal{M}$ be a $\nu\varepsilon\rho$-model and $[\![\_]\!]$ the respective translation of $\nu\varepsilon\rho$. $\mathcal{M}$ is ***adequate*** if, for any typed term $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$, if $[\![M]\!] = [\![\nu\vec{b}.\bar{P}\,;0]\!]$, some $P, \vec{b}$, then there exists P' such that $\vec{a} \vDash M \longrightarrow\!\!\!\!\!\rightarrow P' \vDash 0$. ▲

Proposition 5.18 (Equational Soundness) *Translating $\nu\varepsilon\rho$ into an adequate $\nu\varepsilon\rho$-model $\mathcal{M}$ we obtain:*

$$[\![M]\!] = [\![N]\!] \implies M \lessapprox N\,.$$

■

5.2.2 Full abstraction

Normally we would expect to obtain full-abstraction from soundness by adding further specifications to $\nu\varepsilon\rho$-models, and perhaps doing some quotienting, but this is not the case here. For full-abstraction the model needs to satisfy definability, at least for the arrows defining the semantical preorder. However, θ is clearly not definable — there is no context separating the computational effects in the manner θ does — and its presence affects the semantical preorder in a substantial way. For the latter, note for example that the terms

$$\ddot{a} := 0\,;\texttt{raise}\,\nu\dot{a}.\dot{a} \quad \text{and} \quad \ddot{a} := 1\,;\texttt{raise}\,\nu\dot{a}.\dot{a}$$

are observationally equivalent in the language, but their translations can be distinguished by use of θ: simply discard their inner-component computations by composing them with $\theta\,;T!$, and then return the value of $\ddot{a}$.

Since it is unreasonable to ask definability with θ, we will remove it from our models and thus work in *$\nu\varepsilon\rho$-submodels*, that is, submodels of $\nu\varepsilon\rho$-models that contain the translations of all $\nu\varepsilon\rho$-terms but not problematic arrows like θ.

Definition 5.19 Let $\mathcal{M} = (\mathcal{M}, T, Q)$ be an adequate $\nu\varepsilon\rho$-model and let $[\![\_]\!]$ be the semantic translation of $\nu\varepsilon\rho$ into $\mathcal{M}$. A ***$\nu\varepsilon\rho$-submodel*** is a structure $(\mathcal{M}', T', Q')$ such that:

- $\mathcal{M}'$ is a lluf subcategory of $\mathcal{M}$, and T', Q' are restrictions of T, Q in $\mathcal{M}'$.
- $(\mathcal{M}', T', Q')$ satisfies items I-IV of definitions 5.5,4.8.
- $[\![M]\!] \in \mathcal{M}'(Q'^{\vec{a}}[\![\Gamma]\!], T'[\![A]\!])$, for each typed term $\vec{a} \mid \Gamma \vdash M : A$. ▲

By a slight abuse of notation, we denote $\mathcal{M}'$ by $(\mathcal{M}', T, Q)$. Evidently, a $\nu\varepsilon\rho$-submodel $\mathcal{M}'$ is an adequate model of $\nu\varepsilon\rho$. Regarding observationality, since the intrinsic preorder cannot be shown to be a congruence (the semantic translation comes from $\mathcal{M}$ as a black box), it is stipulated to be so.

Definition 5.20 A $\nu\varepsilon\rho$-submodel $\mathcal{M}' = (\mathcal{M}', T, Q)$ is ***observational*** if:

- for all $\vec{a}$, there exists $O^{\vec{a}} \subseteq \mathcal{M}'(Q^{\vec{a}}1, T\mathbb{N})$ such that, for all $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$,

$$[\![M]\!] \in O^{\vec{a}} \iff \exists P, \vec{b}.\, [\![M]\!] = [\![\nu\vec{b}.\bar{P}\,;0]\!]\,,$$

- the induced intrinsic preorder $\lesssim = (\lesssim^{\vec{a}})_{\vec{a}\in\mathbb{A}^\#}$, defined on arrows in $\mathcal{M}'(Q^{\vec{a}}A, TB)$ by

$$f \lesssim^{\vec{a}} g \iff \forall \rho : Q^{\vec{a}}(TB^{\,A}) \longrightarrow T\mathbb{N}.\, (\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{\vec{a}}(g)\,;\rho \in O^{\vec{a}})\,,$$

  with $\Lambda^{\vec{a}}(f) \triangleq \Lambda^{Q^{\vec{a}},T}(f)$, is a congruence.

We write $\mathcal{M}'$ as $(\mathcal{M}', T, Q, O)$. ▲

Recall from definition 4.17 that $\lesssim$ being a congruence means that, for any pair M, N of terms and any relevant context C,

$$[\![M]\!] \lesssim [\![N]\!] \implies [\![\mathrm{C}[M]]\!] \lesssim [\![\mathrm{C}[N]]\!]\,.$$

We can now prove the following.

Lemma 5.21 (Inequational Soundness) *Translating $\nu\varepsilon\rho$ into an observational $\nu\varepsilon\rho$-submodel $\mathcal{M}'$ we obtain:*

$$[\![M]\!] \lesssim [\![N]\!] \implies M \lessapprox N\,.$$

■

The final step is full-abstraction, which passes through definability for the intrinsic preorder.

Definition 5.22 Let $(\mathcal{M}', T, Q, O)$ be an observational $\nu\varepsilon\rho$-submodel and let $[\![\_]\!]$ be the semantic translation of $\nu\varepsilon\rho$ into $\mathcal{M}'$. $\mathcal{M}'$ satisfies ***ip-definability*** if, for any $\vec{a}, A, B$, there exists $D^{\vec{a}}_{A,B} \subseteq \mathcal{M}'(Q^{\vec{a}}[\![A]\!], T[\![B]\!])$ such that:

- for each $f \in D^{\vec{a}}_{A,B}$ there exists a term M such that $[\![M]\!] = f$,
- for each $f, g \in \mathcal{M}'(Q^{\vec{a}}[\![A]\!], T[\![B]\!])$,

$$f \lesssim^{\vec{a}} g \iff \forall \rho \in D^{\vec{a}}_{A\to B,\mathbb{N}}\,.\,(\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{\vec{a}}(g)\,;\rho \in O^{\vec{a}})\,.$$

We write $\mathcal{M}'$ as $(\mathcal{M}', T, Q, O, D)$. ▲

Proposition 5.23 (FA) *Translating $\nu\varepsilon\rho$ into an ip-definable $\nu\varepsilon\rho$-submodel $\mathcal{M}'$ we obtain:*

$$[\![M]\!] \lesssim [\![N]\!] \iff M \lessapprox N\,.$$

■

5.2.3 The nominal games model

We proceed to construct a fully abstract model of the $\nu\varepsilon\rho$-calculus, that is, an ip-definable $\nu\varepsilon\rho$-submodel, in a category of nominal games. Our basis is the category $\mathcal{V}_{\mathsf{t}}$ of section 3.3, which contains amongst others:

- an arena $\mathbb{A}_e$ for exceptions and, for each type A, an arena $\mathbb{A}_A$ for references to type A,
- finite products, distributive coproducts, partial exponentials, big tensors.

The modelling of storage is monadic by means of a store-monad $\ddot{T}$ built around a store-arena $\xi \triangleq \bigotimes_{A\in \mathrm{TY}}(\mathbb{A}_A \Rightarrow [\![A]\!])$, while exceptions are modelled by use of the coproduct monad $\dot{T}$ of the exception-arena $\mathbb{A}_e$. These specifications lead to the following domain equation.

$$\begin{aligned} [\![A \to B]\!] &= [\![A]\!] \multimap (\xi \Rightarrow ([\![B]\!] + \mathbb{A}_e) \otimes \xi) \\ \xi &= \bigotimes\nolimits_A (\mathbb{A}_A \Rightarrow [\![A]\!]) \end{aligned} \qquad \text{(SE')}$$

The full form of the store-equation (SE') is the following.

$$[\![\mathbb{1}]\!] = 1\,,\quad [\![\mathbb{N}]\!] = \mathbb{N}\,,\quad [\![[A]]\!] = \mathbb{A}_A\,,\quad [\![\mathbb{E}]\!] = \mathbb{A}_e\,,\quad [\![A \times B]\!] = [\![A]\!] \otimes [\![B]\!]\,,$$
$$[\![A \to B]\!] = [\![A]\!] \multimap (\xi \Rightarrow ([\![B]\!] + \mathbb{A}_e) \otimes \xi)\,,\qquad \xi = \textstyle\bigotimes_A (\mathbb{A}_A \Rightarrow [\![A]\!])\,.$$

This is solved in the same way the store-equation for $\nu\rho$ was solved, that is, by expressing it as a fixpoint functorial equation and finding its minimal invariant. We avoid doing the computations again, as they are almost identical to those for $\nu\rho$. Explicitly, the solution is depicted in figure 5.4.

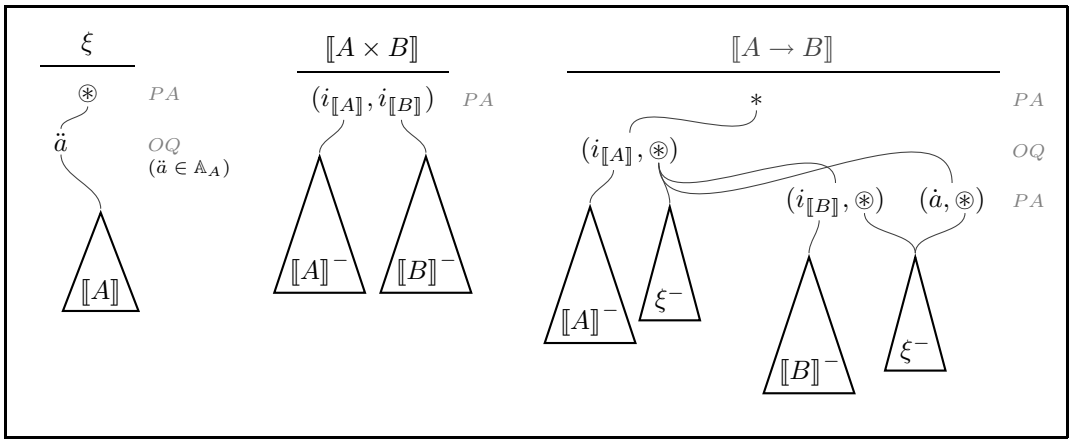


Figure 5.4: The store arena ξ and the translation of $\nu\varepsilon\rho$-types.

The monads $\dot{T}$ and $\ddot{T}$ needed for the semantics are already present in SE′. In particular, their functors are given by:

$$\begin{aligned} \dot{T} &: \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{t}} \triangleq \_ + \mathbb{A}_e \\ \ddot{T} &: \mathcal{V}_{\mathtt{t}} \longrightarrow \mathcal{V}_{\mathtt{t}} \triangleq \xi \Rightarrow \_ \otimes \xi \,. \end{aligned} \tag{5.11}$$

From the above we obtain the exception-monad $(\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ and the store-monad $(\ddot{T}, \ddot{\eta}, \ddot{\mu}, \ddot{\tau})$ following the constructions of sections 2.3.3 and 2.3.5. Composing them (see proposition 2.27) we obtain a computational monad (T, η, μ, τ) for $\nu\varepsilon\rho$, that is, T is a strong monad with exponentials, defined as follows and depicted in figure 5.5 (recall diagrammatic conventions of section 3.2.3).

$$\begin{aligned} T &\triangleq \ddot{T}\dot{T} \\ \eta_A &\triangleq A \xrightarrow{\ddot{\eta}_A} \ddot{T}A \xrightarrow{\ddot{T}\dot{\eta}_A} TA \\ \mu_A &\triangleq T^2A \xrightarrow{\ddot{T}\ddot{\ell}_{\dot{T}A}} \ddot{T}^2\dot{T}^2A \xrightarrow{\ddot{\mu}_{\dot{T}^2A}} \ddot{T}\dot{T}^2A \xrightarrow{\ddot{T}\dot{\mu}_A} TA \\ \tau_{A,B} &\triangleq A \times TB \xrightarrow{\ddot{\tau}_{A,\dot{T}B}} \ddot{T}(A \times \dot{T}B) \xrightarrow{\ddot{T}\dot{\tau}_{A,B}} T(A \times B) \\ \ell_A &\triangleq \dot{T}\ddot{T}A \xrightarrow{[\ddot{T}\iota_1, \iota_2; \ddot{\eta}]} \ddot{T}\dot{T}A \end{aligned} \tag{5.12}$$

Moreover, there is a natural transformation $\beta : (\_)_\perp \longrightarrow T$ given by:

$$\beta_A : A_\perp \xrightarrow{\ddot{\alpha}} \ddot{T}A \xrightarrow{\ddot{T}\dot{\eta}_A} TA\,, \tag{5.13}$$

where $\ddot{\alpha} : (\_)_\perp \longrightarrow T$ is the monad morphism defined in section 4.3.2, that is,

$$\ddot{\alpha}_A = A_\perp \xrightarrow{(\ddot{\eta}_A)_\perp} (\ddot{T}A)_\perp \xrightarrow{\mathtt{pu}_{\ddot{T}A}} \ddot{T}A\,.$$

By proposition 2.27, $\ddot{T}\dot{\eta}$ is also a monad morphism, and therefore β is a monad morphism from $(\_)_\perp$ to T.

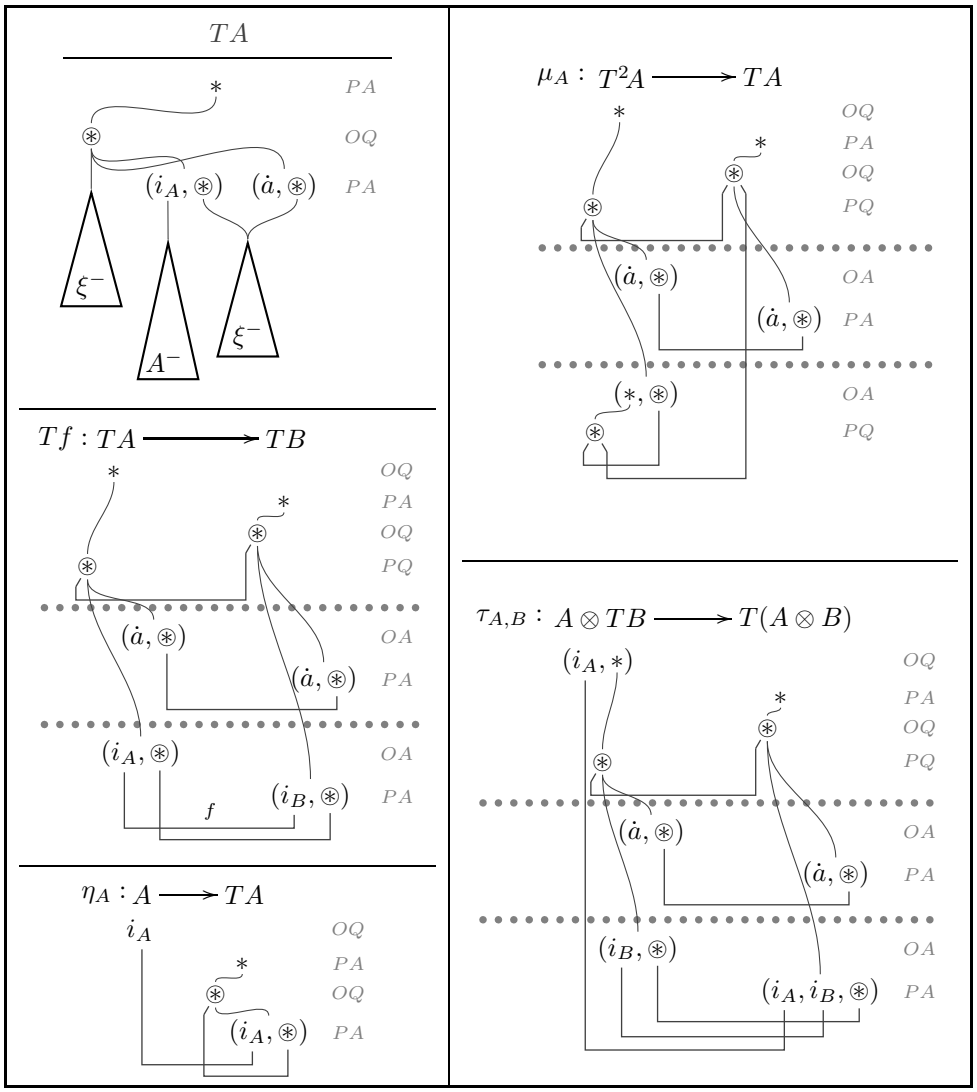


Figure 5.5: The compound monad (T, η, μ, τ) for $\nu\varepsilon\rho$.

5.2.4 The sound model

Regarding the construction of a $\nu\varepsilon\rho$-model in $\mathcal{V}_{\mathtt{t}}$ the situation is as follows (notation follows definition 5.13 and definitions 5.5, 4.8).

A. I-III. $\mathcal{V}_{\mathtt{t}}$ is a category with finite products and an adequate object for natural numbers, and T is a strong monad with exponentials.

IV. There is a family $(Q^{\vec{a}}, \varepsilon, \delta, \zeta)_{\vec{a} \in \mathbb{A}^\#}$ of product comonads, with each $Q^{\vec{a}}$ having basis $\mathbb{A}^{\vec{a}}$ (see section 3.4.2), which fulfills specifications (a,b). There are also fresh-name constructors,

$$\mathtt{new}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow (Q^{\vec{a}a})_\perp \,,$$

given in section 3.4.3, which satisfy (N2).

V. There are name equality arrows, $\mathtt{eq}_e$ and $\mathtt{eq}_A$ for each type A, making the (N1) diagram commute (section 3.4.2).

VI$_{\nu\rho}$ There are update and dereferencing arrows, $\ddot{\mathtt{upd}}_A$ and $\ddot{\mathtt{drf}}_A$ for each type A, over the store-arena $\ddot{T}$. These are given as in definition 4.30.

VI$_{\nu\varepsilon}$ There are distributive coproducts and arrows $\mathtt{eq}_e$, which essentially carry all the structure that we need.

B. By lemma 5.4, T is precompound. θ is depicted in figure 5.6.

We therefore need only do some work on items IV and VI. For the former, the transition from `new` to `nu` is by use of the monad morphism taking us from $(\_)_\perp$ to T.

Definition 5.24 For each $\vec{a}a \in \mathbb{A}^{\vec{a}}$, define a natural transformation $\mathtt{nu}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow TQ^{\vec{a}a}$ by:

$$\mathtt{nu}^{\vec{a}a}_A \triangleq Q^{\vec{a}}A \xrightarrow{\mathtt{new}_A} (Q^{\vec{a}a}A)_\perp \xrightarrow{\beta_{Q^{\vec{a}a}A}} TQ^{\vec{a}a}A\,.$$

▲

Each arrow $\mathtt{nu}^{\vec{a}a}_A$ is explicitly given as follows, and diagrammatically in figure 5.6.

$$\mathtt{nu}^{\vec{a}a}_A = \mathtt{strat}\{\ [(\vec{a}, i_A) * \circledast (\vec{a}a, i_A, \circledast)^a s^a] \mid a \# i_A \wedge ([i_A i_A s] \in \mathtt{viewf}(\mathtt{id}_A) \\ \vee [\circledast \circledast s] \in \mathtt{viewf}(\mathtt{id}_\xi))\ \} \tag{5.14}$$

Because β is a monad morphism, `nu` satisfies the (N2) diagrams.

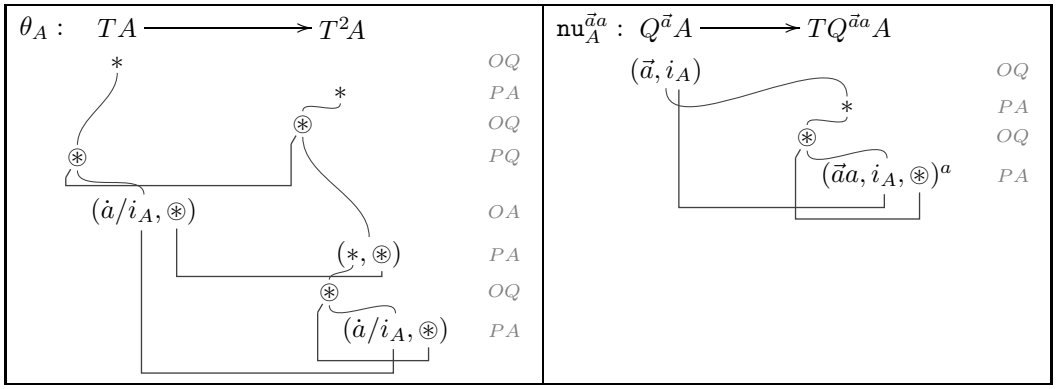


Figure 5.6: Natural transformations θ and `nu` for $\nu\varepsilon\rho$.

Regarding update and dereferencings, we have the following arrows,

$$\ddot{\mathtt{upd}} : \mathtt{A}_A \otimes [\![A]\!] \longrightarrow \ddot{T}1\,, \quad \ddot{\mathtt{drf}} : \mathtt{A}_A \longrightarrow \ddot{T}[\![A]\!]\,,$$

given as in definition 4.30 (modulo the use of a different store ξ). From these, we obtain arrows `upd` and `drf` via a monad morphism.

Definition 5.25 For any type A, define the strategies:

- $\mathtt{upd}_A : \mathtt{A}_A \otimes [\![A]\!] \xrightarrow{\ddot{\mathtt{upd}}_A} \ddot{T}1 \xrightarrow{\ddot{T}\dot{\eta}} T1\,,$
- $\mathtt{drf}_A : \mathtt{A}_A \xrightarrow{\ddot{\mathtt{drf}}_A} \ddot{T}[\![A]\!] \xrightarrow{\ddot{T}\dot{\eta}} T[\![A]\!]\,.$ ▲

The fact that the above strategies factor through the monad transformation $\ddot{T}\dot{\eta}$ implies that these are outer-component arrows, as required. Now, as we can see in the following figure, the strategies work exactly as in the case of $\nu\rho$, except for the fact that the copycat links may also carry raised exceptions. It is therefore not difficult to show that the (NR) diagrams are satisfied.

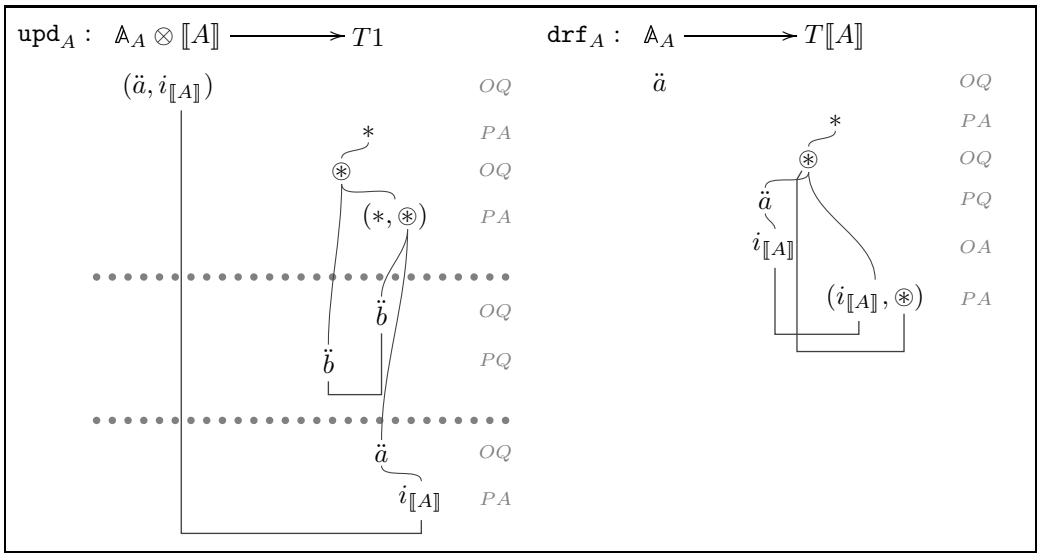


Figure 5.7: Update and dereferencing arrows in $\mathcal{V}_{\mathtt{t}}$.

Finally, we need to provide the structure necessary for exceptions. This is essentially given by the coproducts and the exception-equality arrows.

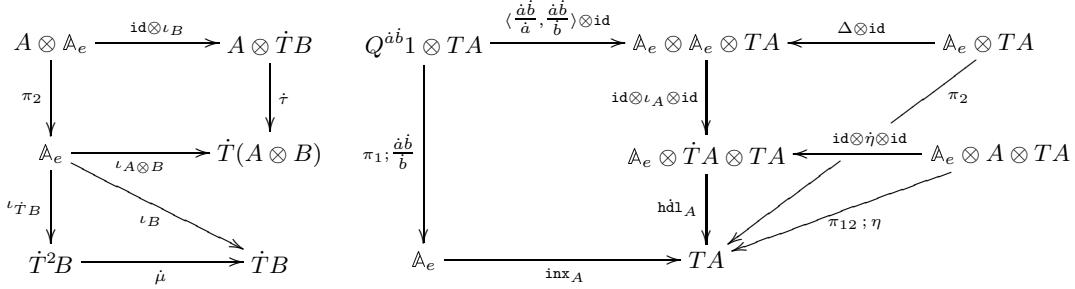
Definition 5.26 For each object A, define the strategies:

- $\mathtt{inx}_A \triangleq \mathbb{A}_e \xrightarrow{\iota_2} \dot{T}A \xrightarrow{\ddot{\eta}} TA$,
- $\mathtt{hdl}_A \triangleq \mathbb{A}_e \otimes TA \otimes TA \xrightarrow{\ddot{\tau} \otimes \mathtt{id}\,;\, \ddot{\tau}'} \ddot{T}(\mathbb{A}_e \otimes \dot{T}A \otimes TA) \xrightarrow{\ddot{T}\dot{\mathtt{hdl}}_A} \ddot{T}TA \xrightarrow{\ddot{\mu}} TA$,
- $\dot{\mathtt{hdl}}_A : \mathbb{A}_e \otimes \dot{T}A \otimes TA \longrightarrow TA \triangleq \{\, [(\dot{a}, i_A, *)\, s] \mid [i_A\, s] \in \eta_A \,\} \cup \{\, [(\dot{a}, \dot{a}, *)\, s] \mid [*\, s] \in \mathtt{id}_{TA} \,\}$
  $\cup \{\, [(\dot{a}, \dot{b}, *)\, s] \mid [\dot{b}\, s] \in \mathtt{inx}_A \wedge \dot{b} \neq \dot{a} \,\}$. ▲

We give a depiction of $\mathtt{hdl}_A$ in figure 5.8. Note also that $\mathtt{inx}$, a composite of natural transformations, is a natural transformation. Moreover, we can show the following.

Proposition 5.27 *The above defined arrows make the* (NE) *diagrams commute.*

Proof: The proof is straightforward, by showing the following diagrams.



■

We have therefore shown the following.

Theorem 5.28 $(\mathcal{V}_{\mathtt{t}}, T, Q)$ *is a* $\nu\varepsilon\rho$*-model.* ■

We proceed to show that $\mathcal{V}_{\mathtt{t}}$ is adequate. This is achieved via O-adequacy (lemma 5.39), which is proven independently, and the following lemma (proven similarly to lemma 4.35).

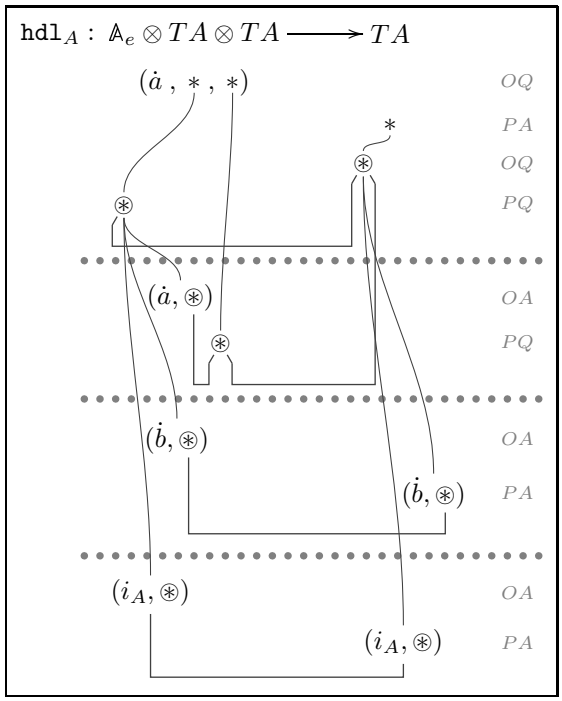


Figure 5.8: Exception-handling in $\mathcal{V}_{\mathtt{t}}$.

Lemma 5.29 *Let $\vec{a} \mid \varnothing \vdash M : A$ be a typed term. For any store P, if $P \models M$ is non-reducing then*

I. *if M is not a value then for no $\vec{b}, i_A$ do we have $[(\vec{a}, *) * \circledast (i_A, \circledast)^{\vec{b}}] \in [\![\bar{P} ; M]\!]$,*

II. *if M is not a raised exception then for no $\vec{b}, \dot{a}$ do we have $[(\vec{a}, *) * \circledast (\dot{a}, \circledast)^{\vec{b}}] \in [\![\bar{P} ; M]\!]$.* ■

Proposition 5.30 (Adequacy) *$\mathcal{V}_{\mathtt{t}}$ is adequate: for any typed term $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$, if $[\![M]\!] = [\![\nu \vec{b}.\bar{P} ; 0]\!]$, for some P, then there exists P' such that $\vec{a} \models M \longrightarrow\!\!\!\!\!\rightarrow P' \models 0$.* ■

5.2.5 Full abstraction

As expected, although $\mathcal{V}_{\mathtt{t}}$ is a sound $\nu\varepsilon\rho$-model for $\nu\varepsilon\rho$, it is not fully abstract due to its strategies not satisfying tidiness conditions (v. section 4.3.5). But even with tidiness our strategies are still missing discipline, this time related to exception-handling. In particular, strategies may well handle fresh (unknown) exceptions, in contrast to what is possible in the operational semantics. Hence, in addition to the tidiness conditions (TD1-3) familiar from the model of $\nu\rho$, we impose on strategies x-tidiness conditions which ensure a certain *fresh-exception-discipline*.

We start by restricting our attention to arenas appearing as type-translations, and classify their moves with regard to their store-behaviour and exception-behaviour (note there is a circularity in $H_{A \multimap \otimes TB}$ and $X_{A \multimap \otimes TB}$; what is meant actually is an inductive definition).

Definition 5.31 Consider $\mathcal{V}_{\nu\varepsilon\rho}$, the full subcategory of $\mathcal{V}_{\mathtt{t}}$ with objects defined as follows.

$$Ob(\mathcal{V}_{\nu\varepsilon\rho}) \ni A, B ::= 1 \mid \mathbb{N} \mid \mathbb{A}^{\vec{a}} \mid A \otimes B \mid A \multimap TB$$

For each such arena A we define its set of ***store-Handles***, H_A, and its set of ***exception-raisers***,

X_A, as follows.

$$
\begin{aligned}
H_1 = H_{\mathbb{N}} = H_{\mathbb{A}^{\ddot{a}}} &\triangleq \varnothing\,,\\
H_{A\otimes B} &\triangleq H_A \cup H_B\,,\\
H_{A\multimap TB} &\triangleq \{(i_A,\circledast_A),(i_B,\circledast_B),(\dot{a},\circledast_B)\} \cup H_A \cup H_B \cup H_{\xi_A} \cup H_{\xi_B}\,,\ \text{with } H_\xi \triangleq \bigcup_C H_{[\![C]\!]}\,;\\
X_1 = X_{\mathbb{N}} = X_{\mathbb{A}^{\ddot{a}}} &\triangleq \varnothing\,,\\
X_{A\otimes B} &\triangleq X_A \cup X_B\,,\\
X_{A\multimap TB} &\triangleq \{(\dot{a},\circledast_B)\} \cup X_A \cup X_B \cup X_{\xi_A} \cup X_{\xi_B}\,,\ \text{with } X_\xi \triangleq \bigcup_C X_{[\![C]\!]}\,;
\end{aligned}
$$

where we write $A \multimap TB$ as $A \multimap (\xi_A \Rightarrow (B + \mathbb{A}_e) \otimes \xi_B)$, and ξ as $\bigotimes_C(\mathbb{A}_C \Rightarrow [\![C]\!])$.
In an arena A, a store-Handle justifies (all) Questions of the form $\ddot{a}$, which we call ***store-Questions***. Answers to store-Questions are called ***store-Answers***. ▲

The classification of moves relatively to the store is familiar from $\nu\rho$. Regarding exceptions, it is obvious that X-raisers are moves that raise an exception — note here that exception names may also appear in a play as values (i.e. not inside X-raisers). We observe that X-raisers are by definition A-store-H's, justified by Q-store-H's, and that every Q-store-H justifies X-raisers. An example of how the above classes of moves are related is given in the next figure.

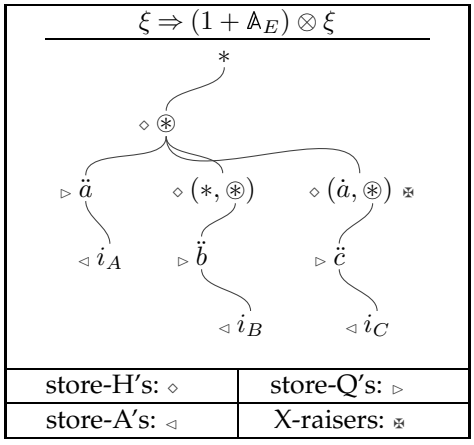


Figure 5.9: Store-H's -Q's -A's and X-raisers in arena $T1$.

From now on we work in $\mathcal{V}_{\nu\varepsilon\rho}$, unless stated otherwise. The above notions can be straightforwardly extended to prearenas, by setting

$$H_{A\to B} \triangleq H_A \cup H_B\,, \quad X_{A\to B} \triangleq X_A \cup X_B\,, \tag{5.15}$$

and similarly for store-Q's and store-A's. As in section 4.3.5, we can show that a move is exclusively either initial or a store-H or a store-Q or a store-A.

Proposition 5.32 *For any type $A \in Ob(\mathcal{V}_{\nu\varepsilon\rho})$,*

$$M_A = I_A \uplus H_A \uplus \{m \in M_A \mid m \textit{ a store-Q}\} \uplus \{m \in M_A \mid m \textit{ a store-A}\}\,.$$

■

Around these notions we define x-tidy strategies. Note that we endorse again the following notational convention. Since stores ξ may occur in several places inside a (pre)arena we may use parenthesised indices to distinguish identical moves from different stores. For example, the same store-question q may be occasionally denoted $q_{(O)}$ or $q_{(P)}$, the particular notation denoting the OP-polarity of the moves.

Definition 5.33 A total strategy σ is ***x-tidy*** if whenever odd-length $[s] \in \sigma$ then:

(TD1) If s ends in a store-Q q then $[sx] \in \sigma$, with x being either a store-A to q introducing no new names, or a copy of q. In particular, if $q = \ddot{a}^{\vec{a}}$ with $\ddot{a} \# \ulcorner s \urcorner^{-}$ then the latter case holds.

(TD2) If $[sq_{(P)}] \in \sigma$ with q a store-Q then $q_{(P)}$ is justified by the last O-store-H in $\ulcorner s \urcorner$.

(TD3) If $\ulcorner s \urcorner = s' q_{(O)} q_{(P)} t\, y_{(O)}$ with q a store-Q then $[s\, y_{(P)}] \in \sigma$ with $y_{(P)}$ justified by $\ulcorner s \urcorner .{-}3$.

(xTD1) If s ends in an X-raiser $(\dot{a}, \circledast)^{\vec{a}}$ with $\dot{a} \# \ulcorner s \urcorner^{-}$ then $[s(\dot{a}, \circledast)^{\vec{a}}] \in \sigma$.

(xTD3) If $\ulcorner s \urcorner = s'(\dot{a}, \circledast)^{\vec{a}}_{(O)} (\dot{a}, \circledast)^{\vec{a}}_{(P)} q_{(O)}$ with q a store-Q, $(\dot{a}, \circledast)_{(O)}$ an X-raiser and $\dot{a} \# s'$, then $[s\, q_{(P)}] \in \sigma$. ▲

The (TD) conditions define tidy strategies as in section 4.3.5, imposing thus a certain *store-discipline*. The (xTD) conditions provide a *fresh-exception-discipline*:

> When a fresh raised exception is encountered, it is simply copycatted.

In (xTD1), the X-raiser $(\dot{a}, \circledast)^{\vec{a}}$ played by Player is an answer and hence needs to be justified by the pending question; the following lemma shows that this is always possible.

Lemma 5.34 *If odd-length $[s] \in \sigma$ ends in an X-raiser $(\dot{a}, \circledast)^{\vec{a}}$ then s has a pending-Q which is an O-store-H, and $s(\dot{a}, \circledast)^{\vec{a}}$ is a play.*

Proof: s being odd-length implies that it has a pending question, say q. If q were a P-move then $s = s_1 q s_2$ with s_1, s_2 being odd-length, so an A in s_2 should be justified by q, $\lightning$. Hence, q an O-move. Moreover, q cannot be initial, by totality, and neither a store-Q: q being unanswered would mean that P copycats after it, so the move following q would be a copy of it answered by an O-store-A y, say. After y is played, P must answer q with a copy of y, thus y can only be the last move in s, i.e. $(\dot{a}, \circledast)^{\vec{a}}$, $\lightning$ as y a store-A. Hence, q an O-store-H. Thus, $s(\dot{a}, \circledast)^{\vec{a}}$ is a justified sequence satisfying well-bracketing, and it clearly satisfies NC's. Finally, it also satisfies visibility since s and $\ulcorner s \urcorner$ have the same pending-Q (see e.g. [McC98]). ■

It is easy to see that identity arrows are x-tidy. Moreover, x-tidy strategies compose and thus we have a category of nominal arenas and x-tidy strategies.

Proposition 5.35 *If $\sigma : A \longrightarrow B$ and $\tau : B \longrightarrow C$ are x-tidy strategies then so is $\sigma ; \tau$.*

Proof: From proposition 4.56 we know that $\sigma ; \tau$ satisfies the (TD) conditions. The (xTD) conditions are shown similarly. We only show (xTD1). So let odd-length $[s ; t] \in \sigma ; \tau$ be ending in an X-raiser $(\dot{a}, \circledast)^{\vec{a}'}$ with $\dot{a} \# \ulcorner s ; t \urcorner^{-}$. Assume, wlog, that $s ; t$ ends in A, so $s.{-}1 = (\dot{a}, \circledast)^{\vec{a}_1}$, some $\vec{a}_1 \preceq \vec{a}'$. Then, similarly to proposition 3.36, $\dot{a} \# \ulcorner s \urcorner^{-}$ so, by x-tidiness, $[s(\dot{a}, \circledast)^{\vec{a}_1}] \in \sigma$. If $(\dot{a}, \circledast)^{\vec{a}_1}$ is in A then we are done. Otherwise, we have that $[t(\dot{a}, \circledast)^{\vec{a}_2}] \in \tau$ some $\vec{a}_2 \preceq \vec{a}'$. Applying the same reasoning consecutively, some $(\dot{a}, \circledast)^{\vec{a}_n}$ is played in AC, giving the required copy of $(\dot{a}, \circledast)^{\vec{a}'}$. ■

Definition 5.36 $\chi\mathcal{T}$ is the lluf subcategory of $\mathcal{V}_{\nu\varepsilon\rho}$ of x-tidy strategies. ▲

Many of the strategies we have encountered thus far are x-tidy, but not all of them: θ_A is not x-tidy, for any object A. But this was exactly the reason for introducing $\nu\varepsilon\rho$-submodels in definition 5.19. Thus, we have the following.

Proposition 5.37 *$\chi\mathcal{T}$ forms a $\nu\varepsilon\rho$-submodel of $\mathcal{V}_{\nu\varepsilon\rho}$.*

Proof: It is not difficult to show the following (see also proposition 4.58).

▷ If $f : A \to B$, $g : A \to C$ are x-tidy then $\langle f, g\rangle$ is. Moreover, projections and terminal arrows are all x-tidy.

▷ $\eta_A, \mu_A, \tau_{A,B}$ are all x-tidy, and if h is x-tidy then Th is. Moreover, $f : A \otimes B \to TC$ is x-tidy iff $\Lambda^T(f)$ is.

▷ ε_A, δ_A are x-tidy, and if h is x-tidy then so is $Q^{\vec{a}}h$.

▷ Successor, predecessor and numeral arrows are x-tidy.

▷ Name-equality arrows are x-tidy. Moreover, $(\frac{\vec{a}}{\vec{a}'})_A$ and $\mathtt{nu}_A^{\vec{a}a}$ are x-tidy.

▷ $\mathtt{upd}_A, \mathtt{drf}_A$ are x-tidy.

▷ $\mathtt{inx}_A$ are x-tidy, and so are the following composite strategies.

$$\mathbb{A}_e \otimes TA \otimes TA \xrightarrow{\mathtt{id} \otimes \theta_A \otimes \mathtt{id}} \mathbb{A}_e \otimes T^2A \otimes TA \xrightarrow{\tau \otimes \mathtt{id}\,;\,\tau'} T(\mathbb{A}_e \otimes TA \otimes TA) \xrightarrow{T\mathtt{hdl}_A\,;\,\mu} TA$$

Hence, $\mathcal{xT}$ is a $\nu\varepsilon\rho$-submodel. ■

Henceforth, by strategies we shall mean x-tidy strategies, unless stated otherwise.

We now proceed to add the structure necessary for an observational $\nu\varepsilon\rho$-submodel.

Definition 5.38 Expand $\mathcal{xT}$ to $(\mathcal{xT}, T, Q, O)$ by setting, for each $\vec{a}$,

$$O^{\vec{a}} \triangleq \{f \in \mathcal{xT}(Q^{\vec{a}}1, T\mathbb{N}) \mid \exists \vec{b}.\, [(\vec{a}, *) * \circledast (0, \circledast)^{\vec{b}}] \in f\}\,.$$

▲

With the above definition, the semantic preorder is given as follows. For each $f, g \in \mathcal{xT}(Q^{\vec{a}}A, TB)$,

$$f \lesssim^{\vec{a}} g \iff \forall \rho \in \mathcal{xT}(Q^{\vec{a}}(A \multimap TB), T\mathbb{N}).\, (\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{\vec{a}}(g)\,;\rho \in O^{\vec{a}})\,,$$

where

$$\Lambda^{\vec{a}}(f) = \Lambda^{Q^{\vec{a}},T}(f) \triangleq Q^{\vec{a}}1 \xrightarrow{\delta} Q^{\vec{a}}Q^{\vec{a}}1 \xrightarrow{Q^{\vec{a}}\Lambda^T(\zeta'\,;\,f)} Q^{\vec{a}}(A \multimap TB)\,.$$

In order to show observationality we need to show O-adequacy and that the semantical preorder is a congruence. The former is shown in the next lemma. The latter can be shown as in the case of $\nu\rho$ (section 4.3.6) and lemma 4.18 (note that we do not need to consider θ itself, but rather the x-tidy arrow of proposition 5.37 which includes it).

Lemma 5.39 (O-Adequacy) *Let $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$ be a typed term. If $[\![M]\!] \in O^{\vec{a}}$ then there exists some P such that $\vec{a} \Vdash M \longrightarrow\!\!\!\!\!\rightarrow P \Vdash 0$.*

Proof: By lemma 5.29 it suffices to show that, for any such M, there is a non-reducing sequent $P \Vdash N$ such that $\vec{a} \Vdash M \longrightarrow\!\!\!\!\!\rightarrow P \Vdash N$, as then N would necessarily be 0. For sake of contradiction suppose the opposite, that is, that there exists an infinite reduction sequence starting from $\vec{a} \Vdash M$.

The sequence must contain infinitely many reductions from the set $\{\mathrm{HL}, \mathrm{NHL}, \mathrm{VHL}, \mathrm{XPN}\}$, or otherwise it would end in an infinite reduction sequence in $\nu\rho$, contradicting the latter's O-adequacy (lemma 4.62). Moreover, if it contained infinitely many reductions from $\{\mathrm{NHL}, \mathrm{XPN}, \mathrm{VHL}\}$ but finitely many HL reductions, then it would have either to terminate at some raised exception or to end in an infinite sequence of reductions in $\nu\rho + \mathrm{VHL}$. The latter would then produce an infinite reduction sequence in $\nu\rho$. We therefore have that $\vec{a} \Vdash M$ has a reduction sequence containing infinitely many HL reductions.

Now we can apply a similar methodology to lemma 4.62. Namely, for each term M, define $(M)^\circ$ by induction as follows.

$$(a)^\circ \triangleq a\,, \quad (x)^\circ \triangleq x\,, \quad \ldots \quad (\lambda x.M)^\circ \triangleq \lambda x.(M)^\circ\,, \quad (M\,N)^\circ \triangleq (M)^\circ(N)^\circ\,, \quad \ldots$$

and $(\texttt{try}\ N_1\ \texttt{handle}\ M \Rightarrow N_2)^\circ \triangleq \texttt{try}\ (N_1)^\circ\ \texttt{handle}\ (M)^\circ \Rightarrow \nu a.(N_2)^\circ$, some $a \notin \texttt{fn}(N_2)$.
We can show that $[\![(M)^\circ]\!] \backsimeq [\![M]\!]$, by induction on M.
Now, for the term M we are examining, $[\![M]\!] \in O^{\vec{a}}$ implies $[\![(M)^\circ]\!] \in O^{\vec{a}}$. Moreover, since $\vec{a} \vDash M$ diverges using infinitely many HL reduction steps, $\vec{a} \vDash (M)^\circ$ diverges using infinitely many NEW reduction steps. But the latter contradicts $[\![(M)^\circ]\!] \in O^{\vec{a}}$. ■

Hence, we can show the following.

Proposition 5.40 (Observationality) *$\chi\mathcal{T}$ is observational.* ■

Our last task is to show ip-definability. Our methodology follows closely that of section 4.3.7, and therefore we will be omitting some proofs which are similar to proofs presented therein.

We start by defining truncation functions for x-tidy strategies, the notion of finitary strategy, and a sub-strategy constructor.

Definition 5.41 Let $\sigma : A \longrightarrow B$ in $\chi\mathcal{T}$ and let $[s] \in \texttt{viewf}(\sigma)$ be of even length. Define $\texttt{trunc}(s)$ and $\texttt{trunc}'(s)$ by induction as follows.

$$\texttt{trunc}(\epsilon) = \texttt{trunc}'(\epsilon) \triangleq \epsilon$$

$$\texttt{trunc}(x_{(O)}y_{(P)}s') \triangleq \begin{cases} \epsilon & \text{, if } x = y \text{ are store-Q's} \\ \epsilon & \text{, if } x = y \text{ are fresh X-raisers} \\ xy\,\texttt{trunc}(s') & \text{, o.w.} \end{cases}$$

$$\texttt{trunc}'(x_{(O)}y_{(P)}s') \triangleq \begin{cases} \epsilon & \text{, if } x = y \text{ are store-Q's} \\ \epsilon & \text{, if } x \text{ a store-Q, } y \text{ a store-A and } s' = \epsilon \\ \epsilon & \text{, if } x \in I_A,\ y \in I_B \text{ and } s' = \epsilon \\ \epsilon & \text{, if } x = y \text{ are fresh X-raisers} \\ xy\,\texttt{trunc}'(s') & \text{, o.w.} \end{cases}$$

Moreover, say σ is ***finitary*** if $\texttt{trunc}(\sigma)$ is finite, where

$$\texttt{trunc}(\sigma) \triangleq \{[\texttt{trunc}(s)] \mid [s] \in \texttt{viewf}(\sigma) \wedge |s| > 3\}\,.$$

Finally, for any $[t] \in \sigma$ define:

$$\sigma_{\leq t} \triangleq \texttt{strat}\{[s] \in \texttt{viewf}(\sigma) \mid \exists\, t' \leq t.\, \texttt{trunc}'(s) = \ulcorner t' \urcorner\}\,.$$

▲

Hence, we call finitary those strategies whose viewfunctions become finite if we delete all the store-copycats, all default initial answers, and all fresh-exception-copycats. On the other hand, the strategy $\sigma_{\leq t}$ is the strategy we are left with if we truncate $\texttt{viewf}(\sigma)$ by removing all its branches of length greater than 3 which are not contained in t, except for:

- the store-copycats and the fresh-exception-copycats, which are left intact,
- the store-A's branches which are truncated to the point of leaving solely the store-A, so that we retain tidiness.

Note that, in general, $\texttt{trunc}'(s) \leq \texttt{trunc}(s) \leq s$. We can now show the following.

Proposition 5.42 *If σ is an x-tidy strategy and $[t] \in \sigma$ is even-length then $\sigma_{\leq t}$ is a finitary x-tidy strategy with $[t] \in \sigma_{\leq t}$ and $\sigma_{\leq t} \sqsubseteq \sigma$.* ■

The proof of definability is facilitated by the following decomposition lemma (cf. lemma 4.66 and its proof).

Lemma 5.43 (Decomposition Lemma) *Let $\sigma : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!]$ be a strategy. We can decompose σ as follows.*

1. *If there exists an $i_{A(0)}$ such that $\exists x_0.\, [(\vec{a}, i_{A(0)}) \ast \circledast x_0] \in \sigma$ then*

$$\sigma = Q^{\vec{a}}[\![A]\!] \xrightarrow{\langle [x \overset{\vec{a}}{=} i_{A(0)}], \langle \sigma_0, \sigma' \rangle \rangle} \mathbb{N} \otimes (T[\![B]\!])^2 \xrightarrow{\mathtt{cnd}} T[\![B]\!]$$

where:

$$\begin{aligned}
[x \overset{\vec{a}}{=} i_{A(0)}] : Q^{\vec{a}}[\![A]\!] \longrightarrow \mathbb{N} &\triangleq \{[(\vec{a}, i_{A(0)})\, 0]\} \cup \{[(\vec{a}, i_A)\, 1] \mid [(\vec{a}, i_A)] \neq [(\vec{a}, i_{A(0)})]\}\,, \\
\sigma_0 : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] &\triangleq \mathtt{strat}\{\ [(\vec{a}, i_{A(0)})\, s] \in \mathtt{viewf}(\sigma)\ \}\,, \\
\sigma' : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] &\triangleq \mathtt{strat}\{\ [(\vec{a}, i_A)\, s] \in \mathtt{viewf}(\sigma) \mid [(\vec{a}, i_A)] \neq [(\vec{a}, i_{A(0)})]\ \}\,.
\end{aligned}$$

2. *If there exists an $i_{A(0)}$ such that $\forall i_A.\, (\exists x_0.\, [(\vec{a}, i_A) \ast \circledast x_0] \in \sigma) \iff [(\vec{a}, i_A)] = [(\vec{a}, i_{A(0)})]$, then $\sigma = \langle\!\langle \vec{b} \rangle\!\rangle\, \sigma_{\vec{b}}$, where:*

$$\sigma_{\vec{b}} : Q^{\vec{a}\vec{b}}[\![A]\!] \longrightarrow T[\![B]\!] \triangleq \mathtt{strat}\{\ [(\vec{a}\vec{b}, i_{A(0)}) \ast \circledast m_0\, s^{\setminus \vec{b}}] \mid [(\vec{a}, i_{A(0)}) \ast \circledast m_0^{\vec{b}}\, s] \in \mathtt{viewf}(\sigma)\ \}\,.$$

3. *If there exist $i_{A(0)}, m_0$ such that $\forall i_A, x.\, [(\vec{a}, i_A) \ast \circledast x] \in \sigma \iff [(\vec{a}, i_A)\, x] = [(\vec{a}, i_{A(0)})\, m_0]$, then one of the following is the case.*

(a) *$m_0 = \ddot{a}$, a store-Q of type C under $\circledast$, in which case we have $\sigma = \sigma' \upharpoonright (\vec{a}, i_{A(0)})$, where:*

$$\begin{aligned}
\sigma' &\triangleq Q^{\vec{a}}[\![A]\!] \xrightarrow{\langle \mathtt{id}, \phi \rangle} Q^{\vec{a}}[\![A]\!] \otimes T[\![C]\!] \xrightarrow{\tau\,;\, T\zeta'} TQ^{\vec{a}}([\![A]\!] \otimes [\![C]\!]) \xrightarrow{T\sigma_{\ddot{a}}} T^2[\![B]\!] \xrightarrow{\mu} T[\![B]\!]\,, \\
\sigma_{\ddot{a}} &\triangleq \mathtt{strat}\{\ [(\vec{a}, i_{A(0)}, i_C) \ast \circledast s] \mid [(\vec{a}, i_{A(0)}) \ast \circledast \ddot{a}\, i_C\, s] \in \mathtt{viewf}(\sigma)\ \}\,, \\
\phi : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![C]\!] &\triangleq \begin{cases} Q^{\vec{a}}!\,;\, \frac{\vec{a}}{\ddot{a}}\,;\, \mathtt{drf}_C & \text{, if } \ddot{a} \in \mathtt{S}(\vec{a}) \\ Q^{\vec{a}}\pi_j\,;\, \frac{\vec{a}}{\epsilon}\,;\, \mathtt{drf}_C & \text{, if } \ddot{a} \# \vec{a}\,. \end{cases}
\end{aligned}$$

(b) *$m_0 = j_A \vee m_0 = (i_B / \dot{a}, \circledast)$, a store-H, in which case if $[(\vec{a}, i_{A(0)}) \ast \circledast m_0\, \ddot{a}\, i_C] \in \sigma$, for some store-Q $\ddot{a}$ and store-A i_C, then*

$$\sigma = Q^{\vec{a}}[\![A]\!] \xrightarrow{\langle \Delta, \sigma_{\ddot{a}} \rangle} Q^{\vec{a}}[\![A]\!] \otimes Q^{\vec{a}}[\![A]\!] \otimes T[\![C]\!] \xrightarrow{\tau; T(\mathtt{id} \otimes \phi; \tau); \mu} TQ^{\vec{a}}[\![A]\!] \xrightarrow{T\sigma'\,;\, \mu} T[\![B]\!]$$

where:

$$\begin{aligned}
\sigma_{\ddot{a}} : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![C]\!] &\triangleq \mathtt{strat}\{\ [(\vec{a}, i_{A(0)}) \ast \circledast (i_C, \circledast)\, s] \mid [(\vec{a}, i_{A(0)}) \ast \circledast m_0\, \ddot{a}\, i_C\, s] \in \mathtt{viewf}(\sigma) \\
&\qquad\qquad \vee\, [\circledast\ \circledast\ s] \in \mathtt{viewf}(\mathtt{id}_\xi)\ \}\,, \\
\sigma' : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] &\triangleq \mathtt{strat}(\ \{[(\vec{a}, i_{A(0)}) \ast \circledast m_0\, y\, s] \in \mathtt{viewf}(\sigma) \mid y \neq \ddot{a}\ \} \\
&\qquad\qquad \cup \{[(\vec{a}, i_{A(0)}) \ast \circledast m_0\, \ddot{a}\, s] \mid [\circledast\ \circledast\ \ddot{a}\, s] \in \mathtt{viewf}(\mathtt{id}_\xi)\}\ )\,, \\
\phi : Q^{\vec{a}}[\![A]\!] \otimes [\![C]\!] \longrightarrow T1 &\triangleq \begin{cases} (Q^{\vec{a}}!\,;\, \frac{\vec{a}}{\ddot{a}}) \otimes \mathtt{id}_{[\![C]\!]}\,;\, \mathtt{upd}_C & \text{, if } \ddot{a} \in \mathtt{S}(\vec{a}) \\ (Q^{\vec{a}}\pi_j\,;\, \frac{\vec{a}}{\epsilon}) \otimes \mathtt{id}_{[\![C]\!]}\,;\, \mathtt{upd}_C & \text{, if } \ddot{a} \# \vec{a}\,. \end{cases}
\end{aligned}$$

In both cases above, we take $j = \min\{j \mid (i_{A(0)})_j = \ddot{a}\}$. ∎

Theorem 5.44 (Definability) *Let A, B be types and $\sigma : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!]$ be finitary. Then σ is definable.*

Proof: The proof follows the same route as the proof of definability in $\nu\rho$ (theorem 4.67). We only show the parts where there is some extra work needed.

The proof is by induction on $(|\mathtt{trunc}(\sigma)|, \|\sigma\|)$. Using the Decomposition Lemma, we reduce the inductive step to showing that for any $\sigma_0 : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!]$ with $(0,0) < (|\mathtt{trunc}(\sigma_0)|, \|\sigma_0\|) \leq (|\mathtt{trunc}(\sigma)|, \|\sigma\|)$ and such that, for some $i_{A(0)}, m_0$,

$$\forall i_A, x.\, [(\vec{a}, i_A) \ast \circledast x] \in \sigma_0 \iff [(\vec{a}, i_A)\, x] = [(\vec{a}, i_{A(0)})\, m_0]\,,$$

with m_0 a store-H and with σ_0 not updating any names before playing m_0, there exists a term M_0 with $\sigma_0 = [\![M_0]\!] \upharpoonright (\vec{a}, i_{A(0)})$. The case of $m_0 = (i_B, \circledast)$ is treated exactly as in theorem 4.67. We need also check the cases $m_0 = (\dot{a}, \circledast)$ and $m_0 = j_A$.

If $m_0 = (\dot{a}, \circledast)$ then $\sigma_0 = [\![M_0]\!] \upharpoonright (\vec{a}, i_{A(0)})$ by taking

$$M_0 \triangleq \begin{cases} \mathtt{raise}\, \dot{a} & \text{, if } \dot{a} \in \mathtt{S}(\vec{a}) \\ \mathtt{raise}\, z'_j & \text{, if } \dot{a} \# \vec{a} \wedge j = \min\{i \mid \dot{a} = (i_{A(0)})_{k+i}\} \end{cases}.$$

If $m_0 = j_A$, played in some $A_{k+k'+i} = A'_i \to A''_i$, then $m_0 = (i_{A'_i}, \circledast)$. Assume that $A'_i = A'_{i,1} \times \cdots \times A'_{i,n_i}$ with $A'_{i,p}$'s being non-products. Now, O can either ask some name $\ddot{a}$ (which would lead to a store-CC), or answer at A''_i, or raise a known exception $\dot{b}$, or raise some fresh exception $\dot{a}$ (which would lead to an exception-CC), or play at some $A'_{i,p}$ of arrow type, say $A'_{i,p} = C_{i,p} \to C'_{i,p}$. Hence, taking $S \triangleq \mathtt{S}(\vec{a}, i_{A(0)})$ we have:

$$\mathtt{viewf}(\sigma_0) = f_A \cup \bigcup\nolimits_{\dot{b} \in S} f_{\dot{b}} \cup \bigcup\nolimits_{p=1}^{n_i} f_p$$

where:

$$\begin{aligned}
f_A &\triangleq f_0 \cup \{\, [(\vec{a}, i_{A(0)}) * \circledast\, (i_{A'_i}, \circledast)\, (i_{A''_i}, \circledast)\, s] \in \mathtt{viewf}(\sigma_0) \,\} \\
f_{\dot{b}} &\triangleq f_0 \cup \{\, [(\vec{a}, i_{A(0)}) * \circledast\, (i_{A'_i}, \circledast)\, (\dot{b}, \circledast)\, s] \in \mathtt{viewf}(\sigma_0) \,\} \\
f_p &\triangleq f_0 \cup \{\, [(\vec{a}, i_{A(0)}) * \circledast\, (i_{A'_i}, \circledast)\, (i_{C_{i,p}}, \circledast)\, s] \in \mathtt{viewf}(\sigma_0) \,\} \\
f_0 &\triangleq \{\, [(\vec{a}, i_{A(0)}) * \circledast\, (i_{A'_i}, \circledast)\, s] \mid [\circledast\ \circledast\ s] \in \mathtt{viewf}(\mathtt{id}_\xi) \\
&\qquad\qquad \vee\, (s.1 = (\dot{a}, \circledast) \wedge \dot{a} \notin S \wedge [s] \in \mathtt{viewf}(\mathtt{id}_{\mathbb{A}_e \otimes \xi})) \,\}
\end{aligned}$$

and where we assume $f_p \triangleq f_0$ if $A'_{i,p}$ is not an arrow type. It is not difficult to see that $f_A, f_{\dot{b}}, f_p$ are viewfunctions. Now, from f_A we obtain $f'_A : Q^{\vec{a}}([\![A]\!] \otimes [\![A''_i]\!]) \longrightarrow T[\![B]\!]$ by:

$$f'_A \triangleq \{\, [(\vec{a}, i_{A(0)}, i_{A''_i}) * \circledast\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, (i_{A'_i}, \circledast)\, (i_{A''_i}, \circledast)\, s] \in f_A \,\}.$$

It is not difficult to see that f'_A is indeed an (x-tidy) viewfunction. By IH, there exists some $\vec{a} \mid \Gamma, y{:}A''_i \vdash M_A : B$ such that $[\![M_A]\!] = \mathtt{strat}(f'_A)$.
From each $f_p \neq f_0$ we obtain a viewfunction $f'_p : Q^{\vec{a}}([\![A]\!] \otimes [\![C_{i,p}]\!]) \longrightarrow T[\![C'_{i,p}]\!]$ by:

$$f'_p \triangleq \{[(\vec{a}, i_{A(0)}, i_{C_{i,p}}) * \circledast\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, (i_{A'_i}, \circledast)\, (i_{C_{i,p}}, \circledast)\, s] \in f_p\}.$$

By IH, there exists some $\vec{a} \mid \Gamma, y' : C_{i,p} \vdash M_p : C'_{i,p}$ such that $[\![M_p]\!] = \mathtt{strat}(f'_p)$, so take $V_p \triangleq \lambda y'.M_p$. For each $A'_{i,p}$ of non-arrow type, the behaviour of σ_0 at $A'_{i,p}$ is fully described by $(i_{A'_i})_p$, so we take V_p to be the denotation of $(i_{A'_i})_p$. $\langle V_1, ..., V_{n_i} \rangle$ is now of type A'_i and describes σ_0's behaviour in A'_i.
Finally, from each $f_{\dot{b}}$ we obtain a viewfunction:

$$f'_{\dot{b}} : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] \triangleq \{[(\vec{a}, i_{A(0)}) * \circledast\, s] \mid [(\vec{a}, i_{A(0)}) * \circledast\, (i_{A'_i}, \circledast)\, (\dot{b}, \circledast)\, s] \in f_{\dot{b}}\}.$$

By IH, there exists some $\vec{a} \mid \Gamma \vdash M_{\dot{b}} : B$ such that $[\![M_{\dot{b}}]\!] = \mathtt{strat}(f'_{\dot{b}})$.
Now, taking for each known exception-name $\dot{b}$

$$N_{\dot{b}} \triangleq \begin{cases} \dot{b} & \text{, if } \dot{b} \in \mathtt{S}(\vec{a}) \\ z'_j & \text{, if } \dot{b} \# \vec{a} \wedge j = \min\{i \mid \dot{b} = (i_{A(0)})_{k+i}\} \end{cases},$$

and

$$M_0 \triangleq (\mathtt{try}\, (\lambda x'.\lambda x.(\lambda y.M_A)\, x')(z''_i \langle V_1, ..., V_{n_i} \rangle)\, \mathtt{handle}\, \overrightarrow{N_{\dot{b}}} \Rightarrow \overrightarrow{\lambda x.M_{\dot{b}}})\, \mathtt{skip},$$

for some x, x' not free in $M_A, M_{\dot{b}}$'s, we obtain $\sigma_0 = [\![M_0]\!] \upharpoonright (\vec{a}, i_{A(0)})$. ∎

Corollary 5.45 (ip-Definability) *$\mathcal{XT} = (\mathcal{XT}, T, Q, O)$ is an ip-definable $\nu\varepsilon\rho$-submodel.*

Proof: For each $\vec{a}, A, B$, define $D^{\vec{a}}_{A,B} \triangleq \{f : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] \mid f \text{ is finitary}\}$. By definability, every $f \in D^{\vec{a}}_{A,B}$ is definable. We need also show:

$$(\forall \rho \in D^{\vec{a}}_{A\to B,\mathbb{N}} \,.\, \Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{\vec{a}}(g)\,;\rho \in O^{\vec{a}}) \implies f \lesssim^{\vec{a}} g\,.$$

Assume the LHS assertion holds and let $\Lambda^{\vec{a}}(f)\,;\rho \in O^{\vec{a}}$, some $\rho : Q^{\vec{a}}([\![A]\!] \multimap T[\![B]\!]) \longrightarrow T\mathbb{N}$. Then, let $[s\,;t] = [(\vec{a},*) * \circledast(0,\circledast)^{\vec{b}}] \in \Lambda^{\vec{a}}(f)\,;\rho$, $[s] \in \Lambda^{\vec{a}}(f)$ and $[t] \in \rho$. Then, by proposition 5.42, $[t] \in \rho_{\leq t}$, so $\Lambda^{\vec{a}}(f)\,;\rho_{\leq t} \in O^{\vec{a}}$. Moreover, $\rho_{\leq t} \in D^{\vec{a}}_{A\to B,\mathbb{N}}$, so $\Lambda^{\vec{a}}(g)\,;\rho_{\leq t} \in O^{\vec{a}}$, by hypothesis. Finally, $\rho_{\leq t} \sqsubseteq \rho$ implies $\Lambda^{\vec{a}}(g)\,;\rho_{\leq t} \sqsubseteq \Lambda^{\vec{a}}(g)\,;\rho$, hence the latter observable, so $f \lesssim^{\vec{a}} g$. ■

Hence, we have finally shown the following.

Theorem 5.46 *$\mathcal{XT} = (\mathcal{XT}, T, Q, O)$ is a fully abstract model of $\nu\varepsilon\rho$.* ■

5.2.6 Equivalences established semantically

Reasoning as in section 4.3.8 we can show that, for any $B \in \mathrm{TY}$ and taking $\ddot{a} \in \mathbb{A}_B$, we have $\nu\ddot{a}.!\,\ddot{a} \cong \mathtt{stop}_B$ in the $\nu\varepsilon\rho$-calculus. Moreover, using the fact that $\nu\varepsilon\rho$-environments are x-tidy we can also show in a similar way that

$$\mathtt{stop}_B \cong \nu\dot{a}.\mathtt{raise}\,\dot{a}\,. \tag{5.16}$$

This implies that $\nu\dot{a}.\mathtt{raise}\,\dot{a} \lessapprox M$ for any $\epsilon \mid \varnothing \vdash M : B$, in $\nu\varepsilon\rho$ *and* in $\nu\varepsilon$.

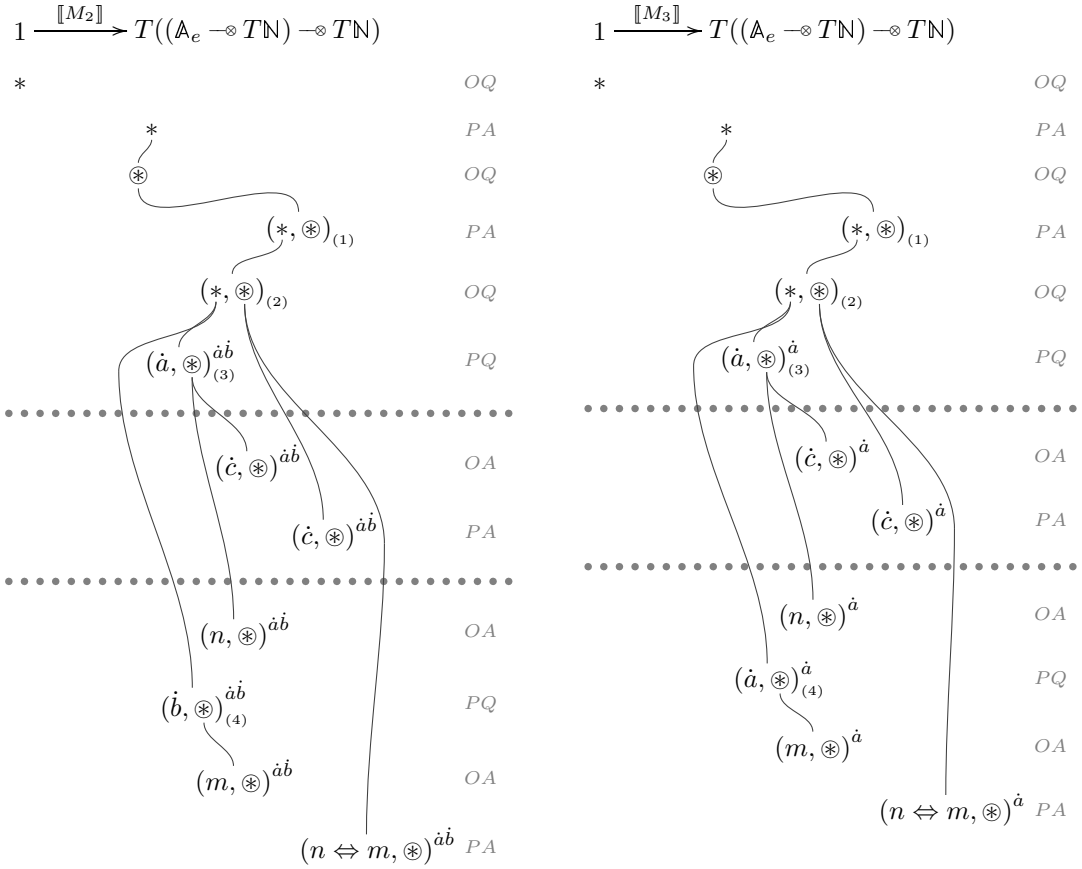
Let us now prove equivalence (5.3) in the $\nu\varepsilon$-calculus using the full-abstraction result for $\nu\varepsilon\rho$. Recall that

$$M_2 \triangleq \lambda f.\nu\dot{a}.\nu\dot{b}.[f\dot{a} \Leftrightarrow f\dot{b}] : (\mathbb{E}\to\mathbb{N})\to\mathbb{N}, \quad M_3 \triangleq \lambda f.\nu\dot{a}.[f\dot{a} \Leftrightarrow f\dot{a}] : (\mathbb{E}\to\mathbb{N})\to\mathbb{N},$$

and that we need to show $M_2 \cong M_3$. By full-abstraction of $\nu\varepsilon\rho$ (in fact, by correctness and adequacy), it suffices to show that, for any $\rho : T((\mathbb{A}_e \multimap T\mathbb{N}) \multimap \mathbb{N}) \longrightarrow T\mathbb{N}$ which is tl4 and does *not* use the store,

$$[\![M_2]\!]\,;\rho \in O^{\epsilon} \iff [\![M_3]\!]\,;\rho \in O^{\epsilon}. \tag{5.17}$$

In fact, it suffices to assume ρ does not use the store for dereferencings, i.e. it does not ask store-Q's unless in a copycat. The viewfunctions of $[\![M_2]\!]$ and $[\![M_3]\!]$ are given below. Note that we have omitted store-copycat links (as we won't be using the store) and also the exception-copycat that occurs if Opponent plays an exception under $(\dot{b},\circledast)^{\dot{a}\dot{b}}_{(4)}/(\dot{a},\circledast)^{\dot{a}}_{(4)}$.



We show only one direction of the equivalence; the other is shown by a similar argument. Let $[* * \circledast (0, \circledast)^{\vec{a}}] \in [\![M_2]\!] ; \rho$, some $\rho, \vec{a}$ with ρ being tl4 and not asking store-Q's. Then, the interaction witnessing this sequence starts with $[* * * \circledast \circledast^{\vec{b}}]$, some $\vec{b}$ introduced by ρ, to which $[\![M_2]\!]$ plays $(*, \circledast)^{\vec{b}}_{(1)}$. At this point, ρ can either play $(0, \circledast)^{\vec{a}}$ or ask some $(*, \circledast)^{\vec{b}_1}_{(2)}$. In the latter case, $[\![M_2]\!]$ plays $(\dot{a}_1, \circledast)^{\vec{b}_1 \dot{a}_1 \dot{b}_1}_{(3)}$ and now ρ has three choices: either play some $(n, \circledast)^{\vec{b}_1 \dot{a}_1 \dot{b}_1 \vec{c}}$, or ask again some $(*, \circledast)^{\vec{b}_2}_{(2)}$, or play some exception $(\dot{c}, \circledast)^{\vec{b}_1 \dot{a}_1 \dot{b}_1 \vec{c}}$. In the latter case, $[\![M_2]\!]$ responds by also playing $(\dot{c}, \circledast)^{\vec{b}_1 \dot{a}_1 \dot{b}_1 \vec{c}}$. Note that $\dot{c}$ cannot be $\dot{b}_1$ as then x-tidiness of ρ would copycat the exception to the output giving $[* * \circledast (\dot{c}, \circledast)^{\vec{b}_1 \dot{a}_1 \dot{b}_1 \vec{c}}] \in [\![M_2]\!] ; \rho$. Hence, the interaction can be simulated (modulo $\_^{\dot{b}_1}$) by $[\![M_3]\!] ; \rho$. At this point, ρ can play either $(*, \circledast)^{\vec{b}_2}_{(2)}$ or $(0, \circledast)^{\vec{a}}$. In the former case, $[\![M_2]\!]$ will play $(\dot{a}_2, \circledast)^{\vec{b}_2 \dot{a}_2 \dot{b}_2}_{(3)}$ with $\dot{a}_1 \neq \dot{a}_2$. Up to now, the interaction can be simulated by $[\![M_3]\!] ; \rho$, as the $\dot{b}_i$'s have not played any part.

So suppose that, after some rounds of Opponent answering to $(\dot{a}_i, \circledast)^{\vec{b}_i \dot{a}_i \dot{b}_i}_{(3)}$ with exceptions or with fresh openings of $(*, \circledast)^{\vec{b}_j}_{(2)}$, Opponent plays some $(n, \circledast)^{\vec{b}_k \dot{a}_k \dot{b}_k \vec{c}}$. At this point, $[\![M_2]\!]$ plays $(\dot{b}_k, \circledast)^{\vec{b}_k \dot{a}_k \dot{b}_k \vec{c}}_{(4)}$ and the play continues. But now $(\dot{b}_k, \circledast)^{\vec{b}_k \dot{a}_k \dot{b}_k \vec{c}}_{(4)}$ has hidden $\dot{a}_k$ from the P-view of ρ and therefore, because of innocence, the latter will play in the same way as if $(\dot{a}_k, \circledast)^{\vec{b} \dot{a}_k \dot{b}_k \vec{c}}_{(4)}$ had been played. Hence, $[\![M_3]\!] ; \rho$ can simulate this appearance of $\dot{b}_k$. Using the precisely the same argument for all appearances of $\dot{b}_i$'s, we have that $[\![M_3]\!] ; \rho$ can simulate the whole interaction.

Chapter 6

Conclusion

E quindi uscimmo a riveder le stelle.

Dante, *La Divina Commedia, Inferno.*

Summary

In this thesis we have examined the semantics of nominal computation, that is, of computation capable of creating fresh names, comparing them and passing them around. Following the work on the ν-calculus, a characteristic feature of our approach is the stipulation of names being constants rather than variables. We find this more adequate not only for denotational reasons (absence of 'bad' constructors) but also because it seems more intuitive: names are just like integers, but can only be compared for equality.

The constants-as-names rationale allows for a simple — syntactic and denotational — modelling of names once an adequate framework for such constants has been laid down. The chosen relevant framework is that of nominal sets, that is, sets supplied with atoms and atom-permutation technology, which provide an elegant foundational mathematical theory for reasoning with names (by atoms). The whole discussion — of nominal computation and its semantics — was made inside the universe of nominal sets.

Our denotational models were built in game semantics, and in particular in nominal games. The latter are stateful, call-by-value games built inside nominal sets: names appear as constants inside games, and states contain precisely the names created along a computation. We have examined names for general references and exceptions. The methodology followed was that of establishing a basic category of games corresponding to a basic nominal calculus, and from that obtain models of the additional nominal computational effects by means of computational monads. Thus, our models differ importantly from previous models of general references and exceptions. While in those models names were semantically modelled as compound objects encapsulating the structure necessary for name-manipulation (i.e. read/write or raise/handle methods), in our models names are elementary objects manipulated by computational monads. This feature allowed us to obtain fully complete models without the need to add 'bad' constructors in the language.

Further directions

This thesis has taken some basic steps in the modelling of nominal computation which can serve as a stable platform for further research on names. A first further direction is that of characterising the nominal effect — i.e. the computational effect that arises from the use of names — in abstract categorical terms. Here we have pursued this task to some extent by introducing the monadic-comonadic description of nominal computation, but it is evident that the description needs further investigation. We see that there are more monad-comonad

connections to be revealed, which will simplify and further substantiate the presentation. The work of Schöpp which examines categories with names [Sch06] seems to be particularly helpful in this direction.

A second direction, which has not been pursued here, is that of decidability of observational equivalence in nominal languages. The use of denotational methods, and game semantics in particular, for attacking the problem has been extremely successful in the 'non-nominal' case, having characterised decidability of (fragments of) Idealized Algol [GM00, Ong02, Mur03]. It would therefore be useful to 'nominalise' that body of work and apply it to nominal calculi. Already from [Mur03] we can deduce that nominal languages with ground store are undecidable, and from [PS93] we know that equivalence is decidable for programs of first-order type in the ν-calculus, but otherwise the problem remains open. A major challenge to be faced is that the fully abstract models we have devised lean heavily on semantical quotientings and therefore disallow direct reasoning on strategies. To this end, Laird's approach to nominal games [Lai08] seems very relevant. A first step, covering Reduced ML, has been taken in [MT09].

A third direction would be to examine nominal languages for concurrent computation. In concurrency, names may also appear in threads, channels etc, and it is therefore natural to extend nominal games to the concurrent setting. Usually, the passage from sequential games to concurrent games is achieved by interleaving of sequential plays [GM04, Lai05], an approach that could be tested in the nominal setting. It would also be interesting to examine formal properties, such as private names and common store, of nominal concurrent languages. Work in this direction has so far only been conducted by Laird [Lai06].

Finally, it would be interesting to examine AJM-games [AJM00] under the lens of nominal sets. A distinctive feature of AJM-games is the use of moves-with-tags inside a play in order to distinguish between different threads of computation. Evidently, strategies need only distinguish between different tags and be impervious to permutation of tags. We see that tags play a role of names and seems therefore natural to use atoms as tags. This approach seems more natural than the usual naturals-as-tags and would greatly simplify the presentation of AJM-games allowing also the nominal consideration of issues arising in Linear Logic and Geometry of Interaction.

Appendix A

Deferred Proofs

Proof of lemma 4.10: The first part is by induction on M, using substantially (N2). For the second part we do again induction on M. The base cases are straightforward; for the induction step we show the following cases:

$\triangleright$ If $M = \nu a.N$ then, assuming $a \# \vec{a}$,

$$\begin{aligned}
[\![M\{\vec{V}/\vec{x}\}]\!] &= [\![\nu a.N\{\vec{V}/\vec{x}\}]\!] = \mathtt{nu}_\Gamma \,;\, T[\![N\{\vec{V}/\vec{x}\}]\!] \,;\, \mu \\
&\overset{IH}{=} \mathtt{nu}_\Gamma \,;\, T(\langle \mathtt{id}, \tfrac{\vec{a}a}{\vec{a}} \,;\, |\vec{V}|\rangle \,;\, \zeta' \,;\, Q^{\vec{a}a}\pi_2 \,;\, [\![N]\!]) \,;\, \mu \\
&\overset{\text{(N2)}}{=} \langle \mathtt{nu}_\Gamma, \mathtt{id}\rangle \,;\, \tau' \,;\, T(\mathtt{id} \times |\vec{V}|) \,;\, T\zeta' \,;\, TQ^{\vec{a}a}\pi_2 \,;\, T[\![N]\!] \,;\, \mu \\
&= \langle \mathtt{nu}_\Gamma, \mathtt{id}\rangle \,;\, \tau' \,;\, T\zeta' \,;\, TQ^{\vec{a}a}(\mathtt{id} \times |\vec{V}| \,;\, \pi_2) \,;\, T[\![N]\!] \,;\, \mu \\
&= \Delta_{Q^{\vec{a}}\Gamma} \,;\, \zeta' \,;\, \mathtt{nu}_{\Gamma \times Q^{\vec{a}}\Gamma} \,;\, TQ^{\vec{a}a}(\mathtt{id} \times |\vec{V}| \,;\, \pi_2) \,;\, T[\![N]\!] \,;\, \mu \\
&= \Delta_{Q^{\vec{a}}\Gamma} \,;\, \zeta' \,;\, Q^{\vec{a}}(\mathtt{id} \times |\vec{V}| \,;\, \pi_2) \,;\, \mathtt{nu}_\Gamma \,;\, T[\![N]\!] \,;\, \mu \\
&= \Delta_{Q^{\vec{a}}\Gamma} \,;\, \mathtt{id} \times |\vec{V}| \,;\, \zeta' \,;\, Q^{\vec{a}}\pi_2 \,;\, \mathtt{nu}_\Gamma \,;\, T[\![N]\!] \,;\, \mu \,.
\end{aligned}$$

$\triangleright$ If $M = \lambda x.N$ then,

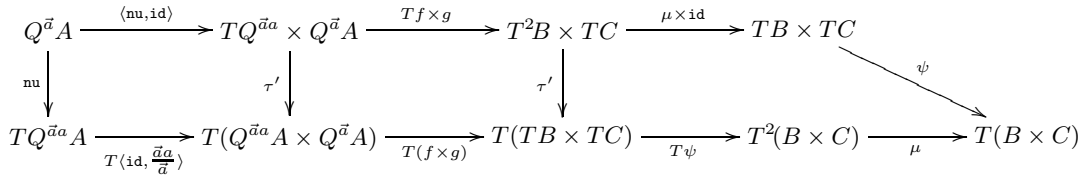
$$\begin{aligned}
[\![M\{\vec{V}/\vec{x}\}]\!] &= [\![\lambda x.(N\{\vec{V}, x/\vec{x}, x\})]\!] = \Lambda^T(\zeta' \,;\, [\![N\{\vec{V}, x/\vec{x}, x\}]\!]) \,;\, \eta \\
&\overset{IH}{=} \Lambda^T(\zeta' \,;\, \langle \mathtt{id}, |\vec{V}|, |x|\rangle \,;\, \zeta' \,;\, Q^{\vec{a}}\pi_2 \,;\, [\![N]\!]) \,;\, \eta \\
&= \Lambda^T(\langle \mathtt{id}, |\vec{V}|\rangle \times \mathtt{id} \,;\, \zeta' \,;\, Q^{\vec{a}}\pi_2 \,;\, [\![N]\!]) \,;\, \eta \\
&= \Lambda^T(\langle \mathtt{id}, |\vec{V}|\rangle \times \mathtt{id} \,;\, \zeta' \times \mathtt{id} \,;\, Q^{\vec{a}}\pi_2 \times \mathtt{id} \,;\, \zeta' \,;\, [\![N]\!]) \,;\, \eta \\
&= \langle \mathtt{id}, |\vec{V}|\rangle \,;\, \zeta' \,;\, Q^{\vec{a}}\pi_2 \,;\, \Lambda^T(\zeta' \,;\, [\![N]\!]) \,;\, \eta \,.
\end{aligned}$$

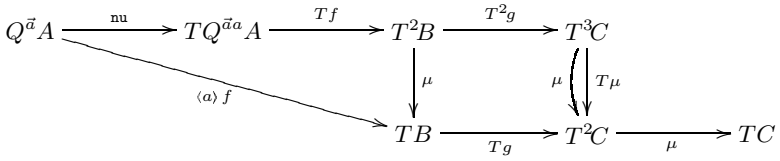
$\triangleright$ The case of $M = N\,K$ is paradigmatic for all other cases:

$$\begin{aligned}
[\![M\{\vec{V}/\vec{x}\}]\!] &= [\![N\{\vec{V}/\vec{x}\}\,K\{\vec{V}/\vec{x}\}]\!] = \langle [\![N\{\vec{V}/\vec{x}\}]\!], [\![K\{\vec{V}/\vec{x}\}]\!]\rangle \,;\, \psi \,;\, T\mathsf{ev}^T \,;\, \mu \\
&\overset{IH}{=} \langle \langle \mathtt{id}, |\vec{V}|\rangle \,;\, \zeta' \,;\, Q^{\vec{a}}\pi_2 \,;\, [\![N]\!], \langle \mathtt{id}, |\vec{V}|\rangle \,;\, \zeta' \,;\, Q^{\vec{a}}\pi_2 \,;\, [\![K]\!]\rangle \,;\, \psi \,;\, T\mathsf{ev}^T \,;\, \mu \\
&= \langle \mathtt{id}, |\vec{V}|\rangle \,;\, \zeta' \,;\, Q^{\vec{a}}\pi_2 \,;\, \langle [\![N]\!], [\![K]\!]\rangle \,;\, \psi \,;\, T\mathsf{ev}^T \,;\, \mu
\end{aligned}$$

■

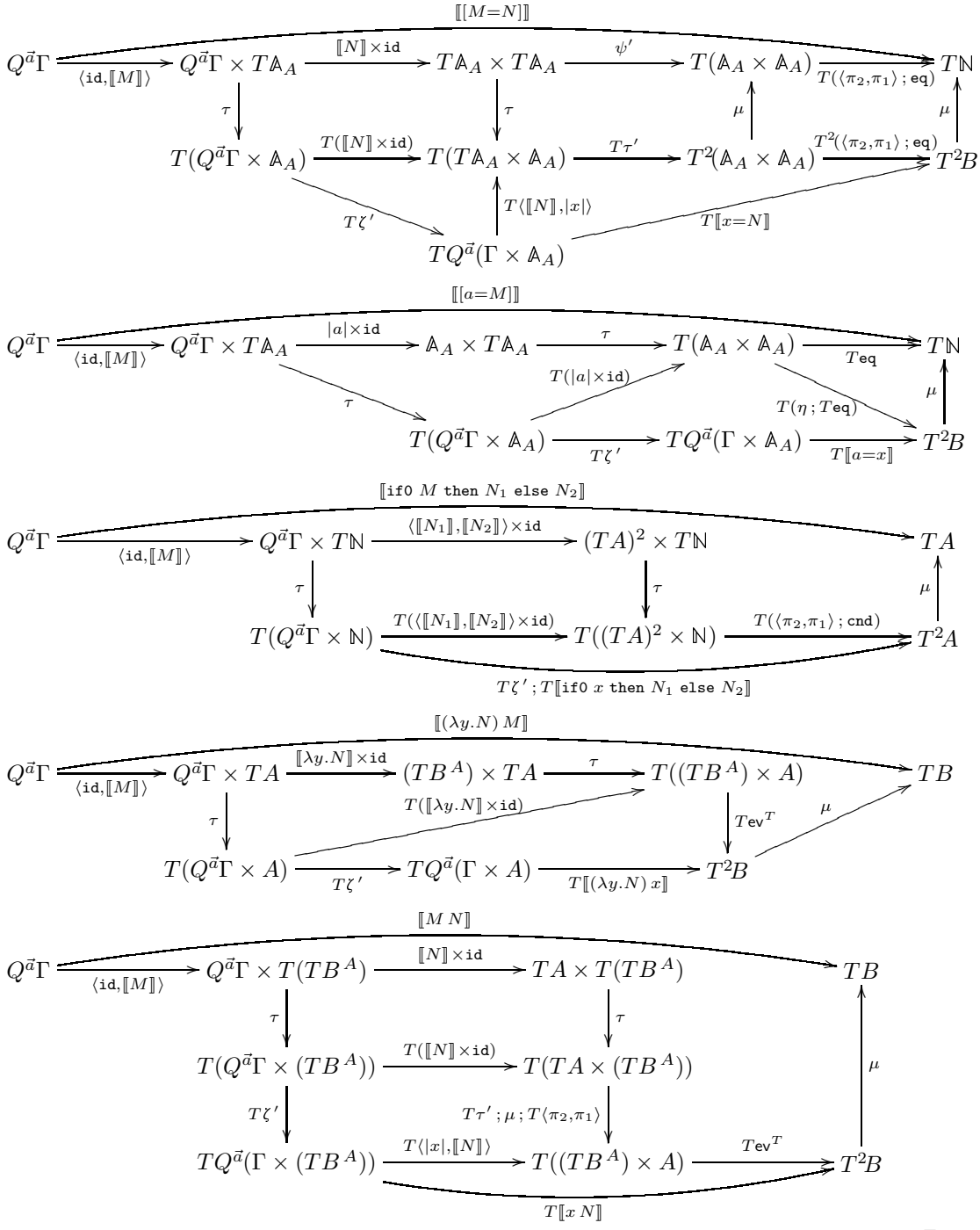
Proof of lemma 4.11: The equalities are shown by the following diagrams.





■

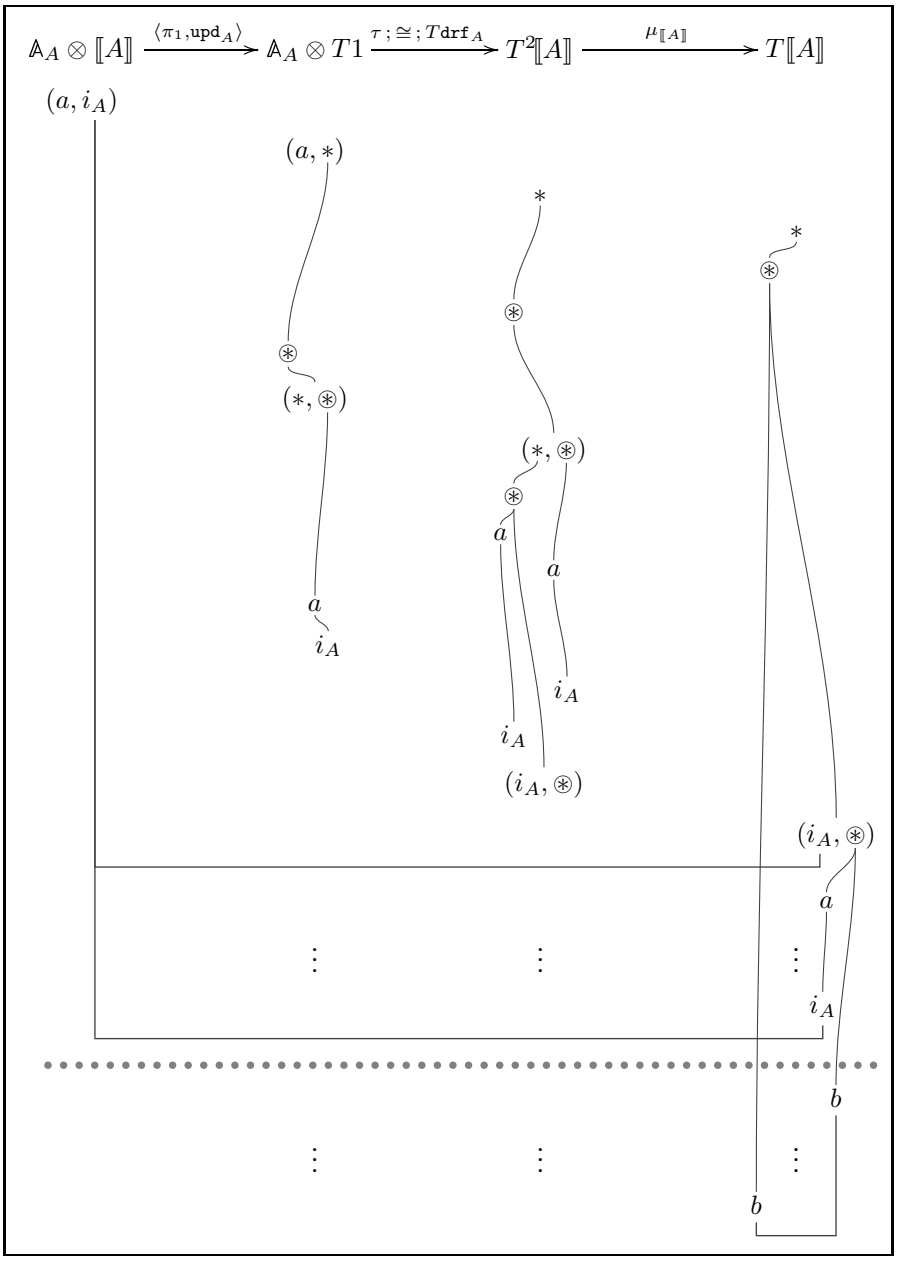
Proof of lemma 4.12: We show the following paradigmatic cases.



■

Proof of proposition 4.31: Commutativity of the (NR) diagrams is shown by direct computation. For example, for the first diagram, the viewfunction of $\langle \pi_1, \mathtt{upd}_A\rangle\,;\tau\,;\cong\,;T\mathtt{drf}_A\,;\mu_{[\![A]\!]}$

is computed in the following figure, from which we see that it equals $\langle \pi_2, \mathtt{upd}_A \rangle ; \tau ; \cong$.



We now show the (SNR) equation holds. We observe that (note we omit superscripts and subscripts for economy)

$$\Lambda^{-1}(\mathtt{upd}) = \Lambda^{-1}(\mathtt{upd}) ; \mathtt{pu} ; \mathtt{up} : \mathbb{A}_A \otimes [\![A]\!] \otimes \xi \longrightarrow (1 \otimes \xi)_\perp$$

and hence $\mathtt{upd} = \Lambda(\Lambda^{-1}(\mathtt{upd}) ; \mathtt{pu} ; \mathtt{up}) = f ; \Lambda(\mathtt{ev} ; \mathtt{up})$, with $f \triangleq \Lambda(\Lambda^{-1}(\mathtt{upd}) ; \mathtt{pu})$. Thus,

$$\mathtt{nu} \times \mathtt{upd} ; \psi = \mathtt{new} \times f ; \alpha \times \Lambda(\mathtt{ev} ; \mathtt{up}) ; \psi \overset{(*)}{=} \mathtt{new} \times f ; \alpha \times \Lambda(\mathtt{ev} ; \mathtt{up}) ; \psi' = \mathtt{nu} \times \mathtt{upd} ; \psi' ,$$

where $(*)$ is proven as follows.

$$\begin{aligned}
\alpha \times \Lambda(\mathtt{ev};\mathtt{up});\psi &= \alpha \times \Lambda(\mathtt{ev};\mathtt{up});\tau';T\tau;\mu = \mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\mathtt{st}';\tau_\perp;\alpha;\mu \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\mathtt{st}';\tau_\perp;\alpha) \times \mathtt{id};\mathtt{ev};\mathtt{ev}_\perp;\mathtt{dn}) \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\mathtt{st}';\tau_\perp) \times \mathtt{id};\mathtt{st}';\mathtt{ev}_\perp;\mathtt{dn}) \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\mathtt{st}') \times \mathtt{id};\mathtt{st}';(\tau \times \mathtt{id})_\perp;\mathtt{ev}_\perp;\mathtt{dn}) \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\mathtt{st}') \times \mathtt{id};\mathtt{st}';(\mathtt{id} \times \mathtt{ev};\mathtt{st})_\perp;\mathtt{dn}) \\
&= \Lambda(\mathtt{st}';(\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up}) \times \mathtt{id})_\perp;(\mathtt{id} \times \mathtt{ev};\mathtt{st})_\perp;\mathtt{dn}) \\
&= \Lambda(\mathtt{st}';(\mathtt{id} \times (\mathtt{ev};\mathtt{up});\mathtt{st})_\perp;\mathtt{dn}) = \Lambda(\mathtt{st}';(\mathtt{id} \times \mathtt{ev})_\perp) \\
\alpha \times \Lambda(\mathtt{ev};\mathtt{up});\psi' &= \alpha \times \Lambda(\mathtt{ev};\mathtt{up});\tau;T\tau';\mu = \mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\tau;T\mathtt{st}';T\alpha;\mu \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\tau;T\mathtt{st}';T\alpha) \times \mathtt{id};\mathtt{ev};\mathtt{ev}_\perp;\mathtt{dn}) \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\tau;T\mathtt{st}') \times \mathtt{id};\mathtt{ev};(\alpha \times \mathtt{id})_\perp;\mathtt{ev}_\perp;\mathtt{dn}) \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\tau;T\mathtt{st}') \times \mathtt{id};\mathtt{ev};\mathtt{st}'_\perp;\mathtt{dn}) \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\tau) \times \mathtt{id};\mathtt{ev};(\mathtt{st}' \times \mathtt{id})_\perp;\mathtt{st}'_\perp;\mathtt{dn}) \\
&= \Lambda((\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up});\tau) \times \mathtt{id};\mathtt{ev};\mathtt{st}'_\perp;\mathtt{dn}) \\
&= \Lambda(\mathtt{id} \times \Lambda(\mathtt{ev};\mathtt{up}) \times \mathtt{id};\mathtt{id} \times \mathtt{ev};\mathtt{st};\mathtt{st}'_\perp;\mathtt{dn}) \\
&= \Lambda(\mathtt{id} \times (\mathtt{ev};\mathtt{up});\mathtt{st};\mathtt{st}'_\perp;\mathtt{dn}) = \Lambda(\mathtt{id} \times \mathtt{ev};\mathtt{st}')
\end{aligned}$$

■

Bibliography

[Abr00] Samson Abramsky, *Axioms for definability and full completeness*, Proof, Language, and Interaction: essays in honour of Robin Milner, MIT Press, 2000, pp. 55–75.

[Abr07] ———, *Domain theory*, Lecture Notes, Oxford University Computing Laboratory, 2007.

[AGM+04] Samson Abramsky, Dan Ghica, Andrzej Murawski, Luke Ong, and Ian Stark, *Nominal games and full abstraction for the nu-calculus*, LICS '04: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science (Turku, Finland), IEEE Computer Society Press, 2004, pp. 150–159.

[AHM98] Samson Abramsky, Kohei Honda, and Guy McCusker, *A fully abstract game semantics for general references*, LICS '98: Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science (Indianapolis, USA), IEEE Computer Society Press, 1998, pp. 334–344.

[AJ94] Samson Abramsky and Radha Jagadeesan, *Games and full completeness for multiplicative linear logic*, Journal of Symbolic Logic **59** (1994), no. 2, 543–574.

[AJM00] Samson Abramsky, Radha Jagadeesan, and Pasquale Malacaria, *Full abstraction for PCF*, Information and Computation **163** (2000), no. 2, 409–470, results first published in *TACS '94: Proceedings of International Symposium on Theoretical Aspects of Computer Software (Sendai, Japan, 1994)*.

[AM97] Samson Abramsky and Guy McCusker, *Linearity, Sharing and State: a fully abstract game semantics for Idealized Algol*, Algol-like languages (Peter O'Hearn and Robert D. Tennent, eds.), vol. 2, Birkhäuser, 1997, 297–329, results first published in *Proceedings of Linear Logic 96 Tokyo Meeting (Tokyo, Japan, 1996)*.

[AM98] ———, *Call-by-value games*, CSL '97: Proceedings of the 11th International Workshop on Computer Science Logic (Aarhus, Denmark), Lecture Notes in Computer Science, vol. 1414, Springer-Verlag, 1998, pp. 1–17.

[AM99] ———, *Game semantics*, Computational Logic: Proceedings of the 1997 Marktoberdorf Summer School (H. Schwichtenberg and U. Berger, eds.), Springer-Verlag, 1999, pp. 1–56.

[Bar84] Hendrik P. Barendregt, *The lambda calculus. Its syntax and semantics*, Studies in Logic and the Foundations of Mathematics, vol. 103, North-Holland, 1984.

[BDE97] Patrick Baillot, Vincent Danos, and Thomas Ehrhard, *Believe it or not, AJM's games model is a model of classical linear logic*, LICS '97: Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science (Warsaw, Poland), IEEE Computer Society Press, 1997, pp. 68–75.

[Bec69] Jon Beck, *Distributive laws*, Seminar on Triples and Categorical Homology Theory, ETH, Zürich, 1966/67 (B. Eckmann, ed.), Lecture Notes in Mathematics, vol. 80, Springer-Verlag, 1969, pp. 119–140.

[BG92] Stephen Brookes and Shai Geva, *Computational comonads and intensional semantics*, Applications of Categories in Computer Science: Proceedings LMS Symposium (Durham, UK, 1991), London Mathematical Society Lecture Note Series, vol. 177, Cambridge University Press, 1992, pp. 1–44.

[BK08] Nick Benton and Vasileios Koutavas, *A mechanized bisimulation for the nu-calculus*, Tech. Report MSR-TR-2008-129, Microsoft Research, September 2008.

[BvS93] Stephen Brookes and Kathryn van Stone, *Monads and comonads in intensional semantics*, Tech. Report CMU-CS-93-140, Carnegie Mellon University, 1993.

[BW85] Michael Barr and Charles Wells, *Toposes, triples and theories*, Grundlehren der mathematischen Wissenschaften, vol. 278, Springer-Verlag, 1985.

[BW99] ———, *Category theory for computing science*, third ed., Les Publications CRM, 1999.

[BW02] ———, *Toposes, triples and theories*, revised version 1.1 of [BW85], available at `http://www.cwru.edu/artsci/math/wells/pub/ttt.html`, November 2002.

[Che04] James Cheney, *Nominal logic programming*, Phd thesis, Cornell University, August 2004.

[Che05] ———, *Nominal logic and abstract syntax*, SIGACT News **36** (2005), no. 4, 47–69.

[CLW93] Aurelio Carboni, Steve Lack, and Robert Walters, *Introduction to extensive and distributive categories*, Journal of Pure and Applied Algebra **84** (1993), 145–158.

[Fil99] Andrzej Filinski, *Representing layered monads*, POPL '99: Proceedings of the 26th ACM SIGPLAN-SIGACT symposium on Principles of programming languages (San Antonio, USA), ACM Press, 1999, pp. 175–188.

[Fre90] Peter J. Freyd, *Recursive types reduced to inductive types*, LICS '90: Proceedings of the 5th Annual IEEE Symposium on Logic in Computer Science (Philadelphia, USA), IEEE Computer Society Press, 1990, pp. 498–507.

[Gab00] Murdoch J. Gabbay, *A theory of inductive definitions with α-equivalence: Semantics, implementation, programming language*, DPhil thesis, University of Cambridge Computing Laboratory, 2000.

[Gab02] ———, *FM-HOL, a higher-order theory of names*, 35 Years of Automath (F. Kamareddine, ed.), Heriot-Watt University, Edinburgh, Scotland, April 2002.

[GM00] Dan R. Ghica and Guy McCusker, *Reasoning about Idealized Algol using regular languages*, ICALP '00: Proceedings of 27th International Colloquium on Automata, Languages and Programming (Geneva, Switzerland), LNCS, vol. 1853, Springer-Verlag, 2000, pp. 103–116.

[GM04] Dan R. Ghica and Andrzej S. Murawski, *Angelic semantics of fine-grained concurrency*, FoSSaCS '04: Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (Barcelona, Spain), Lecture Notes in Computer Science, vol. 2987, Springer, 2004, pp. 211–225.

[GP02] Murdoch J. Gabbay and Andrew M. Pitts, *A new approach to abstract syntax with variable binding*, Formal Aspects of Computing **13** (2002), 341–363, results first published in *LICS '99: Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science (Trento, Italy, 1999)*.

[Har99] Russell Harmer, *Games and full abstraction for nondeterministic languages*, DPhil thesis, University of London, 1999.

[HM99] Russell Harmer and Guy McCusker, *A fully abstract game semantics for finite nondeterminism*, LICS '99: Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science (Trento, Italy), IEEE Computer Society Press, 1999, pp. 422–430.

[HO00] J. Martin E. Hyland and C.-H. Luke Ong, *On full abstraction for PCF: I, II, III*, Information and Computation **163** (2000), no. 2, 285–408, first written in 1994 and published in the authors' domain.

[Hug97] Dominic J. D. Hughes, *Games and definability for system F*, LICS '97: Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science (Warsaw, Poland), IEEE Computer Society Press, 1997, pp. 76–87.

[Hug00] ———, *Hypergame semantics: Full completeness for system F*, DPhil thesis, University of Oxford, 2000.

[HY99] Kohei Honda and Nobuko Yoshida, *Game-theoretic analysis of call-by-value computation*, Theoretical Computer Science **221** (1999), no. 1–2, 393–456, results first published in *ICALP '97: Proceedings of the 24th International Colloquium on Automata, Languages and Programming (Bologna, Italy, 1997)*.

[Jec73] Thomas J. Jech, *The axiom of choice*, Studies in Logic and the Foundations of Mathematics, vol. 75, North-Holland, Amsterdam, 1973.

[Jon03] Simon P. Jones, *Haskell 98 language and libraries: The revised report*, Cambridge University Press, May 2003.

[JR02] Alan Jeffrey and Julian Rathke, *A fully abstract may testing semantics for concurrent objects*, LICS '02: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (Copenhagen, Denmark), IEEE Computer Society Press, 2002, pp. 101–112.

[Kie99] Richard Kieburtz, *Codata and comonads in Haskell*, unpublished manuscript, 1999.

[Kri90] Jean-Louis Krivine, *Lambda-calcul, types et modèles*, Masson, 1990.

[Lai97] James Laird, *Full abstraction for functional languages with control*, LICS '97: Proceeding of the 12th Annual IEEE Symposium on Logic in Computer Science (Warsaw, Poland), IEEE Computer Society Press, 1997, pp. 58–67.

[Lai98] ———, *A semantic analysis of control*, DPhil thesis, University of Edinburgh, 1998.

[Lai01] ———, *A fully abstract game semantics of local exceptions*, LICS '01: Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science (Boston, USA), IEEE Computer Society Press, 2001, pp. 105–114.

[Lai02] ———, *A categorical semantics of higher order store*, CTCS '02: Category Theory and Computer Science (Ottawa, Canada), Electronic Notes in Theoretical Computer Science, vol. 69, Elsevier, 2002, pp. 209–226.

[Lai04] ———, *A game semantics of local names and good variables*, FoSSaCS '04: Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures (Barcelona, Spain), Lecture Notes in Computer Science, vol. 2987, Springer, 2004, pp. 289–303.

[Lai05] ———, *A game semantics of the asynchronous pi-calculus*, CONCUR '05: 16th International Conference on Concurrency Theory (San Francisco, USA), Lecture Notes in Computer Science, vol. 3653, Springer, 2005, pp. 51–65.

[Lai06] ———, *Game semantics for higher-order concurrency*, FSTTCS '06: Proceedings of the 26th International Conference on Foundations of Software Technology and Theoretical Computer Science (Kolkata, India), Lecture Notes in Computer Science, vol. 4337, Springer, 2006, pp. 417–428.

[Lai07] ———, *A fully abstract trace semantics for general references*, ICALP '07: Proceedings of the 34th International Colloquium on Automata, Languages and Programming (Wroclaw, Poland), Lecture Notes in Computer Science, vol. 4596, Springer-Verlag, 2007, pp. 667–679.

[Lai08] ———, *A game semantics of names and pointers*, Annals of Pure and Applied Logic **151** (2008), 151–169, GaLoP '05: First Games for Logic and Programming Languages Workshop (post-proceedings).

[Loa01] Ralph Loader, *Finitary PCF is not decidable*, Theoretical Computer Science **266** (2001), no. 1–2, 341–364.

[LSLM00] Jeffrey R. Lewis, Mark Shields, John Launchbury, and Erik Meijer, *Implicit parameters: Dynamic scoping with static types*, POPL '00: Symposium on Principles of Programming Languages (Boston, USA), 2000, pp. 108–118.

[Mac98] Saunders Mac Lane, *Categories for the working mathematician*, second ed., Graduate texts in mathematics, vol. 5, Springer Verlag, 1998.

[McC96] Guy McCusker, *Games and full abstraction for FPC*, LICS '96: Proceeding of the 11th Annual IEEE Symposium on Logic in Computer Science (New Brunswick, USA), IEEE Computer Society Press, 1996, pp. 174–183.

[McC98] ———, *Games and full abstraction for a functional metalanguage with recursive types*, Distinguished Dissertations, Springer-Verlag, London, 1998.

[McC00] ———, *Games and full abstraction for FPC*, Information and Computation **160** (2000), no. 1-2, 1–61.

[Mog88] Eugenio Moggi, *Computational lambda calculus and monads*, Tech. Report ECS-LFCS-88-86, University of Edinburgh, 1988.

[Mog89] ———, *Computational lambda-calculus and monads*, LICS '89: Proceedings of 4th Annual IEEE Symposium on Logic in Computer Science (Pacific Grove, USA), IEEE Computer Society Press, 1989, pp. 14–23.

[Mog91] ———, *Notions of computation and monads*, Information and Computation **93** (1991), no. 1, 55–92.

[MT09] Andrzej S. Murawski and Nikos Tzevelekos, *Full abstraction for Reduced ML*, FoSSaCS '09: Proceedings of the 12th International Conference on Foundations of Software Science and Computation Structures (York, United Kingdom), Lecture Notes in Computer Science, vol. 5504, Springer, 2009, pp. 32–47.

[MTM97] Robin Milner, Mads Tofte, and David Macqueen, *The definition of Standard ML*, MIT Press, 1997.

[Mur03] Andrzej S. Murawski, *On program equivalence in languages with ground-type references*, LICS '03: Proceedings of the 18th IEEE Symposium on Logic in Computer Science (Ottawa, Canada), IEEE Computer Society Press, 2003, pp. 108–117.

[Nee93] Roger M. Needham, *Names*, Distributed systems, ACM Press/Addison-Wesley Publishing Co., 1993, 2nd edition (1st edition 1989), pp. 315–327.

[Nic96] Hanno Nickau, *Hereditarily sequential functionals: A game-theoretic approach to sequentiality*, Ph.D. thesis, Universität Gesamthochschule Siegen, 1996, results results first published in *LFCS '94: Proceedings of the 3rd Symposium on Logical Foundations of Computer Science (St. Petersburg, Russia, 1994)*.

[Ong02] C.-H. Luke Ong, *Observational equivalence of third-order Idealized Algol is decidable*, LICS '02: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science (Copenhagen, Denmark), IEEE Computer Society Press, 2002, pp. 245–256.

[OT97] Peter W. O'Hearn and Robert D. Tennent (eds.), *ALGOL-like languages*, Birkhäuser, 1997.

[PG00] Andrew M. Pitts and Murdoch J. Gabbay, *A metalanguage for programming with bound names modulo renaming*, MPC2000: Proceedings of 5th International Conference on Mathematics of Program Construction (R. Backhouse and J. N. Oliveira, eds.), Lecture Notes in Computer Science, vol. 1837, Springer-Verlag, 2000, pp. 230–255.

[Pit03] Andrew M. Pitts, *Nominal logic, a first order theory of names and binding*, Information and Computation **186** (2003), 165–193.

[Pit06] ———, *Alpha-structural recursion and induction*, Journal of the ACM **53** (2006), 459–506.

[Plo77] Gordon D. Plotkin, *LCF considered as a programming language*, Theoretical Computer Science **5** (1977), 223–255.

[Pow00] John Power, *Models for the computational lambda-calculus*, MFCSIT2000: Proceedings of First Irish Conference on the Mathematical Foundations of Computer Science and Information Technology (Cork, Ireland), Electronic Notes in Theoretical Computer Science, vol. 40, Elsevier, 2000, pp. 288–301.

[PP02] Gordon D. Plotkin and John Power, *Notions of computation determine monads*, FoSSaCS '02: Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures (Grenoble, France), Springer-Verlag, 2002, pp. 342–356.

[PR97] John Power and Edmund Robinson, *Premonoidal categories and notions of computation*, Mathematical Structures in Computer Science **7** (1997), no. 5, 453–468.

[PS93] Andrew M. Pitts and Ian D. B. Stark, *Observable properties of higher order functions that dynamically create local names, or: What's new?*, MFCS '93: Proceedings of 18th International Symposium on Mathematical Foundations of Computer Science (Gdańsk, Poland), Lecture Notes in Computer Science, vol. 711, Springer-Verlag, 1993, pp. 122–141.

[PW02] John Power and Hiroshi Watanabe, *Combining a monad and a comonad*, Theoretical Computer Science **280** (2002), no. 1-2, 137–162.

[Rey81] John C. Reynolds, *The essence of Algol*, Proceedings of the International Symposium on Algorithmic Languages (Amsterdam, Netherlands), North-Holland, 1981, Reprinted in [OT97, vol. 1, pages 67–88], pp. 345–372.

[Sch06] Ulrich Schöpp, *Names and binding in type theory*, DPhil thesis, University of Edinburgh, 2006.

[Sco70] Dana S. Scott, *Outline of a mathematical theory of computation*, Technical Monograph PRG–2, Oxford University Computing Laboratory, Oxford, England, November 1970.

[Sco93] ———, *A type-theoretical alternative to ISWIM, CUCH, OWHY*, Theoretical Computer Science **121** (1993), no. 1-2, 411–440, First written in 1969 and circulated privately.

[Shi05a] Mark Shinwell, *The fresh approach: functional programming with names and binders*, DPhil thesis, University of Cambridge Computing Laboratory, February 2005, Available also in a more compact form as [Shi05b].

[Shi05b] ———, *The fresh approach: functional programming with names and binders*, Tech. Report UCAM-CL-TR-618, University of Cambridge, UK, February 2005.

[SO07] Sam B. Sanjabi and C.-H. Luke Ong, *Fully abstract semantics of additive aspects by translation*, AOSD '07: Proceedings of the 6th international conference on Aspect-oriented software development (Vancouver, Canada), ACM, 2007, pp. 135–148.

[SP82] Michael B. Smyth and Gordon D. Plotkin, *The category-theoretic solution of recursive domain equations*, SIAM Journal on Computing **11** (1982), no. 4, 761–783.

[Sta94] Ian D. B. Stark, *Names and higher-order functions*, Ph.D. thesis, University of Cambridge, December 1994, Also available as Technical Report 363, University of Cambridge Computer Laboratory.

[Sta96] ———, *Categorical models for local names*, Lisp and Symbolic Computation **9** (1996), no. 1, 77–107.

[Sta97] ———, *Names, equations, relations: Practical ways to reason about* new, TLCA '97: Proceedings of the Third International Conference on Typed Lambda Calculi and Applications (Nancy, France), Lecture Notes in Computer Science, no. 1210, Springer-Verlag, 1997, pp. 336–353.

[Str66] Christopher Strachey, *Towards a formal semantics*, IFIP '64: TC2 Working Conference on Formal Language Description Languages for Computer Programming (Amsterdam), North-Holland, 1966, pp. 198–220.

[SW01] Davide Sangiorgi and David Walker, *The π-calculus: a theory of mobile processes*, Cambridge University Press, 2001.

[Tze07] Nikos Tzevelekos, *Full abstraction for nominal general references*, LICS '07: Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science (Wroclaw, Poland), IEEE Computer Society Press, 2007, pp. 399–410.

[Tze08] ———, *Full abstraction for nominal exceptions and general references*, GaLoP '08: Games for Logic and Programming Languages (Budapest, Hungary), 2008, full version available as *Tech. Report PRG-RR-07-08, Oxford University Computing Laboratory (October 2007, updated December 2009)*.

[UV05] Tarmo Uustalu and Varmo Vene, *Signals and comonads*, Journal of Universal Computer Science **11** (2005), no. 7, 1310–1326.

[Wad92] Philip Wadler, *The essence of functional programming*, POPL '92: Conference Record of the 19th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (Albuquerque, USA), 1992, pp. 1–14.

[Wad95] ———, *Monads for functional programming*, First International Spring School on Advanced Functional Programming Techniques, Tutorial Text (Båstad, Sweden), Lecture Notes in Computer Science, vol. 925, Springer, 1995, pp. 24–52.

[ZN03] Yu Zhang and David Nowak, *Logical relations for dynamic name creation*, CSL '03: Proceedings of the 12th Annual Conference of the European Association for Computer Science Logic (Vienna, Austria), Lecture Notes in Computer Science, vol. 2803, Springer-Verlag, 2003, pp. 575–588.

Index

Symbols

A

B