Full abstraction for nominal general references

Tzevelekos, N

Logical Methods in Computer Science
Vol. 5 (3:8) 2009, pp. 1–69
www.lmcs-online.org

Submitted Jan. 2, 2008
Published Sep. 11, 2009

FULL ABSTRACTION FOR NOMINAL GENERAL REFERENCES

NIKOS TZEVELEKOS

Oxford University Computing Laboratory
e-mail address: nikt@comlab.ox.ac.uk

Abstract. Game semantics has been used with considerable success in formulating fully abstract semantics for languages with higher-order procedures and a wide range of computational effects. Recently, nominal games have been proposed for modelling functional languages with names. These are ordinary, stateful games cast in the theory of nominal sets developed by Pitts and Gabbay. Here we take nominal games one step further, by developing a fully abstract semantics for a language with nominal general references.

Contents

List of Figures

1998 ACM Subject Classification: F.3.2.
Key words and phrases: game semantics, denotational semantics, monads and comonads, ν-calculus, ML.
Research financially supported by the Engineering and Physical Sciences Research Council, the Eugenides Foundation, the A. G. Leventis Foundation and Brasenose College.

DOI:10.2168/LMCS-5 (3:8) 2009

1. Introduction

Functional languages constitute a programming paradigm built around the intuitive notion of a *computational function*, that is, an effectively specified entity assigning values from a codomain to elements of a domain in a *pure manner*: a pure function is not allowed to carry any notion of state or side-effect. This simple notion reveals great computational power if the domains considered are *higher-order*, i.e sets of functions: with the addition of recursive constructs, higher-order functional computation becomes *Turing complete* (PCF [42, 37]). In practice, though, functional programming languages usually contain *impure* features that make programming simpler (*computational effects*), like references, exceptions, etc. While not adding necessarily to its computational power, these effects affect the *expressivity* of a language: two functions which seem to accomplish the same task may have different inner-workings which can be detected by use of effects (e.g. exceptions can distinguish constant functions that do or do not evaluate their inputs). The study of denotational models for effects allows us to better understand their expressive power and to categorise languages with respect to their expressivity.

A computational effect present in most functional programming languages is that of general references. 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 us not only the encoding of recursion (see example 3.4) but also the simulation of a wide range of computational effects and programming paradigms (e.g. object-oriented programming [3, section 2.3] or aspect-oriented programming [40]). The denotational modelling of general 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. In this paper we provide a fully abstract semantics for a language with general references called the *$\nu\rho$-calculus*.

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 [27]. In particular, it extends the basic nominal language of Pitts and Stark [36], the ν-calculus, by using ***names*** for general references. That is, names in $\nu\rho$ are atomic entities which can be (cf. [36]):

> *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.[1]

Fully abstract models for general references were given via game semantics in [3] and via abstract categorical semantics (and games) in [20]. Neither approach used names. The model of [3] 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

[1] In fact, the $\nu\rho$-calculus and its fully abstract model were first presented in [46], of which the present paper is an extended and updated version.

and prohibit the use of equality tests for references.[2] On the other hand, the approach in [20] 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 it is somehow distanced from the common notion of reference in functional programming.

The full-abstraction problem has also been tackled via trace semantics in [23]. The language examined is a version of that in [3] without bad variables. The latter are not needed since the modelling of references is achieved by names pointing to a store (which 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 [17]. 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 [23] and the work herein that should be further examined.

The approach. We model nominal computation in ***nominal games***. These were introduced independently in [2, 21] for producing fully abstract models of the ν-calculus and its extension with pointers respectively. Here we follow the formulation of [2] with rectifications pertaining to the issue of *unordered state* (see remark 4.20).[3] Thus, our nominal games constitute a stateful (cf. Ong [34]) version of Honda-Yoshida call-by-value games [15] built inside the universe of nominal sets of Gabbay and Pitts [12, 35].

A particularly elegant approach to the modelling of names is by use of ***nominal sets*** [12, 35]. These 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 realm (the category) of nominal sets, notions like *atom-permutation*, *atom-freshness* and *atom-abstraction* are built inside the underlying structure. We therefore use nominal sets, with atoms playing the role of names, as a **general foundation for reasoning about names**.

The essential feature of nominal games is the appearance of names *explicitly* in plays as constants (i.e. as atoms), which allows us to directly model names and express name-related notions (name-equality, name-privacy, scope-extrusion, etc.) in the games setting. Thus nominal games can capture the essential features of nominal computation and, in particular, they model the ν-calculus. From that model we can move to a model of $\nu\rho$ by an appropriate *effect-encapsulation* procedure, that is, by use of a *store-monad*. A fully abstract model is then achieved by enforcing appropriate store-discipline conditions on the games.

[2]By "bad variables" we mean read/write constructs of reference type which are not references. They are necessary for obtaining definability and full-abstraction in [3] since read/write-product semantical objects may not necessarily denote references.

[3]The nominal games of [2] use moves attached with finite sets of names. It turns out, however, that this yields discrepancies, as unordered name-creation is incompatible with the deterministic behaviour of strategies and, in fact, nominal games in [2] do not form a category. Here (and also in [46]), we recast nominal games using moves attached with name-lists instead of name-sets. This allows us to restrict our attention to *strong nominal sets* (v. definition 2.6), a restriction necessary for overcoming the complications with determinacy.

The paper is structured as follows. In section 2 we briefly present nominal sets and some of their basic properties. We finally introduce *strong nominal sets*, that is, nominal sets with "ordered involvement" of names, and prove the *strong support lemma*. In section 3 we introduce the $\nu\rho$-calculus and its operational semantics. We then introduce the notion of a *$\nu\rho$-model*, which provides abstract categorical conditions for modelling $\nu\rho$ in a setting involving *local-state comonads* and a *store-monad*. We finally show definability and, by use of a quotienting procedure, full-abstraction in a special class of $\nu\rho$-models. In section 4 we introduce nominal games and show a series of results with the aim of constructing a category $\mathcal{V}_{\mathtt{t}}$ of *total, innocent* nominal strategies. In the end of the section we attempt a comparison with the nominal games presented by Laird in [21, 24]. In section 5 we proceed to construct a specific fully abstract $\nu\rho$-model in the category $\mathcal{V}_{\mathtt{t}}$. The basic ingredients for such a construction have already been obtained in the previous section, except for the construction of the store-monad, which involves solving a recursive domain equation in $\mathcal{V}_{\mathtt{t}}$. Once this has been achieved and the $\nu\rho$-model has been obtained, we further restrict legal strategies to *tidy* ones, i.e. to those that obey a specific store-related discipline; for these strategies we show definability and full-abstraction. We conclude in section 6 with some further directions.

The contributions of this paper are: a) the identification of strong nominal sets as the adequate setting for nominal language semantics; b) the abstract categorical presentation in a monadic-comonadic setting of models of a language with nominal general references; c) the rectification of nominal games of [2] and their use in constructing a specific such model; d) the introduction of a game-discipline (*tidiness*) to capture computation with names-as-references, leading to a definable and hence fully abstract game model.

2. Theory of nominal sets

We give a short overview of nominal sets, which form the basis of all constructions presented in this paper; our presentation generally follows [35]. Nominal sets are an inspiring paradigm of the universality (and reusability) of good mathematics: invented in the 1920's and 1930's by Fraenkel and Mostowski as a model of set theory with atoms (ZFA) for showing its independence from the Axiom of Choice, they were reused in the late 1990's by Gabbay and Pitts [12] as the foundation of a general theory of syntax with binding constructs. The central notion of nominal sets is that of *atoms*, which are to be seen as basic 'particles' present in elements of nominal sets, and of atom-permutations which can act upon those elements. Moreover, there is an infinite supply of atoms, yet each element of a nominal set 'involves' finitely many of them, that is, it has *finite support* with regard to atom-permutations.

We will be expressing the intuitive notion of names by use of atoms, both in the abstract syntax of the language and in its denotational semantics. 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. [24]), but they would constitute an overspecification: 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.

2.1. **Nominal sets.** 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 other atoms). We write $\mathbb{A}$ for the union of all the $\mathbb{A}_i$'s. We take

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

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 $\mathbb{A}_i$. In particular, each $\pi \in \mathrm{PERM}(\mathbb{A})$ is an ω-indexed list of permutations, $\pi \in \prod_{i\in\omega} \mathrm{PERM}(\mathbb{A}_i)$, such that $(\pi)_i \neq \mathtt{id}_{\mathbb{A}_i}$ holds for finitely many indices i. In fact, we will write (non-uniquely) each permutation π as a finite composition

$$\pi = \pi_1 \circ \cdots \circ \pi_n$$

such that each π_i belongs to some $\mathrm{PERM}(\mathbb{A}_{j_i})$ — note that j_i's need not be distinct.

Definition 2.1. 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 $x \in X$,

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

Moreover, for any $x \in X$ there exists a finite set S such that, for all permutations π,

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

For example, $\mathbb{A}$ with the action of permutations being simply permutation-application is a nominal set. Moreover, any set can be trivially rendered into a nominal set of elements with empty support.

Finite support is closed under intersection and hence there is a least finite support for each element x of a nominal set; this we call ***the support of*** x and denote by $\mathtt{S}(x)$.

Proposition and Definition 2.2 ([12])**.** *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. Hence, we can define

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

which can be expressed also as:

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

For example, for each $a \in \mathbb{A}$, $\mathtt{S}(a) = \{a\}$. 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. It follows from the definition that

$$a \# x \iff \text{for cofinitely many } b.\ (a\ b) \circ x = x\,. \tag{2.2}$$

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. This extends to infinite disjoint unions.
- 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 on subsets of X elementwise. In particular, $X' \subseteq X$ is a ***nominal subset*** of X if it has empty support, i.e. if for all $x \in X'$ and permutation π, $\pi \circ x \in X'$.

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

- Using products and infinite unions we obtain the nominal set
$$\mathbb{A}^{\#} \triangleq \bigcup_n \{\, a_1 \dots a_n \mid \forall i,j \in 1..n.\ \ a_i \in \mathbb{A} \wedge (j \neq i \implies a_j \neq a_i) \,\}\,, \tag{2.3}$$
that is, the set of ***finite lists of distinct atoms***. Such lists we denote by $\bar{a}$, $\bar{b}$, $\bar{c}$ and variants.
- 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 $(x,y) \in X \times Y$,
$$x \mathcal{R} y \iff (\pi \circ x) \mathcal{R} (\pi \circ y)\,.$$
For example, it is easy to show that $\# \subseteq \mathbb{A} \times X$ is a nominal relation. Extending this reasoning to functions we obtain the notion of ***nominal functions***.

Definition 2.3 (**The category Nom**)**.** We let **Nom** be the category of nominal sets and nominal functions, where a function $f : X \to Y$ between nominal sets is nominal if $f(\pi \circ x) = \pi \circ f(x)$ for any $\pi \in \mathrm{PERM}(\mathbb{A})$ and $x \in X$. ▲

For example, the support function, $\mathtt{S}(\_) : X \to \mathcal{P}_{\mathsf{fin}}(\mathbb{A})$, 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; we will concentrate on the following.

Definition 2.4 (**Nominal 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
$$[x]_S \triangleq \{\, y \in X \mid \exists \pi.\ (\forall a \in S \cap \mathtt{S}(x).\ \pi(a) = a) \wedge y = \pi \circ x \,\}\,.$$
▲

The abstraction restricts the support of x to $S \cap \mathtt{S}(x)$ by appropriate orbiting of x (note that $[x]_S \in \mathcal{P}_{\mathsf{fs}}(X)$). In particular, we can show the following.

Lemma 2.5 ([48])**.** *For any $x \in X$, $S \subseteq_{\mathsf{fin}} \mathbb{A}$ and $\pi \in \mathrm{PERM}(\mathbb{A})$,*
$$\pi \circ [x]_S = [\pi \circ x]_{\pi \circ S} \ \wedge\ \mathtt{S}([x]_S) = \mathtt{S}(x) \cap S\,.$$
□

Two particular subcases of nominal abstraction are of interest. Firstly, in case $S \subseteq \mathtt{S}(x)$ the abstraction becomes
$$[x]_S = \{\, y \in X \mid \exists \pi.\ (\forall a \in S.\ \pi(a) = a) \wedge y = \pi \circ x \,\}\,. \tag{$*$}$$
This is the mechanism used in [46]. Note that if $S \not\subseteq \mathtt{S}(x) \wedge \mathtt{S}(x) \not\subseteq S$ then $(*)$ does not yield $\mathtt{S}([x]_S) = S \cap \mathtt{S}(x)$. The other case is the simplest possible, that is, of S being empty; it turns out that this last constructor is all we need from nominal abstractions in this paper. We define:
$$[x] \triangleq \{\, y \in X \mid \exists \pi.\ y = \pi \circ x \,\}\,. \tag{2.4}$$

2.2. **Strong support.** Modelling local state in sets of atoms yields a notion of *unordered state*, which is inadequate for our intended semantics. Nominal game semantics is defined by means of nominal strategies for games that model computation. These strategies, however, are deterministic up to choice of fresh names, a feature which is in direct conflict to unordered state. For example, in unordered state the consecutive creation of two atoms a, b is modelled by adding the set $\{a, b\}$ to the local state; on the other hand, by allowing strategies to play such moves we lose determinism in strategies.[4]

Ordered state is therefore more appropriate for our semantical purposes and so we restrict our attention to nominal sets with *ordered presence* of atoms in their elements. This notion is described as *strong support*.[5]

Definition 2.6. For any nominal set X, any $x \in X$ and any $S \subseteq \mathbb{A}$, S ***strongly supports*** x if, for any permutation π,

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

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

Compare the last assertion above with that of definition 2.1, which employs only the left-to-right implication. In fact, strong support coincides with weak support when the former exists.

Proposition 2.7. *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, $\lightning$. □

Thus, for example, the set $\{a, b\} \subseteq \mathbb{A}_i$ of the previous paragraph does not have strong support, since the permutation $(a\ b)$ does not fix the atoms in its support (the set $\{a, b\}$) but still $(a\ b) \circ \{a, b\} = \{a, b\}$. On the other hand, $\{a, b\}$ strongly supports the list ab. In fact, all lists of (distinct) atoms have strong support and therefore $\mathbb{A}^{\#}$ is a strong nominal set (but $\mathcal{P}_{\mathsf{fin}}(\mathbb{A})$ is not).

The main reason for introducing strong nominal sets is the following result, which is a specialised version of the Strong Support Lemma of [48] (with $S = \varnothing$).

Lemma 2.8 (**Strong Support Lemma**)**.** *Let X be a strong nominal set and let $x_1, x_2, y_1, y_2, z_1, z_2 \in X$. Suppose also that $\mathtt{S}(y_i) \cap \mathtt{S}(z_i) \subseteq \mathtt{S}(x_i)$, for $i = 1, 2$, and that there exist π_y, π_z 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 π such that $\pi \circ x_1 = x_2$, $\pi \circ y_1 = y_2$ and $\pi \circ z_1 = z_2$.

Proof: 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 by strong support $\pi'(a) = a$ for all $a \in \mathtt{S}(x_1)$. 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,

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

[4]The problematic behaviour of nominal games in weak support is discussed again in remark 4.20.

[5]An even stricter notion of support is *linear support*, introduced in [31]: a nominal set X is called *linear* if for each $x \in X$ there is a linear order $<_x$ of $\mathtt{S}(x)$ such that $a <_x b \implies \pi(a) <_{\pi \circ x} \pi(b)$.

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

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

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

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

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, considering $\pi \triangleq \pi_y \circ \pi_N^{-1}$ we have:

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

as required. □

A more enlightening formulation of the lemma can be given in terms of abstractions, as in the following table. In the context of nominal games later on, the strong support lemma will guarantee us that composition of abstractions of plays can be reduced to composition of plays.

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 $\mathtt{S}(y_i) \cap \mathtt{S}(z_i) \subseteq \mathtt{S}(x_i)$, for $i = 1, 2$, and moreover that

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

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

3. The language

The language we examine, the $\nu\rho$-calculus, is a call-by-value λ-calculus with nominal general references. It constitutes an extension of the ν-calculus [36] and Reduced ML [44, chapter 5] in which names are used for general references. It is essentially the same calculus of [23], that is, the mkvar-free fragment of the language of [3] extended with reference-equality tests and names.

$$\frac{}{\bar{a} \mid \Gamma \vdash n : \mathbb{N}} \qquad \frac{}{\bar{a} \mid \Gamma, x{:}A \vdash x : A} \qquad \frac{}{\bar{a} \mid \Gamma \vdash \mathtt{skip} : \mathbb{1}}$$

$$\frac{\bar{a} \mid \Gamma \vdash M : A \times B}{\bar{a} \mid \Gamma \vdash \mathtt{fst}\, M : A} \qquad \frac{\bar{a} \mid \Gamma \vdash M : A \times B}{\bar{a} \mid \Gamma \vdash \mathtt{snd}\, M : B} \qquad \frac{\bar{a} \mid \Gamma \vdash M : A \qquad \bar{a} \mid \Gamma \vdash N : B}{\bar{a} \mid \Gamma \vdash \langle M, N \rangle : A \times B}$$

$$\frac{\bar{a} \mid \Gamma \vdash M : \mathbb{N}}{\bar{a} \mid \Gamma \vdash \mathtt{pred}\, M : \mathbb{N}} \qquad \frac{\bar{a} \mid \Gamma \vdash M : \mathbb{N}}{\bar{a} \mid \Gamma \vdash \mathtt{succ}\, M : \mathbb{N}} \qquad \frac{\bar{a} \mid \Gamma \vdash M : \mathbb{N} \qquad \bar{a} \mid \Gamma \vdash N_i : A \quad (i{=}1{,}2)}{\bar{a} \mid \Gamma \vdash \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 : A}$$

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

$$\frac{}{\bar{a} \mid \Gamma \vdash a : [A]}\ {\scriptstyle a \in \mathbb{A}_A \wedge a \in \bar{a}} \qquad \frac{\bar{a}a \mid \Gamma \vdash M : B}{\bar{a} \mid \Gamma \vdash \nu a.M : B} \qquad \frac{\bar{a} \mid \Gamma \vdash M : [A] \qquad \bar{a} \mid \Gamma \vdash N : [A]}{\bar{a} \mid \Gamma \vdash [M = N] : \mathbb{N}}$$

$$\frac{\bar{a} \mid \Gamma \vdash M : [A] \qquad \bar{a} \mid \Gamma \vdash N : A}{\bar{a} \mid \Gamma \vdash M := N : \mathbb{1}} \qquad \frac{\bar{a} \mid \Gamma \vdash M : [A]}{\bar{a} \mid \Gamma \vdash\, !\, M : A}$$

Figure 1: Typing rules.

3.1. **Definitions.** The syntax of the language is built inside **Nom**. In particular, we assume 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 3.1. The $\nu\rho$-calculus is a typed functional language of nominal references. Its types, terms and values are given as follows.

$$\mathrm{TY} \ni A, B ::= \mathbb{1} \mid \mathbb{N} \mid [A] \mid A \to B \mid A \times B$$

$$\begin{array}{lll}
\mathrm{TE} \ni M, N ::= & x \mid \lambda x.M \mid M\, N \mid \langle M, N \rangle \mid \mathtt{fst}\, M \mid \mathtt{snd}\, N & \lambda\text{-calculus} \\
& \mid n \mid \mathtt{pred}\, M \mid \mathtt{succ}\, N & \text{arithmetic} \\
& \mid \mathtt{skip} \mid \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2 & \text{return / if_then_else} \\
& \mid a & \text{reference to type } A\ (a \in \mathbb{A}_A) \\
& \mid [M = N] & \text{name-equality test} \\
& \mid \nu a.M & \nu\text{-abstraction} \\
& \mid M := N & \text{update} \\
& \mid\, !\, M & \text{dereferencing}
\end{array}$$

$$\mathrm{VA} \ni V, W ::= n \mid \mathtt{skip} \mid a \mid x \mid \lambda x.M \mid \langle V, W \rangle$$

The typing system involves terms in environments $\bar{a} \mid \Gamma$, where $\bar{a}$ a list of (distinct) names and Γ a finite set of variable-type pairs. Typing rules are given in figure 1. ▲

The ν-constructor is a ***name-binder***: an occurrence of a name a inside a term M is *bound*

if it is in the scope of some νa. We follow the standard convention of equating terms up to α-equivalence, the latter defined with respect to both variable- and name-binding.

Note that TE and VA are strong nominal sets: each name a of type A is taken from $\mathbb{A}_A$ and all terms contain finitely many atoms — be they free or bound — which form their support. Note also the notion of *ordered state* that is imposed by use of name-lists (instead of name-sets) in type-environments. In fact, we could have used unordered state at the level of syntax (and operational semantics) of $\nu\rho$, and ordered state at the level of denotational semantics. This already happens with contexts: a context Γ is a set of premises, but $[\![\Gamma]\!]$ is an (ordered) product of type-translations. Nevertheless, we think that ordered state does not add much complication while it saves us from some informality.

The operational semantics of the calculus involves computation in some *store environment* where created names have their values stored. Formally, we define store environments S to be lists of the form:

$$S ::= \epsilon \mid a, S \mid a :: V, S\,. \tag{3.1}$$

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:

$$\mathtt{dom}(\epsilon) \triangleq \epsilon\,, \quad \mathtt{dom}(a, S) \triangleq a, \mathtt{dom}(S)\,, \quad \mathtt{dom}(a :: V, S) \triangleq a, \mathtt{dom}(S)\,. \tag{3.2}$$

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 3.2. The operational semantics is given in terms of a small-step reduction, the rules of which are given in figure 2. Evaluation contexts $\mathrm{E}[\_]$ are of the form:

$$[\_ = N]\ ,\ [a = \_]\ ,\ !\_\ ,\ \_ := N\ ,\ a := \_\ ,\ \mathtt{if0}\ \_\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2\ ,$$
$$(\lambda x.N)\,\_\ ,\ \_\,N\ ,\ \mathtt{fst}\,\_\ ,\ \mathtt{snd}\,\_\ ,\ \mathtt{pred}\,\_\ ,\ \mathtt{succ}\,\_\ ,\ \langle \_, N\rangle\ ,\ \langle V, \_\rangle$$ ▲

We can see that $\nu\rho$ is not strongly normalising with the following example. Recall the

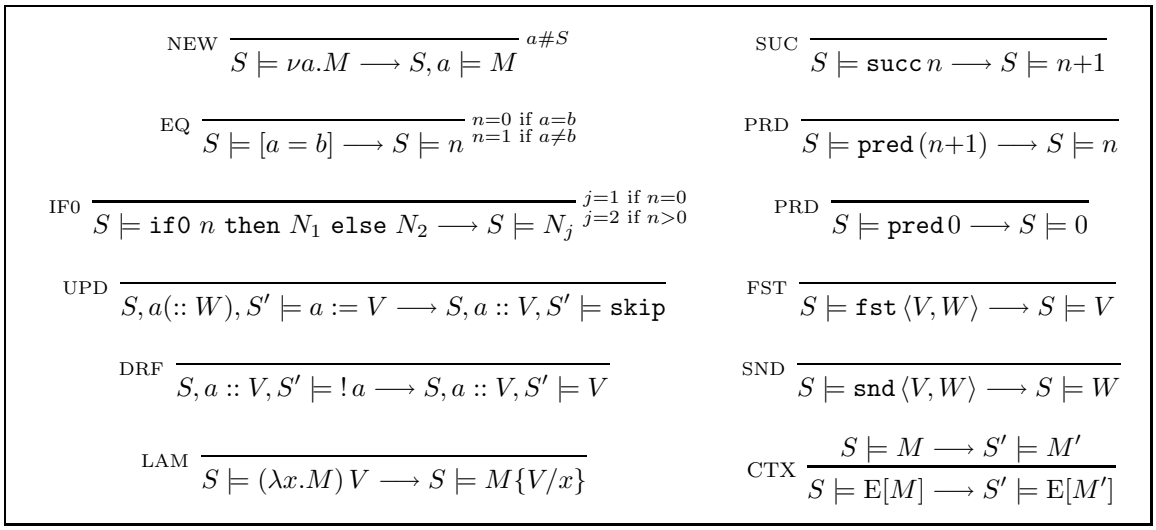


Figure 2: Reduction rules.

standard CBV encoding of sequencing:

$$M\,;N \triangleq (\lambda z.N)M \tag{3.3}$$

with z not free in N.

Example 3.3. 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:

$$\begin{aligned}\models \mathtt{stop}_A &\longrightarrow\!\!\!\!\rightarrow b :: \lambda x.(!\,b)\mathtt{skip} \models (!\,b)\mathtt{skip} \longrightarrow b :: \lambda x.(!\,b)\mathtt{skip} \models (\lambda x.(!\,b)\mathtt{skip})\mathtt{skip}\\ &\longrightarrow b :: \lambda x.(!\,b)\mathtt{skip} \models (!\,b)\mathtt{skip}\,.\end{aligned}$$

□

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 [3].

Example 3.4. 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

$$\begin{aligned}\models (\mathbf{Y}_A(\lambda y.M))V &\longrightarrow\!\!\!\!\rightarrow\ a :: \lambda x.(\lambda y.M)(!\,a)x \models (!\,a)V\\ &\longrightarrow\ a :: \lambda x.(\lambda y.M)(!\,a)x \models (\lambda x.(\lambda y.M)(!\,a)x)V\\ &\longrightarrow\ a :: \lambda x.(\lambda y.M)(!\,a)x \models (\lambda y.M)(!\,a)V\,,\end{aligned}$$

and also $\models (\lambda y.M)(\mathbf{Y}_A(\lambda y.M))V \longrightarrow\!\!\!\!\rightarrow\ a :: \lambda x.(\lambda y.M)(!\,a)x \models (\lambda y.M)(!\,a)V\,.$
For example, setting

$$\begin{aligned}\mathsf{addrec}_{\mathtt{x}} &\triangleq \lambda x.\ \mathtt{if0}\ \mathtt{snd}\,x\ \mathtt{then}\ x\ \mathtt{else}\ \mathtt{x}\langle \mathtt{succ}\,\mathtt{fst}\,x, \mathtt{pred}\,\mathtt{snd}\,x\rangle\,,\\ \mathsf{add} &\triangleq \mathbf{Y}(\lambda h.\mathsf{addrec}_h)\,,\\ S &\triangleq a :: \lambda x.(\lambda h.\mathsf{addrec}_h)(!\,a)x\,,\end{aligned}$$

where $\mathtt{x}$ is a metavariable of relevant type, we have that, for any $n, m \in \mathbb{N}$,

$$\begin{aligned}\models \mathsf{add}\langle n,m\rangle &\longrightarrow\!\!\!\!\rightarrow\ S \models (\lambda h.\mathsf{addrec}_h)(!\,a)\langle n,m\rangle \longrightarrow\!\!\!\!\rightarrow\ S \models \mathsf{addrec}_{S(a)}\langle n,m\rangle\\ &\longrightarrow\!\!\!\!\rightarrow\ S \models \mathtt{if0}\ m\ \mathtt{then}\ \langle n,m\rangle\ \mathtt{else}\ S(a)\langle \mathtt{succ}\,\mathtt{fst}\,\langle n,m\rangle, \mathtt{pred}\,\mathtt{snd}\,\langle n,m\rangle\rangle\\ &\longrightarrow\!\!\!\!\rightarrow\ S \models S(a)\langle n{+}1, m{-}1\rangle \longrightarrow\ S \models (\lambda h.\mathsf{addrec}_h)(!\,a)\langle n{+}1, m{-}1\rangle\\ \cdots &\longrightarrow\!\!\!\!\rightarrow\ S \models (\lambda h.\mathsf{addrec}_h)(!\,a)\langle n{+}m, 0\rangle \longrightarrow\!\!\!\!\rightarrow\ S \models \langle n{+}m, 0\rangle\,.\end{aligned}$$

□

The notions of ***observational approximation*** and ***observational equivalence*** are built around the *observable type* $\mathbb{N}$. Two terms are equivalent if, whenever they are put inside a variable- and name-closing context of resulting type $\mathbb{N}$, called a ***program context***, they reduce to the same natural number. The formal definition follows; note that we usually omit $\bar{a}$ and Γ and write simply $M \lessapprox N$.

Definition 3.5. For typed terms $\bar{a} \mid \Gamma \vdash M : A$ and $\bar{a} \mid \Gamma \vdash N : A$, define

$$\bar{a} \mid \Gamma \vdash M \lessapprox N \iff \forall \mathrm{C}.\,(\exists S'.\ \models \mathrm{C}[M] \longrightarrow\!\!\!\!\rightarrow\ S' \models 0) \implies (\exists S''.\ \models \mathrm{C}[N] \longrightarrow\!\!\!\!\rightarrow\ S'' \models 0)$$

where C is a program context. Moreover, $\approx \triangleq \lessapprox \cap \gtrapprox$. ▲

3.2. **Categorical 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 5 we will apply this knowledge in constructing a concrete such model in nominal games.

Translating each term M into a semantical entity $[\![M]\!]$ and assuming a preorder "$\lesssim$" in the semantics, full-abstraction amounts to the assertion:

$$M \lessapprox N \iff [\![M]\!] \lesssim [\![N]\!] \tag{FA}$$

Note that this formulation is weaker than equational full abstraction, which is given by:

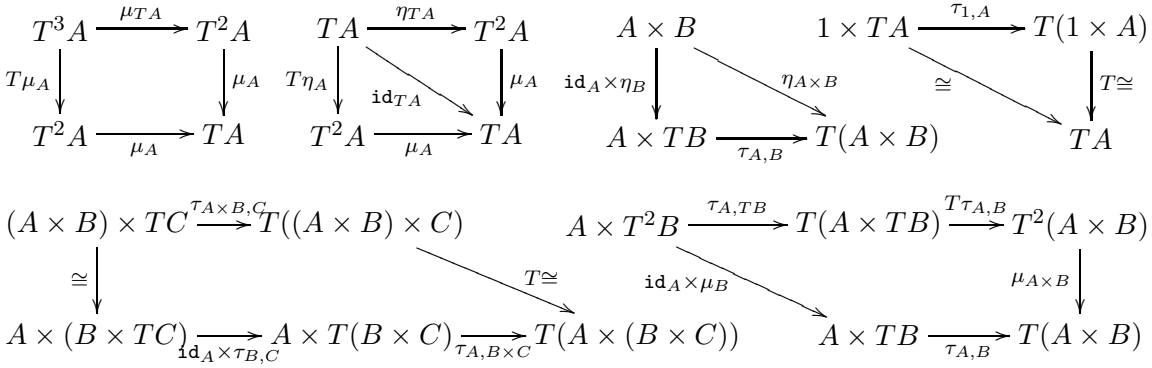
$$M \cong N \iff [\![M]\!] = [\![N]\!]\,. \tag{EFA}$$

Nevertheless, once we achieve (FA) we can construct an *extensional model*, via a quotienting construction, for which EFA 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 (EFA) would be preferred but this cannot be achieved in our nominal games. Therefore, our categorical models will be guided by the (FA) formulation.

3.2.1. *Monads and comonads.* The abstract categorical semantics we put forward is based on the notions of monads and comonads. These are standard categorical notions (v. [25], and [8, *Triples*]) which have been used extensively in denotational semantics of programming languages. We present here some basic definitions and properties.

Monads. Monads were introduced in denotational semantics through the work of Moggi [29, 30] as a generic tool for encapsulating computational effects. Wadler [49] 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 [18].

Definition 3.6. A ***strong monad*** over a category $\mathcal{C}$ with finite products is a quadruple (T, η, μ, τ), where T is an endofunctor in $\mathcal{C}$ and $\eta : Id_{\mathcal{C}} \to T$, $\mu : T^2 \to T$ and $\tau : \_ \times T\_ \to T(\_ \times \_)$ are natural transformations such that the following diagrams commute.



We say that $\mathcal{C}$ has T***-exponentials*** if, for every pair B, C of objects, 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 can define the following transformations.

$$\begin{aligned}
\tau'_{A,B} &\triangleq TA \times B \xrightarrow{\cong} B \times TA \xrightarrow{\tau_{A,B}} T(B \times A) \xrightarrow{\cong} T(A \times B)\,,\\
\psi_{A,B} &\triangleq TA \times TB \xrightarrow{\tau'_{A,TB}} T(A \times TB) \xrightarrow{T\tau_{A,B}} T^2(A \times B) \xrightarrow{\mu_{A\times B}} T(A \times B)\,,\\
\psi'_{A,B} &\triangleq TA \times TB \xrightarrow{\tau_{TA,B}} T(TA \times B) \xrightarrow{T\tau'_{A,B}} T^2(A \times B) \xrightarrow{\mu_{A\times B}} T(A \times B)\,.
\end{aligned} \tag{3.4}$$

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

$$\mathsf{ev}^T_{B,C} : TC^{\,B} \times B \to TC \triangleq \Lambda^{T^{-1}}(\mathtt{id}_{TC^B}) \tag{3.5}$$

so that, for each $f : A \times B \to TC$,

$$f = \Lambda^T(f) \times_B ; \mathsf{ev}^T_{B,C}\,.$$

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

$$Tg^{\,f} : TB'^{\,A} \to TB^{\,A'} \triangleq \Lambda^T(TB'^{\,A} \times A' \xrightarrow{\mathtt{id}\times f} TB'^{\,A} \times A \xrightarrow{\mathsf{ev}^T} TB' \xrightarrow{Tg} TB)\,. \tag{3.6}$$

Naturality of $\Lambda^T_{A,B,C}$ in A implies its naturality in B, C too, by use of the above construct.

Comonads. Comonads are the dual notion of monads. They were first used in denotational semantics by Brookes and Geva [9] for modelling programs *intensionally*, that is, as mechanisms which receive external computation data and decide on an output. Monadic-comonadic approaches were examined by Brookes and van Stone [10].

Definition 3.7. A ***comonad*** on a category $\mathcal{C}$ is a triple (Q, ε, δ), where Q is an endofunctor in $\mathcal{C}$ and $\varepsilon : Q \to Id_{\mathcal{C}}$, $\delta : Q \to Q^2$ are natural transformations such that the following diagrams commute.

$$\begin{array}{ccc} Q^3A & \xleftarrow{\delta_{QA}} & Q^2A \\ {\scriptstyle Q\delta_A}\uparrow & & \uparrow{\scriptstyle \delta_A} \\ Q^2A & \xleftarrow[\delta_A]{} & QA \end{array} \qquad\qquad \begin{array}{ccccc} QA & \xleftarrow{\varepsilon_{QA}} & Q^2A & \xrightarrow{Q\varepsilon_A} & QA \\ & {\scriptstyle \mathtt{id}_{QA}}\nwarrow & \uparrow{\scriptstyle \delta_A} & \nearrow{\scriptstyle \mathtt{id}_{QA}} & \\ & & QA & & \end{array}$$

Now assume $\mathcal{C}$ has binary products. We define a transformation $\bar{\zeta} : Q(\_ \times \_) \to \_ \times Q(\_)$,

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

Q is called a ***product comonad*** if $\bar{\zeta}$ is a natural isomorphism, and is written $(Q, \varepsilon, \delta, \zeta)$ where ζ is the inverse of $\bar{\zeta}$. ▲

It is easy to see that the transformation $\bar{\zeta}$ makes the relevant (dualised) diagrams of definition 3.6 commute, even without stipulating the existence of the inverse ζ. Note that we write $\zeta', \bar{\zeta}'$ for the symmetric counterparts of $\zeta, \bar{\zeta}$.

Product comonads are a stronger version of "strong comonads" of [10]. A product comonad Q can be written as:

$$Q\_ \cong Q1 \times \_$$

hence the name.[6] We say that $Q1$ is the ***basis of the comonad***.

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

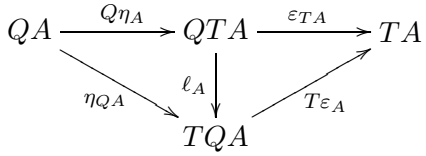
Monadic-comonadic setting. In the presence of both a strong monad (T,η,μ,τ) and a product comonad $(Q,\varepsilon,\delta,\zeta)$ in a cartesian category $\mathcal{C}$, one may want to solely consider arrows from some initial computation data (i.e., some *initial state*) of type A to some computation of type B, that is, arrows of type:

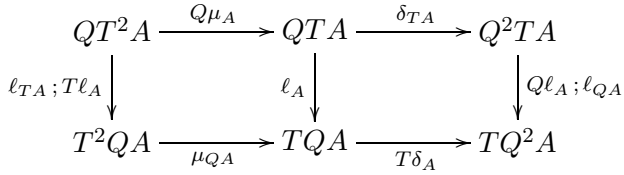
$$QA \to TB$$

This amounts to applying the *biKleisli* construction on $\mathcal{C}$, that is, 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 need a distributive law between Q and T, that is, a natural transformation $\ell : QT \to TQ$ making the following diagrams commute.





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

$$QA \xrightarrow{\delta} Q^2A \xrightarrow{Qf} QTB \xrightarrow{\ell_B} TQB \xrightarrow{Tg} T^2C \xrightarrow{\mu_C} TC$$

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

$$\ell : Q1 \times T\_ \to T(Q1 \times \_)\,,$$

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

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

Definition 3.8. Let $\mathcal{C}$ be a category with finite products and let (T,η,μ,τ), (Q,ε,δ) be a strong monad and comonad, respectively, on $\mathcal{C}$. We say that $\mathcal{C}$ has (Q,T)-***exponentials*** if, for each pair B,C of 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. ▲

Assume now we are in a monadic-comonadic setting $(\mathcal{C},Q,T)$ with T a strong monad with T-exponentials and Q a product comonad. (Q,T)-exponentials then come for free.

Proposition 3.9. *In the setting of the previous definition, if T is a strong monad with exponentials and Q is a product comonad then $\mathcal{C}$ has (Q,T)-exponentials defined by:*

$$(Q,T)C^B \triangleq TC^B,$$

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

ϕ is a bijection with its inverse sending each $g : QA \to TC^{\,B}$ to the arrow:

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

□

In the same setting, we can define a notion of ***intrinsic preorder***. Assuming an object O of *observables* and a collection $\mathcal{O} \subseteq \mathcal{C}(1, TO)$ of observable arrows, we can have the following.

Definition 3.10. Let $\mathcal{C}, Q, T, O, \mathcal{O}$ 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:

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

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

We have the following enrichment properties.

Proposition 3.11. *Let $\mathcal{C}, Q, T, O, \mathcal{O}$ and $\lesssim$ be as above. Then, for any $f, g : QA \to TB$ and any arrow h, if $f \lesssim g$ then:*

- *if $h : QB \to TB'$ then $\delta ; Qf ; \ell ; Th ; \mu \lesssim \delta ; Qg ; \ell ; Th ; \mu$,*
- *if $h : QA' \to TA$ then $\delta ; Qh ; \ell ; Tf ; \mu \lesssim \delta ; Qh ; \ell ; Tg ; \mu$,*
- *if $h : QA \to TC$ 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 $\Lambda^T_{QA_1, A_2, B}(\zeta' ; f) ; \eta \lesssim \Lambda^T_{QA_1, A_2, B}(\zeta' ; g) ; \eta$.* □

3.2.2. *Soundness.* We proceed to present categorical models of the $\nu\rho$-calculus. The approach we take is a monadic and comonadic one, over a computational monad T and a family of local-state comonads $Q = (Q^{\bar{a}})_{\bar{a} \in \mathbb{A}^\#}$, so that the morphism related to each $\bar{a} \mid \Gamma \vdash M : A$ be of the form $[\![M]\!] : Q^{\bar{a}}[\![\Gamma]\!] \to T[\![A]\!]$. Computation in $\nu\rho$ is store-update and fresh-name creation, so T is a store monad, while initial state is given by product comonads.

Definition 3.12. 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 an appropriate natural numbers object $\mathbb{N}$ equipped with successor and predecessor arrows and $\tilde{n} : 1 \to \mathbb{N}$, each $n \in \mathbb{N}$. Moreover, for each object A, there is an arrow $\mathtt{cnd}_A : \mathbb{N} \times TA \times TA \to TA$ for zero-equality tests.

IV. Q is a family of product comonads $(Q^{\bar{a}}, \varepsilon, \delta, \zeta)_{\bar{a} \in \mathbb{A}^\#}$ on $\mathcal{M}$ such that:

(a) the basis of Q^ϵ is 1, and $Q^{\bar{a}} = Q^{\bar{a}'}$ whenever $[\bar{a}] = [\bar{a}']$ (i.e., whenever $\pi \circ \bar{a} = \bar{a}'$),

(b) if $\mathtt{S}(\bar{a}') \subseteq \mathtt{S}(\bar{a})$ then there exists a comonad morphism $\frac{\bar{a}}{\bar{a}'} : Q^{\bar{a}} \to Q^{\bar{a}'}$ such that $\frac{\bar{a}}{\epsilon} = \varepsilon$, $\frac{\bar{a}}{\bar{a}} = \mathtt{id}$ and, whenever $\mathtt{S}(\bar{a}') \subseteq \mathtt{S}(\bar{a}'') \subseteq \mathtt{S}(\bar{a})$,

$$\frac{\bar{a}}{\bar{a}''} ; \frac{\bar{a}''}{\bar{a}'} = \frac{\bar{a}}{\bar{a}'}$$

(c) for each $\bar{a}a \in \mathbb{A}^{\#}$ there exists a natural transformation $\mathtt{nu}^{\bar{a}a} : Q^{\bar{a}} \to TQ^{\bar{a}a}$ such that, for each $A, B \in Ob(\mathcal{M})$ and $\bar{a}a, \bar{a}'a$ with $\mathtt{S}(\bar{a}a) \subseteq \mathtt{S}(\bar{a}'a)$, the following diagrams commute.

$$\begin{array}{ccccc} Q^{\bar{a}'}A & \xrightarrow{\frac{\bar{a}'}{\bar{a}}} & Q^{\bar{a}}A & \xrightarrow{\langle \mathtt{id}, \mathtt{nu}^{\bar{a}a}\rangle} & Q^{\bar{a}}A \times TQ^{\bar{a}a}A \\ \downarrow{\scriptstyle \mathtt{nu}^{\bar{a}'a}} & & \downarrow{\scriptstyle \mathtt{nu}^{\bar{a}a}} & & \downarrow{\scriptstyle \tau} \\ TQ^{\bar{a}'a}A & \xrightarrow[T\frac{\bar{a}'a}{\bar{a}a}]{} & TQ^{\bar{a}a}A & \xrightarrow[T\langle \frac{\bar{a}a}{\bar{a}}, \mathtt{id}\rangle]{} & T(Q^{\bar{a}}A \times Q^{\bar{a}a}A) \end{array} \qquad \begin{array}{ccc} A \times Q^{\bar{a}}B & \xrightarrow{\zeta} & Q^{\bar{a}}(A \times B) \\ \downarrow{\scriptstyle \mathtt{id} \times \mathtt{nu}_B} & & \downarrow{\scriptstyle \mathtt{nu}_{A\times B}} \\ A \times TQ^{\bar{a}a}B & \xrightarrow[\tau\,;\, T\zeta]{} & TQ^{\bar{a}a}(A \times B) \end{array} \tag{N2}$$

V. Setting $\mathbb{A}_A \triangleq Q^a 1$, for each $a \in \mathbb{A}_A$, there is a name-equality arrow $\mathtt{eq}_A : \mathbb{A}_A \times \mathbb{A}_A \to \mathbb{N}$ such that, for any distinct $a, b \in \mathbb{A}_A$, the following diagram commutes.

$$\begin{array}{ccccc} Q^a 1 & \xrightarrow{\Delta} & \mathbb{A}_A \times \mathbb{A}_A & \xleftarrow{\langle \frac{ab}{a}, \frac{ab}{b}\rangle} & Q^{ab}1 \\ \downarrow{\scriptstyle !} & & \downarrow{\scriptstyle \mathtt{eq}_A} & & \downarrow{\scriptstyle !} \\ 1 & \xrightarrow{\tilde{0}} & \mathbb{N} & \xleftarrow{\tilde{1}} & 1 \end{array} \tag{N1}$$

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

$$\mathtt{drf}_A : \mathbb{A}_A \to T[\![A]\!] \quad \text{and} \quad \mathtt{upd}_A : \mathbb{A}_A \times [\![A]\!] \to T1$$

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,

$$(\mathtt{nu}_A^{\bar{a}a} \times \mathtt{upd}_B)\,;\psi = (\mathtt{nu}_A^{\bar{a}a} \times \mathtt{upd}_B)\,;\psi'\,, \tag{SNR}$$

i.e., updates and fresh names are independent effects. ▲

The second subcondition of (N2) above essentially states that, for each object A, $\mathtt{nu}_A$ can be expressed as:

$$Q^{\bar{a}}A \xrightarrow{\cong} Q^{\bar{a}}1 \times A \xrightarrow{\mathtt{nu}_1\times\mathtt{id}} TQ^{\bar{a}a}1 \times A \xrightarrow{\tau'} T(Q^{\bar{a}a}1 \times A) \xrightarrow{\cong} TQ^{\bar{a}a}A$$

It is evident that the role reserved for $\mathtt{nu}$ in our semantics is that of fresh name creation. Accordingly, $\mathtt{nu}$ gives rise to a categorical name-abstraction operation: for any arrow $f : Q^{\bar{a}a}A \to TB$ in $\mathcal{M}$, we define

$$\langle\!\langle\, a \,\rangle\!\rangle\, f \triangleq Q^{\bar{a}}A \xrightarrow{\mathtt{nu}_A} TQ^{\bar{a}a}A \xrightarrow{Tf} T^2B \xrightarrow{\mu} TB\,. \tag{3.7}$$

The (NR) diagrams give the basic equations for dereferencings and updates (cf. [38, definition 1] and [44, section 5.8]). 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 3.13. 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 typing judgement $\bar{a} \mid \Gamma \vdash M : A$ is translated to an arrow $[\![M]\!]_{\bar{a}|\Gamma} : Q^{\bar{a}}[\![\Gamma]\!] \to T[\![A]\!]$ in $\mathcal{M}$, which we write simply as $[\![M]\!] : Q^{\bar{a}}\Gamma \to TA$, as in figure 3. ▲

We note that the translation of values follows a common pattern: for any $\bar{a} \mid \Gamma \vdash V : B$, we have $[\![V]\!] = |V|\,;\eta$, where

$$\begin{array}{llll} |x| \triangleq Q^{\bar{a}}\pi\,;\frac{\bar{a}}{\epsilon} & |\tilde{n}| \triangleq Q^{\bar{a}}!\,;\frac{\bar{a}}{\epsilon}\,;\tilde{n} & |\lambda x.M| \triangleq \Lambda^T(\zeta'\,;[\![M]\!]) \\ |a| \triangleq Q^{\bar{a}}!\,;\frac{\bar{a}}{a} & |\mathtt{skip}| \triangleq Q^{\bar{a}}!\,;\frac{\bar{a}}{\epsilon} & |\langle V, W\rangle| \triangleq \langle |V|, |W|\rangle\,. \end{array} \tag{3.8}$$

We can show the following lemmas, which will be used in the proof of Correctness.

Lemma 3.14. *For any* $\bar{a} \mid \Gamma \vdash M : A$ *and* $\mathtt{S}(\bar{a}) \subseteq \mathtt{S}(\bar{a}')$, $[\![M]\!]_{\bar{a}'|\Gamma} = \frac{\bar{a}'}{\bar{a}}\,;[\![M]\!]_{\bar{a}|\Gamma}$.
Moreover, if $\Gamma = x_1 : B_1, ..., x_n : B_n$, *and* $\bar{a} \mid \Gamma \vdash M : A$ *and* $\bar{a} \mid \Gamma \vdash V_i : B_i$ *are derivable,*

$$[\![M\{\vec{V}/\vec{x}\}]\!] = Q^{\bar{a}}\Gamma \xrightarrow{\langle \mathtt{id}, |V_1|, ..., |V_n|\rangle} Q^{\bar{a}}\Gamma \times \Gamma \xrightarrow{\zeta'\,;Q^{\bar{a}}\pi_2} Q^{\bar{a}}\Gamma \xrightarrow{[\![M]\!]} TA\,. \qquad \square$$

Lemma 3.15. *For any relevant* f, g,

$$\langle\!\langle a \rangle\!\rangle \left(Q^{\bar{a}a}A \xrightarrow{\langle f, \frac{\bar{a}a}{\bar{a}}\,; g\rangle} TB \times TC \xrightarrow{\psi} T(B \times C)\right) = Q^{\bar{a}}A \xrightarrow{\langle \langle\!\langle a \rangle\!\rangle f, g\rangle} TB \times TC \xrightarrow{\psi} T(B \times C)\,,$$

$$\langle\!\langle a \rangle\!\rangle \left(Q^{\bar{a}a}A \xrightarrow{f} TB \xrightarrow{Tg} T^2C \xrightarrow{\mu} TC\right) = Q^{\bar{a}}A \xrightarrow{\langle\!\langle a \rangle\!\rangle f} TB \xrightarrow{Tg} T^2C \xrightarrow{\mu} TC\,. \qquad \square$$

Lemma 3.16. *Let* $\bar{a} \mid \Gamma \vdash M : A$ *and* $\bar{a} \mid \Gamma \vdash \mathrm{E}[M] : B$ *be derivable, with* $\mathrm{E}[\_]$ *being an evaluation context. Then* $[\![\mathrm{E}[M]]\!]$ *is equal to:*

$$Q^{\bar{a}}\Gamma \xrightarrow{\langle \mathtt{id}, [\![M]\!]\rangle} Q^{\bar{a}}\Gamma \times TA \xrightarrow{\tau} T(Q^{\bar{a}}\Gamma \times A) \xrightarrow{T\zeta'} TQ^{\bar{a}}(\Gamma \times A) \xrightarrow{T[\![\mathrm{E}[x]]\!]} T^2B \xrightarrow{\mu} TB\,. \qquad \square$$

We write $S \models M \xrightarrow{\mathrm{r}} S' \models M'$ with $\mathrm{r} \in \{\mathrm{NEW}, \mathrm{SUC}, \mathrm{EQ}, ..., \mathrm{LAM}'\}$ if the last non-CTX rule in the related derivation is r. Also, to any store S, we relate the term $\bar{S}$ of type $\mathbb{1}$ as:

$$\bar{\epsilon} \triangleq \mathtt{skip}\,, \quad \overline{a, S} \triangleq \bar{S}\,, \quad \overline{a :: V, S} \triangleq (a := V\,;\bar{S}) \tag{3.9}$$

Proposition 3.17 (**Correctness**). *For any typed term* $\bar{a} \mid \Gamma \vdash M : A$, *and* S *with* $\mathtt{dom}(S) = \bar{a}$, *and* r *as above,*

1. *if* $\mathrm{r} \notin \{\mathrm{NEW}, \mathrm{UPD}, \mathrm{DRF}\}$ *then* $S \models M \xrightarrow{\mathrm{r}} S \models M' \implies [\![M]\!] = [\![M']\!]$,
2. *if* $\mathrm{r} \in \{\mathrm{UPD}, \mathrm{DRF}\}$ *then* $S \models M \xrightarrow{\mathrm{r}} S' \models M' \implies [\![\bar{S}\,; M]\!] = [\![\bar{S}'\,; M']\!]$,
3. $S \models M \xrightarrow{\mathrm{NEW}} S, a \models M' \implies [\![\bar{S}\,; M]\!] = \langle\!\langle a \rangle\!\rangle\, [\![\bar{S}\,; M']\!]$.
Therefore, $S \models M \to S' \models M' \implies [\![\bar{S}\,; M]\!] = \langle\!\langle \bar{a}' \rangle\!\rangle\, [\![\bar{S}'\,; M']\!]$, *with* $\mathtt{dom}(S') = \bar{a}\bar{a}'$.

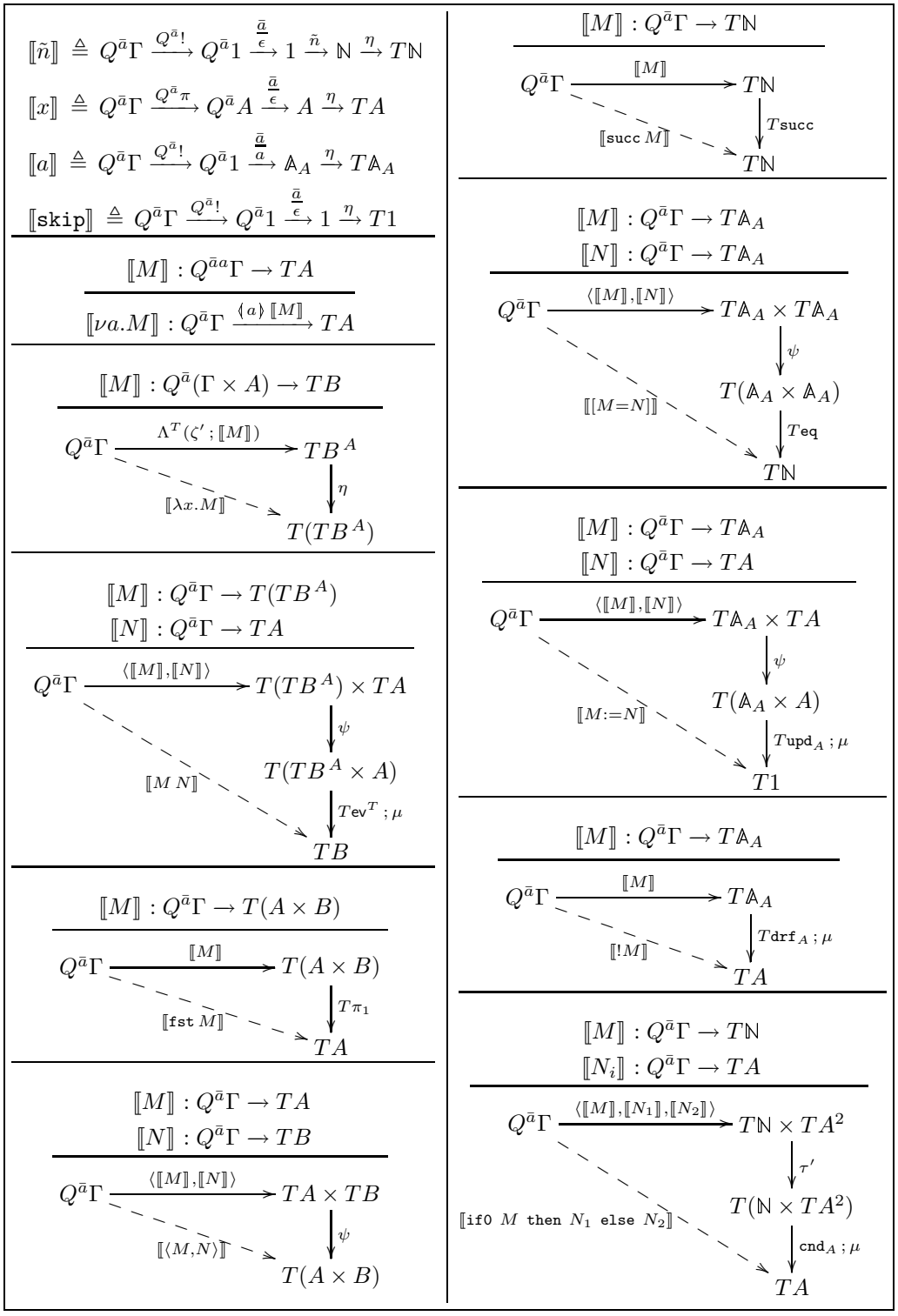
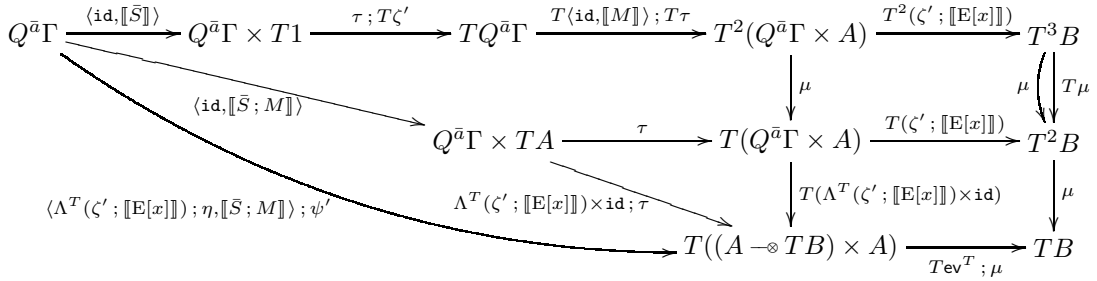


Figure 3: The semantic translation.

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 3.12 and lemma 3.14. 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 $\langle \mathtt{id}, [\![\bar{S}]\!] \rangle\,;\tau\,;T\zeta'\,;T[\![\mathrm{E}[M]]\!]\,;\mu$ and therefore to $[\![\bar{S}\,;\mathrm{E}[M]]\!]$. Hence, we can immediately show the inductive steps of 1-2. For 3, assuming $S \models \mathrm{E}[M] \xrightarrow{\text{NEW}} S, a \models \mathrm{E}[M']$ and $[\![\bar{S}\,;M]\!] = \langle\!|\, a \,|\!\rangle\, [\![\bar{S}\,;M']\!]$, we have, using also lemmas 3.14 and 3.15,

$$\begin{aligned}
\langle\!|\, a \,|\!\rangle\, [\![\bar{S}\,;\mathrm{E}[M']]\!] &= \langle\!|\, a \,|\!\rangle\, (\langle \Lambda^T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\eta, [\![\bar{S}\,;M']\!] \rangle\,;\psi'\,;T\mathsf{ev}^T\,;\mu) \\
&= \langle\!|\, a \,|\!\rangle\, (\langle \Lambda^T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\eta, [\![\bar{S}\,;M']\!] \rangle\,;\psi')\,;T\mathsf{ev}^T\,;\mu \\
&= \langle \Lambda^T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\eta, \langle\!|\, a \,|\!\rangle\, [\![\bar{S}\,;M']\!] \rangle\,;\psi'\,;T\mathsf{ev}^T\,;\mu \\
&= \langle \Lambda^T(\zeta'\,;[\![\mathrm{E}[x]]\!])\,;\eta, [\![\bar{S}\,;M]\!] \rangle\,;\psi'\,;T\mathsf{ev}^T\,;\mu = [\![\bar{S}\,;\mathrm{E}[M]]\!]\,.
\end{aligned}$$

□

In order for the model to be sound, we need computational adequacy. This is added explicitly as a specification.

Definition 3.18. Let $\mathcal{M}$ be a $\nu\rho$-model and $[\![\_]\!]$ the respective translation of $\nu\rho$. $\mathcal{M}$ is ***adequate*** if

$$\exists S, \bar{b}.\ [\![M]\!] = \langle\!|\, \bar{b} \,|\!\rangle\, [\![\bar{S}\,;\tilde{0}]\!] \implies \exists S'.\ \bar{a} \models M \longrightarrow\!\!\!\!\!\rightarrow S' \models \tilde{0}\,,$$

for any typed term $\bar{a} \mid \varnothing \vdash M : \mathbb{N}$. ▲

Proposition 3.19 (**Equational Soundness**). *If $\mathcal{M}$ is an adequate $\nu\rho$-model,*

$$[\![M]\!] = [\![N]\!] \implies M \lessapprox N\,.$$

□

3.2.3. *Completeness.* We equip the semantics with a preorder to match the observational preorder of the syntax as in (FA). The chosen preorder is the intrinsic preorder with regard to a collection of observable arrows in the biKleisli monadic-comonadic setting (cf. definition 3.10). In particular, since we have a collection of monad-comonad pairs, we also need a collection of sets of observable arrows.

Definition 3.20. An adequate $\nu\rho$-model $\mathcal{M} = (\mathcal{M}, T, Q)$ is ***observational*** if, for all $\bar{a}$:

- There exists $O^{\bar{a}} \subseteq \mathcal{M}(Q^{\bar{a}}1, T\mathbb{N})$ such that, for all $\bar{a} \mid \varnothing \vdash M : \mathbb{N}$,

$$[\![M]\!] \in O^{\bar{a}} \iff \exists S, \bar{b}.\ [\![M]\!] = \langle\!|\, \bar{b} \,|\!\rangle\, [\![\bar{S}\,;\tilde{0}]\!]\,.$$

- The induced intrinsic preorder on arrows in $\mathcal{M}(Q^{\bar a}A, TB)$ defined by
$$f \lesssim^{\bar a} g \iff \forall \rho : Q^{\bar a}(TB^{\,A}) \to T\mathbb{N}.\, (\Lambda^{\bar a}(f)\,;\rho \in O^{\bar a} \implies \Lambda^{\bar a}(g)\,;\rho \in O^{\bar a})$$
with $\Lambda^{\bar a}(f) \triangleq \Lambda^{Q^{\bar a},T}(f)$, satisfies, for all relevant $a, \bar a', f, f'$,
$$f \lesssim^{\bar a a} f' \implies \langle\!\langle\, a \,\rangle\!\rangle f \lesssim^{\bar a} \langle\!\langle\, a \,\rangle\!\rangle f' \quad \wedge \quad f \lesssim^{\bar a} f' \implies \tfrac{\bar a'}{\bar a}\,; f \lesssim^{a'} \tfrac{\bar a'}{\bar a}\,; f'\,.$$

We write $\mathcal{M}$ as $(\mathcal{M}, T, Q, O)$. ▲

Recurring to $\Lambda^{Q^{\bar a},T}$ of definition 3.10, we have that $\Lambda^{\bar a}(f)$ is the arrow:
$$Q^{\bar a}1 \xrightarrow{\delta} Q^{\bar a}Q^{\bar a}1 \xrightarrow{Q^{\bar a}\Lambda^T(\zeta'\,;\,f)} Q^{\bar a}(TB^{\,A})\,. \tag{3.10}$$
Hence, $O^{\bar a}$ contains those arrows that have a specific *observable behaviour* in the model, and the semantic preorder is built over this notion. In particular, terms that yield a number 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. Congruences for $\nu\rho$, along with typed contexts, are defined properly in [48]. For now, we state the following.

Lemma 3.21. *Let $(\mathcal{M}, T, Q, O)$ be an observational $\nu\rho$-model. Then, for any pair $\bar a \mid \Gamma \vdash M, N : A$ of typed terms and any context C such that $\bar a' \mid \Gamma' \vdash \mathrm{C}[M], \mathrm{C}[N] : B$ are valid,*
$$[\![M]\!] \lesssim^{\bar a} [\![N]\!] \implies [\![\mathrm{C}[M]]\!] \lesssim^{\bar a'} [\![\mathrm{C}[N]]\!]\,.$$
□

Assuming that we translate $\nu\rho$ into an observational $\nu\rho$-model, we can now show one direction of (FA).

Proposition 3.22 (**Inequational Soundness**). *For typed terms $\bar a \mid \Gamma \vdash M, N : A$,*
$$[\![M]\!] \lesssim [\![N]\!] \implies M \lessapprox N\,.$$

Proof: Assume $[\![M]\!] \lesssim^{\bar a} [\![N]\!]$ and $\models \mathrm{C}[M] \longrightarrow\!\!\!\!\!\rightarrow S' \models \tilde 0$, so $[\![\mathrm{C}[M]]\!] = \langle\!\langle\, \bar a' \,\rangle\!\rangle [\![\bar S'\,; \tilde 0]\!]$ with $\bar a' = \mathtt{dom}(S')$. $[\![M]\!] \lesssim^{\bar a} [\![N]\!]$ implies $[\![\mathrm{C}[M]]\!] \lesssim [\![\mathrm{C}[N]]\!]$, and hence $[\![\mathrm{C}[N]]\!] \in O^{\epsilon}$. Thus, by adequacy, there exists S'' such that $\models \mathrm{C}[N] \longrightarrow\!\!\!\!\!\rightarrow S'' \models \tilde 0$. □

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 3.23. 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 any $\bar a, A, B$, there exists $D^{\bar a}_{A,B} \subseteq \mathcal{M}(Q^{\bar a}[\![A]\!], T[\![B]\!])$ such that:

- For each $f \in D^{\bar a}_{A,B}$ there exists a term M such that $[\![M]\!] = f$.
- For each $f, g \in \mathcal{M}(Q^{\bar a}A, TB)$,
$$f \lesssim^{\bar a} g \iff \forall \rho \in D^{\bar a}_{A\to B,\mathbb{N}}\,.\,(\Lambda^{\bar a}(f)\,;\rho \in O^{\bar a} \implies \Lambda^{\bar a}(g)\,;\rho \in O^{\bar a})\,.$$

We write $\mathcal{M}$ as $(\mathcal{M}, T, Q, O, D)$. ▲

For such a model $\mathcal{M}$ we achieve full abstraction.

Theorem 3.24 (**FA**). *For typed terms $\bar a \mid \Gamma \vdash M, N : A$,*
$$[\![M]\!] \lesssim [\![N]\!] \iff M \lessapprox N\,.$$

Proof: Soundness is by previous proposition. For completeness ("$\Longleftarrow$"), we do induction on the size of Γ.
For the base case suppose $\bar{a} \mid \varnothing \vdash M \lessapprox N$ and take any $\rho \in D_{\mathbb{1}\to A,\mathbb{N}}$ such that $\Lambda^{\bar{a}}([\![M]\!])\,;\rho \in O^{\bar{a}}$. Let $\rho = [\![\bar{a} \mid y : \mathbb{1} \to A \vdash L : \mathbb{N}]\!]$, some term L, so $\Lambda^{\bar{a}}([\![M]\!])\,;\rho$ is

$$\Lambda^{\bar{a}}([\![M]\!])\,;[\![L]\!] = \delta\,;Q^{\bar{a}}|\lambda z.M|\,;[\![L]\!] = [\![(\lambda y.L)(\lambda z.M)]\!]$$

for some $z : \mathbb{1}$. The latter being in $O^{\bar{a}}$ implies that it equals $\langle\!\langle \bar{b} \rangle\!\rangle\, [\![\bar{S}\,;\tilde{0}]\!]$, some S. Now, $M \lessapprox N$ implies $(\lambda y.L)(\lambda z.M) \lessapprox (\lambda y.L)(\lambda z.N)$, hence $\nu\bar{b}.(\bar{S}\,;\tilde{0}) \lessapprox (\lambda y.L)(\lambda z.N)$, by soundness. But this implies that $\bar{a} \models (\lambda y.L)(\lambda z.N) \longrightarrow\!\!\!\!\!\rightarrow\; S' \models \tilde{0}$, so $[\![(\lambda y.L)(\lambda z.N)]\!] \in O^{\bar{a}}$, by correctness. Hence, $\Lambda^{\bar{a}}([\![N]\!])\,;\rho \in O^{\bar{a}}$, so $[\![M]\!] \lesssim^{\bar{a}} [\![N]\!]$, by ip-definability.

For the inductive step, if $\Gamma = x{:}B, \Gamma'$ then

$$\begin{aligned} \bar{a} \mid \Gamma \vdash M \lessapprox N &\implies \bar{a} \mid \Gamma' \vdash \lambda x.M \lessapprox \lambda x.N \overset{IH}{\implies} [\![\lambda x.M]\!] \lesssim^{\bar{a}} [\![\lambda x.N]\!] \\ &\implies [\![M]\!] = [\![(\lambda x.M)x]\!] \lesssim^{\bar{a}} [\![(\lambda x.N)x]\!] = [\![N]\!] \end{aligned}$$

where the last approximation follows from lemma 3.21. □

4. Nominal games

In this section we introduce nominal games and strategies, and construct the basic structure from which a fully abstract model of $\nu\rho$ will be obtained in the next section. We first introduce nominal arenas and strategies, which form the category $\mathcal{G}$. We afterwards refine $\mathcal{G}$ by restricting to *innocent, total* strategies, obtaining thus the category $\mathcal{V}_{\mathtt{t}}$.

$\mathcal{V}_{\mathtt{t}}$ is essentially a semantical basis for call-by-value nominal computation in general. In fact, from it we can obtain not only fully abstract models of $\nu\rho$, but also of the ν-calculus [2], the $\nu\varepsilon\rho$-calculus [47] ($\nu\rho$+exceptions), etc.

4.1. The basic category $\mathcal{G}$.

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

Definition 4.1. 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, so initial moves have level 0.
(l1) Initial moves are P-Answers.
(l2) If $m_1, m_2 \in M_A$ are at consecutive levels then they have complementary OP-labels.
(l3) Answers may only justify Questions. ▲

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. 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 $M_A \setminus J_A$.

Note that, although the nominal arenas of [2] are defined by use of a set of weaker conditions than those above, the actual arenas used there fall within the above definition. We move on to prearenas, which are the 'boards' on which nominal games are played.

Definition 4.2. 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 $A \to B$ as:

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

where $\bar\lambda_A$ is the OP-complement of λ_A. ▲

It is useful to think of the (pre)arena A as a vertex-labelled directed graph with vertex-set M_A and edge-set $\vdash_A$ such that the labels on vertices are 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.

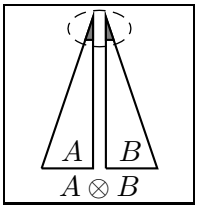
The simplest arena is $0 \triangleq (\varnothing,\varnothing,\varnothing,\varnothing)$. Other (flat) arenas are 1 (*unit arena*), $\mathbb{N}$ (*arena of naturals*) and $\mathbb{A}^{\bar a}$ (*arena of $\bar a$-names*), for any $\bar a \in \mathbb{A}^{\#}$, which we define as

$$M_1 = I_1 \triangleq \{*\}, \qquad M_{\mathbb{N}} = I_{\mathbb{N}} \triangleq \mathbb{N}, \qquad M_{\mathbb{A}^{\bar a}} = I_{\mathbb{A}^{\bar a}} \triangleq \mathbb{A}^{\bar a}, \tag{4.1}$$

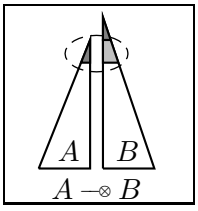
where $\mathbb{A}^{\bar a} \triangleq \{\,\pi \circ \bar a \,|\, \pi \in \mathrm{PERM}(\mathbb{A})\,\}$. Note that for $\bar a$ empty we get $\mathbb{A}^{\epsilon} = 1$, and that we write $\mathbb{A}_i$ for $\mathbb{A}^a$ with a being of type i.

More involved are the following constructions. For arenas A, B, define the arenas $A \otimes B$, $A_\perp$, $A \multimap B$ and $A \Rightarrow B$ as follows.

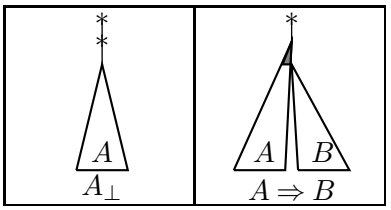
$$\begin{aligned} M_{A\otimes B} &\triangleq I_A \times I_B + \bar I_A + \bar I_B \\ I_{A\otimes B} &\triangleq I_A \times I_B \\ \lambda_{A\otimes B} &\triangleq [\,((i_A,i_B) \mapsto PA)\,, \lambda_A \upharpoonright \bar I_A\,, \lambda_B \upharpoonright \bar I_B] \\ \vdash_{A\otimes B} &\triangleq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



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



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



In the constructions above it is assumed that all moves which are not hereditarily justified by initial moves are discarded. 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,

$$1 \multimap A = A\,, \quad 0 \Rightarrow A = A_\perp\,.$$

Using the latter convention, the construction of $A \Rightarrow B$ in the previous definition is equivalent to $A \Rightarrow B$ of [15, 2]; concretely, it is given by:

$$
\begin{aligned}
M_{A\Rightarrow B} &\triangleq \{*\} + I_A + \bar{I}_A + M_B \\
I_{A\Rightarrow B} &\triangleq \{*\} \\
\lambda_{A\Rightarrow B} &\triangleq [\,(* \mapsto PA)\,,(i_A \mapsto OQ)\,,\bar{\lambda}_A\,,\lambda_B] \\
\vdash_{A\Rightarrow B} &\triangleq \{(*,i_A)\} \cup \{\,(i_A,m) \mid i_A \vdash_A m \vee m \in I_B\,\} \cup (\vdash_A \restriction \bar{I}_A{}^2) \cup (\vdash_B \restriction M_B{}^2)
\end{aligned}
\tag{4.2}
$$

Of the previous constructors all look familiar apart from $\multimap$ (which in [46] appears as $\tilde{\Rightarrow}$). 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.

We move on to describe how are nominal games played. Plays of a game consist of sequences of moves from some prearena. These moves are attached with name-lists to the effect of capturing name-environments.

Definition 4.3. A ***move-with-names*** of a (pre)arena A is a pair, written $m^{\bar{a}}$, where m is a move of A and $\bar{a}$ 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,

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

We introduce some notation for sequences (and lists).

Notation 4.4 (**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 $t = s(t \setminus s)$,
- $s \preceq t$ denotes that s is a (not necessarily initial or contiguous) subsequence of t,
- 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 end of s (for example, $s.{-}1$ is s_n),
- $s_{\leq s_i}$ denotes $s_1 \dots s_i$, and so does $s_{<s_{i+1}}$,
- if s is a sequence of moves-with-names then, by extending our previous notation, we have $s = \underline{s}^{\mathsf{nlist}(s)}$, 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 $s_{<i}$ (called the ***pending question***). 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.

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

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

Definition 4.5. Let A be a prearena. A legal sequence on A is 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 (cf. [34]):

(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 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 [2]). Note also that plays are the ϵ-plays of [46]. Now, some further notation.

Notation 4.6 (**Name-introduction**)**.** A name a is introduced (by Player) in a play s, written $a \in \mathcal{L}(s)$, if there exist consecutive moves yx in s such that x is a P-move and $a \in \mathsf{S}(\mathsf{nlist}(x) \setminus \mathsf{nlist}(y))$. ▲

From plays we move on to strategies. Recall the notion of name-restriction we introduced in definition 2.4; for any nominal set X and any $x \in X$, $[x] = \{\, \pi \circ x \mid \pi \in \mathrm{PERM}(\mathbb{A}) \,\}$.

Definition 4.7. Let A be a prearena. A ***strategy*** σ on A is a 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. Some basic strategies are the following — note that we give definitions *modulo prefix closure*.

Definition 4.8. For any $\bar{a}', \bar{a} \in \mathbb{A}^{\#}$ with $\mathtt{S}(\bar{a}') \subseteq \mathtt{S}(\bar{a})$, $n \in \mathbb{N}$ and any arena B, define the following strategies.

- $\tilde{n} : 1 \to \mathbb{N} \triangleq \{[* \, n]\}$
- $!_B : B \to 1 \triangleq \{[i_B \, *]\}$
- $\frac{\bar{a}}{\bar{a}'} : \mathbb{A}^{\bar{a}} \to \mathbb{A}^{\bar{a}'} \triangleq \{[\bar{a} \, \bar{a}']\}$
- $\mathtt{id}_B : B \to B \triangleq \{\, [s] \mid s \in P_{B_{(1)} \to B_{(2)}} \wedge \forall t \leq^{\text{even}} s.\ t \upharpoonright B_{(1)} = t \upharpoonright B_{(2)} \,\}$ ▲

It is easy to see that the aforedefined are indeed strategies. That definitions are given modulo prefix closure means that e.g. $\tilde{n}$ is in fact:

$$\tilde{n} = \{\, [\epsilon],\, [*],\, [* \, n] \,\}.$$

We proceed to composition of plays and strategies. In ordinary games, plays are composed by doing "parallel composition plus hiding" (v. [4]); in nominal games we need also take some extra care for names.

Definition 4.9. 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 $\underline{s} \upharpoonright B = \underline{t} \upharpoonright B$.
- 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 [15], adapted from [7].

Lemma 4.10 (Zipper lemma). *If $s \in P_{A \to B}$ and $t \in P_{B \to C}$ with $s \smile t$ then either $\underline{s} \upharpoonright B = \underline{t} = \epsilon$, 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. [15, 5]) without further mention. Composable plays are composed as below. Note that we may tag a move m as $m_{(O)}$ (or $m_{(P)}$) to specify it is an O-move (a P-move).

Definition 4.11. 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 follows. We set $\epsilon \parallel \epsilon \triangleq \epsilon$, $\epsilon \bullet \epsilon \triangleq \epsilon$, and:

$$sm^{\bar{b}}_A \parallel t \triangleq (s \parallel t)m^{sm^{\bar{b}}_A \bullet t}_A \qquad sm^{\bar{b}}_B \parallel tm^{\bar{c}}_B \triangleq (s \parallel t)m^{sm^{\bar{b}}_B \bullet tm^{\bar{c}}_B}_B \qquad s \parallel tm^{\bar{c}}_C \triangleq (s \parallel t)m^{s \bullet tm^{\bar{c}}_C}_C$$

$$sm^{\bar{b}_s\bar{b}}_{A(P)} \bullet t \triangleq (s \bullet t)\,\bar{b} \qquad sm^{\bar{b}_s\bar{b}}_{B(P)} \bullet tm^{\bar{c}}_{B(O)} \triangleq (s \bullet t)\,\bar{b} \qquad s \bullet tm^{\bar{c}_t\bar{c}}_{C(P)} \triangleq (s \bullet t)\,\bar{c}$$

$$sm^{\bar{b}}_{A(O)} \bullet t \triangleq \bar{b}' \qquad sm^{\bar{b}}_{B(O)} \bullet tm^{\bar{c}_t\bar{c}}_{B(P)} \triangleq (s \bullet t)\,\bar{c} \qquad s \bullet tm^{\bar{c}}_{C(O)} \triangleq \bar{c}\,',$$

where $\bar{b}_s$ is the name-list of the last move in s, and $\bar{b}'$ is the name-list of $m_{A(O)}$'s justifier inside $s \parallel t$; similarly for $\bar{c}_t$ and $\bar{c}\,'$.
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 as:

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

When composing compatible plays s and t, although their parts appearing in the common component (B) are hidden, the names appearing in (the support of) s and t are not lost but rather propagated to the output components (A and C). This is shown in the following lemma (the proof of which is tedious but not difficult, see [48]).

Lemma 4.12. *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 $m^{\bar{b}}$ then $\bar{b}$ 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 $m^{\bar{b}}$ then $\bar{b}$ 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 $m^{\bar{b}}$ then $\bar{b}$ contains as a prefix the name-list of the move explicitly justifying $m^{\bar{b}}$.*
(d) *If $s = s'm^{\bar{b}}$ ends in A and t in B then $\bar{b} \preceq s \bullet t$,*
if $s = s'm^{\bar{b}}$ and $t = t'm^{\bar{c}}$ end in B then $\bar{b} \preceq s \bullet t$ and $\bar{c} \preceq s \bullet t$,
if s ends in B and $t = t'm^{\bar{c}}$ in C then $\bar{c} \preceq s \bullet t$.
(e) $\mathtt{S}(s) \cup \mathtt{S}(t) = \mathtt{S}(s \parallel t) = \mathtt{S}(s\,;t) \cup \mathtt{S}(s \bullet t)\,.$ □

Proposition 4.13 (**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 $s = s'm^{\bar{b}}$, and so $(s\,;t)^- = s'\,;t$. Now, if $a \# s' \bullet t$ then, by part (e) of previous lemma, $a \# s', t$ and therefore $a \in \bar{b}$, by (NC2) of s. By part (d) then, $a \in \mathtt{S}(s \bullet t)$. Otherwise, $a \in \mathtt{S}(s' \bullet t)$ and hence, by part (a), $a \in \mathtt{S}(s \bullet t)$. □

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

Definition 4.14. For strategies $\sigma : A \to B$ and $\tau : B \to C$, their composition is defined as

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

and is a candidate strategy on $A \to C$. ▲

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 indeed strategies. Again, the proofs of the following technical lemmata are omitted for economy (but see [48]).

Lemma 4.15. *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 $t_1 \leq t_2$.* □

Lemma 4.16. *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 $\pi \circ (s_1 \parallel t_1) \leq s_2 \parallel t_2$.* □

Proposition 4.17 (**Strategies compose**). *If $\sigma : A \to B$ and $\tau : B \to C$ are strategies then so is $\sigma\,;\tau$.*

Proof: By definition and proposition 4.13, $\sigma\,;\tau$ contains equivalence classes of plays. We need also check prefix closure, contingency completeness and determinacy. The former two are rather straightforward, so we concentrate on the latter.
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 $s_i = s_i' n_i^{\bar{b}_i}$ and $s_i\,;t_i = (s_i'\,;t_i) n_i^{\bar{b}_i'}$, $i = 1, 2$. Now, $[s_1'\,;t_1] = [u_1] = [u_2] = [s_2'\,;t_2]$, so, by lemma 4.16 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 $[s_2''] = [s_1']$, $\therefore\ [s_2''(s_2''' n_2^{\bar{b}_2}).1] = [s_1' n_1^{\bar{b}_1}]$, by determinacy of σ, and hence $|s_2'''| = 0$, $s_2' = \pi \circ s_1'$ and $t_2 = \pi \circ t_1$. Moreover, $\pi' \circ s_1' n_1^{\bar{b}_1} = s_2' n_2^{\bar{b}_2}$, some permutation π'. Now we can apply the Strong Support Lemma, as (C1) implies $(\mathtt{S}(n_i^{\bar{b}_i}) \setminus \mathtt{S}(s_i')) \cap \mathtt{S}(t_i) = \varnothing$. 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 4.16, $\pi \circ s_1^- \leq s_2$, 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 order to obtain a category of nominal games, we still need to show that strategy composition is associative. We omit the (rather long) proof and refer the interested reader to [48].

Proposition 4.18. *For any $\sigma : A \to B$, $\sigma_1 : A' \to A$ and $\sigma_3 : B \to B'$,*

$$\mathtt{id}_A\,;\sigma = \sigma = \sigma\,;\mathtt{id}_B \quad \wedge \quad (\sigma_1\,;\sigma)\,;\sigma_3 = \sigma_1\,;(\sigma\,;\sigma_3)\,.$$ □

Definition 4.19. The ***category $\mathcal{G}$ of nominal games*** contains nominal arenas as objects and nominal strategies as arrows. ▲

In the rest of this section let us examine closer the proof of proposition 4.17 in order identify where exactly is strong support needed, and for which reasons is the nominal games model of [2] flawed.

Remark 4.20 (**The need for strong support**). The nominal games presented here differ from those of [2] crucially in one aspect; namely, the modelling of local state. In [2] local state is modelled by finite sets of names, so a move-with-names is a move attached with a finite set of names, 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 4.16; in the (omitted) 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 that of [2]. For take some $i \in \omega$ and consider the following AGMOS-strategies (i.e. strategies of [2]).

$$\begin{aligned} \sigma &: 1 \to \mathbb{A}_i \triangleq \{\, [* \, a^{\{a,b\}}] \mid a, b \in \mathbb{A}_i \wedge a \neq b \,\}, \\ \tau &: \mathbb{A}_i \to \mathbb{A}_i \Rightarrow \mathbb{A}_i \triangleq \{\, [a * c\, a] \mid a, c \in \mathbb{A}_i \,\}. \end{aligned} \tag{4.20:A}$$

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):

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

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

The above statement does not hold for AGMOS-games. To show this, we need to introduce[7] 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.

$$\begin{aligned} \sigma &: \mathbb{A}_i \otimes \mathbb{A}_i \to \mathbb{A}_i \odot \mathbb{A}_i \triangleq \{\, [\,(a,b)\,\{a,b\}] \mid a, b \in \mathbb{A}_i \wedge a \neq b \,\} \\ \tau &: \mathbb{A}_i \odot \mathbb{A}_i \to \mathbb{A}_i \triangleq \{\, [\{a,b\}\, a] \mid a, b \in \mathbb{A}_i \wedge a \neq b \,\} \end{aligned} \tag{4.20:B}$$

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 $[\,(a,b)\, a], [\,(a,b)\, b] \in \sigma ; \tau$. ▲

[7] This is because our presentation of nominal games does not include plays and strategies with non-empty initial local state. In the AGMOS setting we could have used to the same effect the $\{a,b\}$-strategies:

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

4.2. **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 consider (subset) orderings for arenas and strategies, the latter enriching $\mathcal{G}$ over Cpo.[8] These will prove useful for solving domain equations in categories of nominal games.

Definition 4.21. For any arenas A, B and each $\sigma, \tau \in \mathcal{G}(A, B)$ define $\sigma \sqsubseteq \tau \iff \sigma \subseteq \tau$. For each $\sqsubseteq$-increasing sequence $(\sigma_i)_{i\in\omega}$ take $\bigsqcup_i \sigma_i \triangleq \bigcup_i \sigma_i$. ▲

It is straightforward to see that each such $\bigsqcup_i \sigma_i$ 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 one containing only $[\epsilon]$). More than that, these cpo's enrich $\mathcal{G}$.

Proposition 4.22. *$\mathcal{G}$ is Cpo-enriched wrt $\sqsubseteq$.*

Proof: Enrichment amounts to showing the following straightforward assertions.

$$\sigma \sqsubseteq \sigma' \wedge \tau \sqsubseteq \tau' \implies \sigma\,;\tau \sqsubseteq \sigma'\,;\tau'$$

$$(\sigma_i)_{i\in\omega} \text{ an } \omega\text{-chain} \implies (\bigsqcup_{i\in\omega} \sigma_i)\,;\tau \sqsubseteq \bigsqcup_{i\in\omega}(\sigma_i\,;\tau)$$

$$(\tau_i)_{i\in\omega} \text{ an } \omega\text{-chain} \implies \sigma\,;(\bigsqcup_{i\in\omega} \tau_i) \sqsubseteq \bigsqcup_{i\in\omega}(\sigma\,;\tau_i)$$

□

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

Definition 4.23. 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:

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

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

There is also an indexed version of $\trianglelefteq$, for any $k \in \mathbb{N}$,

$$A \trianglelefteq_k B \iff A \trianglelefteq B \wedge \{\, m \in M_B \,|\, level(m) < k \,\} \subseteq M_A\,.$$

▲

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:

$$\mathtt{incl}_{A,B}\,;\mathtt{proj}_{B,A} = \mathtt{id}_A \quad \wedge \quad \mathtt{proj}_{B,A}\,;\mathtt{incl}_{A,B} \sqsubseteq \mathtt{id}_B \tag{4.3}$$

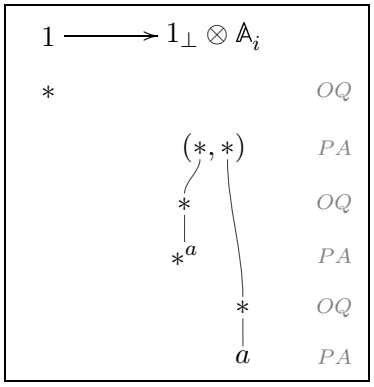
[8] By cpo we mean a partially ordered set with least element and least upper bounds for increasing ω-sequences. Cpo is the category of cpos and continuous functions.

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.

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

4.3. **Innocence: the category $\mathcal{V}$.** In game semantics for pure functional languages (e.g. PCF [16]), the absence of computational effects corresponds to innocence of the strategies. Here, although our aim is to model a language with effects, our model will use innocent strategies: the effects will still be achieved, by using monads.

Innocence is the condition stipulating that the 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 play on the side (where curved lines represent justification pointers) is $*\,(*,*)\,*\,a$ and violates (NC2). Consequently, we need to explicitly impose innocence on plays.



Definition 4.24. 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 justifying it.

The set of innocent plays of A is denoted by $P_A^{\mathtt{i}}$. ▲

It is not difficult to show now that a play s is innocent iff, for any $t \leq s$, $\ulcorner t \urcorner$ is a play. We can obtain the following characterisation of name-introduction in innocent plays.

Proposition 4.25 (**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 $a \in \mathtt{S}(x)$ 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 $a \in \mathtt{S}(x)$ and $a \# \ulcorner s_{<x} \urcorner$ then, by (NC2′), $a \in \mathsf{nlist}(x)$, while $a \# \ulcorner s_{<x} \urcorner$ implies $a \# \mathsf{nlist}(s_{<x}.{-}1)$. □

Innocent plays are closed under composition (proof omitted, v. [48]).

Proposition 4.26. *If $s \in P_{A \to B}$, $t \in P_{B \to C}$ are innocent and $s \asymp t$ then $s ; t$ is innocent.* □

We now move on to innocent strategies and show some basic properties.

Definition 4.27. A strategy σ is an ***innocent strategy*** 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 $[\ulcorner s_1 x_1 \urcorner] = [\ulcorner s_2 x_2 \urcorner]$. ▲

Lemma 4.28. *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 $s' \leq^{even} s$, $[\ulcorner s' \urcorner] \in \sigma$ 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 can now show that innocent strategies are closed under composition (details in [48]).

Proposition 4.29. *If $\sigma : A \to B, \tau : B \to C$ are innocent strategies then so is $\sigma\,;\tau$.* □

Definition 4.30. $\mathcal{V}$ is the lluf subcategory of $\mathcal{G}$ of innocent strategies. ▲

Henceforth, when we consider plays and strategies we presuppose them being innocent.

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 4.31. 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 and let f be a viewfunction. Then, we can define a corresponding viewfunction and a strategy by:

$$\mathtt{viewf}(\sigma) \triangleq \{\, [s] \in \sigma \mid |s| \text{ even} \wedge \ulcorner s \urcorner = s \,\}\,,$$
$$\mathtt{strat}(f) \triangleq \bigcup\nolimits_n \mathtt{strat}_n(f)\,,$$

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

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

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

Proposition 4.32. *For any innocent strategy σ, $\mathtt{viewf}(\sigma)$ is a viewfunction. Conversely, for any viewfunction f, $\mathtt{strat}(f)$ is an innocent strategy. Moreover,*

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

Recall the subset ordering $\sqsubseteq$ of strategies given in definition 4.21. It is easy to see that the ordering induces a cpo on innocent strategies and that $\mathcal{V}$ is Cpo-enriched. We can also show the following.

Corollary 4.33. *For all viewfunctions f, g and innocent strategies σ, τ,*

(1) $f \subseteq \mathtt{strat}(f)$,
(2) $\sigma \subseteq \tau \iff \mathtt{viewf}(\sigma) \subseteq \mathtt{viewf}(\tau)$, $\quad f \subseteq g \iff \mathtt{strat}(f) \subseteq \mathtt{strat}(g)$,
(3) $\mathtt{viewf}(\sigma) \subseteq \tau \wedge \mathtt{viewf}(\tau) \subseteq \sigma \implies \sigma = \tau$.

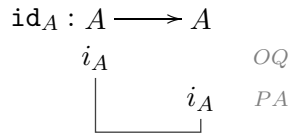
Moreover, $\sqsubseteq$ yields a cpo on viewfunctions, and `viewf` *and* `strat` *are continuous with respect to $\sqsubseteq$.* $\square$

Notation 4.34 (**Diagrams of viewfunctions**)**.** We saw previously 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

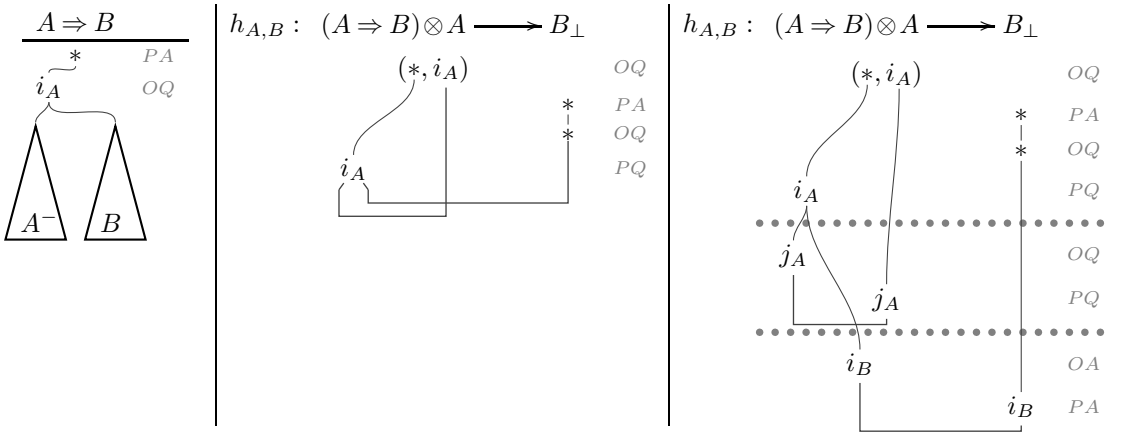
$$\mathtt{viewf}(\mathtt{id}_A) = \{\, [s m_{(1)} m_{(2)}] \,|\, [s] \in \mathtt{viewf}(\mathtt{id}_A) \wedge (m \in I_A \vee (s.\text{-}1 \vdash_A m_{(1)} \wedge s.\text{-}2 \vdash_A m_{(2)}))\,\}\,.$$

The above behaviour is called ***copycat*** (v. [4]) 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 in the middle 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 left 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),

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

Another way to depict $h_{A,B}$ is by cases with regard to Opponent's next move after l-i_A, as seen on the right diagram above.

Finally, we will sometimes label copycat links by strategies (e.g. in the proof of proposition 4.42). 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.

4.4. **Totality: the category $\mathcal{V}_{\mathtt{t}}$.** 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 level-1 moves, yielding several subclasses of innocent strategies. Note that an arena A is ***pointed*** if I_A is singleton.

Definition 4.35. An innocent strategy $\sigma : A \to B$ is ***total*** if for any $[i_A] \in \sigma$ there exists $[i_A\, i_B] \in \sigma$. A total strategy $\sigma : A \to B$ 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 $[i_A\, i_B\, j_B\, j_A^{\bar{b}}] \in \sigma$,
- ***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 $\tau : C \otimes A \to B$ 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 $[\,(i_C, i_A) i_B\, j_B\, j_A^{\bar{b}}] \in \tau$,
- ***tl4**** if it is both t4* and l4*.

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. ▲

The above subclasses of strategies will be demystified in the sequel. For now, we show a technical lemma. Let us define, for each arena A, the diagonal strategy Δ_A as follows.

$$\Delta_A : A \to A \otimes A \triangleq \mathtt{strat}\{\,[\,i_A\,(i_A, i_A)\,s\,]\,|\,[\,i_A\, i_A\ s\,] \in \mathtt{viewf}(\mathtt{id}_A)\,\} \tag{4.4}$$

Lemma 4.36 (**Separation of Head Occurrence**). *Let A be a pointed arena and let $f : A \to B$ be a t4 strategy. There exists a tl4* strategy $\tilde{f} : A \otimes A \to B$ 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

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

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 $f = \Delta\,;\tilde{f}$. □

We proceed to examine $\mathcal{V}_{\mathtt{t}}$. Eventually, we will see that it contains finite products and that it contains *some* exponentials, and that lifting promotes to a functor.

Lifting and product. We first promote the lifting and tensor arena-constructions to functors. In the following definition recall $\mathcal{L}$ from notation 4.6 and note that we write $\mathcal{L}(m) \# m'$ for $\mathcal{L}(m) \cap \mathtt{S}(m') = \varnothing$.

Definition 4.37. Let $f : A \to A'$, $g : B \to B'$ in $\mathcal{V}_{\mathtt{t}}$. Define the arrows

$$\begin{aligned} f \otimes g \;\triangleq\; \mathtt{strat}\{\, [\,(i_A, i_B)\,(i_{A'}, i_{B'})\, s\,] \,|\, & \\ & (\,[\,i_A\, i_{A'}\, s\,] \in \mathtt{viewf}(f) \wedge [i_B\, i_{B'}] \in g \wedge \mathcal{L}(i_A\, i_{A'}\, s) \# i_B) \\ & \vee\, (\,[\,i_B\, i_{B'}\, s\,] \in \mathtt{viewf}(g) \wedge [i_A\, i_{A'}] \in f \wedge \mathcal{L}(i_B\, i_{B'}\, s) \# i_A)\,\}\,, \\ f_\perp \;\triangleq\; \mathtt{strat}\{\, [* *' *' * s] \,|\, [s] \in \mathtt{viewf}(f)\,\}\,, & \end{aligned}$$

of types $f \otimes g : A \otimes B \to A' \otimes B'$ and $f_\perp : A_\perp \to A'_\perp$. ▲

Let us 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.

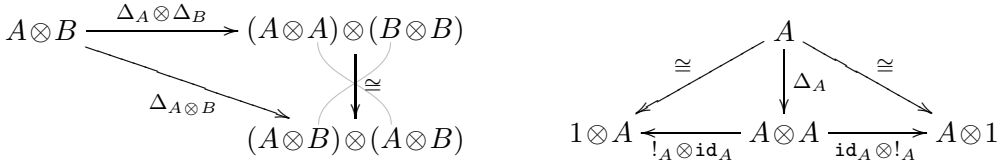
Note that $f_\perp$ is always ttotal. We can show the following.

Proposition 4.38. *$\_ \otimes \_ : \mathcal{V}_{\mathtt{t}} \times \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{t}}$ and $(\_)_\perp : \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{tt}*}$ are functors.* □

Moreover, $\otimes$ yields products and hence $\mathcal{V}_{\mathtt{t}}$ is cartesian.

Proposition 4.39. *$\mathcal{V}_{\mathtt{t}}$ 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 $\Delta : Id_{\mathcal{V}_{\mathtt{t}}} \to \otimes \circ \langle Id_{\mathcal{V}_{\mathtt{t}}}, Id_{\mathcal{V}_{\mathtt{t}}} \rangle$ (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 that the diagonal of (4.4) makes the above diagrams commute. Naturality follows from the single-threaded nature of strategies (v. [14]). □

Products are concretely given by triples $A \xleftarrow{\pi_1} A \otimes B \xrightarrow{\pi_2} B$, where

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

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

$$\begin{aligned}\langle f, g\rangle : C \to A \otimes B = \mathtt{strat}\{\,[\,i_C\,(i_A, i_B)\, s\,] \,|\, & \\ & (\,[i_C\ i_A\ s] \in \mathtt{viewf}(f) \wedge [i_C\ i_B] \in \mathtt{viewf}(g)\,) \\ & \vee\,(\,[i_C\ i_A] \in \mathtt{viewf}(f) \wedge [i_C\ i_B\ s] \in \mathtt{viewf}(g)\,)\,\}\,.\end{aligned}$$

Finally, we want 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 4.40. *For pointed arenas $\{A_i\}_{i\in\omega}$ define $\bigotimes_i A_i$ by:*

$$M_{\bigotimes_i A_i} \triangleq \{*\} + \biguplus_i \bar{I}_{A_i}\,, \qquad \lambda_{\bigotimes_i A_i} \triangleq [\,(* \mapsto PA), [\lambda_{A_i}{}^{i\in\omega}]]\,,$$

$$I_{\bigotimes_i A_i} \triangleq \{*\}\,, \qquad \vdash_{\bigotimes_i A_i} \triangleq \{(\dagger, *)\} \cup \{\,(*, j_{A_i}) \,|\, i \in \omega\,\} \cup \bigcup_i (\vdash_{A_i} \restriction \bar{I}_{A_i}{}^2)\,.$$

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

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

Then, $\bigotimes_{\_} : \prod \mathcal{V}_{\mathtt{t}*} \to \mathcal{V}_{\mathtt{t}*}$ is a functor. □

In fact, we could proceed and show that the aforedefined tensor yields general products of pointed objects, but this will not be of use here.

Partial exponentials. We saw that $\mathcal{V}_{\mathtt{t}}$ has products, given by the tensor functor $\otimes$. We now show that the arrow constructor yields appropriate partial exponentials, which will be sufficient for our modelling tasks.

Let us introduce the following transformations on strategies.

Definition 4.41. 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 $h : A \otimes B \to C$ and $g : A \to B \multimap C$,[9]

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

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

For each $(f, g) : (A, B) \to (A', B')$, define the arrows

$$\mathtt{ev}_{A,B} : (A \multimap B) \otimes A \to B \triangleq \Lambda^{A\ -1}_{A\multimap B, B}(\mathtt{id}_{A\multimap B})\,,$$

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

[9] 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 $\mathtt{viewf}(h)$.

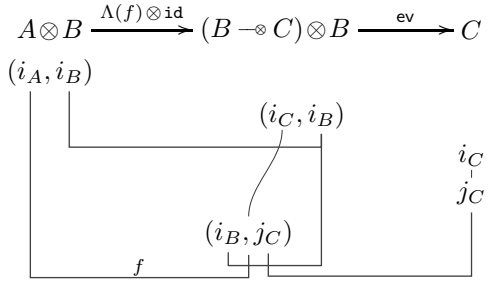
It is not difficult to see that Λ and Λ^{-1} are well-defined and mutual inverses. What is more, they supply us with exponentials.

Proposition 4.42. *$\mathcal{V}_{\mathtt{t}}$ has partial exponentials wrt to $\otimes$, in the following sense. For any object B, the functor $\_ \otimes B : \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{t}}$ has a partial right adjoint $B \multimap \_ : \mathcal{V}_{\mathtt{t}*} \to \mathcal{V}_{\mathtt{t}}$, that is, for any object A and any pointed object C the bijection $\Lambda^B_{A,C}$ is natural in A.*

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

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

These equalities are straightforward. For example, the viewfunction of $\Lambda(f) \otimes \mathtt{id} \,;\, \mathtt{ev}$ is given by the diagram on the side, which also gives the viewfunction of f. □



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

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

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

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

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:

$$\_ \Rightarrow \_ : (\mathcal{V}_{\mathtt{t}})^{\mathrm{op}} \times \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{tt}*} \triangleq \_ \multimap (\_)_\perp\,. \tag{4.5}$$

Remark 4.43. In the work on CBV games of Honda & Yoshida [15] the following version of partial exponentiation is shown.

$$\mathcal{V}(A \otimes B, C) \cong \mathcal{V}_{\mathtt{t}}(A, B \Rightarrow C) \tag{4.6}$$

Interestingly, that version can be derived from ours (using also another bijection shown in [15]),

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

But also vice versa, if C is pointed then $C \cong C_2 \Rightarrow C_1$, for some arenas C_1, C_2,[10] and

$$\mathcal{V}_{\mathtt{t}}(A \otimes B, C_2 \Rightarrow C_1) \overset{(4.6)}{\cong} \mathcal{V}(A \otimes B \otimes C_2, C_1) \overset{(4.6)}{\cong} \mathcal{V}_{\mathtt{t}}(A, (B \otimes C_2) \Rightarrow C_1) = \mathcal{V}_{\mathtt{t}}(A, B \multimap (C_2 \Rightarrow C_1))\,.$$

Strategy and arena orders. Recall the orders defined for strategies ($\sqsubseteq$) and arenas ($\trianglelefteq$) in section 4.2. These being subset orderings are automatically inherited by $\mathcal{V}_{\mathtt{t}}$. Moreover, by use of corollary 4.33 we can easily show that the aforedefined functors are continuous. Note that, although the strategy order $\sqsubseteq$ is inherited from $\mathcal{V}$, the least element (the empty strategy) is lost, as it is not total.

Proposition 4.44. *$\mathcal{V}_{\mathtt{t}}$ and $\mathcal{V}_{\mathtt{tt}}$ are PreCpo-enriched wrt $\sqsubseteq$.*[11] *Moreover,*

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

$$(\_ \multimap \_) : \mathcal{V}_{\mathtt{t}}^{\mathrm{op}} \times \mathcal{V}_{\mathtt{tt*}} \to \mathcal{V}_{\mathtt{tt*}}\,, \quad (\_ \Rightarrow \_) : \mathcal{V}_{\mathtt{t}}^{\mathrm{op}} \times \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{tt*}}$$

are locally continuous functors. □

The order of 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 that $A \trianglelefteq B$ does not imply that the corresponding projection is a total strategy — but $A \trianglelefteq_1 B$ does imply it. In fact,

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

Moreover, we have the following.

Proposition 4.45. *All of the functors of proposition 4.44 are continuous wrt $\trianglelefteq$. Moreover,*

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

Proof: All the clauses are in effect functoriality statements, since the underlying sets of inclusions and projections correspond to identity strategies. □

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

(f') For each $m \in M_A$, there exists unique $k \geq 0$ and a unique sequence $x_1 \dots x_n \in \{Q, A\}^*$ 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

$M_{C_1} \triangleq K_C^A + \{\, m \in M_C \mid \exists k \in K_C^A.\, k \vdash_C \cdots \vdash_C m \,\}$ $I_{C_1} \triangleq K_C^A$ $\vdash_{C_1} \triangleq \vdash_C \restriction (M_{C_1} \times \bar{I}_{C_1})$ $\lambda_{C_1} \triangleq \lambda_C \restriction M_{C_1}$

$M_{C_2} \triangleq \bar{I}_C \setminus M_{C_1}$ $\lambda_{C_2} \triangleq [i_{C_2} \mapsto PA, m \mapsto \bar{\lambda}_C(m)]$ $I_{C_2} \triangleq J_C$ $\vdash_{C_2} \triangleq \vdash_C \restriction (M_{C_2} \times \bar{I}_{C_2})$.

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

4.5. **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 sections. Specifically, we will upgrade lifting to a monad and introduce a family of product comonads for initial state.

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

Definition 4.46. Define the natural transformations $\mathtt{up}, \mathtt{dn}, \mathtt{st}$ as follows.

$$\mathtt{up}_A : A \to A_\perp = \mathtt{strat}\{\,[i_A *_1 *_2 i_A\, s] \,|\, [i_A\, i_A\, s] \in \mathtt{viewf}(\mathtt{id}_A)\,\}$$

$$\mathtt{dn}_A : A_{\perp\perp} \to A_\perp \triangleq \mathtt{strat}\{\,[*_1 *'_1 *'_2 *_2 *_3 *_4\, s] \,|\, [s] \in \mathtt{viewf}(\mathtt{id}_A)\,\}$$

$$\mathtt{st}_{A,B} : A \otimes B_\perp \to (A\otimes B)_\perp \triangleq \mathtt{strat}\{\,[\,(i_A, *_1) *'_1 *'_2 *_2\ i_B\,(i_A, i_B)\, s] \,|\, [\,(i_A, i_B)\,(i_A, i_B)\, s] \in \mathtt{viewf}(\mathtt{id}_{A\otimes B})\,\}$$

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

Proposition 4.47. *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, by setting:

$$\mathtt{pu}_A : A_\perp \to A \triangleq \mathtt{strat}\{\,[* i_A\, j_A\ *\ i_A\, j_A\, s] \,|\, [i_A\, i_A\, j_A\, j_A\, s] \in \mathtt{viewf}(\mathtt{id}_A)\,\}. \tag{4.7}$$

Lemma 4.48. *$\mathtt{pu}_A$ yields a natural transformation $\mathtt{pu} : (\_)_{\perp(\mathcal{V}_{\mathtt{tt}*})} \to Id_{\mathcal{V}_{\mathtt{tt}*}}$. Moreover, for any arenas A, B with B pointed, $\mathtt{up}_A ; \mathtt{pu}_A = \mathtt{id}_A$, $\mathtt{pu}_{A_\perp} = \mathtt{dn}_A$ and*

$$\mathtt{pu}_{A\multimap B} = \Lambda\Big((A \multimap B)_\perp \otimes A \xrightarrow{\mathtt{st}'} ((A \multimap B)\otimes A)_\perp \xrightarrow{\mathtt{ev}_\perp} B_\perp \xrightarrow{\mathtt{pu}_B} B\Big).$$ □

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 [9]. In our setting, the initial state can be any list $\bar a$ of distinct names; we define a comonad for each one of those lists.

Definition 4.49 (**Initial-state comonads**)**.** For each $\bar a \in \mathbb{A}^\#$ define the triple $(Q^{\bar a}, \varepsilon, \delta)$ by taking $Q^{\bar a} : \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{t}} \triangleq \mathbb{A}^{\bar a} \otimes \_$ and

$$\varepsilon : Q^{\bar a} \to Id_{\mathcal{V}_{\mathtt{t}}} \triangleq \{\,\varepsilon_A : \mathbb{A}^{\bar a}\otimes A \xrightarrow{\pi_2} A\,\},$$

$$\delta : Q^{\bar a} \to (Q^{\bar a})^2 \triangleq \{\,\delta_A : \mathbb{A}^{\bar a}\otimes A \xrightarrow{\Delta\otimes \mathtt{id}} \mathbb{A}^{\bar a}\otimes\mathbb{A}^{\bar a}\otimes A\,\}.$$

For each $\mathtt{S}(\bar a') \subseteq \mathtt{S}(\bar a)$ define the natural transformation $\frac{\bar a}{\bar a'} : Q^{\bar a} \to Q^{\bar a'}$ by taking

$$(\tfrac{\bar a}{\bar a'})_A : \mathbb{A}^{\bar a}\otimes A \to \mathbb{A}^{\bar a'}\otimes A \triangleq (\tfrac{\bar a}{\bar a'})_1 \otimes \mathtt{id}_A,$$

where $(\frac{\bar a}{\bar a'})_1$ is $\frac{\bar a}{\bar a'}$ of definition 4.8, that is, $(\frac{\bar a}{\bar a'})_1 \triangleq \{\,[\,(\bar a, *)\,(\bar a', *)\,]\,\}$. ▲

Note that Q^{ϵ}, the comonad for empty initial state, is the identity comonad. Note also that we have suppressed indices $\bar{a}$ from transformations ε, δ for notational economy.

Clearly, each triple $(Q^{\bar{a}}, \varepsilon, \delta)$ forms a product comonad on $\mathcal{V}_{\mathtt{t}}$. Moreover, it is straightforward to show the following.

Proposition 4.50 (**Chain rule**). *For each* $\mathtt{S}(\bar{a}') \subseteq \mathtt{S}(\bar{a}) \in \mathbb{A}^{\#}$*, the transformation* $\frac{\bar{a}}{\bar{a}'}$ *is a comonad morphism. Moreover,* $\frac{\bar{a}}{\epsilon} = \varepsilon : Q^{\bar{a}} \to Id_{\mathcal{V}_{\mathtt{t}}}$, $\frac{\bar{a}}{\bar{a}} = \mathtt{id} : Q^{\bar{a}} \to Q^{\bar{a}}$ *and, for each* $\mathtt{S}(\bar{a}') \subseteq \mathtt{S}(\bar{a}'') \subseteq \mathtt{S}(\bar{a})$, $\frac{\bar{a}}{\bar{a}''} ; \frac{\bar{a}''}{\bar{a}'} = \frac{\bar{a}}{\bar{a}'}$. □

Finally, for each name-type i, we can define a name-test arrow:

$$\mathtt{eq}_i : \mathbb{A}_i \otimes \mathbb{A}_i \to \mathbb{N} \triangleq \{\, [\,(a,a)\,0\,]\,\} \cup \{\, [\,(a,b)\,1\,] \,|\, a \neq b\,\}\,, \tag{4.8}$$

which clearly makes the (N1) diagram (definition 3.12) commute.

Fresh-name constructors. Combining the monad and comonads defined previously we can obtain a monadic-comonadic setting $(\mathcal{V}_{\mathtt{t}}, (\_)_{\perp}, Q)$, where Q denotes the family $(Q^{\bar{a}})_{\bar{a} \in \mathbb{A}^{\#}}$. This setting, which in fact yields a sound model of the ν-calculus [2, 48], will be used as the basis of our semantics of nominal computation in the sequel. Nominal computation of type A, in name-environment $\bar{a}$ and variable-environment Γ, will be translated into the set of strategies

$$\{\, \sigma : Q^{\bar{a}}[\![\Gamma]\!] \to [\![A]\!]_{\perp}\,\}\,.$$

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 semantical expression of fresh-name creation. Fresh names are created by means of natural transformations which transform a comonad $Q^{\bar{a}}$, say, to a monad-comonad composite $(Q^{\bar{a}a}\_)_{\perp}$.

Definition 4.51. Consider the setting $(\mathcal{V}_{\mathtt{t}}, (\_)_{\perp}, Q)$. We define natural transformations $\mathtt{new}^{\bar{a}a} : Q^{\bar{a}} \to (Q^{\bar{a}a}\_)_{\perp}$ by

$$\mathtt{new}^{\bar{a}a}_A \triangleq \mathbb{A}^{\bar{a}} \otimes A \xrightarrow{\mathtt{new}^{\bar{a}a}_1 \otimes \mathtt{id}_A} (\mathbb{A}^{\bar{a}a})_{\perp} \otimes A \xrightarrow{\mathtt{st}'} (\mathbb{A}^{\bar{a}a} \otimes A)_{\perp}\,,$$

$$\mathtt{new}^{\bar{a}a}_1 : \mathbb{A}^{\bar{a}} \otimes 1 \to (\mathbb{A}^{\bar{a}a} \otimes 1)_{\perp} \triangleq \mathtt{strat}\{\, [\,(\bar{a}, *) * *\,(\bar{a}a, *)^a]\,\}\,,$$

for each $\bar{a}a \in \mathbb{A}^{\#}$. ▲

That `new` is a natural transformation is straightforward: for any $f : A \to B$ we can form the following commutative diagram.

$$\begin{array}{ccccc}
\mathbb{A}^{\bar{a}} \otimes A & \xrightarrow{\mathtt{new}_1 \otimes \mathtt{id}} & (\mathbb{A}^{\bar{a}a})_{\perp} \otimes A & \xrightarrow{\mathtt{st}'} & (\mathbb{A}^{\bar{a}a} \otimes A)_{\perp} \\
\downarrow{\scriptstyle \mathtt{id} \otimes f} & & \downarrow{\scriptstyle \mathtt{id} \otimes f} & & \downarrow{\scriptstyle (\mathtt{id} \otimes f)_{\perp}} \\
\mathbb{A}^{\bar{a}} \otimes B & \xrightarrow[\mathtt{new}_1 \otimes \mathtt{id}]{} & (\mathbb{A}^{\bar{a}a})_{\perp} \otimes B & \xrightarrow[\mathtt{st}']{} & (\mathbb{A}^{\bar{a}a} \otimes B)_{\perp}
\end{array}$$

Moreover, we can show the following.

Proposition 4.52. *In the setting* $(\mathcal{V}_{\mathtt{t}}, (\_)_{\perp}, Q)$ *with* `new` *defined as above, the* (N2) *diagrams (definition 3.12) commute.* □

The fresh-name constructor allows us to define name-abstraction on strategies by taking:

$$\langle\!\langle a \rangle\!\rangle \sigma \triangleq Q^{\bar{a}}B \xrightarrow{\mathtt{new}^{\bar{a}a}_B} (Q^{\bar{a}a}B)_\perp \xrightarrow{\sigma_\perp} C_\perp \xrightarrow{\mathtt{pu}_C} C\,. \tag{4.9}$$

Name-abstraction can be given an explicit description as follows. For any sequence of moves-with-names s and any name $a \# \mathsf{nlist}(a)$, let s^a be s with a in the head of all of its name-lists. Then, for σ as above, we can show that:

$$\mathtt{viewf}(\langle\!\langle a \rangle\!\rangle \sigma) = \{\, [\,(\bar{a}, i_B)\, i_C\, j_C\, m^{a\bar{b}}\, s^a] \,|\, [\,(\bar{a}a, i_B)\, i_C\, j_C\, m^{\bar{b}}\, s] \in \mathtt{viewf}(\sigma) \wedge a \# i_B, j_C \,\} \tag{4.10}$$

We end our discussion on fresh-name constructors with a technical lemma stating that name-abstraction and currying commute.

Lemma 4.53. *Let $f : Q^{\bar{a}a}(A \otimes B) \to C$ with C a pointed arena. Then,*

$$\langle\!\langle a \rangle\!\rangle \Lambda(\zeta' ; f) = \Lambda(\zeta' ; \langle\!\langle a \rangle\!\rangle f) : Q^{\bar{a}}A \to B \multimap C\,.$$

Proof: As follows.

$$\begin{aligned}
\langle\!\langle a \rangle\!\rangle \Lambda(\zeta' ; f) &= \mathtt{new}^{\bar{a}a}_A ; (\Lambda(\zeta' ; f))_\perp ; \mathtt{pu}_{B \multimap C} = \mathtt{new}^{\bar{a}a}_A ; (\Lambda(\zeta' ; f))_\perp ; \Lambda(\mathtt{st}' ; \mathtt{ev}_\perp ; \mathtt{pu}_C) \\
&= \Lambda(\mathtt{new}^{\bar{a}a}_A \otimes \mathtt{id}_B ; (\Lambda(\zeta' ; f))_\perp \otimes \mathtt{id}_B ; \mathtt{st}' ; \mathtt{ev}_\perp ; \mathtt{pu}_C) \\
&= \Lambda(\mathtt{new}^{\bar{a}a}_A \otimes \mathtt{id}_B ; \mathtt{st}' ; (\Lambda(\zeta' ; f) \otimes \mathtt{id}_B)_\perp ; \mathtt{ev}_\perp ; \mathtt{pu}_C) \\
&= \Lambda(\mathtt{new}^{\bar{a}a}_A \otimes \mathtt{id}_B ; \mathtt{st}' ; (\zeta' ; f)_\perp ; \mathtt{pu}_C) \overset{(\mathrm{N2})}{=} \Lambda(\zeta' ; \mathtt{new}^{\bar{a}a}_{A \otimes B} ; f_\perp ; \mathtt{pu}_C)
\end{aligned}$$

and the latter equals $\Lambda(\zeta' ; \langle\!\langle a \rangle\!\rangle f)$. □

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 sequel, λ-abstraction is not simply currying, because of the use of monads.

4.6. **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 (AGMOS) [2] and another one due to Laird [21, 24]. Although Laird's constructions are 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 AGMOS, and those of [21, 24].

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. [31]). 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.[12] 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

[12]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 absence of innocence.

and therefore the two terms above are identified only at the extensional level (i.e. after quotienting).[13]

The major difference between [21] and [24] lies in the modelling of (ground-type, name-storing) store. In [21] 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 [24], where a carefully formulated setting of moves-with-store[14] 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.[15] 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 [21, 24], 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 or the ν-calculus: a name may have been used in a play but may still be inaccessible to some participant (that is, if it is outside his view [31]). On the other hand, our approach is advantageous in its simplicity and its applicability on a wide rage of nominal effects (see [48]), but suffers from the accuracy issues discussed above.

5. 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. 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.

[13]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 5.3 and compare e.g. with respective proof in [3]) by exploiting precisely this extra information!

[14]Inter alia, frugality of names implies that sequences of moves-with-store have strong support even if stores are represented by sets!

[15]In our model they correspond to the strategies (see also section 5):

$$\sigma_1 \triangleq \{\,[\,(a,b) * \circledast(*,\circledast)(n,\circledast)\,a\,c\,0]\,\}\,, \quad \sigma_2 \triangleq \{\,[\,(a,b) * \circledast(*,\circledast)(n,\circledast)\,0]\,\}\,.$$

Thus, the inner-workings of the store revealled 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 innaccuracies to be the most stubborn ones.

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 4.46) by use of a side-effect monad constructor, that is,

$$TA \triangleq \xi \multimap (A \otimes \xi)_\perp \qquad \text{i.e. } TA = \xi \Rightarrow A \otimes \xi\,.$$

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 (wrt type-structure) 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 $[\![A \to B]\!] = [\![A]\!] \otimes \xi \Rightarrow [\![B]\!] \otimes \xi\,$.

5.1. **Solving the Store Equation.** The full form of the store equation (SE) is:

$$\begin{gathered} [\![\mathbb{1}]\!] = 1\,, \qquad [\![\mathbb{N}]\!] = \mathbb{N}\,, \qquad [\![[A]]\!] = \mathbb{A}_A\,, \qquad [\![A \to B]\!] = [\![A]\!] \otimes [\![B]\!]\,, \\ [\![A \to B]\!] = [\![A]\!] \multimap (\xi \Rightarrow [\![B]\!] \otimes \xi)\,, \qquad \xi = \textstyle\bigotimes_A(\mathbb{A}_A \Rightarrow [\![A]\!])\,. \end{gathered}$$

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 [26]). It uses the categorical constructions of [43, 11] for solving recursive domain equations, as adapted to games in [26].

Definition 5.1. 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:

$$\begin{array}{l|l|l} [\![\mathbb{1}]\!]_{D,E} \triangleq 1 & [\![A \times B]\!]_{D,E} \triangleq [\![A]\!]_{D,E} \otimes [\![B]\!]_{D,E} & [\![[A]]\!]_{D,E} \triangleq \mathbb{A}_A \\ [\![\mathbb{N}]\!]_{D,E} \triangleq \mathbb{N} & [\![A \to B]\!]_{D,E} \triangleq D_A \multimap (\xi_{E,D} \Rightarrow E_B \otimes \xi_{D,E}) & \xi_{D,E} \triangleq \bigotimes_{A\in\mathrm{TY}}(\mathbb{A}_A \Rightarrow E_A) \end{array}$$

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 4.44), and continuous wrt the arena ordering (proposition 4.45). The solution to (SE*) is given via a *local bilimit* construction to the following ω-chain in $\mathcal{C}$.[16]

[16]Recall that we call an arrow $e : A \to B$ an ***embedding*** if there exists $e^R : B \to A$ such that

$$e\,;e^R = \mathtt{id}_A \;\wedge\; e^R\,;e \sqsubseteq \mathtt{id}_B\,.$$

Definition 5.2. In $\mathcal{C}$ form the sequence $(D_i)_{i\in\omega}$ taking D_0 as below and $D_{i+1} \triangleq F(D_i, D_i)$.

$$D_{0,\mathbb{1}} \triangleq 1 \qquad D_{0,\mathbb{N}} \triangleq \mathbb{N} \qquad D_{0,[A]} \triangleq \mathbb{A}_A$$
$$D_{0,A\to B} \triangleq 1 \qquad D_{0,A\times B} \triangleq D_{0,A} \otimes D_{0,B} \qquad D_{0,\xi} \triangleq \bigotimes\nolimits_A (\mathbb{A}_A \Rightarrow 0)$$

Moreover, define arrows $e_i : D_i \to D_{i+1}$ and $e_i^R : D_{i+1} \to D_i$ as:

$$e_0 \triangleq \mathtt{incl}_{D_0,D_1} \qquad e_0^R \triangleq \mathtt{proj}_{D_1,D_0} \qquad e_{i+1} \triangleq F(e_i^R, e_i) \qquad e_{i+1}^R \triangleq F(e_i, e_i^R)\,. \qquad \blacktriangle$$

The above inclusion and projection arrows are defined componentwise. In fact, there is a hidden lemma here 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).

$$(\Delta) \qquad D_0 \xrightarrow{e_0} D_1 \xrightarrow{e_1} D_2 \xrightarrow{e_2} D_3 \xrightarrow{e_3} \cdots$$

Thus, we have formed the ω-chain Δ. We show that Δ is a $\trianglelefteq$-increasing sequence of objects and embeddings, and proceed to the main result.

Lemma 5.3. *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: It is easy to see that $D_i \trianglelefteq_1 D_{i+1}$, all $i \in \omega$, so the above are well-defined. We now do induction on i; the base case is true by definition. The inductive step follows easily from proposition 4.45. □

Theorem 5.4. *We obtain a local bilimit $(D^*, {\eta_i}^{i\in\omega})$ for Δ by taking:*

$$D^* \triangleq \bigsqcup\nolimits_i D_i\,, \qquad \eta_i \triangleq \mathtt{incl}_{D_i,D^*} \quad \textit{(each } i \in \omega\textit{)}.$$

Hence, $\mathtt{id}_{D^*} : F(D^*, D^*) \to D^*$ 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 $\alpha : F(D^*, D^*) \to D^*$ by taking (see e.g. [1]):

$$\alpha \triangleq \bigsqcup\nolimits_i \alpha_i\,, \quad \alpha_i \triangleq F(\eta_i, \eta_i^R) ; \eta_{i+1} \overset{\text{prop. 4.45}}{=} \mathtt{proj}_{F(D^*,D^*),D_{i+1}} ; \mathtt{incl}_{D_{i+1},D^*}\,.$$

Moreover, $D^* = F(D^*, D^*)$ by the Tarski-Knaster theorem, and therefore $\alpha_i = \eta_{i+1}^R ; \eta_{i+1}$, which implies $\alpha = \mathtt{id}_{D^*}$. □

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 $(\eta_i : D_i \to D)_{i\in\omega}$ 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 $i \in \omega$,

$$\eta_i^R ; \eta_i \sqsubseteq \eta_{i+1}^R ; \eta_{i+1} \;\wedge\; \bigsqcup\nolimits_{i\in\omega}(\eta_i^R ; \eta_i) = \mathtt{id}_D\,.$$

Thus, D^* is the canonical solution to $D = F(D, D)$, and in particular it solves:

$$D_{A\to B} = D_A \multimap (D_\xi \Rightarrow D_B \otimes D_\xi)\,, \quad D_\xi = \bigotimes\nolimits_A (\mathbb{A}_A \Rightarrow D_A)\,.$$

Definition 5.5. Taking D^* as in the previous theorem define, for each type A,

$$\xi \triangleq D^*_\xi\,, \quad [\![A]\!] \triangleq D^*_A\,. \qquad \blacktriangle$$

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*.

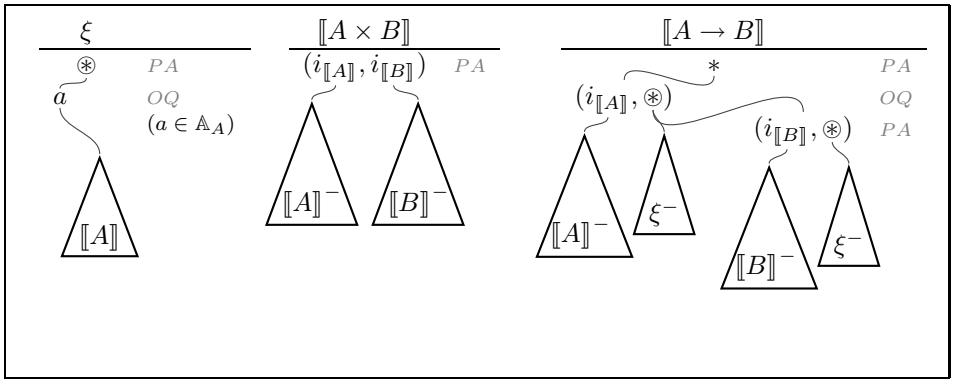


Figure 4: The store arena and the type translation.

The store monad T. There is a standard construction (v. [28]) for defining a monad of A-side-effects (any object A) starting from a given strong monad with exponentials. Here we define a store monad, i.e. a ξ-side-effects monad, from the lifting monad as follows.

$$\begin{aligned}
&T : \mathcal{C} \to \mathcal{C} \triangleq \xi \Rightarrow (_ \otimes \xi)\\
&\eta_A : A \to TA \triangleq \Lambda\left(A \otimes \xi \xrightarrow{\texttt{up}} (A \otimes \xi)_\perp\right)\\
&\mu_A : T^2A \to TA \triangleq \Lambda\left(T^2A \otimes \xi \xrightarrow{\texttt{ev}} (TA \otimes \xi)_\perp \xrightarrow{\texttt{ev}_\perp} (A \otimes \xi)_{\perp\perp} \xrightarrow{\texttt{dn}} (A \otimes \xi)_\perp\right)\\
&\tau_{A,B} : A \otimes TB \to T(A \otimes B) \triangleq \Lambda\left(A \otimes TB \otimes \xi \xrightarrow{\texttt{id}\otimes\texttt{ev}} A \otimes (B \otimes \xi)_\perp \xrightarrow{\texttt{st}} (A \otimes B \otimes \xi)_\perp\right)
\end{aligned} \tag{5.1}$$

A concrete description of the store monad is given in figure 5 (the diagrams of strategies depict their viewfunctions, as described in notation 4.34). 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 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 $\circledast$.

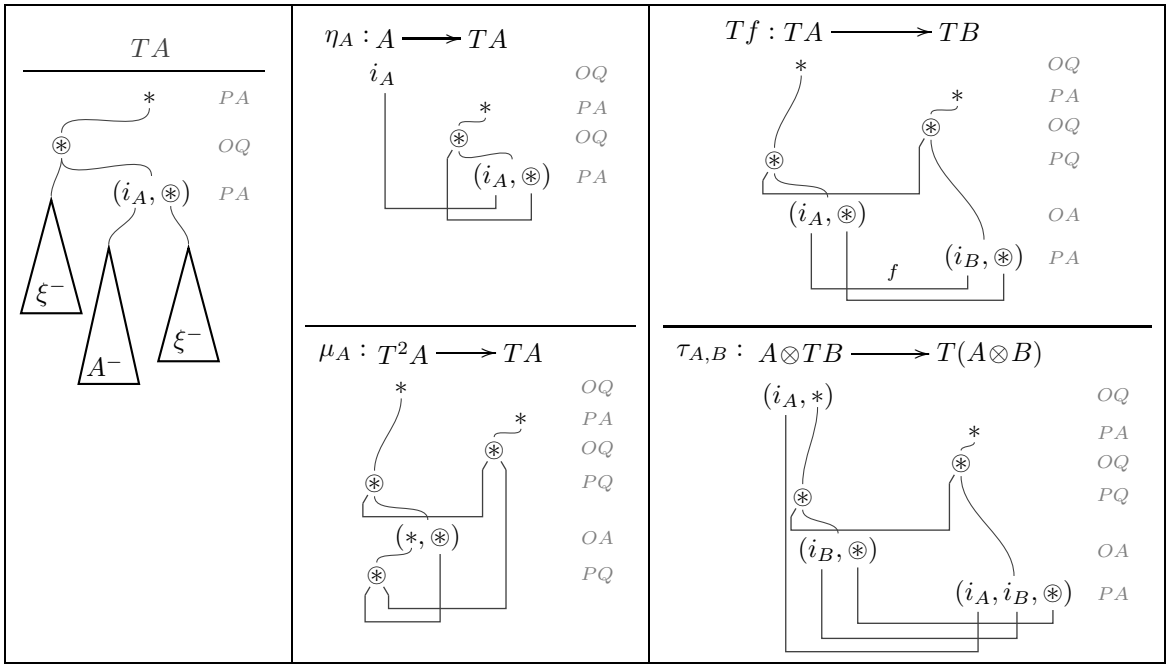


Figure 5: The store monad.

Thus, for example, μ_A is given by:

$$\mu_A = \mathtt{strat}(\ \{\, [* * \circledast \circledast s] \,|\, [\circledast \circledast s] \in \mathtt{viewf}(\mathtt{id}_\xi) \,\} \\ \cup \{\, [* * \circledast \circledast (*, \circledast') \circledast' s] \,|\, [\circledast' \circledast' s] \in \mathtt{viewf}(\mathtt{id}_\xi) \vee [s] \in \mathtt{viewf}(\mathtt{id}_{A \otimes \xi}) \,\}\ ) .$$

A consequence of lifting being a strong monad with exponentials is that the store monad is also a strong monad with exponentials. T-exponentials are given by:

$$TB^A \triangleq A \multimap TB , \qquad \Lambda^T(f : A \otimes B \to TC) \triangleq \Lambda(f) . \tag{5.2}$$

Moreover, for each arena A we can define an arrow:

$$\alpha_A \triangleq A_\perp \xrightarrow{(\eta_A)_\perp} (TA)_\perp \xrightarrow{\mathtt{pu}_{TA}} TA . \tag{5.3}$$

The transformation $\mathtt{pu}$ was introduced in (4.7). Using lemma 4.48 we obtain $\alpha_A = \Lambda(\mathtt{st}'_{A,\xi})$. Moreover, we can show that $\alpha : (\_)_\perp \to T$ is a monad morphism.

5.2. **Obtaining the $\nu\rho$-model.** Let us recapitulate the structure that 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 3.12.

I. $\mathcal{V}_\mathtt{t}$ is a category with finite products (proposition 4.39).
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 $(Q^{\bar{a}}, \varepsilon, \delta, \zeta)_{\bar{a} \in \mathbb{A}^\#}$ of product comonads, with each $Q^{\bar{a}}$ having basis $\mathbb{A}^{\bar{a}}$ (see section 4.5), which fulfils specifications (a,b). There are also fresh-name constructors, $\mathtt{new}^{\bar{a}a} : Q^{\bar{a}} \to (Q^{\bar{a}a})_\perp$, which satisfy (N2).

V. There are name-equality arrows, $\mathtt{eq}_A$ for each type A, making the (N1) diagram commute (section 4.5).

From `new` we can obtain a fresh-name transformation for the store monad.

Definition 5.6. For each $\bar{a}a \in \mathbb{A}^{\#}$, define a natural transformation $\mathtt{nu}^{\bar{a}a} : Q^{\bar{a}} \to TQ^{\bar{a}a}$ by:

$$\mathtt{nu}^{\bar{a}a}_A \triangleq Q^{\bar{a}}A \xrightarrow{\mathtt{new}_A} (Q^{\bar{a}a}A)_\perp \xrightarrow{\alpha_{Q^{\bar{a}a}A}} TQ^{\bar{a}a}A\,.$$

Moreover, for each $f : Q^{\bar{a}a}A \to TB$, take $\langle\!\langle\, a \,\rangle\!\rangle\, f \triangleq Q^{\bar{a}}A \xrightarrow{\mathtt{nu}_A} TQ^{\bar{a}a}A \xrightarrow{Tf} T^2B \xrightarrow{\mu_B} TB\,.$ ▲

Each arrow $\mathtt{nu}^{\bar{a}a}_A$ is explicitly given by (note we use the same conventions as in (4.10)):

$$\mathtt{nu}^{\bar{a}a}_A = \mathtt{strat}\{\,[(\bar{a}, i_A) * \circledast\, (\bar{a}a, i_A, \circledast)^a s^a] \mid a \# i_A \wedge ([i_A i_A s] \in \mathtt{viewf}(\mathtt{id}_A) \vee [\circledast\circledast s] \in \mathtt{viewf}(\mathtt{id}_\xi))\,\}$$

and diagrammatically as in figure 6. Moreover, using the fact that α is a monad morphism and lemma 4.48 we can show that, in fact, $\langle\!\langle\, a \,\rangle\!\rangle\, f$ is given exactly as in (4.9), that is,

$$\langle\!\langle\, a \,\rangle\!\rangle\, f = \mathtt{new}_A\,;\, f_\perp\,;\, \mathtt{pu}_{TB}\,.$$

Finally, α being is a monad morphism implies also the following.

Proposition 5.7. *The* `nu` *transformation satisfies the* (N2) *diagrams of definition 3.12.* □

What we are only missing for a $\nu\rho$-model is update and dereferencing maps.

Definition 5.8. For any type A we define the following arrows in $\mathcal{V}_{\mathtt{t}}$,

$$\begin{aligned}
\mathtt{drf}_A &\triangleq \mathtt{strat}\{\,[a * \circledast\, a\, i_{[\![A]\!]}\, (i_{[\![A]\!]}, \circledast)\, s\,] \mid [\circledast\circledast s] \in \mathtt{viewf}(\mathtt{id}_\xi) \vee [i_{[\![A]\!]} i_{[\![A]\!]} s] \in \mathtt{viewf}(\mathtt{id}_{[\![A]\!]})\,\}\,,\\
\mathtt{upd}_A &\triangleq \mathtt{strat}\big(\{\,[(a, i_{[\![A]\!]}) * \circledast\, b\, b\, s\,] \mid [\circledast\ \circledast\ b\, b\, s] \in \mathtt{viewf}(\mathtt{id}_\xi) \wedge b\#a\,\}\\
&\qquad \cup \{\,[(a, i_{[\![A]\!]}) * \circledast\, a\, i_{[\![A]\!]}\ s\,] \mid [i_{[\![A]\!]}\ i_{[\![A]\!]}\ s] \in \mathtt{viewf}(\mathtt{id}_{[\![A]\!]})\,\}\big)\,,
\end{aligned}$$

depicted also in figure 6. ▲

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 the following.

Proposition 5.9. *The* (NR) *and* (SNR) *diagrams of definition 3.12 commute.* □

We have therefore established the following.

Theorem 5.10. $(\mathcal{V}_{\mathtt{t}}, T, Q)$ *is a* $\nu\rho$*-model.* □

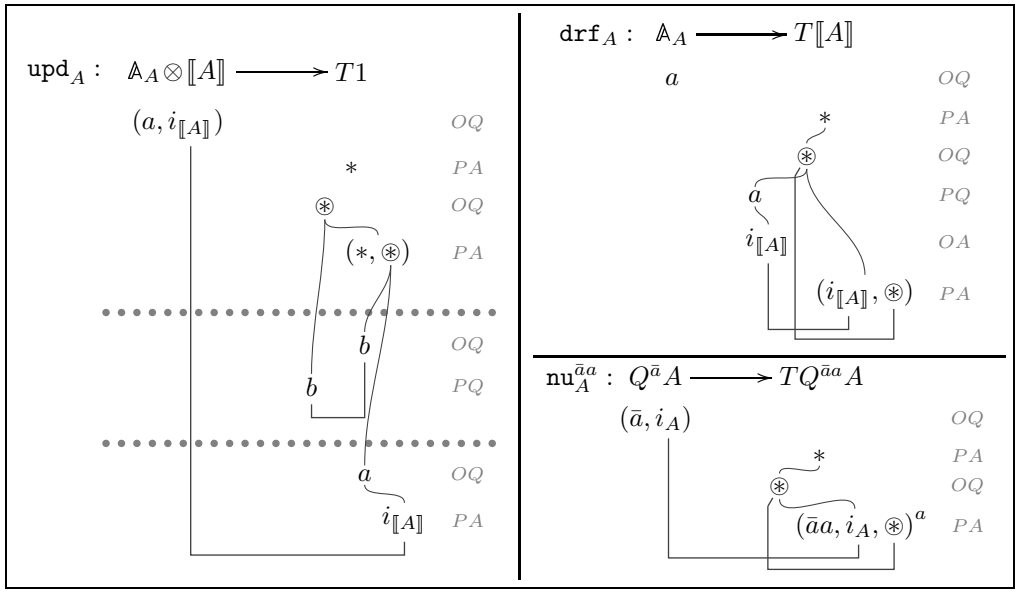


Figure 6: Strategies for update, dereferencing and fresh-name creation.

We close this section with a discussion on how the store-effect is achieved in our innocent setting, and with some examples of translations of $\nu\rho$-terms in $\mathcal{V}_{\mathtt{t}}$.

Remark 5.11 (**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 [6, 3], 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 cell 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, 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,[17]

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 7: A dialogue in innocent store.

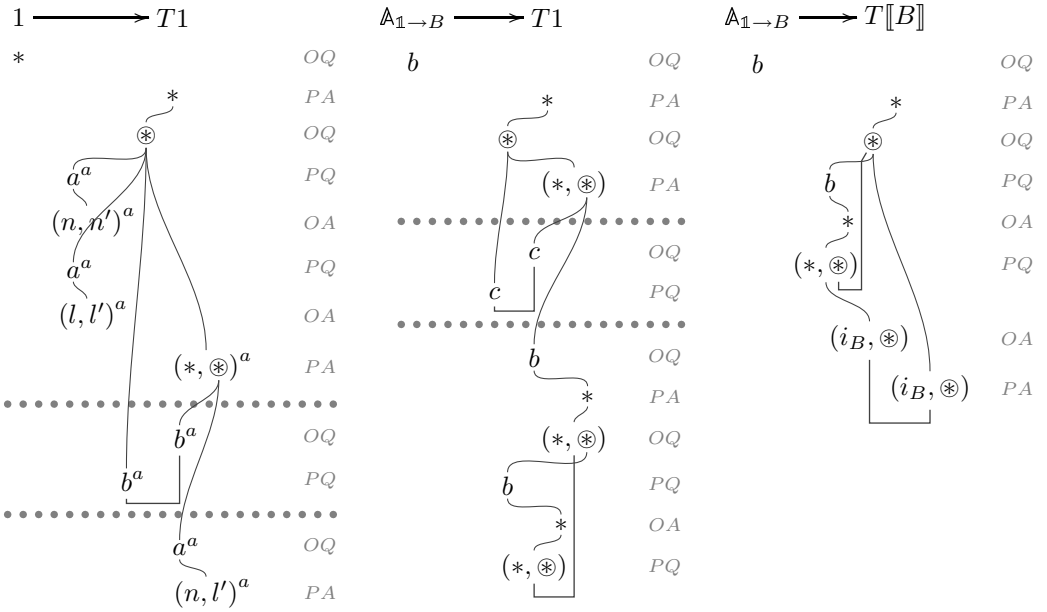
[17] i.e. at the last store-opening move played by the other participant.

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. 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.

Example 5.12. 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.



In the first example we see that, although the strategy is looking up the fresh (and therefore uninitialised) reference a, the play does not deadlock: if Opponent answered the question a^a then the play would proceed as depicted. In practice, however, Opponent will never be able to answer that question and the play will halt indeed (this is because Opponent must play *tidily*, see section 5.4). Moreover, from the latter two examples we can compute $[\![\mathtt{stop}_B]\!] : 1 \to T[\![B]\!] = \{\,[* * \circledast]\,\}$. □

5.3. **Adequacy.** We proceed to show that $\mathcal{V}_{\mathtt{t}}$ is adequate (v. definition 3.18). First we characterise non-reducing terms as follows.

Lemma 5.13. *Let $\bar{a} \mid \varnothing \vdash M : A$ be a typed term. M is a value iff there exists a store S such that $S \models M$ has no reducts and $[(\bar{a}, *) * \circledast (i_A, \circledast)^{\bar{b}}] \in [\![\bar{S}\,; M]\!]$, for some $i_A, \bar{b}$.*

Proof: The "only if"-part is straightforward. For the "if"-part assume that M is a non-value and take any S such that $S \models M$ has no reducts. We show by induction on M that there exist no $i_A, \bar{b}$ such that $[(\bar{a}, *) * \circledast (i_A, \circledast)^{\bar{b}}] \in [\![\bar{S}\,; M]\!]$. The base case follows trivially from M not being a value. Now, for the inductive step, the specifications of $S \models M$ (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 $\bar{a} \mid \varnothing \vdash K : B$ and such that $S \models K$ non-reducing.

In case of $M \equiv !\, a$, we have that $[(\bar{a}, *) * \circledast\, a] \in [\![\bar{S}\,; M]\!]$, which proves the claim because of determinacy. On the other hand, if $M \equiv \mathrm{E}[K]$ then, as in proof of proposition 3.17, we have:

$$[\![\bar{S}\,; M]\!] = \langle \Lambda(\zeta'\,; [\![\mathrm{E}[x]]\!]), [\![\bar{S}\,; K]\!] \rangle\,; \tau\,; T\mathsf{ev}\,; \mu = \langle \mathtt{id}, [\![\bar{S}\,; K]\!] \rangle\,; \tau\,; T(\zeta'\,; [\![\mathrm{E}[x]]\!])\,; \mu$$

By IH, there are no $i_B, \bar{c}$ such that $[(\bar{a}, *) * \circledast\, (i_B, \circledast)^{\bar{c}}] \in [\![\bar{S}\,; K]\!]$, which implies that there are no $i_A, \bar{b}$ such that $[(\bar{a}, *) * \circledast\, (i_A, \circledast)^{\bar{b}}] \in [\![\bar{S}\,; M]\!]$. □

Because of the previous result, in order to show adequacy it suffices to show that, whenever $[\![M]\!] = \langle\!\langle \bar{b} \rangle\!\rangle\, [\![\bar{S}\,; \tilde{0}]\!]$, there is no infinite reduction sequence starting from $\bar{a} \models M$. 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 ν-calculus — it suffices to show there is no reduction sequence starting from $\bar{a} \models M$ and containing infinitely many DRF reduction steps.
- In fact, the problem can be further reduced to showing that, whenever $[(\bar{a}, *) * \circledast\, (0, \circledast)^{\bar{b}}] \in [\![M]\!]$, there is no reduction sequence starting from $\bar{a} \models M$ and containing infinitely many NEW reduction steps. The latter clearly holds, since M cannot create more than $|\bar{b}|$ 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 $S \models M$, there is a sequence with infinitely many NEW's starting from $S \models M'$. The reduction is completed by finally showing that, whenever we have $[(\bar{a}, *) * \circledast\, (0, \circledast)^{\bar{b}}] \in [\![M]\!]$, we also have $[(\bar{a}, *) * \circledast\, (0, \circledast)^{\bar{b}'}] \in [\![M']\!]$.

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 is carried out by using the observational equivalence relation on strategies, defined later in this section. Note, though, that a direct proof can also be given (see [48]).

Proposition 5.14 (Adequacy). *$(\mathcal{V}_{\mathtt{t}}, T, Q)$ is adequate.*

Proof: This follows from O-adequacy (lemma 5.28), which is proved independently. □

Hence, $(\mathcal{V}_{\mathtt{t}}, T, Q)$ is a sound model for $\nu\rho$ and thus, for all terms M, N,

$$[\![M]\!] = [\![N]\!] \implies M \lessapprox N\,.$$

5.4. **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, $\mathcal{V}_{\mathsf{t}}$ 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 5.15. Consider $\mathcal{V}_{\nu\rho}$, the full subcategory of $\mathcal{V}_{\mathsf{t}}$ with objects given by:

$$Ob(\mathcal{V}_{\nu\rho}) \ni A, B ::= 1 \mid \mathbb{N} \mid \mathbb{A}^{\bar{a}} \mid A \otimes B \mid A \multimap TB$$

For each such arena A we define its set of ***store-Handles***, H_A, as follows.

$$H_1 = H_{\mathbb{N}} = H_{\mathbb{A}^{\bar{a}}} \triangleq \varnothing\,, \quad H_{A\otimes B} \triangleq H_A \cup H_B\,,$$

$$H_{A\multimap TB} \triangleq \{(i_A, \circledast_A), (i_B, \circledast_B)\} \cup H_A \cup H_B \cup H_{\xi_A} \cup H_{\xi_B} \quad \text{with } H_\xi \triangleq \bigcup\nolimits_C H_{[\![C]\!]}\,,$$

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 $[\![A]\!], Q^{\bar{a}}[\![A]\!], T[\![A]\!] \in Ob(\mathcal{V}_{\nu\rho})$, 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 is a definition by induction: we take $H_A \triangleq \bigcup_{i\in\omega} H^i_A$ and,

$$H^i_1 = H^i_{\mathbb{N}} = H^i_{\mathbb{A}^{\bar{a}}} = H^0_A \triangleq \varnothing\,, \quad H^i_{A\otimes B} \triangleq H^i_A \cup H^i_B\,,$$

$$H^{i+1}_{A\multimap TB} \triangleq \{(i_A, \circledast_A), (i_B, \circledast_B)\} \cup H^i_A \cup H^i_B \cup H^{i+1}_{\xi_A} \cup H^{i+1}_{\xi_B} \quad \text{with } H^{i+1}_\xi \triangleq \bigcup\nolimits_C H^i_{[\![C]\!]}\,.$$

Intuitively, store-H's are store-opening moves, while store-Q's and store-A's are obtained from unfolding the store structure. On the side we give examples of store-related moves in a simple arena.

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 an element of the aforedefined move-classes.

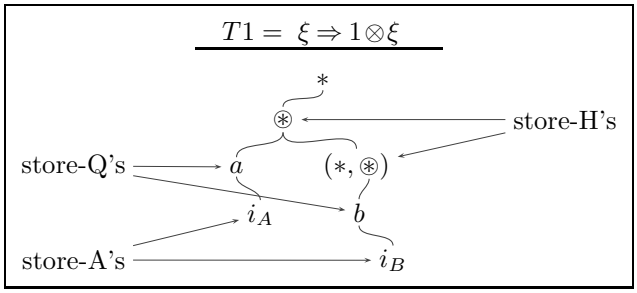


Figure 8: Store-H's -Q's -A's in arena $T1$.

Proposition 5.16. *For any $A \in Ob(\mathcal{V}_{\nu\rho})$,*

$$M_A = I_A \uplus H_A \uplus \{\, m \in M_A \,|\, m \text{ a store-Q}\} \uplus \{m \in M_A \,|\, m \text{ a store-A}\,\}.$$

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 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 5.17. 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:

$$I_A \uplus I_B \uplus H_{A\to B} \uplus \{\, m \in M_{A\to B} \,|\, m \text{ a store-Q}\,\} \uplus \{\, m \in M_{A\to B} \,|\, m \text{ a store-A}\,\} \qquad (5.4)$$

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 5.18 (**Tidy strategies**)**.** 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 $q = a^{\bar{a}}$ 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 $[sy_{(P)}] \in \sigma$, where $y_{(P)}$ is justified by $\ulcorner s \urcorner .\text{-}3$. ▲

(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 5.11). (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.

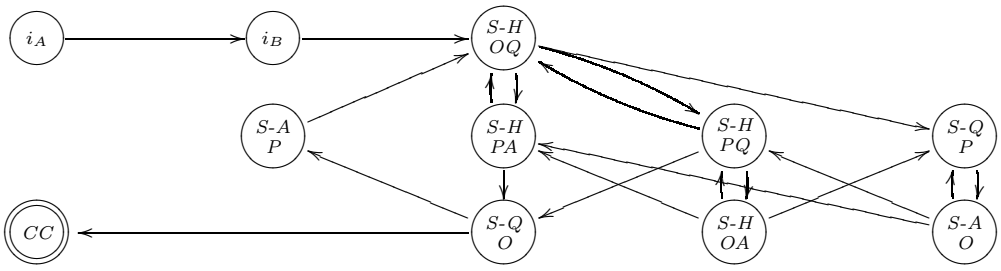
The following straightforward result shows that (TD3), as stated, provides the intended copycat behaviour.

Proposition 5.19. *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 5.20 (**Tidy Discipline**). *Let $\sigma : A \to B$ 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 5.21 (**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.* □

Observe that strategies that mostly do copycats are tidy; in particular, identities are tidy. Moreover, tidy strategies are closed under composition (proof delegated to the appendix).

Proposition 5.22. *If $\sigma : A \to B$ and $\tau : B \to C$ are tidy strategies then so is $\sigma\,;\tau$.* □

Definition 5.23. $\mathcal{T}$ is the lluf subcategory of $\mathcal{V}_{\nu\rho}$ of tidy strategies. ▲

Finally, we need to check that all structure required for a sound $\nu\rho$-model pass from $\mathcal{V}_{\mathtt{t}}$ to $\mathcal{T}$. It is not difficult to see that all such structure which does not handle the store remains safely within the tidy universe. On the other hand, strategies for update and dereferencing are tidy by construction. (A fully formal proof is given in [48].)

Proposition 5.24. *$\mathcal{T}$ forms an adequate $\nu\rho$-model by inheriting all the necessary structure from $\mathcal{V}_{\mathtt{t}}$.* □

Henceforth, by strategies we shall mean tidy strategies, unless stated otherwise.

5.5. **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. In fact, it is not only because of the treatment of name-creation that the semantics is not complete. Terms like

$$a := 1\,;\lambda x.!\,a\,;2 \;\approxeq\; a := 1\,;\lambda x.2$$

are distinguished because of the 'explicit' way in which the store works.

So there are many ways in which our semantics is too expressive for our language. We therefore proceed to a quotienting by the intrinsic preorder and prove full-abstraction in the extensional model. Following the steps described in section 3.2, 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 5.25. Expand $\mathcal{T}$ to $(\mathcal{T}, T, Q, O)$ by setting, for each $\bar{a} \in \mathbb{A}^{\#}$,

$$O^{\bar{a}} \;\triangleq\; \{\, f \in \mathcal{T}(Q^{\bar{a}}1, T\mathbb{N}) \mid \exists \bar{b}.\, [(\bar{a}, *) \,*\, \circledast\, (0, \circledast)^{\bar{b}}] \in f \,\}\,.$$

Then, for each $f, g \in \mathcal{T}(Q^{\bar{a}}A, TB)$, $f \lesssim^{\bar{a}} g$ if

$$\forall \rho : Q^{\bar{a}}(A \multimap TB) \to T\mathbb{N}.\;\; (\Lambda^{\bar{a}}(f)\,;\rho \in O^{\bar{a}} \implies \Lambda^{\bar{a}}(g)\,;\rho \in O^{\bar{a}})\,. \qquad ▲$$

Thus, the observability predicate O is a family $(O^{\bar{a}})_{\bar{a}\in\mathbb{A}^{\#}}$, and the intrinsic preorder $\lesssim$ is a family $(\lesssim^{\bar{a}})_{\bar{a}\in\mathbb{A}^{\#}}$. Recall that by $\Lambda^{\bar{a}}(f)$ we mean $\Lambda^{Q^{\bar{a}},T}(f)$, that is,

$$\Lambda^{\bar{a}}(f) = Q^{\bar{a}}1 \xrightarrow{\delta} Q^{\bar{a}}Q^{\bar{a}}1 \xrightarrow{Q^{\bar{a}}\Lambda(\zeta'\,;f)} Q^{\bar{a}}(A \multimap TB)\,.$$

Note in particular that $f \sqsubseteq g$ implies $\Lambda^{\bar{a}}(f)\,;\rho \sqsubseteq \Lambda^{\bar{a}}(g)\,;\rho$, for any relevant ρ, and therefore:

$$f \sqsubseteq g \implies f \lesssim^{\bar{a}} g \tag{5.5}$$

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 5.26 (tl4 tests suffice). *Let $f,g \in \mathcal{T}(Q^{\bar a}1, B)$ with B pointed. The following are equivalent (recall definition 4.35).*

I. $\forall \rho : Q^{\bar a}B \to T\mathbb{N}.\ \ \delta\,;Q^{\bar a}f\,;\rho \in O^{\bar a} \implies \delta\,;Q^{\bar a}g\,;\rho \in O^{\bar a}$

II. $\forall \rho : Q^{\bar a}B \to T\mathbb{N}.\ \ \rho$ *is tl4* $\implies (\delta\,;Q^{\bar a}f\,;\rho \in O^{\bar a} \implies \delta\,;Q^{\bar a}g\,;\rho \in O^{\bar a})$

Hence, for each $\bar a$ and $f,g \in \mathcal{T}(Q^{\bar a}A, TB)$, $f \lesssim^{\bar a} g$ iff

$$\forall \rho : Q^{\bar a}(A \multimap TB) \to T\mathbb{N}.\ \ \rho \text{ is tl4} \implies (\Lambda^{\bar a}(f)\,;\rho \in O^{\bar a} \implies \Lambda^{\bar a}(g)\,;\rho \in O^{\bar a})\,.$$

Proof: I $\Rightarrow$ II is trivial. Now assume II holds and let $\rho : Q^{\bar a}B \to T\mathbb{N}$ be any strategy such that $\delta\,;Q^{\bar a}f\,;\rho \in O^{\bar a}$. Then, there exist $[s] \in \delta\,;Q^{\bar a}f$ and $[t] \in \rho$ such that $[s\,;t] = [(\bar a,*) * \circledast\,(0,\circledast)^{\bar b}] \in (\delta\,;Q^{\bar a}f)\,;\rho$. We show by induction on the number of J_B-moves appearing in $s \parallel t$ that $\delta\,;Q^{\bar a}g\,;\rho \in O^{\bar a}$.

If no such moves appear then $t = (\bar a, i_B) * \circledast\,(0,\circledast)^{\bar b}$, so done. If $n+1$ such moves appear then ρ is necessarily t4, as B is pointed, so by lemma 4.36 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, $\delta\,;Q^{\bar a}f\,;\rho = \delta\,;Q^{\bar a}f\,;\Delta\,;\tilde\rho = \delta\,;Q^{\bar a}f\,;\langle \mathtt{id}, Q^{\bar a}!\,;\delta\,;Q^{\bar a}f\rangle\,;\tilde\rho = \delta\,;Q^{\bar a}f\,;\rho'$, with ρ' being $\langle \mathtt{id}, Q^{\bar a}!\,;\delta\,;Q^{\bar a}f\rangle\,;\tilde\rho$. Now, by definition of $\tilde\rho$, $[(\bar a,*) * \circledast\,(0,\circledast)^{\bar b}] = [s'\,;t'] \in \delta\,;Q^{\bar a}f\,;\rho'$ with $s' \parallel t'$ containing n J_B-moves so, by IH, $\delta\,;Q^{\bar a}g\,;\rho' \in O^{\bar a}$. But $\delta\,;Q^{\bar a}g\,;\rho' = \delta\,;Q^{\bar a}g\,;\langle \mathtt{id}, Q^{\bar a}!\,;\delta\,;Q^{\bar a}f\rangle\,;\tilde\rho = \delta\,;Q^{\bar a}f\,;\langle Q^{\bar a}!\,;\delta\,;Q^{\bar a}g, \mathtt{id}\rangle\,;\tilde\rho = \delta\,;Q^{\bar a}f\,;\rho''$, where ρ'' is given by $\langle Q^{\bar a}!\,;\delta\,;Q^{\bar a}g, \mathtt{id}\rangle\,;\tilde\rho$. But ρ'' is tl4, thus, by hypothesis, $O^{\bar a} \ni \delta\,;Q^{\bar a}g\,;\rho'' = \delta\,;Q^{\bar a}g\,;\rho$, as required. □

We can now prove the second half of observationality.

Lemma 5.27. *For any morphism $f : Q^{\bar a a}1 \to B$, with B pointed, and any tl4 morphism $\rho : Q^{\bar a}B \to T\mathbb{N}$,*

$$\delta\,;Q^{\bar a}\langle\!|\, a \,|\!\rangle\, f\,;\rho \in O^{\bar a} \iff \delta\,;Q^{\bar a a}f\,;\tfrac{\bar a a}{\bar a}\,;\rho \in O^{\bar a a}$$

Moreover, for each $\bar a$ and relevant $a, \bar a', f, g$,

$$f \lesssim^{\bar a a} g \implies \langle\!|\, a \,|\!\rangle\, f \lesssim^{\bar a} \langle\!|\, a \,|\!\rangle\, g\,, \quad f \lesssim^{\bar a} g \implies \tfrac{\bar a'}{\bar a}\,;f \lesssim^{\bar a'} \tfrac{\bar a'}{\bar a}\,;g\,.$$

Proof: For the first part, ρ being tl4 and B being pointed imply that there exists some $\bar b \# \bar a$ and a ttotal strategy ρ' such that $\rho = \langle\!|\, \bar b \,|\!\rangle\, \rho'$. Now let $\delta\,;Q^{\bar a}\langle\!|\, a \,|\!\rangle\, f\,;\rho \in O^{\bar a}$, so there exists $[s\,;t] = [(\bar a,*) * \circledast\,(0,\circledast)^{\bar b a \bar c}] \in (\delta\,;Q^{\bar a}\langle\!|\, a \,|\!\rangle\, f)\,;\rho$, and let $s = (\bar a,*)\,(\bar a, i_B)\,j_B\,m^{a\bar d}s'$ and $t = (\bar a, i_B) * \circledast\, j_B^{\bar b}\,t'$. Letting $s^{\smallsetminus a}$ be $\underline{s}^{\mathsf{nlist}(s)\smallsetminus a}$, we can see that $[(\bar a a,*)\,i_B\,j_B\,m^{\bar d}s'^{\smallsetminus a}] \in f$ and thus $[s''] \triangleq [(\bar a a,*)\,(\bar a, i_B)\,j_B\,m^{\bar d}s'^{\smallsetminus a}] \in \delta\,;Q^{\bar a a}f\,;\frac{\bar a a}{\bar a}$. Hence, $[s''\,;t] = [(\bar a a,*) * \circledast\,(0,\circledast)^{\bar b \bar c}] \in \delta\,;Q^{\bar a a}f\,;\frac{\bar a a}{\bar a}\,;\rho$, as required. The converse is shown similarly.

For the second part, suppose $f \lesssim^{\bar a a} g : Q^{\bar a a}A \to TB$ and take any tl4 morphism $\rho : Q^{\bar a}(A \multimap TB) \to T\mathbb{N}$. Then,

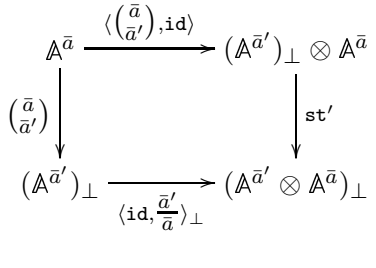
$$\begin{aligned}
\Lambda^{\bar a}(\langle\!|\, a \,|\!\rangle\, f)\,;\rho \in O^{\bar a} &\iff \delta\,;Q^{\bar a}\Lambda(\zeta'\,;\langle\!|\, a \,|\!\rangle\, f)\,;\rho \in O^{\bar a} \overset{\text{lem 4.53}}{\iff} \delta\,;Q^{\bar a}\langle\!|\, a \,|\!\rangle\, (\Lambda(\zeta'\,;f))\,;\rho \in O^{\bar a} \\
&\iff \delta\,;Q^{\bar a a}\Lambda(\zeta'\,;f)\,;\tfrac{\bar a a}{\bar a}\,;\rho \in O^{\bar a a} \\
&\overset{f \lesssim^{\bar a a} g}{\implies} \delta\,;Q^{\bar a a}\Lambda(\zeta'\,;g)\,;\tfrac{\bar a a}{\bar a}\,;\rho \in O^{\bar a a} \iff \Lambda^{\bar a}(\langle\!|\, a \,|\!\rangle\, g)\,;\rho \in O^{\bar a}\,.
\end{aligned}$$

For the other claim, let us generalise the fresh-name constructors `new` to:

$$\binom{\bar a}{\bar a'} : \mathbb{A}^{\bar a} \to (\mathbb{A}^{\bar a'})_\perp \triangleq \{\,[\,(\bar a,*) * *\,(\bar a',*)^{\bar a' \smallsetminus \bar a}]\,\}$$

for any $\mathtt{S}(\bar{a}) \subseteq \mathtt{S}(\bar{a}')$. The above yields a natural transformation of type $Q^{\bar{a}} \to Q^{\bar{a}'}_{\perp}$. It is easy to see that, for any $h : Q^{\bar{a}'}1 \to T\mathbb{N}$, $h \in O^{\bar{a}'}$ iff $\binom{\bar{a}}{\bar{a}'} ; h_\perp ; \mathtt{pu} \in O^{\bar{a}}$ and, moreover, that the diagram on the right below commutes. Hence, if $f \lesssim^{\bar{a}} g$ then

$$
\begin{aligned}
&\delta ; Q^{\bar{a}'}\Lambda(\zeta' ; \tfrac{\bar{a}'}{\bar{a}} ; f) ; \rho \in O^{\bar{a}'} \\
&\quad\iff \delta ; Q^{\bar{a}'}\tfrac{\bar{a}'}{\bar{a}} ; Q^{\bar{a}'}\Lambda(\zeta' ; f) ; \rho \in O^{\bar{a}'} \\
&\quad\iff \tbinom{\bar{a}}{\bar{a}'} ; (\delta ; Q^{\bar{a}'}\tfrac{\bar{a}'}{\bar{a}} ; Q^{\bar{a}'}\Lambda(\zeta' ; f) ; \rho)_\perp ; \mathtt{pu} \in O^{\bar{a}} \\
&\quad\iff \delta ; Q^{\bar{a}}\Lambda(\zeta' ; f) ; \tbinom{\bar{a}}{\bar{a}'} ; \rho_\perp ; \mathtt{pu} \in O^{\bar{a}} \\
&\quad\overset{f \lesssim^{\bar{a}} g}{\Longrightarrow} \delta ; Q^{\bar{a}}\Lambda(\zeta' ; g) ; \tbinom{\bar{a}}{\bar{a}'} ; \rho_\perp ; \mathtt{pu} \in O^{\bar{a}} \\
&\quad\iff \delta ; Q^{\bar{a}'}\Lambda(\zeta' ; \tfrac{\bar{a}'}{\bar{a}} ; g) ; \rho \in O^{\bar{a}'},
\end{aligned}
$$



as required. □

In order to prove that $\mathcal{T}$ is observational, we are only left to show that

$$[\![M]\!] \in O^{\bar{a}} \iff \exists \bar{b}, S. [\![M]\!] = \langle\!| \bar{b} |\!\rangle [\![\bar{S} ; 0]\!]$$

for any $\bar{a} \mid \varnothing \vdash M : \mathbb{N}$. The "$\Longleftarrow$" direction is trivial. For the converse, because of correctness, it suffices to show the following generalisation of adequacy.

Lemma 5.28 (**O-Adequacy**). *Let $\bar{a} \mid \varnothing \vdash M : \mathbb{N}$ be a typed term. If $[\![M]\!] \in O^{\epsilon}$ then there exists some S such that $\bar{a} \models M \longrightarrow\!\!\!\!\!\rightarrow S \models 0$.*

Proof: The idea behind the proof is given above proposition 5.14. It suffices to show that, for any such M, there is a non-reducing sequent $S \models N$ such that $\bar{a} \models M \longrightarrow\!\!\!\!\!\rightarrow S \models N$; therefore, because of Strong Normalisation in the ν-calculus, it suffices to show that there is no infinite reduction sequence starting from $\bar{a} \models M$ 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, $\bar{a} \models (M)^\circ$ 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 $a \# N$.

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 $[\![(M)^\circ]\!] = \langle\!| a |\!\rangle (\frac{\bar{a}a}{\bar{a}} ; [\![!(N)^\circ]\!])$, while the IH implies that $[\![M]\!] \backsimeq [\![!(N)^\circ]\!]$. Hence, it sts that for each $f : Q^{\bar{a}}A \to TB$ we have $f \backsimeq \langle\!| a |\!\rangle (\frac{\bar{a}a}{\bar{a}} ; f)$. Indeed, for any relevant ρ which is tl4,

$$
\begin{aligned}
\Lambda^{\bar{a}}(\langle\!| a |\!\rangle (\tfrac{\bar{a}a}{\bar{a}} ; f)) ; \rho \in O^{\bar{a}} &\overset{\text{lem 5.27}}{\iff} \delta ; Q^{\bar{a}a}\Lambda(\zeta' ; \tfrac{\bar{a}a}{\bar{a}} ; f) ; \tfrac{\bar{a}a}{\bar{a}} ; \rho \in O^{\bar{a}a} \\
&\iff \delta ; Q^{\bar{a}a}\tfrac{\bar{a}a}{\bar{a}} ; \tfrac{\bar{a}a}{\bar{a}} ; Q^{\bar{a}}\Lambda(\zeta' ; f) ; \rho \in O^{\bar{a}a} \\
&\iff \tfrac{\bar{a}a}{\bar{a}} ; \Lambda^{\bar{a}}(f) ; \rho \in O^{\bar{a}a} \iff \Lambda^{\bar{a}}(f) ; \rho \in O^{\bar{a}}.
\end{aligned}
$$

Now, take any $\bar{a} \mid \varnothing \vdash M : \mathbb{N}$ and assume $[\![M]\!] \in O^{\bar{a}}$, and that $\bar{a} \models M$ diverges using infinitely many DRF reduction steps. Then, $\bar{a} \models (M)^\circ$ diverges using infinitely many NEW reduction steps. However, since $[\![(M)^\circ]\!] \backsimeq [\![M]\!]$, we have $[\![(M)^\circ]\!] \in O^{\bar{a}}$ and therefore

$[(\bar{a},*) * \circledast (\tilde{0},\circledast)^{\bar{b}}] \in [\![(M)^\circ]\!]$ for some $\bar{b}$. However, $\bar{a} \models (M)^\circ$ reduces to some $S \models M'$ using $|\bar{b}|+1$ NEW reduction steps, so $[\![(M)^\circ]\!] = \langle\!\langle\, \bar{c} \,\rangle\!\rangle\, [\![\bar{S}\,; M']\!]$ with $|\bar{c}| = |\bar{b}|+1$, $\lightning$ to determinacy. □

We have therefore shown observationality.

Proposition 5.29 (**Observationality**). *$(\mathcal{T}, T, Q, O)$ is observational.* □

5.6. **Definability and full-abstraction.** We now proceed to show definability for $\mathcal{T}$, and through it ip-definability. According to the results of section 3.2.3, this will suffice for full abstraction.

We first make precise the notion of *finitary strategy*, that is, of (tidy) strategy with finite description, by introducing truncation functions that remove inessential branches from a strategy's description.

Definition 5.30. Let $\sigma : A \to B$ 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

$$\mathtt{trunc}(\sigma) \triangleq \{\, [\mathtt{trunc}(s)] \,|\, [s] \in \mathtt{viewf}(\sigma) \wedge |s| > 3 \,\}.$$

Finally, for any $[t] \in \sigma$ define:

$$\sigma_{\leq t} \triangleq \mathtt{strat}\{\, [s] \in \mathtt{viewf}(\sigma) \,|\, \exists\, t' \leq t.\ \mathtt{trunc}'(s) = \ulcorner t' \urcorner \,\}. \qquad \blacktriangle$$

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 (proof in [48]).

Proposition 5.31. *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 $\sigma_{\leq t} \sqsubseteq \sigma$.* □

We proceed to show definability. The proof is facilitated by the following lemma, the proof of which is delegated to the appendix. 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 $s^{\setminus \bar{b}}$ for s with $\bar{b}$ removed from all of its name-lists.

Lemma 5.32 (**Decomposition Lemma**). *Let $\sigma : Q^{\bar{a}}[\![A]\!] \to T[\![B]\!]$ be a strategy. We can decompose σ as follows.*

1. *If there exists an $i_{A(0)}$ such that $\exists x_0.\, [(\bar{a}, i_{A(0)}) * \circledast x_0] \in \sigma$ then*

$$\begin{array}{ccc} Q^{\bar{a}}[\![A]\!] & & \\ {\scriptstyle\sigma}\downarrow & \searrow^{\langle [x \stackrel{\bar{a}}{=} i_{A(0)}], \langle \sigma_0, \sigma' \rangle \rangle} & \\ T[\![B]\!] & \xleftarrow[\mathtt{cnd}]{} & \mathbb{N} \otimes (T[\![B]\!])^2 \end{array}$$

where:

$$\begin{aligned} [x \stackrel{\bar{a}}{=} i_{A(0)}] : Q^{\bar{a}}[\![A]\!] \to \mathbb{N} &\triangleq \{\, [(\bar{a}, i_{A(0)})\, 0\,] \} \cup \{\, [\,(\bar{a}, i_A)\, 1\,] \mid [\,(\bar{a}, i_A)\,] \neq [\,(\bar{a}, i_{A(0)})\,] \,\}, \\ \sigma_0 : Q^{\bar{a}}[\![A]\!] \to T[\![B]\!] &\triangleq \mathtt{strat}\{\, [\,(\bar{a}, i_{A(0)})\, s\,] \in \mathtt{viewf}(\sigma) \,\}, \\ \sigma' : Q^{\bar{a}}[\![A]\!] \to T[\![B]\!] &\triangleq \mathtt{strat}\{\, [\,(\bar{a}, i_A)\, s\,] \in \mathtt{viewf}(\sigma) \mid [\,(\bar{a}, i_A)\,] \neq [\,(\bar{a}, i_{A(0)})\,] \,\}. \end{aligned}$$

2. *If there exists $i_{A(0)}$ such that $\forall i_A.\, (\exists x_0.\, [(\bar{a}, i_A) * \circledast x_0] \in \sigma) \iff [(\bar{a}, i_A)] = [(\bar{a}, i_{A(0)})]$, then $\sigma = \langle\!\langle \bar{b} \rangle\!\rangle\, \sigma_{\bar{b}}$ where:*

$$\begin{aligned} \sigma_{\bar{b}} : Q^{\bar{a}\bar{b}}[\![A]\!] \to T[\![B]\!] \triangleq \mathtt{strat}\{\, [\,(\bar{a}\bar{b}, i_{A(0)}) * \circledast m_0\, s^{\setminus \bar{b}}\,] \mid \\ [\,(\bar{a}, i_{A(0)}) * \circledast m_0^{\bar{b}}\, s\,] \in \mathtt{viewf}(\sigma) \,\}. \end{aligned}$$

3. *If there exist $i_{A(0)}, m_0$ such that $\forall i_A, x.\, [(\bar{a}, i_A) * \circledast x] \in \sigma \iff [(\bar{a}, i_A)\, x] = [(\bar{a}, i_{A(0)})\, m_0]$, then one of the following is the case.*

(a) *$m_0 = a$, a store-Q of type C under $\circledast$, in which case $\sigma = \sigma' \upharpoonright (\bar{a}, i_{A(0)})$ where*

$$\begin{aligned} \sigma' : Q^{\bar{a}}[\![A]\!] \to T[\![B]\!] &\triangleq \langle \mathtt{id}, \phi \rangle ; \tau ; T\zeta' ; T\sigma_a ; \mu \\ \sigma_a : Q^{\bar{a}}([\![A]\!] \otimes [\![C]\!]) \to T[\![B]\!] &\triangleq \mathtt{strat}\{\, [\,(\bar{a}, i_{A(0)}, i_C) * \circledast s\,] \mid \\ &\qquad [\,(\bar{a}, i_{A(0)}) * \circledast a\, i_C\, s\,] \in \mathtt{viewf}(\sigma) \,\}, \\ \phi : Q^{\bar{a}}[\![A]\!] \to T[\![C]\!] &\triangleq \begin{cases} Q^{\bar{a}}!\,;\, \frac{\bar{a}}{a}\,;\, \mathtt{drf}_C & , \text{ if } a \in \mathtt{S}(\bar{a}) \\ Q^{\bar{a}}\pi_j\,;\, \frac{\bar{a}}{\epsilon}\,;\, \mathtt{drf}_C & , \text{ if } a \# \bar{a}\,. \end{cases} \end{aligned}$$

(b) *$m_0 = j_A \vee m_0 = (i_B, \circledast)$, a store-$H$, in which case if $[\,(\bar{a}, i_{A(0)}) * \circledast m_0\, a\, i_C\,] \in \sigma$, for some store-$Q$ a and store-A i_C, then*

$$\begin{array}{ccc} Q^{\bar{a}}[\![A]\!] & \xrightarrow{\langle \Delta, \sigma_a \rangle} & Q^{\bar{a}}[\![A]\!] \otimes Q^{\bar{a}}[\![A]\!] \otimes T[\![C]\!] \\ {\scriptstyle\sigma}\downarrow & & \downarrow{\scriptstyle \tau ; T(\mathtt{id} \otimes \phi ; \tau) ; \mu} \\ T[\![B]\!] & \xleftarrow[T\sigma' ; \mu]{} & TQ^{\bar{a}}[\![A]\!] \end{array}$$

where:

$$\begin{aligned}
\sigma_a : Q^{\bar a}[\![A]\!] \to T[\![C]\!] \;\triangleq\; \mathtt{strat}\{\,[\,(\bar a, i_{A(0)}) \ast \circledast\,(i_C, \circledast)\,s\,] \mid{}& \\
[\,(\bar a, i_{A(0)}) \ast \circledast\, m_0\, a\, i_C\, s\,] \in \mathtt{viewf}(\sigma)& \\
\vee\, [\,\circledast\ \circledast\ s\,] \in \mathtt{viewf}(\mathtt{id}_\xi)\,\}\,,& \\
\sigma' : Q^{\bar a}[\![A]\!] \to T[\![B]\!] \;\triangleq\; \mathtt{strat}(\,\{\,[\,(\bar a, i_{A(0)}) \ast \circledast\, m_0\, y\, s\,] \in \mathtt{viewf}(\sigma) \mid y \neq a\,\}& \\
\cup\,\{\,[\,(\bar a, i_{A(0)}) \ast \circledast\, m_0\, a\, s\,] \mid{}& \\
[\,\circledast\ \circledast\ a\, s\,] \in \mathtt{viewf}(\mathtt{id}_\xi)\}\,)\,,& \\
\phi : Q^{\bar a}[\![A]\!] \otimes [\![C]\!] \to T1 \;\triangleq\; \begin{cases} (Q^{\bar a}!\,;\tfrac{\bar a}{a}) \otimes \mathtt{id}_{[\![C]\!]}\,;\mathtt{upd}_C & ,\ \textit{if } a \in \mathtt{S}(\bar a) \\ (Q^{\bar a}\pi_j\,;\tfrac{\bar a}{\epsilon}) \otimes \mathtt{id}_{[\![C]\!]}\,;\mathtt{upd}_C & ,\ \textit{if } a \,\#\, \bar a\,. \end{cases}
\end{aligned}$$

In both cases above, we take $j = \min\{\, j \mid (i_{A(0)})_j = a\,\}$. □

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 [15] — itself based on previous proofs of definability.

Theorem 5.33 (**Definability**). *Let* A, B *be types and* $\sigma : Q^{\bar a}[\![A]\!] \to T[\![B]\!]$ *be finitary. Then* σ *is definable.*

Proof: We do induction on $(|\mathtt{trunc}(\sigma)|, \|\sigma\|)$, where we let $\|\sigma\| \triangleq \max\{\,|\mathcal{L}(s)| \mid [s] \in \mathtt{viewf}(\sigma)\,\}$, i.e. the maximum number of names introduced in any play of $\mathtt{trunc}(\sigma)$. If $|\mathtt{trunc}(\sigma)| = 0$ then $\sigma = [\![\mathtt{stop}_B]\!]$; otherwise, there exist $x_0, i_{A(0)}$ such that $[(\bar a, i_{A(0)}) \ast \circledast\, x_0] \in \sigma$. By Decomposition Lemma,

$$\sigma = \langle [x \stackrel{\bar a}{=} i_{A(0)}], \langle \sigma_0, \sigma' \rangle \rangle ; \mathtt{cnd}$$

with $|\mathtt{trunc}(\sigma')| < |\mathtt{trunc}(\sigma)|$ and $(0,0) < (|\mathtt{trunc}(\sigma_0)|, \|\sigma_0\|) \leq (|\mathtt{trunc}(\sigma)|, \|\sigma\|)$, so by IH there exists term M' such that $[\![M']\!] = \sigma'$. Hence, if there exist terms M_0, N_0 with $[\![M_0]\!] \upharpoonright (\bar a, i_{A(0)}) = \sigma_0$ and $[\![N_0]\!] = [x \stackrel{\bar a}{=} i_{A(0)}]; \eta$, then we can see that

$$\sigma = [\![\mathtt{if0}\ N_0\ \mathtt{then}\ M_0\ \mathtt{else}\ M']\!]\,.$$

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 without loss of generality 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 $\bar z, \bar z', \bar z'', \bar z'''$ 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 $i_{A(0)}$,

$$\phi_0' \triangleq \kappa_1', ..., \kappa_{k'}', \text{ with each } \kappa_i' \triangleq \begin{cases} (i_{A(0)})_{k+i} & , \text{if } (i_{A(0)})_{k+i} \in \mathtt{S}(\bar a) \\ z_j' & , \text{if } (i_{A(0)})_{k+i} \,\#\, \bar a \\ & \quad \wedge\, j = \min\{\, j < i \mid (i_{A(0)})_{k+i} = (i_{A(0)})_{k+j}\,\} \\ \mathtt{fresh}(i) & , \text{otherwise}. \end{cases}$$

$\mathtt{fresh}(i)$ 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

$$\begin{aligned} N_0 &\triangleq [\langle \bar{z}, \bar{z}' \rangle = \langle \phi_0, \phi_0' \rangle] \quad \text{where:} \\ [\langle \bar{z}, \bar{z}' \rangle = \langle \vec{\kappa}, \vec{\kappa}' \rangle] &\triangleq [z_1 = \kappa_1] \wedge \cdots \wedge [z_k = \kappa_k] \wedge [z_1' = \kappa_1'] \wedge \cdots \wedge [z_{k'}' = \kappa_{k'}'] \,, \\ [z' = \mathtt{fresh}(i)] &\triangleq [z' \neq a_1] \wedge \cdots \wedge [z' \neq a_{|\bar{a}|}] \wedge [z' \neq z_1'] \wedge \cdots \wedge [z' \neq z_{i-1}'] \,, \end{aligned}$$

with the logical connectives $\wedge$ and $\neg$ defined using $\mathtt{if0}$'s, and $[z_i = \kappa_i]$ using $\mathtt{pred}$'s, in the standard way. It is not difficult to show that indeed $[\![N_0]\!] \stackrel{\bar{a}}{=} [x = i_{A(0)}]; \eta$.

We proceed to find M_0 . By second part of Decomposition Lemma, $\sigma_0 = \langle\!| \bar{b} |\!\rangle \, \sigma_{\bar{b}}$ with $\bar{b} = \mathsf{nlist}(x_0)$, $|\mathtt{trunc}(\sigma_{\bar{b}})| = |\mathtt{trunc}(\sigma_0)|$ and $\|\sigma_{\bar{b}}\| = \|\sigma_0\| - |\bar{b}|$. If $|\bar{b}| > 0$ then, by IH, there exists term $M_{\bar{b}}$ such that $[\![M_{\bar{b}}]\!] = \sigma_{\bar{b}}$, so taking

$$M_0 \triangleq \nu \bar{b}. M_{\bar{b}}$$

we have $\sigma_0 = [\![M_0]\!]$.

Assume now $|\bar{b}| = 0$, 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

$$\sigma_0 = (\langle \mathtt{id}, \phi \rangle ; \tau ; T\zeta' ; T\sigma_a ; \mu) \upharpoonright (\bar{a}, i_{A(0)})$$

with $\mathtt{trunc}(\sigma_a) < \mathtt{trunc}(\sigma_0)$. Then, by IH, there exists $\bar{a} \mid \Gamma, y : C \vdash M_a : B$ such that $\sigma_a = [\![M_a]\!]$, and taking

$$M_0 \triangleq \begin{cases} (\lambda y. M_a)(!\, a) & \text{, if } a \in \mathtt{S}(\bar{a}) \\ (\lambda y. M_a)(!\, z_j') & \text{, if } a \# \bar{a} \wedge j = \min\{\, j \mid a = (i_{A(0)})_{k+j} \,\} \end{cases}$$

we have $\sigma_0 = [\![M_0]\!] \upharpoonright (\bar{a}, i_{A(0)})$.

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 $\bar{a} \mid \Gamma \vdash M_a : C$ and $\bar{a} \mid \Gamma \vdash M' : B$ such that $\sigma_a = [\![M_a]\!]$ and $\sigma' = [\![M']\!]$. Taking

$$M_0 \triangleq \begin{cases} (a := M_a); M' & \text{, if } a \in \mathtt{S}(\bar{a}) \\ (z_j' := M_a); M' & \text{, if } a \# \bar{a} \wedge j = \min\{\, j \mid a = (i_{A(0)})_{k+j} \,\} \end{cases}$$

we obtain $\sigma_0 = [\![M_0]\!]$. Note here that σ_a blocks initial moves $[\bar{a}, i_A] \neq [\bar{a}, i_{A(0)}]$ 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 $a \# \bar{a}$, 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 $f : Q^{\bar{a}}([\![A]\!] \otimes [\![B_p']\!]) \to T[\![B_p'']\!]$ by:

$$f \triangleq \{\, [\, (\bar{a}, i_{A(0)}, i_{B_p'}) * \circledast\, s \,] \mid [\, (\bar{a}, i_{A(0)}) * \circledast\, (i_B, \circledast)\, (i_{B_p'}, \circledast)\, s \,] \in \mathtt{viewf}(\sigma_0) \,\}.$$

Note that, for any $[(\bar{a}, i_A) * \circledast (i_B, \circledast)\,(i_{B'_p}, \circledast)\,s] \in \mathtt{viewf}(\sigma_0)$, 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 $\bar{a} \mid \Gamma, y\!:\!B'_p \vdash N : B''_p$ 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 $\sigma_0 = [\![M_0]\!] \upharpoonright (\bar{a}, i_{A(0)})$.

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,

$$\mathtt{viewf}(\sigma_0) = f_A \cup \bigcup\nolimits_{p=1}^{n_i} f_p \quad \text{where:}$$

$$\begin{aligned}
f_A &\triangleq f_0 \cup \{\, [\,(\bar{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,(i_{A''_i}, \circledast)\,s\,] \in \mathtt{viewf}(\sigma_0) \,\} \\
f_p &\triangleq f_0 \cup \{\, [\,(\bar{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,(i_{C_{i,p}}, \circledast)\,s\,] \in \mathtt{viewf}(\sigma_0) \,\} \\
f_0 &\triangleq \{\, [\,(\bar{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,s] \,|\, [\circledast \circledast s\,] \in \mathtt{viewf}(\mathtt{id}_\xi) \,\}
\end{aligned}$$

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_p are viewfunctions. Now, from f_A we obtain:

$$\begin{aligned}
f'_A : Q^{\bar{a}}([\![A]\!] \otimes [\![A''_i]\!]) \to T[\![B]\!] \triangleq \{\, [\,(\bar{a}, i_{A(0)}, i_{A''_i}) * \circledast s\,] \,|\, \\
[(\bar{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,(i_{A''_i}, \circledast)\,s\,] \in f_A \,\}.
\end{aligned}$$

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 $\bar{a} \mid \Gamma, y\!:\!A''_i \vdash M_A : B$ such that $[\![M_A]\!] = \mathtt{strat}(f'_A)$.

From each $f_p \neq f_0$ we obtain a viewfunction $f'_p : Q^{\bar{a}}([\![A]\!] \otimes [\![C_{i,p}]\!]) \to T[\![C'_{i,p}]\!]$ by:

$$f'_p \triangleq \{\, [\,(\bar{a}, i_{A(0)}, i_{C_{i,p}}) * \circledast s\,] \,|\, [\,(\bar{a}, i_{A(0)}) * \circledast (i_{A'_i}, \circledast)\,(i_{C_{i,p}}, \circledast)\,s\,] \in f_p \,\}.$$

By IH, there exists some $\bar{a} \mid \Gamma, y' : C_{i,p} \vdash M_p : C'_{i,p}$ 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 $\sigma_0 = [\![M_0]\!] \upharpoonright (\bar{a}, i_{A(0)})$. □

Finally, using the definability result and proposition 5.31 we can now show the following.

Corollary 5.34. *$\mathcal{T} = (\mathcal{T}, T, Q, O)$ satisfies ip-definability.*

Proof: For each $\bar{a}, A, B$, define $D^{\bar{a}}_{A,B} \triangleq \{\, f : Q^{\bar{a}}[\![A]\!] \to T[\![B]\!] \mid f \text{ is finitary} \,\}$. By definability, every $f \in D^{\bar{a}}_{A,B}$ is definable. We need also show:

$$(\forall \rho \in D^{\bar{a}}_{A \to B, \mathbb{N}}\,.\, \Lambda^{\bar{a}}(f)\,;\rho \in O^{\bar{a}} \implies \Lambda^{\bar{a}}(g)\,;\rho \in O^{\bar{a}}) \implies f \lesssim^{\bar{a}} g\,.$$

Assume the LHS assertion holds and let $\Lambda^{\bar{a}}(f)\,;\rho \in O^{\bar{a}}$, some $\rho : Q^{\bar{a}}([\![A]\!] \multimap T[\![B]\!]) \to T\mathbb{N}$. Then, let $[s\,;t] = [(\bar{a}, *) * \circledast (0, \circledast)^{\bar{b}}] \in \Lambda^{\bar{a}}(f)\,;\rho$, $[s] \in \Lambda^{\bar{a}}(f)$ and $[t] \in \rho$. By proposition 5.31,

$[t] \in \rho_{\leq t}$, so $\Lambda^{\bar{a}}(f) ; \rho_{\leq t} \in O^{\bar{a}}$. Moreover, $\rho_{\leq t} \in D^{\bar{a}}_{A \to B, \mathbb{N}}$, so $\Lambda^{\bar{a}}(g) ; \rho_{\leq t} \in O^{\bar{a}}$, by hypothesis. Finally, $\rho_{\leq t} \sqsubseteq \rho$ implies $\Lambda^{\bar{a}}(g) ; \rho_{\leq t} \sqsubseteq \Lambda^{\bar{a}}(g) ; \rho$, hence the latter observable, so $f \lesssim^{\bar{a}} g$. □

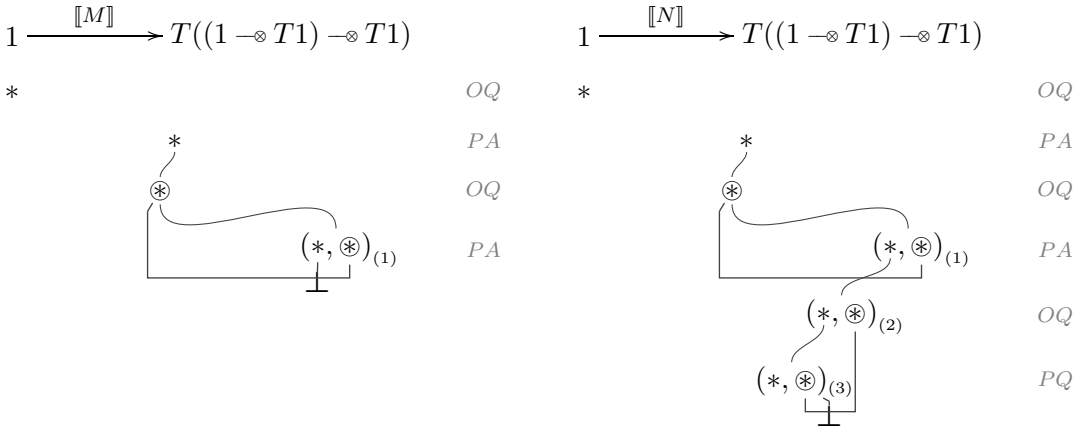
Hence, we have shown full abstraction.

Theorem 5.35. *$\mathcal{T} = (\mathcal{T}, T, Q, O)$ is a fully abstract model of $\nu\rho$.* □

5.7. **An equivalence established semantically.** In this last section we prove that the following terms M and N are equivalent. The particular equivalence exemplifies the fact that exceptional behaviour cannot be simulated in general by use of references, even of higher-order.

$$M \triangleq \lambda f.\, \mathtt{stop} : (\mathbb{1} \to \mathbb{1}) \to \mathbb{1}\,, \quad N \triangleq \lambda f.\, f\,\mathtt{skip}\,;\mathtt{stop} : (\mathbb{1} \to \mathbb{1}) \to \mathbb{1}\,.$$

By full-abstraction, it suffices to show $[\![M]\!] \backsimeq [\![N]\!]$, where the latter are given as follows.



Bottom links stand for deadlocks: if Opponent plays a move $(*, \circledast)_{(2)}$ under the last $*$ in $[\![M]\!]$ (thus providing the function f) then Player must play $[\![\mathtt{stop}]\!]$, i.e. remain idle. Similarly for $[\![N]\!]$: if Opponent gives an answer to $(*, \circledast)_{(3)}$ (providing thus the outcome of $f\mathtt{skip}$) then Player deadlocks the play.

We have that $[\![M]\!] \sqsubseteq [\![N]\!]$ and therefore, by (5.5), $[\![M]\!] \lesssim [\![N]\!]$. Conversely, let $\rho : T((1 \multimap T1) \multimap T1) \to T\mathbb{N}$ be a tl4 tidy strategy such that $[* * \circledast (0, \circledast)^{\bar{a}}] \in [\![N]\!] ; \rho$ for some $\bar{a}$. Then, because of the form of $[\![N]\!]$, ρ can only play initial moves up to $(*, \circledast)_{(1)}$, then possibly ask some names to $(*, \circledast)_{(1)}$, and finally play $(0, \circledast)^{\bar{a}}$. Crucially, ρ cannot play $(*, \circledast)_{(2)}$ under $*$: this would introduce a question that could never be answered by $[\![N]\!]$, and therefore ρ would not be able to play $(0, \circledast)^{\bar{a}}$ without breaking well-bracketing. Hence, $[\![M]\!]$ and ρ can simulate the whole interaction and therefore $[* * \circledast (0, \circledast)^{\bar{a}}] \in [\![M]\!] ; \rho$.

6. Conclusion

Until recently, names used to be bypassed in Denotational Semantics: most approaches focussed on the effect achieved by use of names rather than names themselves. Characteristic of this attitude was the 'object-oriented' modelling of references [6, 3] and exceptions [19] as products of their effect-related *methods* (in the spirit of [39]). These approaches were unsatisfactory to some extent, due to the need for 'bad' syntactic constructors in the examined languages. Moreover, they could not apply to the simplest nominal language, the

ν-calculus [36], since there the achieved effect could not be given an extensional, name-free description. These issues revealed the need that names be treated as a proper computational effect [44], and led to the advent of nominal games [2, 21].

In this paper we have taken some further steps in the semantics of nominal computation by examining the effect of (nominal) general references. We have shown that nominal games provide a framework expressive enough that, by use of appropriate monadic (and comonadic) constructions, one can model general references without moving too far from the model of the ν-calculus [2]. This approach can be extended to other nominal effects too; e.g. in [47] it is applied to exceptions (with and without references). Moreover, we have examined abstract categorical models for nominal computation, and references in particular (in the spirit of [45, 44]).

There are many threads in the semantics of nominal computation which need to be pursued further. Firstly, there are many nominal games models to build yet: research in this direction has already been undertaken in [24, 22, 47, 31]. By constructing models for more nominal languages we better understand the essential features of nominal computation (e.g. *name-availability* [31]) and build stronger intuitions on nominal games. Another direction for further research is that of characterising the nominal effect — i.e. the computational effect that rises 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 [41] seems to be particularly helpful in this direction.

A 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 [13, 34, 32]. It would therefore be useful to 'nominalise' that body of work and apply it to nominal calculi. Already from [32] we can deduce that nominal languages with ground store are undecidable, and from [36] we know that equivalence is decidable for programs of first-order type in the ν-calculus, but otherwise the problem remains open.

Acknowledgements. I would like to thank Samson Abramsky for his constant encouragement, support and guidance. I would also like to thank Andy Pitts, Andrzej Murawski, Dan Ghica, Ian Stark, Luke Ong, Guy McCusker, Jim Laird, Paul Levy, Sam Sanjabi and the anonymous reviewers for fruitful discussions, suggestions and criticisms.

Appendix A. Deferred proofs

I. Proof of closure of tidiness under composition.

Lemma A.1. *Let $\sigma : A \to B$ and $\tau : B \to C$ 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 A.2. *Let $\sigma : A \to B$ and $\tau : B \to C$ 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 without loss of generality $(s \parallel t)\tilde{q}_1 ... \tilde{q}_i = (sq_1 ... q_i) \parallel (tq'_1 ... q'_{i-1})$ 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 without loss of generality 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 5.21, q_1 is a copy of $s.\text{–}1 = q_0$. If q_0 is in A then its justifier is

$s.\text{–}2 = x_0$ 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 A.3. *If $\sigma : A \to B$ and $\tau : B \to C$ 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 need only show the other two conditions. Assume without loss of generality 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 $\tilde{q} = a^{\bar{a}}$ 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 $sq_1 \asymp tq_1'$, 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 $q = a^{\bar{a}}$ 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 $\tilde{q}_{(O)}\tilde{q}_1 \dots \tilde{q}_i\tilde{q}_{(P)}$ 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 $(sy_1 \dots y_k \parallel ty_1' \dots y_k') \in \sigma \parallel \tau$ 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 $x_1 = \ulcorner s \urcorner.\text{–}3 = s.\text{–}3$. Then, $[ty_1'] \in \tau$ with $\underline{y_1'} = \underline{y_1}$. By proposition 5.19, $q_{(O)} q_1 s_2'$ 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 $(sy_1 \dots y_{k-1} \parallel ty_1' \dots y_{k-1}') \in \sigma \parallel \tau$ 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, $[sy_1 \dots y_{k-1}y_k] \in \sigma$ with $\underline{y_k} = \underline{y'_k}$. The case of k odd is entirely dual.

Now, just 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 $\tilde{x}_{i+1} = (s\,;t)$.–3. Finally, as above, $\tilde{y}_{i+1} = \tilde{y}_j = \tilde{y}$, all j, as required. □

II. *Proof of Decomposition Lemma 5.32:* 1 is straightforward: we just partition σ into σ_0 and σ' and recover it by use of $[x \overset{\bar{a}}{=} i_{A(0)}]$ and $\mathtt{cnd}$. For 2, we just use the definition of name-abstraction for strategies and the condition on σ.

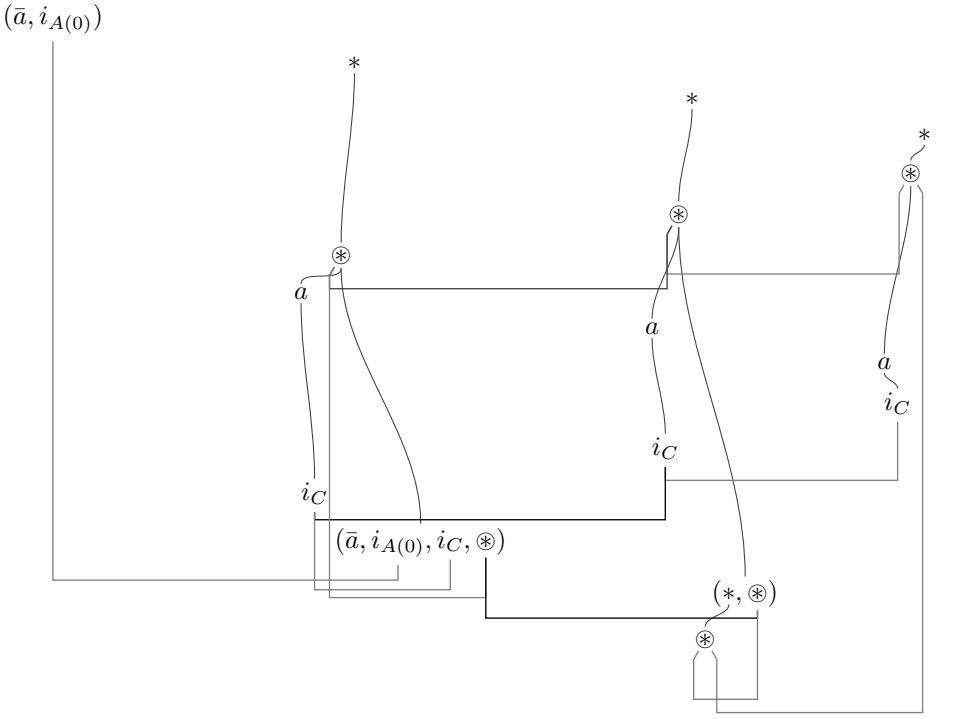
For 3, it is clear that m_0 is either a store-Q a under $\circledast$, or a store-H j_A, or a store-H $(i_B, \circledast)$.

In case $m_0 = a$ with $a \in \mathbb{A}_C$, we define $\sigma_a : Q^{\bar{a}}(\llbracket A \rrbracket \otimes \llbracket C \rrbracket) \to T\llbracket B \rrbracket \triangleq \mathtt{strat}(f_a)$, where

$$f_a \triangleq \{\, [\,(\bar{a}, i_{A(0)}, i_C) * \circledast s\,] \mid [\,(\bar{a}, i_{A(0)}) * \circledast a\, i_C\, s\,] \in \mathtt{viewf}(\sigma) \,\}.$$

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 $[(\bar{a}, i_{A(0)}, i_C) * \circledast s] \in f_a$ with $[(\bar{a}, i_{A(0)}) * \circledast a\, i_C\, s] \in \mathtt{viewf}(\sigma)$, $(\bar{a}, i_{A(0)}, i_C) * \circledast s$ is a justified sequence and satisfies well-bracketing, as its open Q's outside s are the same as those in $(\bar{a}, i_{A(0)}) * \circledast a\, i_C\, s$, 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.

$$Q^{\bar{a}}\llbracket A \rrbracket \xrightarrow{\langle \mathtt{id}, \phi \rangle\,;\,\tau\,;\,T\zeta'} TQ^{\bar{a}}(\llbracket A \rrbracket \otimes \llbracket C \rrbracket) \xrightarrow{T\sigma_a} T^2\llbracket B \rrbracket \xrightarrow{\mu} T\llbracket B \rrbracket$$



Because of the copycat links, we see that

$$\begin{aligned}\mathtt{viewf}(\langle \mathtt{id}, \phi\rangle ; \tau ; T\zeta' ; T\sigma_a ; \mu) \upharpoonright (\bar{a}, i_{A(0)}) \\ = \{[(\bar{a}, i_{A(0)}) * \circledast\, a\, i_C\, s] \,|\, [(\bar{a}, i_{A(0)}, i_C) * \circledast\, s] \in \mathtt{viewf}(\sigma_a)\} = \mathtt{viewf}(\sigma)\,,\end{aligned}$$

as required. Note that the restriction to initial moves $[\bar{a}, i_{A(0)}]$ 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 $[(\bar{a}, i_{A(0)}) * \circledast\, m_0\, a\, i_C] \in \sigma$, we have that

$$\sigma = \mathtt{strat}(f_a \cup (f' \setminus f'_a))\,,$$

where f_a, f' are viewfunctions of type $Q^{\bar{a}}[\![A]\!] \to T[\![B]\!]$, 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,

$$\begin{aligned} f_a &\triangleq \{\,[\,(\bar{a}, i_{A(0)}) * \circledast\, j_A\, a\, i_C\, s\,] \in \mathtt{viewf}(\sigma)\,\} \\ f'_a &\triangleq \{\,[\,(\bar{a}, i_{A(0)}) * \circledast\, j_A\, a\, s\,] \,|\, [\circledast\ \circledast\ a\, s] \in \mathtt{viewf}(\mathtt{id}_\xi)\,\} \\ f' &\triangleq f'_a \cup \{\,[\,(\bar{a}, i_{A(0)}) * \circledast\, j_A\, y\, s\,] \in \mathtt{viewf}(\sigma) \,|\, y \neq a\,\}\,. \end{aligned}$$

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

$$\begin{aligned} f''_a : Q^{\bar{a}}[\![A]\!] \to T[\![C]\!] &\triangleq \{\,[\,(\bar{a}, i_{A(0)}) * \circledast(i_C, \circledast)\, s\,] \,| \\ &\qquad [\,(\bar{a}, i_{A(0)}) * \circledast j_A\, a\, i_C\, s\,] \in f_a \vee [\circledast\ \circledast\ s] \in \mathtt{viewf}(\mathtt{id}_\xi)\,\} \\ \sigma_a : Q^{\bar{a}}[\![A]\!] \to T[\![C]\!] &\triangleq \mathtt{strat}(f''_a) \\ \sigma' : Q^{\bar{a}}[\![A]\!] \to T[\![B]\!] &\triangleq \mathtt{strat}(f') \\ \sigma'' : Q^{\bar{a}}[\![A]\!] \to T[\![B]\!] &\triangleq \langle \Delta, \sigma_a\rangle ; \tau ; T(\mathtt{id} \otimes \phi ; \tau) ; \mu ; \cong ; T\sigma' ; \mu\,. \end{aligned}$$

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 $[(\bar{a}, i_{A(0)}) * \circledast\, (i_C, \circledast)\, s] \in f''_a$ with $[(\bar{a}, i_{A(0)}) * \circledast\, j_A\, a\, i_C\, s] \in \mathtt{viewf}(\sigma)$. We have that $(\bar{a}, i_{A(0)}) * \circledast\, (i_C, \circledast)\, s$ 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, visibility and NC's are straightforward.

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

$$\sigma'_a : Q^{\bar{a}}[\![A]\!] \to [\![C]\!] \triangleq \mathtt{strat}\{\,[\,(\bar{a}, i_{A(0)})\, i_C\, s\,] \,|\, [\,(\bar{a}, i_{A(0)}) * \circledast\, j_A\, a\, i_C\,] \in f_a\,\}\,,$$

and hence $\sigma'' \upharpoonright (\bar{a}, i_{A(0)}) = \langle \Delta, \sigma'_a\rangle ; \mathtt{id} \otimes \phi ; \tau ; \cong ; T\sigma' ; \mu\,.$ Analysing the behaviour of the latter composite strategy and observing that the response of σ'' to inputs different than

$[\bar{a}, i_{A(0)}]$ is merely the initial answer $*$ imposed by totality, we obtain:

$$\begin{aligned}\mathtt{viewf}(\sigma'') &= \{\, [\,(\bar{a}, i_{A(0)}) \ast \circledast\, j_A\, a\, s\,], [(\bar{a}, i_{A(0)}) \ast \circledast\, j_A\, y\, s] \in \mathtt{viewf}(\sigma'') \mid y \neq a \,\} \\ &= \{\, [\,(\bar{a}, i_{A(0)}) \ast \circledast\, j_A\, a\, i_C\, s\,] \mid [\,(\bar{a}, i_{A(0)}) \ast \circledast\, (i_C, \circledast)\, s] \in f''_a \wedge s.1 \in J_{[\![C]\!]} \,\} \\ &\qquad \cup \{\, [\,(\bar{a}, i_{A(0)}) \ast \circledast\, j_A\, y\, s\,] \in f' \mid y \neq a \,\} \\ &= f_a \cup (f' \setminus f'_a) = \mathtt{viewf}(\sigma)\end{aligned}$$

as required.

In case $x = (i_B, \circledast)$ we work similarly as above. □

References

[1] Abramsky, S. Domain theory. Lecture Notes, Oxford University Computing Laboratory, 2007.

[2] Abramsky, S., Ghica, D., Murawski, A., Ong, L., and Stark, I. Nominal games and full abstraction for the nu-calculus. In *LICS '04: Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science* (Turku, 2004), IEEE Computer Society Press, pp. 150–159.

[3] Abramsky, S., Honda, K., and McCusker, G. A fully abstract game semantics for general references. In *LICS '98: Proceedings of the 13th Annual IEEE Symposium on Logic in Computer Science* (Indianapolis, 1998), IEEE Computer Society Press, pp. 334–344.

[4] Abramsky, S., and Jagadeesan, R. Games and full completeness for multiplicative linear logic. *Journal of Symbolic Logic 59*, 2 (1994), 543–574.

[5] Abramsky, S., Jagadeesan, R., and Malacaria, P. Full abstraction for PCF. *Information and Computation 163*, 2 (2000), 409–470.

[6] Abramsky, S., and McCusker, G. Linearity, Sharing and State: a fully abstract game semantics for Idealized Algol. In O'Hearn and Tennent [33], pp. 297–329. Vol. 2, 1997.

[7] Baillot, P., Danos, V., and Ehrhard, T. Believe it or not, AJM's games model is a model of classical linear logic. In *LICS '97: Proceedings of the 12th Annual IEEE Symposium on Logic in Computer Science* (Warsaw, 1997), IEEE Computer Society Press, pp. 68–75.

[8] Barr, M., and Wells, C. *Category theory for computing science*, third ed. Les Publications CRM, Montreal, 1999.

[9] Brookes, S., and Geva, S. Computational comonads and intensional semantics. In *Applications of Categories in Computer Science: Proceedings LMS Symposium* (Durham, 1991), vol. 177, Cambridge University Press, pp. 1–44.

[10] Brookes, S., and van Stone, K. Monads and comonads in intensional semantics. Tech. Rep. CMU-CS-93-140, Carnegie Mellon University, 1993.

[11] Freyd, P. J. Recursive types reduced to inductive types. In *LICS'90: Proceedings of the 5th Annual IEEE Symposium on Logic in Computer Science* (Philadelphia, 1990), IEEE CS Press, pp. 498–507.

[12] Gabbay, M. J., and Pitts, A. M. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing 13* (2002), 341–363.

[13] Ghica, D. R., and McCusker, G. Reasoning about Idealized Algol using regular languages. In *ICALP '00: Proceedings of 27th International Colloquium on Automata, Languages and Programming* (Geneva, 2000), vol. 1853 of *LNCS*, Springer-Verlag, pp. 103–116.

[14] Harmer, R. *Games and full abstraction for nondeterministic languages*. DPhil thesis, University of London, 1999.

[15] Honda, K., and Yoshida, N. Game-theoretic analysis of call-by-value computation. *Theoretical Computer Science 221*, 1–2 (1999), 393–456.

[16] Hyland, J. M. E., and Ong, C.-H. L. On full abstraction for PCF: I, II, III. *Information and Computation 163*, 2 (2000), 285–408.

[17] Jeffrey, A., and Rathke, J. A fully abstract may testing semantics for concurrent objects. In *LICS '02: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science* (Copenhagen, 2002), IEEE Computer Society Press, pp. 101–112.

[18] Jones, S. P. *Haskell 98 Language and Libraries: The Revised Report*. Cambridge University Press, May 2003.

[19] Laird, J. A fully abstract game semantics of local exceptions. In *LICS '01: Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science* (Boston, 2001), IEEE CS Press, p. 105.
[20] Laird, J. A categorical semantics of higher order store. In *CTCS '02: Category Theory and Computer Science* (Ottawa, 2002), vol. 69 of *Electronic Notes in Theoretical Computer Science*, pp. 209–226.
[21] Laird, J. A game semantics of local names and good variables. In *FoSSaCS '04: Proceedings of the 7th International Conference on Foundations of Software Science and Computation Structures* (Barcelona, 2004), vol. 2987 of *Lecture Notes in Computer Science*, Springer, pp. 289–303.
[22] Laird, J. Game semantics for higher-order concurrency. In *FSTTCS '06: Proceedings of the 26th International Conference on Foundations of Software Technology and Theoretical Computer Science* (Kolkata, 2006), vol. 4337 of *Lecture Notes in Computer Science*, Springer, pp. 417–428.
[23] Laird, J. A fully abstract trace semantics for general references. In *ICALP '07: Proceedings of the 34th International Colloquium on Automata, Languages and Programming* (Wroclaw, 2007), vol. 4596 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 667–679.
[24] Laird, J. 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).
[25] Mac Lane, S. *Categories for the working mathematician*, second ed., vol. 5 of *Graduate texts in mathematics*. Springer Verlag, 1998.
[26] McCusker, G. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Distinguished Dissertations. Springer-Verlag, London, 1998.
[27] Milner, R., Tofte, M., and Macqueen, D. *The Definition of Standard ML*. MIT Press, 1997.
[28] Moggi, E. Computational lambda calculus and monads. Tech. Rep. ECS-LFCS-88-86, University of Edinburgh, 1988.
[29] Moggi, E. Computational lambda-calculus and monads. In *LICS '89: Proceedings of 4th Annual IEEE Symposium on Logic in Computer Science* (Pacific Grove, 1989), IEEE CS Press, pp. 14–23.
[30] Moggi, E. Notions of computation and monads. *Information and Computation 93*, 1 (1991), 55–92.
[31] Murawski, A., and Tzevelekos, N. Full abstraction for Reduced ML. In *FoSSaCS '09: Proceedings of the 12th International Conference on Foundations of Software Science and Computation Structures* (York, 2009), vol. 5504 of *Lecture Notes in Computer Science*, Springer, pp. 32–47.
[32] Murawski, A. S. On program equivalence in languages with ground-type references. In *LICS '03: Proceedings of the 18th IEEE Symposium on Logic in Computer Science* (Ottawa, 2003), pp. 108–117.
[33] O'Hearn, P. W., and Tennent, R. D., Eds. *ALGOL-like Languages*. Birkhäuser, 1997.
[34] Ong, C.-H. L. Observational equivalence of third-order Idealized Algol is decidable. In *LICS '02: Proceedings of the 17th Annual IEEE Symposium on Logic in Computer Science* (Copenhagen, 2002), IEEE Computer Society Press, pp. 245–256.
[35] Pitts, A. M. Nominal logic, a first order theory of names and binding. *Information and Computation 186* (2003), 165–193.
[36] Pitts, A. M., and Stark, I. D. B. Observable properties of higher order functions that dynamically create local names, or: What's new? In *MFCS '93: Proceedings of 18th International Symposium on Mathematical Foundations of Computer Science* (Gdańsk, 1993), vol. 711 of *Lecture Notes in Computer Science*, Springer-Verlag, Berlin, pp. 122–141.
[37] Plotkin, G. D. LCF considered as a programming language. *Theoretical Computer Science 5* (1977), 223–255.
[38] Plotkin, G. D., and Power, J. Notions of computation determine monads. In *FoSSaCS '02: Proceedings of the 5th International Conference on Foundations of Software Science and Computation Structures* (Grenoble, 2002), Springer-Verlag, pp. 342–356.
[39] Reynolds, J. C. The essence of Algol. In *Proceedings of the International Symposium on Algorithmic Languages* (Amsterdam, 1981), North-Holland, pp. 345–372. Reprinted in [33, vol. 1, pages 67–88].
[40] Sanjabi, S. B., and Ong, C.-H. L. Fully abstract semantics of additive aspects by translation. In *AOSD '07: Proceedings of the 6th international conference on Aspect-oriented software development* (Vancouver, 2007), ACM, pp. 135–148.
[41] Schöpp, U. *Names and Binding in Type Theory*. DPhil thesis, University of Edinburgh, 2006.
[42] Scott, D. S. A type-theoretical alternative to ISWIM, CUCH, OWHY. *Theoretical Computer Science 121*, 1-2 (1993), 411–440. First written in 1969 and circulated privately.
[43] Smyth, M. B., and Plotkin, G. D. The category-theoretic solution of recursive domain equations. *SIAM Journal on Computing 11*, 4 (1982), 761–783.

[44] Stark, I. D. B. *Names and Higher-Order Functions*. PhD thesis, University of Cambridge, Dec. 1994. Also available as Technical Report 363, University of Cambridge Computer Laboratory.

[45] Stark, I. D. B. Categorical models for local names. *Lisp and Symbolic Computation 9*, 1 (Feb. 1996), 77–107.

[46] Tzevelekos, N. Full abstraction for nominal general references. In *LICS '07: Proceedings of the 22nd Annual IEEE Symposium on Logic in Computer Science* (Wroclaw, 2007), IEEE Computer Society Press, pp. 399–410.

[47] Tzevelekos, N. Full abstraction for nominal exceptions and general references. In *GaLoP '08: Games for Logic and Programming Languages* (Budapest, 2008). Journal version submitted to *APAL*.

[48] Tzevelekos, N. *Nominal game semantics*. DPhil thesis, Oxford University, 2008.

[49] Wadler, P. The essence of functional programming. In *POPL '92: Conference Record of the 19th ACM Symposium on Principles of Programming Languages* (Albuquerque, 1992), pp. 1–14.