Full abstraction for nominal general references

Nikos Tzevelekos
Oxford University Computing Laboratory

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 modeling functional languages with names. These are ordinary 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.

1 Introduction

One of the most challenging problems in denotational semantics of programming languages is that of modeling languages with general references. General references are references which can store not only values of ground types (integers, booleans, etc.) but also of higher types (procedures, higher-order functions, or references themselves). The general reference is a very useful and powerful programming construct, and it can be used to encode a wide range of computational effects and programming paradigms (e.g. object-oriented programming). The added expressiveness of general references makes their denotational models complicated, mainly because of the phenomena of dynamic update and interference present in the language.

Fully abstract models for general references have been achieved via game semantics in [3], and via abstract categorical semantics in [9]. The presentation in [9] does not distinguish between λ-abstraction and local fresh-reference creation (ν-abstraction), and hence is distanced from the common use of references in programming languages. On the other hand, the calculus examined in [3] distinguishes between λ- and ν-abstractions, yet encodes references as variables of a read/write product type. This leads to the presence of *bad variables*[1], as read/write-product semantical objects may not necessarily denote references. Bad variables lead to unwanted behaviors and prohibit the use of *equality tests* for references.

In this paper we obtain the *first full-abstraction result for a statically-scoped language with general references, good variables and reference-equality tests,* which faithfully reflects the practice of real programming languages such as ML. We follow the alternative (nominal) approach of treating references separately from variables, as *names*, extending the ν-calculus of Pitts and Stark [14]. The ν-calculus is a paradigmatic λ-calculus with names, in which names are constant terms of ground type that "...are created with local scope, can be tested for equality and can be passed around via function application, but that is all". Here we use names for references, so names are of reference types and may also be dereferenced and updated, introducing thus a λ-calculus with *nominal general references,* the $\nu\rho$-calculus.

Nominal games were introduced in [2] as the basis for the first fully abstract model of the ν-calculus.[2] They constitute a version of Honda-Yoshida CBV-games [6] built in the universe of nominal sets of Pitts and Gabbay [5, 13]. Nominal sets are sets whose elements entail a finite number of *names,* and which are acted upon by finite name-permutations. Thus, the nominal games of [2] are CBV-games played using moves-with-names, that is moves attached with a finite set of names representing the names introduced so far. Our intention was to build a model for the $\nu\rho$-calculus using nominal games, yet we discovered discrepancies arising from the use of name-sets in moves: the unordered nature of name-creation is incompatible with the deterministic behavior of strategies and, in fact, nominal games do not form a category.

Hence, we recast nominal games using moves attached with name-lists instead of name-sets, and rectifying other discrepancies. Moreover, since names model references of several types, our construction is based on nominal sets over countably infinitely many sets of names –one for each type.[3] From the basic category of nominal games we obtain an adequate model for $\nu\rho$ by using a store arena, which is obtained as the canonical solution to the domain equation (SE)

[1] By "bad variables" we mean read/write constructs of reference type which do not yield references, like `mkvar` of [3].

[2] A different version of nominal games was introduced in [8], yet it did not yield a fully abstract model for the ν-calculus.

[3] Also, the use of name-lists allows us to construct nominal games in nominal sets with strongly supported elements (v. definition 1).

Updated version, 23/07/07.

of page 5. For full abstraction we need to apply some further constraints on the way the store is accessed in nominal games, obtaining thus *tidy strategies*.

Summarising, the contributions of this paper are: a) the introduction of a λ-calculus with nominal general references, name-equality test and good variables; b) the rectification of nominal games; c) the construction of a fully abstract model using nominal games with tidy strategies. An appealing further direction is that of abstracting the basic nominal games model to a categorical level, in the spirit of [1, 9]; a first step in this direction has already been taken in the abstract description of a $\lambda_{\nu\rho}$-model (section 3.1).

2 Theory of nominal sets

We give a short overview of nominal sets, which will be used as the basis for all constructions presented in this paper. Intuitively, nominal sets are sets whose elements entail a finite number of *names*, and which are acted upon by finite name-permutations.[4] We present these following [13].

Assume a countably infinite set TY of types $A, B, ...$, and for each type A assume a countably infinite set of names N_A. The elements of N_A are *names to type* A and are denoted by $\mathtt{a}^A, \mathtt{b}^A, ...$. We write $\text{PERM}(\mathsf{N}_A)$ for the group of finite permutations of N_A. We let $\mathsf{N} \triangleq \bigcup_{A\in\text{TY}} \mathsf{N}_A$ be the set of (general) names and $\text{PERM}(\mathsf{N}) \triangleq \bigoplus_{A\in\text{TY}} \text{PERM}(\mathsf{N}_A)$ be the group of (finite) permutations. Names are denoted by $\alpha, \beta, \gamma, ...$, and permutations by $\pi, \pi', ...$; in particular, $(\alpha\,\beta)$ denotes the permutation that only swaps names α and β (of same type) and $\mathtt{id}$ denotes the identity permutation.

A ***nominal set*** X is a set equipped with an action from $\text{PERM}(\mathsf{N})$, that is a function $\_ \circ \_ : \text{PERM}(\mathsf{N}) \times X \to X$ such that, for any $\pi, \pi' \in \text{PERM}(\mathsf{N})$ and $x \in X$,

$$\pi \circ (\pi' \circ x) = (\pi \circ \pi') \circ x \qquad \mathtt{id} \circ x = x \qquad \text{(P)}$$

Moreover, all $x \in X$ have finite support $\mathsf{S}(x)$, where

$$\mathsf{S}(x) \triangleq \{\alpha \in \mathsf{N} \mid \text{for infinitely many } \beta.\ (\alpha\,\beta) \circ x \neq x\} \quad \text{(S)}$$

We can see that N in particular is a nominal set. For $x \in X$ and $\alpha \in \mathsf{N}$, α ***is fresh for*** x, written $\alpha\#x$, iff $\alpha \notin \mathsf{S}(x)$. x is ***equivariant*** iff it has empty support. $\mathsf{N}^\#$ stands for the nominal set of finite lists of distinct (i.e. pairwise fresh) names.

If Y is a nominal set and $X \subseteq Y$ then X is a ***nominal subset*** of Y iff X is closed under permutations, these acting as on Y. If X, Y are nominal sets then their product $X \times Y$ is also a nominal set, with permutations defined componentwise. Moreover, a relation $R \subseteq X \times Y$ is a nominal subset of $X \times Y$ iff, for any permutation π and $(x, y) \in X \times Y$, $xRy \iff (\pi \circ x) R (\pi \circ y)$. We call such an R a ***nominal relation***. Accordingly, $f : X \to Y$ is a ***nominal function*** iff $f(\pi \circ x) = \pi \circ f(x)$, for any $x \in X$ and π. For example, $\mathsf{S}(\_) : X \to \mathcal{P}_{\text{fin}}(\mathsf{N})$ is a nominal function.

We let $\mathbf{Nom}_{\text{TY}}$ be the category of nominal sets (on N) and nominal functions. In nominal sets we can succinctly define *name-abstraction*: for each $\alpha \in \mathsf{N}$ and $x \in X$ let

$$\langle\alpha\rangle x \triangleq \{(\beta, y) \in \mathsf{N} \times X \mid (\beta = \alpha \vee \beta\#x) \wedge y = (\alpha\,\beta) \circ x\}$$

We can show $\mathsf{S}(\langle\alpha\rangle x) = \mathsf{S}(x) \setminus \{\alpha\}$. Another form of abstraction involves restricting the support of an element to that of a given name-list: for any $x \in X$ and $\vec{\alpha} \in \mathsf{N}^\#$ let

$$[x]_{\vec{\alpha}} \triangleq \{y \in X \mid \exists\pi.\ \pi \circ \vec{\alpha} = \vec{\alpha} \ \wedge\ y = \pi \circ x\}$$

If $\mathsf{S}(x) \supseteq \mathsf{S}(\vec{\alpha})$ then $\mathsf{S}([x]_{\vec{\alpha}}) = \mathsf{S}(\vec{\alpha})$. The notion of support can be strengthened to model *ordered* entailment of names.

Definition 1 (Strong support) For any nominal set X, any $x \in X$ and any $S \subseteq \mathsf{N}$, S strongly supports x if, for any permutation π, π fixes x iff π fixes each element in S. ▲

The notion of strong support is indeed stronger than that of support, which employs only the "if"-part of the above assertion. For example, if $\alpha, \beta \in \mathsf{N}$ then the set $\{\alpha, \beta\}$ only has weak support $\{\alpha, \beta\}$, whereas the list α, β has strong support $\{\alpha, \beta\}$. Strong support coincides with weak support when the former exists.

3 The $\nu\rho$-calculus

The $\nu\rho$-calculus is a λ-calculus with nominal general references. Leaving aside the use of name-lists instead of name-sets in the operational semantics, it is an extension of the ν-calculus of Pitts and Stark [14] (and of the ν-calculus with `int_ref` of [15, chapter 5]) using names for general references. We present its syntax in nominal sets, and thus obtain nominal notions such as name-freshness and name-permutation for free.

Definition 2 The $\nu\rho$-calculus is a functional calculus of nominal references. Its types are given as follows.

$$\text{TY} \ni A, B ::= \ \mathbf{1} \mid \mathbb{N} \mid [A] \mid A \to B \mid A \otimes B$$

So references to type A are of type $[A]$. Terms compose TE:

$\text{TE} \ni M, N ::= \ x \mid \lambda x.M \mid M\,N$	λ-term
$\mid \mathtt{skip} \mid \tilde{n} \mid \mathtt{pred}\,M \mid \mathtt{succ}\,N$	return / arithmetic
$\mid \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2$	if_then_else
$\mid \langle M, N\rangle \mid \mathtt{fst}\,M \mid \mathtt{snd}\,N$	pair / projections
$\mid \alpha \mid \nu\alpha.M$	name / ν-abstraction
$\mid [M = N]$	name-equality test
$\mid M := N \mid\ !M$	update / dereferencing

TE is a nominal set in $\mathbf{Nom}_{\text{TY}}$: each name $\alpha = \mathtt{a}^A$ is taken from N_A and $\nu\alpha.M$ stands for $\nu(\langle\alpha\rangle M)$. Of the terms

[4] In fact, nominal sets are sets in a Fraenkel-Mostowski permutation model of ZFA set theory with a countably infinite set of names and a group of finite permutations of names.

above, the values are:

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

The typing system involves terms in environments $\vec{\alpha}\,|\,\Gamma$, where $\vec{\alpha}$ a list of (distinct) names and Γ a finite set of variable-type pairs. Some of its rules are the following.

$$\frac{}{\vec{\alpha} \mid \Gamma, x{:}A \vdash x : A} \qquad \frac{}{\vec{\alpha} \mid \Gamma \vdash \alpha : [A]}\ {\scriptstyle \alpha = \mathtt{a}^A \not\# \vec{\alpha}}$$

$$\frac{\vec{\alpha}, \alpha \mid \Gamma \vdash M : B}{\vec{\alpha} \mid \Gamma \vdash \nu\alpha.M : B} \qquad \frac{\vec{\alpha} \mid \Gamma \vdash M : [A]}{\vec{\alpha} \mid \Gamma \vdash\, !M : A}$$

$$\frac{\vec{\alpha} \mid \Gamma \vdash M : [A] \qquad \vec{\alpha} \mid \Gamma \vdash N : [A]}{\vec{\alpha} \mid \Gamma \vdash [M = N] : \mathbb{N}}$$

$$\frac{\vec{\alpha} \mid \Gamma \vdash M : [A] \qquad \vec{\alpha} \mid \Gamma \vdash N : A}{\vec{\alpha} \mid \Gamma \vdash M := N : \mathbf{1}}$$

The reduction calculus is defined in store environment S:

$$S ::= \epsilon \mid \alpha, S \mid \alpha :: V, S$$

For each store environment S we define its domain, $\mathtt{dom}(S)$, to be the list of names stored in S. We only consider environments with domains in $\mathsf{N}^{\#}$ (i.e. lists of distinct names). Reduction rules are as below,

$$\text{DRF}\ \frac{}{S, \alpha :: V, S' \vDash\, !\alpha \longrightarrow S, \alpha :: V, S' \vDash V}$$

$$\text{NEW}\ \frac{}{S \vDash \nu\alpha.M \longrightarrow S, \beta \vDash (\alpha\ \beta) \circ M}\ \text{(any } \beta \# S)$$

$$\text{UPD}\ \frac{}{S, \alpha(:: W), S' \vDash \alpha := V \longrightarrow S, \alpha :: V, S' \vDash \texttt{skip}}$$

$$\text{EQ}\ \frac{}{S \vDash [\alpha = \beta] \longrightarrow S \vDash \tilde{n}}\ {\scriptstyle n=1 \text{ if } \alpha\#\beta \atop n=0 \text{ if } \alpha=\beta}$$

$$\text{PRD}\ \frac{}{S \vDash \texttt{pred}\,\tilde{0} \longrightarrow S \vDash \tilde{0}}$$

$$\text{LAM}\ \frac{}{S \vDash (\lambda x.M)\, V \longrightarrow S \vDash M\{V/x\}}$$

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

plus standard CBV rules for $\texttt{fst}$, $\texttt{snd}$, $\texttt{if0}$, $\texttt{pred}$ and $\texttt{succ}$. Evaluation contexts $E[\_]$ are of the forms:

$$\begin{array}{llllll}
[_ = N] \ , & [\alpha = _] \ , & !_ \ , & _ := N \ , & \alpha := _ \\
(\lambda x.N)\,_ \ , & _\, N \ , & \texttt{fst}\ _ \ , & \texttt{if0}\ _\ \texttt{then}\ N\ \texttt{else}\ N' \\
\texttt{snd}\ _ \ , & \texttt{pred}\ _ \ , & \texttt{succ}\ _ \ , & \langle _, N\rangle \ , & \langle V, _\rangle
\end{array}$$

▲

We take *observable terms* to be the constants of type $\mathbb{N}$, and around them we build the notion of observational equivalence.

Definition 3 ($\lessapprox$) For typed terms $\vec{\alpha} \mid \Gamma \vdash M\,,\,N : A$ define $\vec{\alpha} \mid \Gamma \vdash M \lessapprox N$ to be the assertion:

for any variable- and name-closing context $C[\_] : \mathbb{N}$,

$$\exists S'.(\vDash C[M] \twoheadrightarrow S' \vDash \tilde{0}) \implies \exists S''.(\vDash C[N] \twoheadrightarrow S'' \vDash \tilde{0})$$

We usually omit $\vec{\alpha}\,|\,\Gamma$ and write simply $M \lessapprox N$. ▲

3.1 Semantics

We examine sufficient conditions for a fully abstract categorical semantics of $\nu\rho$, following a development similar to that of [15, chapter 3]. Note that, translating each term M into a morphism $[\![M]\!]$ and assuming a preorder "$\lesssim$" in the semantics, full-abstraction will amount to the assertion:

$$M \lessapprox N \iff [\![M]\!] \lesssim [\![N]\!] \qquad \text{(FA)}$$

Soundness. We examine semantics in a family of categories $\langle \mathcal{M}^{\vec{\alpha}} \rangle_{\vec{\alpha}\in\mathsf{N}^{\#}}$ so that each typed term $\vec{\alpha} \mid \Gamma \vdash M : A$ is translated into a map $[\![M]\!] : [\![\Gamma]\!] \to T[\![A]\!]$ in $\mathcal{M}^{\vec{\alpha}}$. T is a computational monad, so our semantics is a *monadic* one (v. [11]). Computation in $\nu\rho$ is store-update and fresh-name creation. These requirements define $\lambda_{\nu\rho}$-models.

Definition 4 A $\lambda_{\nu\rho}$-model $\mathcal{M}$ is a family of categories and monads $\langle \mathcal{M}^{\vec{\alpha}}, (T, \eta, \mu, \tau)^{\vec{\alpha}} \rangle_{\vec{\alpha}\in\mathsf{N}^{\#}}$ such that, for each $\vec{\alpha}$:

I. $\mathcal{M}^{\vec{\alpha}}$ has finite products, with 1 being the terminal object and $A \otimes B$ the product of A and B.

II. $\mathcal{M}^{\vec{\alpha}}$ and $(T, \eta, \mu, \tau)^{\vec{\alpha}}$ form a λ_c-model (v. [11]). The T-exponential TB^A is denoted by $A \Rrightarrow TB$.

III. $\mathcal{M}^{\vec{\alpha}}$ contains a natural numbers object $\mathbb{N}$ equipped with successor/predecessor arrows and $\tilde{n} : 1 \to \mathbb{N}$, each $n \in \mathbb{N}$.

IV. $\mathcal{M}^{\vec{\alpha}}$ contains, for each $A \in \mathrm{TY}$, an A-names object N_A, a $\tilde{0}/\tilde{1}$-valued name-equality arrow $\mathtt{eq}_A : \mathsf{N}_A \otimes \mathsf{N}_A \to \mathbb{N}$, and, for each $\alpha \in (\mathsf{N}_A \cap \mathsf{S}(\vec{\alpha}))$, an arrow $\alpha : 1 \to \mathsf{N}_A$.

These make
$$\begin{tikzcd} \mathsf{N}_A \arrow[r, "\Delta"] \arrow[d, "!"'] & \mathsf{N}_A \otimes \mathsf{N}_A \arrow[d, "\mathtt{eq}_A"] \\ 1 \arrow[r, "\tilde{0}"'] & \mathbb{N} \end{tikzcd}$$
a pullback.

V. Taking $[\![\mathbf{1}]\!] \triangleq 1$, $[\![\mathbb{N}]\!] \triangleq \mathbb{N}$, $[\![[A]]\!] \triangleq \mathsf{N}_A$, $[\![A \to B]\!] \triangleq [\![A]\!] \Rrightarrow T[\![B]\!]$ and $[\![A \otimes B]\!] \triangleq [\![A]\!] \otimes [\![B]\!]$, $\mathcal{M}^{\vec{\alpha}}$ contains, for each $A \in \mathrm{TY}$, arrows

$$\mathtt{drf}_A : \mathsf{N}_A \to T[\![A]\!] \quad \text{and} \quad \mathtt{upd}_A : \mathsf{N}_A \otimes [\![A]\!] \to T1$$

such that, for $\alpha\#\beta$ and $\mathtt{upd}^{\alpha}_A \triangleq \langle !; \alpha, \mathtt{id}\rangle \,;\mathtt{upd}_A$, the following diagrams (which describe the specifications for *dereferencing* and *update*) commute.

$$\begin{tikzcd} \left[\!\left[A\right]\!\right] \arrow[r, "\langle \mathtt{id}, \mathtt{upd}^{\alpha}_A\rangle"] & \left[\!\left[A\right]\!\right] \otimes T1 \arrow[r, "\pi_2\,;T\alpha"] \arrow[d, "\tau\,;\cong"] & T\mathsf{N}_A \arrow[d, "T\mathtt{drf}_A"] \\ & T\left[\!\left[A\right]\!\right] & TT\left[\!\left[A\right]\!\right] \arrow[l, "\mu"'] \end{tikzcd}$$

$$\begin{tikzcd} \left[\!\left[A\right]\!\right] \otimes \left[\!\left[B\right]\!\right] \arrow[r, "\mathtt{id}\otimes\mathtt{upd}^{\beta}_B"] \arrow[d, "\mathtt{upd}^{\alpha}_A\otimes\mathtt{id}"] & \left[\!\left[A\right]\!\right] \otimes T1 \arrow[r, "\tau\,;\cong"] & T\left[\!\left[A\right]\!\right] \arrow[d, "T\mathtt{upd}^{\alpha}_A"] & \\ T1 \otimes \left[\!\left[B\right]\!\right] \arrow[r, "\tau'\,;\cong"] & T\left[\!\left[B\right]\!\right] \arrow[r, "T\mathtt{upd}^{\beta}_B"] & TT1 \arrow[r, "\mu"] & T1 \end{tikzcd} \qquad \text{(NR)}$$

$$\begin{tikzcd} \left[\!\left[A\right]\!\right] \otimes \left[\!\left[A\right]\!\right] \arrow[r, "\mathtt{id}\otimes\mathtt{upd}^{\alpha}_A"] \arrow[d, "\pi_1"] & \left[\!\left[A\right]\!\right] \otimes T1 \arrow[r, "\tau\,;\cong"] & T\left[\!\left[A\right]\!\right] \arrow[d, "T\mathtt{upd}^{\alpha}_A"] \\ \left[\!\left[A\right]\!\right] \arrow[r, "\mathtt{upd}^{\alpha}_A"] & T1 & TT1 \arrow[l, "\mu"'] \end{tikzcd}$$

Moreover, $Ob(\mathcal{M}^{\vec{\alpha}})$ is a nominal set with equivariant elements and all $\mathcal{M}^{\vec{\alpha}}$'s contain the same objects, so we let $Ob(\mathcal{M}) \triangleq Ob(\mathcal{M}^{\vec{\alpha}})$, any $\vec{\alpha}$. For each $A, B \in Ob(\mathcal{M})$ there exists a nominal set $\mathcal{M}(A,B)$, such that

$$\mathcal{M}^{\vec{\alpha}}(A,B) = \{(x,\vec{\alpha}) \mid x \in \mathcal{M}(A,B) \wedge \mathsf{S}(x) \subseteq \mathsf{S}(\vec{\alpha})\}$$

We write $f = ((f)^\circ, \vec{\alpha})$, each $f \in \mathcal{M}^{\vec{\alpha}}(A,B)$. Moreover, the structure defined in I-V above is equivariant in the following sense:

$$\begin{array}{ccc} \mathsf{S}((\mathtt{id}_A^{(\vec{\alpha})})^\circ) = \varnothing & \wedge & ((\mathtt{id}_A^{(\vec{\alpha})})^\circ, \vec{\beta}) = \mathtt{id}_A^{(\vec{\beta})} \\ \mathsf{S}((\eta_A^{(\vec{\alpha})})^\circ) = \varnothing & \wedge & ((\eta_A^{(\vec{\alpha})})^\circ, \vec{\beta}) = \eta_A^{(\vec{\beta})} \\ & etc. & \\ \mathsf{S}((\alpha^{(\vec{\alpha})})^\circ) = \{\alpha\} & \wedge & ((\alpha^{(\vec{\alpha})})^\circ, \vec{\beta}) = \alpha^{(\vec{\beta})} \quad (\text{if } \alpha \not\# \vec{\beta}) \end{array}$$

Also, for each $\alpha \# \vec{\alpha}$ and each A, B, the nominal mapping

$$(\_)^{+\alpha} : \mathcal{M}^{\vec{\alpha}}(A,B) \to \mathcal{M}^{\vec{\alpha},\alpha}(A,B) \triangleq (f,\vec{\alpha}) \mapsto (f,\vec{\alpha},\alpha)$$

is functorial and commutes with pairing, currying and T. Finally, there exists a nominal mapping

$$\langle\!\langle\alpha\rangle\!\rangle(\_) : \mathcal{M}^{\vec{\alpha},\alpha}(A,TB) \to \mathcal{M}^{\vec{\alpha}}(A,TB)$$

such that, for all relevant f, g, β, the SN-diagrams commute:

$$f \,;\langle\!\langle\alpha\rangle\!\rangle g = \langle\!\langle\alpha\rangle\!\rangle(f^{+\alpha}\,; g) \qquad \langle\!\langle\alpha\rangle\!\rangle f \,; Tg = \langle\!\langle\alpha\rangle\!\rangle(f\,; T(g^{+\alpha}))$$
$$\langle\!\langle\alpha\rangle\!\rangle f \,; \mu = \langle\!\langle\alpha\rangle\!\rangle(f\,;\mu) \qquad (\mathtt{id}\otimes\langle\!\langle\alpha\rangle\!\rangle f)\,;\tau = \langle\!\langle\alpha\rangle\!\rangle((\mathtt{id}\otimes f)\,;\tau)$$
$$(\mathtt{upd}_B^\beta \otimes \langle\!\langle\alpha\rangle\!\rangle f)\,;\psi = \langle\!\langle\alpha\rangle\!\rangle((\mathtt{upd}_B^\beta\otimes f)\,;\psi)$$

(where $\psi = \tau' ; T\tau ; \mu$, see [11]). $\_^{+\alpha}$ is *name-addition* and $\langle\!\langle\alpha\rangle\!\rangle\_$ is *name-abstraction*, not to be confused with nominal name-abstraction $\langle\alpha\rangle\_$. ▲

Our semantics is cast inside $\mathbf{Nom}_{\mathrm{TY}}$. The reason for describing morphisms as pairs $(x, \vec{\alpha})$ comes from the fact that we give semantic translations of sequents, not terms, and sequents may contain superfluous names in their name-environments. Thus, if f models $\vec{\alpha} \mid \Gamma \vdash M : A$ then $(f)^\circ$ models $\vec{\alpha}' \mid \Gamma \vdash M : A$, where $\vec{\alpha}'$ is $\vec{\alpha}$ with all names that are fresh for M removed. Moreover, this description allows us to form a family of categories that *have essentially the same structure*, and gives us a means to relate the semantics of sequents like $\vec{\alpha} \mid \Gamma \vdash M : A$ and $\vec{\alpha}, \alpha \mid \Gamma \vdash M : A$.

Recall that in a λ_c-model $\mathcal{M}^{\vec{\alpha}}$ there exists, for each A, B, C, a bijection natural in A:

$$\Lambda^{T,B}_{A,C} : \mathcal{M}^{\vec{\alpha}}(A\otimes B, TC) \xrightarrow{\cong} \mathcal{M}^{\vec{\alpha}}(A, B \Rrightarrow TC)$$

Let $\mathsf{ev}^T_{A,B} : (A \Rrightarrow TB) \otimes A \to TB \triangleq (\Lambda^T)^{-1}(\mathtt{id}_{A\Rrightarrow TB})$. We give the semantics of $\nu\rho$ in a $\lambda_{\nu\rho}$-model $\mathcal{M}$.

Definition 5 Let $\langle \mathcal{M}^{\vec{\alpha}}, T^{\vec{\alpha}} \rangle_{\vec{\alpha}\in\mathsf{N}^\#}$ be a $\lambda_{\nu\rho}$-model. A typing judgement $\vec{\alpha} \mid \Gamma \vdash M : A$ is translated into an arrow $[\![M]\!] : [\![\Gamma]\!] \to T[\![A]\!]$ in $\mathcal{M}^{\vec{\alpha}}$ as follows.

$$[\![\tilde{n}]\!] : \Gamma \xrightarrow{!} 1 \xrightarrow{\tilde{n}\,;\,\eta} T\mathbb{N} \qquad [\![\alpha]\!] : \Gamma \xrightarrow{!} 1 \xrightarrow{\alpha\,;\,\eta} T\mathsf{N}_A$$

$$\frac{[\![M]\!] : \Gamma\otimes A \to TB}{[\![\lambda x.M]\!] : \Gamma \xrightarrow{\Lambda^T([\![M]\!])} A \Rrightarrow TB \xrightarrow{\eta} T(A \Rrightarrow TB)}$$

$$\frac{[\![M]\!] : \Gamma \to T(A\Rrightarrow TB) \qquad [\![N]\!] : \Gamma \to TA}{[\![M\,N]\!] : \Gamma \xrightarrow{\langle [\![M]\!], [\![N]\!]\rangle\,;\,\psi} T((A\Rrightarrow TB)\otimes A) \xrightarrow{T\mathsf{ev}^T;\,\mu} TB}$$

$$\frac{[\![M]\!] : \Gamma \to TA}{[\![\nu\alpha.M]\!] = \langle\!\langle\alpha\rangle\!\rangle[\![M]\!] : \Gamma \to TA}$$

$$\frac{[\![M]\!] : \Gamma \to T\mathsf{N}_A \qquad [\![N]\!] : \Gamma \to T\mathsf{N}_A}{[\![[M=N]]\!] : \Gamma \xrightarrow{\langle [\![M]\!], [\![N]\!]\rangle\,;\,\psi} T(\mathsf{N}_A\otimes\mathsf{N}_A) \xrightarrow{T\mathtt{eq}} T\mathbb{N}}$$

$$\frac{[\![M]\!] : \Gamma \to T\mathsf{N}_A \qquad [\![N]\!] : \Gamma \to TA}{[\![M := N]\!] : \Gamma \xrightarrow{\langle [\![M]\!], [\![N]\!]\rangle\,;\,\psi} T(\mathsf{N}_A\otimes A) \xrightarrow{T\mathtt{upd}_A\,;\,\mu} T1}$$

$$\frac{[\![M]\!] : \Gamma \to T\mathsf{N}_A}{[\![!M]\!] : \Gamma \xrightarrow{[\![M]\!]} T\mathsf{N}_A \xrightarrow{T\mathtt{drf}_A} TTA \xrightarrow{\mu} TA}$$

plus standard translations for other term constructors. ▲

We proceed to show correctness. Note that we write $S \vDash M \xrightarrow{\mathrm{r}} S' \vDash M'$, with r being a reduction rule different from CTX, if the non-CTX rule in the related derivation is r. We write $M\,;N$ for the term $(\lambda d.N)M$, some d not in N, and relate to any store S a term $\bar{S}$ of type $\mathbf{1}$, by: $\bar{\epsilon} \triangleq \mathtt{skip}$, $\overline{\alpha, S} \triangleq \bar{S}$, $\overline{\alpha :: V, S} \triangleq (\alpha := V\,; \bar{S})$.

Proposition 6 (Correctness) *For any* $\vec{\alpha} \mid \Gamma \vdash M : A$, *any* S *with* $\mathtt{dom}(S) = \vec{\alpha}$ *and any* $\mathrm{r} \neq \mathrm{NEW}$,

- $S \vDash M \xrightarrow{\mathrm{r}} S' \vDash M' \implies [\![\bar{S}\,; M]\!] = [\![\bar{S}'\,; M']\!]$,
- $S \vDash M \xrightarrow{\mathrm{NEW}} S, \alpha \vDash M' \implies [\![\bar{S}\,; M]\!] = \langle\!\langle\alpha\rangle\!\rangle[\![\bar{S}\,; M']\!]$.

Hence, $S \vDash M \twoheadrightarrow S' \vDash M' \implies [\![\nu\vec{\alpha}.(\bar{S}\,;M)]\!] = [\![\nu\vec{\alpha}'.(\bar{S}'\,;M')]\!]$, *with* $\mathtt{dom}(S') = \vec{\alpha}'$. ■

Soundness does not follow from correctness; we need to add an adequacy specification.

Definition 7 (Adequacy) Let $\mathcal{M} = \langle \mathcal{M}^{\vec{\alpha}}, T^{\vec{\alpha}} \rangle_{\vec{\alpha}\in\mathsf{N}^\#}$ be a $\lambda_{\nu\rho}$-model and $[\![\_]\!]$ the respective translation of $\nu\rho$. $\mathcal{M}$ is adequate if, for any typed term $\vec{\alpha} \mid \varnothing \vdash M : \mathbb{N}$, if $[\![M]\!] = \langle\!\langle\vec{\beta}\rangle\!\rangle[\![\bar{S}\,;\tilde{0}]\!]$, some S, then there exists S' such that $\vec{\alpha} \vDash M \twoheadrightarrow S' \vDash \tilde{0}$. ▲

Assume now our running $\mathcal{M}$ is an adequate $\lambda_{\nu\rho}$-model.

Proposition 8 (Equational Soundness)

$$[\![M]\!] = [\![N]\!] \implies M \lessapprox N \qquad ■$$

Completeness. To achieve completeness we need to introduce a preorder in the semantics to match the observational preorder of the syntax, as in (FA). This step, which is essentially a *quotiening* procedure, is found in many (but by no means all) fully abstract models based on game semantics.

Definition 9 (p-Observationality) An adequate $\lambda_{\nu\rho}$-model $\mathcal{M}$ is p(reorder)-observational if, for all $\vec{\alpha}$:

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

$$[\![M]\!] \in O^{\vec{\alpha}} \iff \exists S, \vec{\beta}.\, [\![M]\!] = \langle\!\langle\vec{\beta}\rangle\!\rangle[\![\bar{S}\,;\tilde{0}]\!] \tag{9a}$$

- The induced intrinsic preorder on arrows in $\mathcal{M}^{\vec{\alpha}}(A, TB)$, defined by $f \lesssim^{\vec{\alpha}} g \iff$

$$\forall \rho : A \Rightarrow TB \to T\mathbb{N}.\, (\Lambda^T(f)\,;\rho \in O^{\vec{\alpha}} \Longrightarrow \Lambda^T(g)\,;\rho \in O^{\vec{\alpha}})$$

satisfies, for all $\alpha \# \vec{\alpha}$ and relevant f, g, f', g',

$$f \lesssim^{\vec{\alpha}} g \implies f^{+\alpha} \lesssim^{\vec{\alpha},\alpha} g^{+\alpha} \tag{9b}$$

$$f' \lesssim^{\vec{\alpha},\alpha} g' \implies \langle\!\langle\alpha\rangle\!\rangle f' \lesssim^{\vec{\alpha}} \langle\!\langle\alpha\rangle\!\rangle g' \tag{9c}$$

We write $\mathcal{M}$ as $\langle \mathcal{M}^{\vec{\alpha}}, T^{\vec{\alpha}}, O^{\vec{\alpha}}, \lesssim^{\vec{\alpha}} \rangle_{\vec{\alpha} \in \mathsf{N}^{\#}}$. ▲

So $O^{\vec{\alpha}}$ contains those arrows that have an *observable behavior* in the model, and the semantic preorder is built around this notion. In particular, due to (9a), terms that yield $\tilde{0}$ have observable behavior. On the other hand, by (9b) and (9c) we have that $\lesssim$ is a *congruence*, i.e. it passes through contexts. Now assume our running model is p-observational.

Lemma 10 (Inequational Soundness)

$$[\![M]\!] \lesssim [\![N]\!] \implies M \lessapprox N$$ ■

In order to achieve completeness, and hence full-abstraction, we need some definability requirement.

Definition 11 (p-Definability) A p-observational $\lambda_{\nu\rho}$-model $\mathcal{M}$ satisfies p-definability if, for any $\vec{\alpha}, A, B$, there exists $D^{\vec{\alpha}}_{A,B} \subseteq \mathcal{M}^{\vec{\alpha}}([\![A]\!], T[\![B]\!])$ such that:

- For each $f \in D^{\vec{\alpha}}_{A,B}$ there exists term M with $[\![M]\!] = f$
- For any $f, g \in \mathcal{M}^{\vec{\alpha}}([\![A]\!], T[\![B]\!])$, $f \lesssim^{\vec{\alpha}} g$ iff $\forall \rho \in D^{\vec{\alpha}}_{A \to B, \mathbb{N}}\,.\, (\Lambda^T(f)\,;\rho \in O^{\vec{\alpha}} \Longrightarrow \Lambda^T(g)\,;\rho \in O^{\vec{\alpha}})$ ▲

p-Definability states that *definable test-arrows suffice* for defining the semantic preorder. Now assume our model satisfies p-definability.

Proposition 12 (Full-Abstraction)

$$[\![M]\!] \lesssim [\![N]\!] \iff M \lessapprox N$$

Proof: Soundness is by previous lemma. For completeness ("$\Longleftarrow$"), assume $\vec{\alpha} \mid \Gamma \vdash M \lessapprox N$; we do induction on the size of Γ. The base case of $\Gamma = \varnothing$ is encompassed in the case of $\Gamma = \{x : A\}$ –just add a dummy x, which we now show. Suppose $\vec{\alpha} \mid x : A \vdash M \lessapprox N$ and take any $\rho \in D^{\vec{\alpha}}_{A \to B, \mathbb{N}}$ such that $\Lambda^T([\![M]\!])\,;\rho \in O^{\vec{\alpha}}$. Let $\rho = [\![\vec{\alpha} \mid y : A \to B \vdash L]\!]$, some L, so $\Lambda^T([\![M]\!])\,;\rho$ is $\Lambda^T([\![M]\!])\,;[\![L]\!] = |\lambda x.M|\,;[\![L]\!] = [\![(\lambda y.L)(\lambda x.M)]\!]$. The latter being in $O^{\vec{\alpha}}$ implies that it equals $\langle\!\langle\vec{\beta}\rangle\!\rangle[\![\bar{S}\,;\tilde{0}]\!]$, some S. Now, $M \lessapprox N$ implies $(\lambda y.L)(\lambda x.M) \lessapprox (\lambda y.L)(\lambda x.N)$, hence $\nu\vec{\beta}.(\bar{S}\,;\tilde{0}) \lessapprox (\lambda y.L)(\lambda x.N)$, by soundness. But this implies that $\vec{\alpha} \vDash (\lambda y.L)(\lambda x.N) \twoheadrightarrow S' \vDash \tilde{0}$, so $[\![(\lambda y.L)(\lambda x.N)]\!] \in O^{\vec{\alpha}}$, by correctness. Hence, $\Lambda^T([\![N]\!])\,;\rho \in O^{\vec{\alpha}}$, so $[\![M]\!] \lesssim [\![N]\!]$, by p-definability.
For the inductive step, let $\Gamma = x_1 : A_1, x_2 : A_2, \Gamma'$, and let z not appear in Γ; if $\vec{\alpha} \mid \Gamma \vdash M \lessapprox N$ then

$$\vec{\alpha} \mid z : A_1 \otimes A_2, \Gamma' \vdash ((\lambda xy.M)\mathtt{fst}\,(z))\mathtt{snd}\,(z) \lessapprox ((\lambda xy.N)\mathtt{fst}\,(z))\mathtt{snd}\,(z)$$

$$\overset{IH}{\therefore}\ [\![\vec{\alpha} \mid z : A_1 \otimes A_2, \Gamma' \vdash ((\lambda xy.M)\mathtt{fst}\,(z))\mathtt{snd}\,(z)]\!] = [\![\vec{\alpha} \mid \Gamma \vdash M]\!]$$

$$\wr\wedge$$

$$[\![\vec{\alpha} \mid z : A_1 \otimes A_2, \Gamma' \vdash ((\lambda xy.N)\mathtt{fst}\,(z))\mathtt{snd}\,(z)]\!] = [\![\vec{\alpha} \mid \Gamma \vdash N]\!]$$

as required. ■

4 The nominal games model

We embark on the adventure of modeling $\nu\rho$ in a category of nominal arenas and strategies. Our presentation of nominal games rectifies the one presented in [2] by using name-lists instead of name-sets and introducing innocent plays. The basic category construction will be $\mathcal{V}^{\vec{\alpha}}_{\mathtt{t}}$, the category of nominal arenas and total $\vec{\alpha}$-strategies. $\mathcal{V}^{\vec{\alpha}}_{\mathtt{t}}$ will be constructed in $\mathbf{Nom}_{\mathrm{TY}}$, so there will be, for each type A, an arena N_A for references to type A. The translation $[\![A]\!]$ of a general type will make use of a *store arena* $\xi \triangleq \bigotimes_A (\mathsf{N}_A \Rightarrow [\![A]\!])$, which will literally serve as a reference store. This will naturally lead us to a monadic semantics, with computation monad T defined (on arenas) by $TA \triangleq \xi \Rightarrow A \otimes \xi$. Since arrow types involve the monad in their translation and the monad involves all types, we will have to first solve the domain equation:[5]

$$\begin{aligned} [\![A \to B]\!] &= [\![A]\!] \Rightarrow (\xi \Rightarrow [\![B]\!] \otimes \xi) \\ \xi &= \bigotimes_A (\mathsf{N}_A \Rightarrow [\![A]\!]) \end{aligned} \tag{SE}$$

Construct a category. We assume a set of types TY and build our constructions inside $\mathbf{Nom}_{\mathrm{TY}}$. We start with nominal arenas and prearenas. Arenas will be used for type-translation, while terms will be translated to *strategies between arenas*, i.e. strategies for prearenas.

Definition 13 A nominal arena $A \triangleq (M_A, \vdash_A, \lambda_A)$ contains:

- A nominal set of moves M_A, the elements of which have strong support.[6] Moves are denoted by $m, n, \ldots$.
- A nominal justification relation $\vdash_A \subseteq (M_A + \{\dagger\}) \times M_A$.
- A nominal labeling function $\lambda_A : M_A \to \{O, P\} \times \{Q, A\}$, so that each move can be played by Opponent or Player, and is a Question or an Answer.

[5] Notation will be clarified below.

[6] Strong support is essential in proving basic properties of nominal games, e.g. that these form a category, yet their proofs are omitted here.

These satisfy the conditions:

(f) For each $m \in M_A$, there exists unique $k \geq 0$ such that $\dagger \vdash_A m_1 \vdash_A \cdots \vdash_A m_k \vdash_A m$, for some m_l's in M_A. k is called the level of m.
Level-0 moves, denoted by i, i', ..., are called *initial*.

(l1) Initial moves are P-Answers.

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

(l3) Answers may only justify Questions.

A *prearena* is an arena with its initial moves labeled OQ. Given arenas A and B, construct the prearena $A \to B$ as:

$$M_{A\to B} \triangleq M_A + M_B$$
$$\lambda_{A\to B} \triangleq [(i_A \mapsto OQ\ ,\ \overline{i_A} \mapsto \overline{\lambda_A}(m_A))\ ,\ \lambda_B]$$
$$\vdash_{A\to B} \triangleq \{(\dagger, i_A), (i_A, i_B)\} \cup \{(m,n) \mid m \vdash_{A,B} n\}$$ ▲

I_A is the set of ***initial*** (level-0) moves of A, and J_A the set of ***jnitial*** (level-1) moves. Then, $\overline{I_A} = M_A \setminus I_A$, and $\overline{J_A} = M_A \setminus J_A$. In general, we use m_A to denote moves in M_A, i_A for moves in I_A, $\overline{i_A}$ for moves in $\overline{I_A}$, j_A for moves in J_A, etc. Finally, $\overline{\lambda_A}$ denotes the OP-complement of λ_A.

Condition (f) states that arenas can be represented by directed connected graphs with no directed cycles. Note that the nominal arenas of [2] do satisfy the above conditions, although a different set of conditions is used there.

Example 14 (Basic arenas) *The simplest arena is* $0 = (\varnothing, \varnothing, \varnothing)$. *Now let A be an arbitrary type. Define the (flat) arenas N_A, $\mathbb{N}$ and 1 as follows.*

$$M_{\mathsf{N}_A} \triangleq \mathsf{N}_A \qquad M_{\mathbb{N}} \triangleq \mathbb{N} \qquad M_1 \triangleq \{*\}$$
$$\lambda_{\mathsf{N}_A}(m) \triangleq PA \qquad \lambda_{\mathbb{N}}(m) \triangleq PA \qquad \lambda_1(*) \triangleq PA$$
$$\vdash_{\mathsf{N}_A} \triangleq \{(\dagger, m)\} \qquad \vdash_{\mathbb{N}} \triangleq \{(\dagger, m)\} \qquad \vdash_1 \triangleq \{(\dagger, *)\}$$

Nominal games are played using sequences of *moves-with-names*, that is moves attached with name-lists. Name-lists capture name-environments; this idea of attaching state-information explicitly to moves first appeared in [12] and was later followed in [2].

Definition 15 A move-with-names of a (pre)arena A is a pair, $m^{\vec{\alpha}}$, where $m \in M_A$ and $\vec{\alpha} \in \mathsf{N}^{\#}$ (i.e. $\vec{\alpha}$ a name-list). Writing $m^{\vec{\alpha}}$ as x, we have $\underline{x} \triangleq m$ and $\mathsf{nlist}(x) \triangleq \vec{\alpha}$. ▲

At this point, let us introduce some handy notation for sequences. Let s, t be sequences, then:

- $s \leq t$ denotes that s is a prefix of t, and then $t = s(t-s)$,
- s^- denotes s with its last element removed,
- if $s = s_1 \cdots s_n$ then
  - n is the *length* of s, and is denoted by $|s|$,
  - $s.i$ denotes s_i and $s.\text{-}i$ denotes s_{n+1-i}, e.g. $s.\text{-}1$ is s_n,
  - $s_{\leq s_i}$ denotes $s_1 \cdots s_i$, and so does $s_{<s_{i+1}}$.

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* $s.i$. We can now proceed to plays.

Definition 16 (Plays) Let A be a prearena. *A legal sequence* on A is a justified sequence of moves-with-names that satisfies Visibility and Well-Bracketing (v. [10, 7]). A legal sequence s is ***a play*** if it 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 its preceding O-move. It possibly contains 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 P-move explicitly justifying it.

An $\vec{\alpha}$-play is a play that opens with a move with name-list $\vec{\alpha}$. The set of $\vec{\alpha}$-plays on a prearena A is denoted by $P_A^{\vec{\alpha}}$. ▲

With s and x as above, P ***introduces a name*** α at x iff $\alpha \not\# x$ and $\alpha \# s_{<x}$. $\mathcal{L}(s)$ contains all names introduced by P in s. Note also that, for any move x in an $\vec{\alpha}$-play, $\vec{\alpha} \leq \mathsf{nlist}(x)$. We proceed to strategies.

Definition 17 (Strategies) An $\vec{\alpha}$-strategy σ is a set of equivalence classes $[s]_{\vec{\alpha}}$ of $\vec{\alpha}$-plays, written $[s]$, satisfying *prefix closure, contingency completeness* and *determinacy*:

- If $[su] \in \sigma$ then $[s] \in \sigma$.
- If even-length $[s] \in \sigma$ and sx is an $\vec{\alpha}$-play then $[sx] \in \sigma$.
- If even-length $[s_1 x_1], [s_2 x_2] \in \sigma$ and $[s_1] = [s_2]$ then $[s_1 x_1] = [s_2 x_2]$.

An $\vec{\alpha}$-strategy σ on $A \to B$ is written $\sigma : A \to B$. ▲

For example, for $\alpha \not\# \vec{\alpha}$ and $n \in \mathbb{N}$, define the $\vec{\alpha}$-strategies:

$$\alpha : 1 \to \mathsf{N}_A \triangleq \{[*^{\vec{\alpha}} \alpha^{\vec{\alpha}}]\} \quad \text{and} \quad \tilde{n} : 1 \to \mathbb{N} \triangleq \{[*^{\vec{\alpha}} n^{\vec{\alpha}}]\}$$

Note that $\vec{\alpha}$-strategies have (strong) support $\mathsf{S}(\vec{\alpha})$. We define play- and strategy-composition building on [6, 10]. We let $\underline{s}$ be s without its name-lists, and $\underline{s}^{\mathsf{nlist}(s)}$ be s.

Definition 18 (Composable plays) Let $s \in P_{A\to B}^{\vec{\alpha}}$ and $t \in P_{B\to C}^{\vec{\alpha}}$. These are *almost composable*, $s \smile t$, if $\underline{s} \upharpoonright B = \underline{t} \upharpoonright B$. They are *composable*, $s \asymp t$, if $s \smile t$ and, for any $s' \leq s$ and $t' \leq t$ with $s' \smile t'$,

(C1) If s' ends in a P-move in A introducing some name α then $\alpha \# t'$; dually, if t' ends in a P-move in C introducing some name α then $\alpha \# s'$.

(C2) If both s', t' end in B and s' ends in a P-move introducing some name α then $\alpha \# t'^-$; dually, if t' ends in a P-move introducing some name α then $\alpha \# s'^-$. ▲

If $s \in P^{\vec{\alpha}}_{A\to B}$ and $t \in P^{\vec{\alpha}}_{B\to C}$ are composable then either $s \upharpoonright B = t = \epsilon$, 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 (cf. *zipper lemma* of [6]). In the following we state that m is an O-move by writing $m_{(O)}$, and similarly for P-moves.

Definition 19 (Composition) Let $s \in P^{\vec{\alpha}}_{A\to B}$ and $t \in P^{\vec{\alpha}}_{B\to C}$ with $s \asymp t$. Their parallel interaction $s \parallel t$ and their mix $s \bullet t$, which returns the name-list of the final move in $s \parallel t$, are defined by mutual induction as below.

$$\epsilon \parallel \epsilon \triangleq \epsilon \qquad sm^{\vec{\beta}}_B \parallel tm^{\vec{\gamma}}_B \triangleq (s \parallel t)m_B^{sm^{\vec{\beta}}_B \bullet tm^{\vec{\gamma}}_B}$$

$$s \parallel tm^{\vec{\gamma}}_C \triangleq (s \parallel t)m_C^{s \bullet tm^{\vec{\gamma}}_C} \qquad sm^{\vec{\beta}}_A \parallel t \triangleq (s \parallel t)m_A^{sm^{\vec{\beta}}_A \bullet t}$$

$$s \bullet tm^{\vec{\gamma}}_{C(O)} \triangleq \vec{\gamma}'' \qquad sm^{\vec{\beta}}_{A(O)} \bullet t \triangleq \vec{\beta}''$$

$$sm^{\vec{\beta}}_{A(P)} \bullet t \triangleq s \bullet t, \vec{\beta}' \qquad sm^{\vec{\beta}}_{B(P)} \bullet tm^{\vec{\gamma}}_{B(O)} \triangleq s \bullet t, \vec{\beta}'$$

$$s \bullet tm^{\vec{\gamma}}_{C(P)} \triangleq s \bullet t, \vec{\gamma}' \qquad sm^{\vec{\beta}}_{B(O)} \bullet tm^{\vec{\gamma}}_{B(P)} \triangleq s \bullet t, \vec{\gamma}'$$

where $\vec{\beta}'$ is $\vec{\beta} - \mathsf{nlist}(s.\text{-}1)$, and $\vec{\beta}''$ is the name-list of $m_{A(O)}$'s justifier in $s \parallel t$; similarly for $\vec{\gamma}', \vec{\gamma}''$.
The composition of s and t is: $s\,;t \triangleq (s \parallel t) \upharpoonright AC$.

For $\vec{\alpha}$-strategies $\sigma : A \to B$ and $\tau : B \to C$, their composition is: $\sigma\,;\tau \triangleq \{\, [s\,;t] \mid [s] \in \sigma \wedge [t] \in \tau \wedge s \asymp t \,\}$. ▲

Proposition 20 *If $s \in P^{\vec{\alpha}}_{A\to B}$ and $t \in P^{\vec{\alpha}}_{B\to C}$ with $s \asymp t$, then $s\,;t \in P^{\vec{\alpha}}_{A\to C}$.*
If $\sigma : A \to B$ and $\tau : B \to C$ are $\vec{\alpha}$-strategies then so is $\sigma\,;\tau$. Moreover, if $\sigma_1 : A_1 \to A_2$, $\sigma_2 : A_2 \to A_3$ and $\sigma_3 : A_3 \to A_4$ are $\vec{\alpha}$-strategies then $(\sigma_1\,;\sigma_2)\,;\sigma_3 = \sigma_1\,;(\sigma_2\,;\sigma_3)$. ■

We are interested in *innocent strategies*, that is strategies in which P-moves depend solely on current P-views. Recall that the P-view, $\ulcorner s \urcorner$, of a justified sequence s is:

$$\ulcorner sx \urcorner \triangleq \ulcorner s \urcorner x \qquad \text{if } x \text{ a P-move}$$

$$\ulcorner x \urcorner \triangleq x \qquad \text{if } x \text{ is initial}$$

$$\ulcorner sxs'y \urcorner \triangleq \ulcorner s \urcorner xy \qquad \text{if } y \text{ an O-move justified by } x$$

Note that the P-view of a play is *not* necessarily itself a play; hence, we further restrict plays.

Definition 21 A play s is *innocent* if, for any $t \le s$, $\ulcorner t \urcorner$ is a play. ▲

It is not difficult to see that innocent plays are legal sequences satisfying (NC1), (NC3) and (NC2′), where

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

From innocent plays we move on to innocent strategies.

Definition 22 An $\vec{\alpha}$-strategy σ is innocent if $[s] \in \sigma$ implies that s is innocent, and if even-length $[s_1 n_1^{\vec{\gamma}_1}] \in \sigma$ and odd-length $[s_2] \in \sigma$ have $[\ulcorner s_1 \urcorner] = [\ulcorner s_2 \urcorner]$ then there exists $n_2^{\vec{\gamma}_2}$ such that $[s_2 n_2^{\vec{\gamma}_2}] \in \sigma$ and $[\ulcorner s_1 n_1^{\vec{\gamma}_1} \urcorner] = [\ulcorner s_2 n_2^{\vec{\gamma}_2} \urcorner]$. ▲

Proposition 23 *If $s \in P^{\vec{\alpha}}_{A\to B}$, $t \in P^{\vec{\alpha}}_{B\to C}$ are innocent and $s \asymp t$ then $s\,;t$ is innocent. If $\sigma : A \to B, \tau : B \to C$ are innocent $\vec{\alpha}$-strategies then so is $\sigma\,;\tau$.* ■

We can now define our basic category of nominal games.

Definition 24 ($\mathcal{V}^{\vec{\alpha}}_{\mathtt{t}}, \mathcal{V}^{\vec{\alpha}}_{\mathtt{tt}}$) An innocent $\vec{\alpha}$-strategy $f : A \to B$ is *total* if for any $[i^{\vec{\alpha}}_A] \in f$ there exists $[i^{\vec{\alpha}}_A i^{\vec{\alpha}}_B] \in f$.
It is also *ttotal* if for any $[i^{\vec{\alpha}}_A i^{\vec{\alpha}}_B j^{\vec{\alpha}}_B] \in f$ there exists $[i^{\vec{\alpha}}_A i^{\vec{\alpha}}_B j^{\vec{\alpha}}_B j^{\vec{\alpha}}_A] \in f$, and whenever $[s j'^{\vec{\alpha}'}_A] \in f$ then $\underline{s.\text{-}1} \in J_B$.
$\mathcal{V}^{\vec{\alpha}}_{\mathtt{t}}$ is the category of nominal arenas and total $\vec{\alpha}$-strategies, and $\mathcal{V}^{\vec{\alpha}}_{\mathtt{tt}}$ is its lluf subcategory of ttotal strategies. ▲

Thus, a strategy is total iff it immediately answers any initial question without introducing new names. It is ttotal iff it follows a similar pattern for jnitial moves. Now, innocent strategies are conveniently represented using *viewfunctions*.

Definition 25 An $\vec{\alpha}$-viewfunction f is a set of equivalence classes of innocent $\vec{\alpha}$-plays that are even-length P-views, which satisfies *even-prefix closure* and *single-valuedness*:

- If $[s] \in f$ and t is an even-length prefix of s then $[t] \in f$.
- If $[s_1 x_1], [s_2 x_2] \in f$ and $[s_1] = [s_2]$ then $[s_1 x_1] = [s_2 x_2]$. ▲

There are maps `viewf` and `strat` from innocent $\vec{\alpha}$-strategies to $\vec{\alpha}$-viewfunctions and viceversa, such that $f = \mathtt{viewf}(\mathtt{strat}(f))$, and $\sigma = \mathtt{strat}(\mathtt{viewf}(\sigma))$. From now on, will be defining strategies via their viewfunctions.

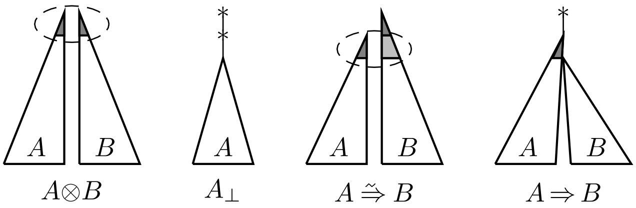
Constructions in $\mathcal{V}^{\vec{\alpha}}_{\mathtt{t}}$. In $\mathcal{V}^{\vec{\alpha}}_{\mathtt{t}}$ we construct tensor product, lifting and function space arenas as follows. For nominal arenas A, B, define $A \otimes B$, $A_\perp$, $A \mathrel{\tilde{\Rrightarrow}} B$:

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

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

$$\begin{aligned} M_{A \tilde{\Rrightarrow} B} &\triangleq I_B + I_A \times J_B + \overline{I_A} + \overline{I_B} \cap \overline{J_B} && (A \mathrel{\tilde{\Rrightarrow}} B)\\ \lambda_{A \tilde{\Rrightarrow} B} &\triangleq [(i_B \mapsto PA)\,, ((i_A, j_B) \mapsto OQ)\,, \overline{\lambda_A}\,, \lambda_B]\\ \vdash_{A \tilde{\Rrightarrow} B} &\triangleq \{(\dagger, i_B)\} \cup \{(i_B, (i_A, j_B)) \mid i_B \vdash_B j_B\}\\ &\quad \cup \{((i_A, j_B), m) \mid (i_A \vdash_A m \vee j_B \vdash_B m)\}\\ &\quad \cup (\vdash_A \upharpoonright \overline{I_A}^{\,2}) \cup (\vdash_B \upharpoonright (\overline{I_B} \cap \overline{J_B})^2) \end{aligned}$$

Moreover, let $A \Rightarrow B \triangleq A \mathrel{\bar{\Rightarrow}} B_\perp$ be the lifted function space. Note that we will usually *identify graph-isomorphic arenas* related by isomorphisms which simply manipulate $*$'s. With this convention, the last construction corresponds precisely to $A \Rightarrow B$ of [2]; also, for any A, $1 \mathrel{\bar{\Rightarrow}} A = A$. The previous constructions are sketched below.



We also have arrow-counterparts. Let $f : A \to A'$, $g : B \to B'$ in $\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$ and $h : B \to B'$ in $\mathcal{V}_{\mathtt{tt}}^{\vec{\alpha}}$, then

- $f_\perp : A_\perp \to A'_\perp$ initially plays a sequence of asterisks $[*_1^{\vec{\alpha}} *_1'^{\vec{\alpha}} *_2'^{\vec{\alpha}} *_2^{\vec{\alpha}}]$ and then continues playing like f.
- $f \otimes g : A \otimes B \to A' \otimes B'$ answers initial moves $[(i_A, i_B)^{\vec{\alpha}}]$ with f's answer to $[i_A^{\vec{\alpha}}]$ and g's answer to $[i_B^{\vec{\alpha}}]$. Then, according to whether Opponent plays in $J_{A'}$ or in $J_{B'}$, Player plays like f or like g respectively.
- $f \mathrel{\bar{\Rightarrow}} h : A' \mathrel{\bar{\Rightarrow}} B \to A \mathrel{\bar{\Rightarrow}} B'$ answers initial moves $[i_B^{\vec{\alpha}}]$ like h and then responds to $[i_B^{\vec{\alpha}} i_{B'}^{\vec{\alpha}} (i_A, j_{B'})^{\vec{\alpha}}]$ with f's answer to $[i_A^{\vec{\alpha}}]$ and h's response to $[i_B^{\vec{\alpha}} i_{B'}^{\vec{\alpha}} j_{B'}^{\vec{\alpha}}]$ (hence the need for ttotality of h). It then plays like f or like h, according to Opponent's next move.

We can also define infinite tensor products of pointed arenas, where an arena A is ***pointed*** iff I_A is singleton (in which case the unique initial move is necessarily equivariant). For pointed arenas $\{A_i\}_{i\in\omega}$ construct their product $\bigotimes_i A_i$ by 'gluing them together' at their initial moves. Since these are equivariant, the resulting initial move is also equivariant, and we denote it by "$*$". For any pointed A_i's and B_i's and any $\{f_i : A_i \to B_i\}_{i\in\omega}$ define:

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

Take $\mathcal{V}_{\mathtt{t}*}^{\vec{\alpha}}$ to be the full subcategory of $\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$ of pointed arenas.

Our constructions enjoy the following properties.

Proposition 26 *All of the following are functors.*

$$\_ \otimes \_ : \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}} \times \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}} \to \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}\ ,\ \ \_ \mathrel{\bar{\Rightarrow}} \_ : (\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}})^{\mathrm{op}} \times \mathcal{V}_{\mathtt{tt}}^{\vec{\alpha}} \to \mathcal{V}_{\mathtt{tt}}^{\vec{\alpha}}$$
$$(\_)_\perp : \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}} \to \mathcal{V}_{\mathtt{tt}}^{\vec{\alpha}} \qquad , \qquad \bigotimes \_ : \textstyle\prod_{i\in\omega} \mathcal{V}_{\mathtt{t}*}^{\vec{\alpha}} \to \mathcal{V}_{\mathtt{t}*}^{\vec{\alpha}}$$

Moreover, $\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$ is a symmetric monoidal category under $\otimes$, and is partially closed in the following sense. For any object B, the functor $\_ \otimes B : \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}} \longrightarrow \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$ has a partial right adjoint $B \mathrel{\bar{\Rightarrow}} \_ : \mathcal{V}_{\mathtt{tt}*}^{\vec{\alpha}} \longrightarrow \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$, that is for any object A and any pointed object C there exists a bijection

$$\Lambda_{A,C}^B : \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}(A \otimes B, C) \xrightarrow{\cong} \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}(A, B \mathrel{\bar{\Rightarrow}} C)$$

natural in A, C. Moreover, 1 is a terminal object and $\otimes$ is a product constructor in $\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$, so $\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$ has finite products. ■

Solving (SE)**.** The full form of (SE) is the following.

$$[\![\mathbf{1}]\!] = 1\ ,\ [\![\mathbb{N}]\!] = \mathbb{N}\ ,\ [\![[A]]\!] = \mathsf{N}_A\ ,\ [\![A \otimes B]\!] = [\![A]\!] \otimes [\![B]\!]$$
$$[\![A \to B]\!] = [\![A]\!] \mathrel{\bar{\Rightarrow}} (\xi \Rightarrow [\![B]\!] \otimes \xi)\ ,\ \xi = \bigotimes_A (\mathsf{N}_A \Rightarrow [\![A]\!])$$

To solve it, we will upgrade it to a recursive functor equation and then recur to minimal-invariants theory for games (v. [10]). Let us first define the following preorders on games.

Definition 27 For any $A, B \in Ob(\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}})$ and any $\sigma, \tau \in \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}(A, B)$, define

$$A \trianglelefteq B \iff M_A \subseteq M_B \wedge \lambda_A \subseteq \lambda_B \wedge \vdash_A \subseteq \vdash_B$$
$$\sigma \sqsubseteq \tau \iff \sigma \subseteq \tau$$
▲

It follows that $\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$ is PreCpo-enriched, with $\bigsqcup_i \sigma_i = \bigcup_i \sigma_i$ for any ω-chain $\{\sigma_i\}_{i\in\omega}$, and that $Ob(\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}})$ is a cpo,[7] with least $0 \triangleq (\varnothing, \varnothing, \varnothing)$, and $\bigsqcup_i A_i = (\bigcup_i M_{A_i}, \bigcup_i \lambda_{A_i}, \bigcup_i \vdash_{A_i})$ for any ω-chain $\{A_i\}_{i\in\omega}$. Moreover, if $A \trianglelefteq B$ then we can define an embedding-projection pair of copycat maps $\mathtt{incl}_{A,B} : A \to B$ and $\mathtt{proj}_{B,A} : B \to A$.

Let $\mathcal{C}^{\vec{\alpha}} \triangleq \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}} \times \prod_{A\in \mathrm{TY}} \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$, with objects D of the form $(D_\xi, {D_A}^{A\in\mathrm{TY}})$ and arrows f of the form $(f_\xi, {f_A}^{A\in\mathrm{TY}})$. Define $F : (\mathcal{C}^{\vec{\alpha}})^{\mathrm{op}} \times \mathcal{C}^{\vec{\alpha}} \to \mathcal{C}^{\vec{\alpha}}$ on objects by taking $F(D,E) \triangleq (\xi_{D,E}, {[\![A]\!]_{D,E}}^{A\in\mathrm{TY}})$, where

$$\xi_{D,E} = \bigotimes_{A\in\mathrm{TY}} (\mathsf{N}_A \Rightarrow E_A) \qquad [\![[A]]\!]_{D,E} = \mathsf{N}_A$$
$$[\![A \otimes B]\!]_{D,E} = [\![A]\!]_{D,E} \otimes [\![B]\!]_{D,E} \qquad [\![\mathbb{N}]\!]_{D,E} = \mathbb{N}$$
$$[\![A \to B]\!]_{D,E} = D_A \mathrel{\bar{\Rightarrow}} (\xi_{E,D} \Rightarrow E_B \otimes \xi_{D,E}) \quad [\![\mathbf{1}]\!]_{D,E} = 1$$

and similarly for $F(f,g) \triangleq (\xi_{f,g}, {[\![A]\!]_{f,g}}^{A\in\mathrm{TY}})$. Now (SE) has been reduced to $D \cong F(D,D)$. We can show that F is a locally continuous functor, and continuous wrt $\trianglelefteq$. Hence the following.

Theorem 28 *In $\mathcal{C}^{\vec{\alpha}}$ we can form a $\trianglelefteq$-increasing sequence $\{e_i : D_i \to D_{i+1}\}_{i\in\omega}$ of objects and embeddings as follows.*

$$D_{0,\mathbf{1}} = D_{0,A\to B} \triangleq 1 \qquad D_{0,\mathbb{N}} \triangleq \mathbb{N}\ ,\ D_{0,[A]} \triangleq \mathsf{N}_A$$
$$D_{0,\xi} \triangleq \bigotimes_A (\mathsf{N}_A \Rightarrow 0) \qquad D_{0,A\otimes B} \triangleq D_{0,A} \otimes D_{0,B}$$
$$D_{i+1} \triangleq F(D_i, D_i) \qquad e_0 \triangleq \mathtt{incl}_{D_0,D_1}\ ,\ e_{i+1} \triangleq F(e_i^R, e_i)$$

Taking $D^* \triangleq \bigsqcup_i D_i$ and, for each i, $\eta_i \triangleq \mathtt{incl}_{D_i,D^*}$ we obtain a local bilimit $(D^*, {\eta_i}^{i\in\omega})$. ■

Hence, D^* is the canonical solution to $D \cong F(D,D)$, and it solves (SE) with the following notation.

Definition 29 (ξ, $\circledast$ and $[\![A]\!]$) Let D^* be as in the previous theorem. Define the store arena ξ to be D^*_ξ and, for each type A, the translation $[\![A]\!]$ to be D^*_A.

ξ is pointed; we denote its unique initial move by $\circledast$. ▲

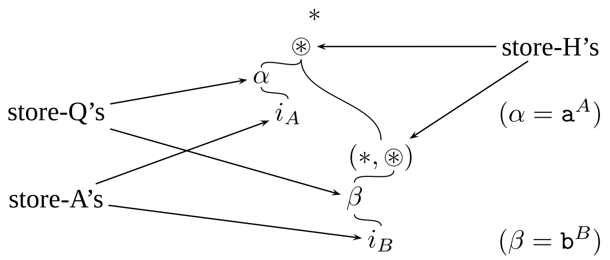
[7] At this point note that there is a simpler, yet less elegant, solution to (SE), simply by taking the least (in fact, the unique) fixpoint of the map $G : \prod Ob(\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}) \to \prod Ob(\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}})$ induced by (SE) in the cpo $\prod_{A\in\mathrm{TY}} (Ob(\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}), \trianglelefteq)$.

Tidy strategies. Using the solution D^* to (SE) we can model $\nu\rho$ in the family $\langle \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}, T^{\vec{\alpha}} \rangle_{\vec{\alpha} \in \mathsf{N}^{\#}}$, with T being the store monad on ξ (i.e. $T = \xi \Rightarrow \_ \otimes \xi$). However, thus we do not obtain a fully abstract model. In the reduction calculus the treatment of the store follows a specific *store-discipline*; for example, if a store S is updated to S' then the original store S is not accessible any more. 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 is that, when the store is asked a name, it 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 elsewhere. We will therefore constrain total strategies with further conditions, defining *tidy strategies*.

Let $\mathcal{V}_{\mathtt{t},\mathrm{TY}}^{\vec{\alpha}}$ be the full subcategory of $\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$ with objects $[\![A]\!]$, $A \in \mathrm{TY}$. For each $[\![A]\!]$ let its set of ***store-Handles***, H_A, be:[8]

$$H_{A\otimes B} \triangleq H_A \cup H_B \qquad H_{\mathbb{N}} = H_{\mathbf{1}} = H_{[A]} \triangleq \varnothing$$
$$H_{A\to B} \triangleq \{(i_A, \circledast_A), (i_B, \circledast_2)\} \cup H_A \cup H_B \cup H_{\xi_A} \cup H_{\xi_B}$$

where we let $[\![A \to B]\!]$ be $[\![A]\!] \mathrel{\tilde{\Rightarrow}} (\xi_A \Rightarrow [\![B]\!] \otimes \xi_B)$, and $H_\xi = \bigcup_C H_C$ if $\xi = \bigotimes_C (\mathsf{N}_C \Rightarrow [\![C]\!])$. In any arena $[\![A]\!]$, a store-H justifies name-questions α, which we call ***store-Questions***. Answers to store-Q's are called ***store-Answers***.
For example: $\quad T1 = \quad \xi \Rightarrow 1 \otimes \xi \qquad$ (note $T1 = 1 \mathrel{\tilde{\Rightarrow}} T1$)



We can show that a move $m \in M_{[\![A]\!]}$ is exactly one of the following: initial, store-H, store-Q or store-A.

As store-H's occur in several places in a play, we may use parenthesised indices to distinguish moves from different store-H's. For example, a store-Q w may be denoted $w_{(O)}$ or $w_{(P)}$, the notation denoting also the OP-polarity. Note also that from now on we work in $\mathcal{V}_{\mathtt{t},\mathrm{TY}}^{\vec{\alpha}}$, unless stated otherwise.

Definition 30 (Tidy strategies) A total $\vec{\alpha}$-strategy σ is *tidy* if whenever odd-length $[s] \in \sigma$ then:

(TD1) If s ends in a store-Q w then $[sx] \in \sigma$, with x being either a store-A to w introducing no new names, or a copy of w. In particular, if $w = \alpha^{\vec{\alpha}'}$ with $\alpha \# \ulcorner s \urcorner^-$ then the latter case holds.

(TD2) If $[sw_{(P)}] \in \sigma$ with w a store-Q then $w_{(P)}$ is justified by last O-store-H in $\ulcorner s \urcorner$.

(TD3) If $\ulcorner s \urcorner = s' w_{(O)} w_{(P)} t\, y_{(O)}$ with w a store-Q then $[s y_{(P)}] \in \sigma$ with $y_{(P)}$ justified by $\ulcorner s \urcorner$.-3. ▲

TD1 states that, whenever O(pponent) asks the value of a name, P either immediately answers with its value or copy-cats the question to the previous store-H. The former case corresponds to P having updated the given name lastly (i.e. between the previous O-store-H and the last one), while the latter to P not having done so, and hence asking its value to the previous store configuration. Hence, the current store is, in fact, composed by layers of stores –one on top of the other– and only when a name has not been updated at the top layer is P allowed to search for it in layers underneath. TD3 further guarantees the above-described behavior. It states that when P starts a store-Q copycat then he must copy-cat the store-A he receives and all proceeding moves. TD2 guarantees the multi-layer discipline of the store: P can only see the store-H played last by O in the P-view.

Proposition 31 *If $\sigma : A \to B$ and $\tau : B \to C$ are tidy strategies then so is $\sigma ; \tau$.* ■

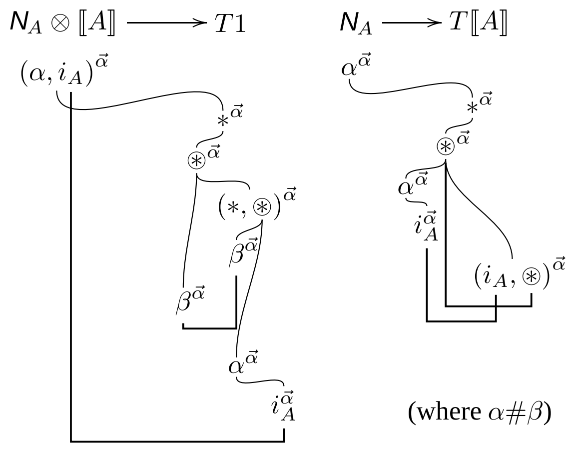
Full-abstraction with tidy strategies. Let $\mathcal{T}^{\vec{\alpha}}$ be the lluf subcategory of $\mathcal{V}_{\mathtt{t},\mathrm{TY}}^{\vec{\alpha}}$ of tidy strategies. $\mathcal{T}^{\vec{\alpha}}$ inherits finite products from $\mathcal{V}_{\mathtt{t}}^{\vec{\alpha}}$. Moreover, the endofunctor

$$T : \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}} \to \mathcal{V}_{\mathtt{t}}^{\vec{\alpha}} \triangleq \xi \Rightarrow \_ \otimes \xi$$

restricts to $\mathcal{T}^{\vec{\alpha}}$, and induces a strong monad $(T, \eta, \mu, \tau)^{\vec{\alpha}}$ on it (by a more-or-less standard monad construction). Furthermore, setting $(TB)^A \triangleq A \mathrel{\tilde{\Rightarrow}} TB$ we obtain a λ_c-model.

We take $\mathcal{T} \triangleq \langle \mathcal{T}^{\vec{\alpha}}, (T, \eta, \mu, \tau)^{\vec{\alpha}} \rangle_{\vec{\alpha} \in \mathsf{N}^{\#}}$ and proceed to update and dereferencing arrows.

Definition 32 In $\mathcal{T}^{\vec{\alpha}}$, define $\mathtt{upd}_A : \mathsf{N}_A \otimes [\![A]\!] \longrightarrow T1$ and $\mathtt{drf}_A : \mathsf{N}_A \longrightarrow T[\![A]\!]$, for any type A, as follows.[9]



▲

Proposition 33 *In each $\mathcal{T}^{\vec{\alpha}}$ the* NR-*diagrams of definition 4 commute.* ■

[8] The definition of H_A is informal, note circularity in $H_{A\to B}$; a formal definition is given by induction on the level of moves in $[\![A]\!]$ and on A.

[9] In the diagrams we use curved lines for justification pointers; polygonic lines denote that the strategy copycats between the connected moves.

We introduce *name-abstraction* and *name-addition* transformations for nominal strategies.

Definition 34 Let $f : A \longrightarrow B$ in $\mathcal{T}^{\vec{\alpha},\alpha}$ and $g : A \longrightarrow B$ in $\mathcal{T}^{\vec{\alpha}}$. Define $\langle\!\langle\alpha\rangle\!\rangle f : A \longrightarrow B$ in $\mathcal{T}^{\vec{\alpha}}$ and $g^{+\alpha} : A \longrightarrow B$ in $\mathcal{T}^{\vec{\alpha},\alpha}$ as:

$$\langle\!\langle\alpha\rangle\!\rangle f \triangleq \mathtt{strat}\{[i_A^{\vec{\alpha}} i_B^{\vec{\alpha}} j_B^{\vec{\alpha}} s] \mid [i_A^{\vec{\alpha},\alpha} i_B^{\vec{\alpha},\alpha} j_B^{\vec{\alpha},\alpha} s] \in \mathtt{viewf}(f) \wedge \alpha \# i_A\}$$

$$g^{+\alpha} \triangleq \{[s^{+\alpha}] \mid [s] \in g \,\wedge\, \alpha \# \mathcal{L}(s)\}$$

where $s^{+\alpha}$ is s with $\vec{\alpha}$ replaced by $\vec{\alpha}, \alpha$ in its name-lists. ▲

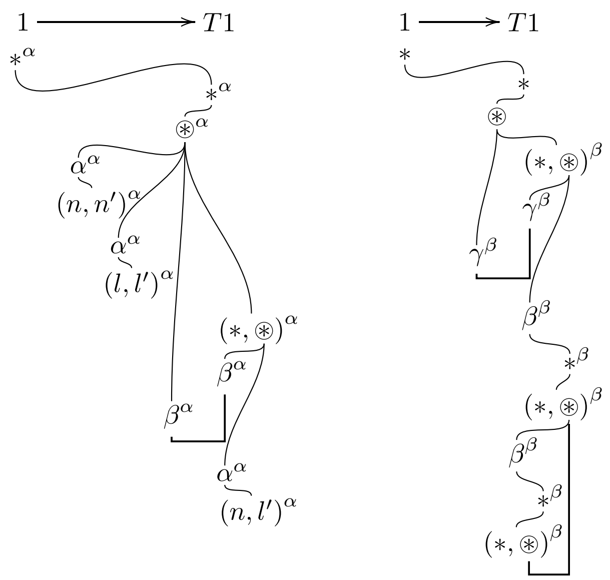
Proposition 35 *For any $\vec{\alpha}, \alpha$-strategy f and any $\vec{\alpha}$-strategy g, $\langle\!\langle\alpha\rangle\!\rangle f$ is an $\vec{\alpha}$-strategy and $g^{+\alpha}$ is an $\vec{\alpha}$,α-strategy. Moreover, $\mathcal{T}$ satisfies the* SN-*equations of definition 4.* ■

Using name-deletion, a transformation dual to name-addition, we represent an $\vec{\alpha}$-strategy f as a pair $((f)^\circ, \vec{\alpha})$ by deleting from f all names that are *essentially-fresh* for it[10] and orbiting the result under all permutations with domain $\mathsf{S}(\vec{\alpha})$. We then have a $\lambda_{\nu\rho}$-model, which is also adequate.

Theorem 36 *$\mathcal{T}$ is an adequate $\lambda_{\nu\rho}$-model.* ■

The (omitted) proof of adequacy proceeds by showing that if $[\![M]\!] = \langle\!\langle\vec{\beta}\rangle\!\rangle[\![\bar{S}\,;\tilde{0}]\!]$ then $\vec{\alpha} \vDash M$ cannot have a reduction sequence with infinitely many DRF-reduction steps; omitting DRF's we are left with a strongly normalising calculus.

Let us proceed to an example. Consider the typed terms $\alpha \mid \varnothing \vdash \alpha := \langle \mathtt{fst}\,!\alpha, \mathtt{snd}\,!\alpha\rangle$ and $\mid \varnothing \vdash \nu\beta.\, \beta := \lambda x.(!\beta)\mathtt{skip}$ with $\alpha = \mathtt{a}^{\mathbb{N}\otimes\mathbb{N}}$ and $\beta = \mathtt{b}^{\mathtt{1}\to A}$. Their translations in $\mathcal{T}^{\alpha}$ and $\mathcal{T}$ respectively are as follows.



The reader may want to check now that the bottom arrow, $\{[* * \circledast]\}$, equals $[\![\nu\beta.(\beta := \lambda x.(!\beta)\mathtt{skip}); (!\beta)\mathtt{skip}]\!]$.

Finally, we add observationality to $\mathcal{T}^{\vec{\alpha}}$ as follows.

[10] We say that α is *essentially fresh* for $f \in \mathcal{T}^{\vec{\alpha}}(A, B)$, and write $\alpha \overset{ess}{\#} f$, if $\alpha \# \vec{\alpha}$ or, for any $[s] \in f$ and any $\beta \# \mathsf{nlist}(s)$, $[((\alpha\ \beta) \circ \underline{s})^{\mathsf{nlist}(s)}] \in f$.

Definition 37 Expand $\mathcal{T}$ to $\langle \mathcal{T}^{\vec{\alpha}}, T^{\vec{\alpha}}, O^{\vec{\alpha}}, \lesssim^{\vec{\alpha}} \rangle_{\vec{\alpha}\in\mathsf{N}^\#}$ by

$$O^{\vec{\alpha}} \triangleq \{f \in \mathcal{T}^{\vec{\alpha}}(1, T\mathbb{N}) \mid \exists \vec{\beta}.\, [*^{\vec{\alpha}} *^{\vec{\alpha}} \circledast^{\vec{\alpha}} (0, \circledast)^{\vec{\alpha},\vec{\beta}}] \in f\}$$

and $\lesssim^{\vec{\alpha}}$ as in definition 9. ▲

Following a technique which involves appropriate *Separation of Head Occurrence* and *Function Space Decomposition* lemmata (v. [4, 10]), we can show p-observationality of $\mathcal{T}$. For p-definability, the subset of definable morphisms $D_{A,B}^{\vec{\alpha}} \subseteq \mathcal{T}^{\vec{\alpha}}([\![A]\!], T[\![B]\!])$ we use is that of ***finitary strategies***: a strategy σ is finitary iff its viewfunction becomes finite when we remove from it store-copycats and initial $T[\![B]\!]$-answers. At last, full abstraction:

Theorem 38 *For any A, B and finitary $\sigma : [\![A]\!] \to T[\![B]\!]$, σ is definable. Taking $D_{A,B}^{\vec{\alpha}}$ to contain all finitary arrows, $\mathcal{T}$ satisfies p-definability, and is therefore fully abstract for $\nu\rho$.* ■

Acknowledgements. We thank Samson Abramsky for his constant support and guidance; Andy Pitts and Luke Ong for their advice; the anonymous referee for the instructive comments; and the EPSRC, the AG Leventis Foundation and Brasenose College for their financial support.

References

[1] S. Abramsky. Axioms for definability and full completeness. In *Proof, Language, and Interaction: essays in honour of Robin Milner*, pages 55–75. MIT Press, 2000.

[2] S. Abramsky, D. Ghica, A. Murawski, L. Ong, and I. Stark. Nominal games and full abstraction for the nu-calculus. In *Proceedings of LICS '04*, 2004.

[3] S. Abramsky, K. Honda, and G. McCusker. A fully abstract game semantics for general references. In *LICS '98*, 1998.

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

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

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

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

[8] J. Laird. A game semantics of names and pointers. Submitted.

[9] J. Laird. A categorical semantics of higher order store. *ENTCS*, 69, 2002.

[10] G. McCusker. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Distinguished Dissertations. Springer-Verlag, London, 1998.

[11] E. Moggi. Computational lambda-calculus and monads. In *Proceedings of LICS'89*, 1989.

[12] L. Ong. Observational equivalence of third-order idealized algol is decidable. In *Proceedings of LICS '02*, 2002.

[13] A. M. Pitts. Nominal logic, a first order theory of names and binding. *Information and Computation*, 186:165–193, 2003.

[14] A. M. Pitts and I. D. B. Stark. Observable properties of higher order functions that dynamically create local names, or: What's new? In *Proc. 18th MFCS*, 1993. LNCS 711.

[15] I. Stark. *Names and Higher-Order Functions*. PhD thesis, University of Cambridge, Dec. 1994.