Nominal Games and Full Abstraction for the Nu-Calculus

S. Abramsky
Oxford University

D. R. Ghica
Oxford University

A. S. Murawski
Oxford University

C.-H. L. Ong
Oxford University

I. D. B. Stark
Edinburgh University

Abstract

We introduce nominal games for modelling programming languages with dynamically generated local names, as exemplified by Pitts and Stark's nu-calculus. Inspired by Pitts and Gabbay's recent work on nominal sets, we construct arenas and strategies in the world (or topos) of Fraenkel-Mostowski sets (or simply FM-sets). We fix an infinite set $\mathcal{N}$ of names to be the "atoms" of the FM-theory, and interpret the type ν of names as the flat arena whose move-set is $\mathcal{N}$. This approach leads to a clean and precise treatment of fresh names and standard game constructions (such as plays, views, innocent strategies, etc.) that are considered invariant under renaming. The main result is the construction of the first fully-abstract model for the nu-calculus.

1 Introduction

Many convenient features of modern programming languages involve some notion of *generativity*: the idea that an entity (e.g. identifier, reference, object, exception, thread, channel, etc.) may be freshly created, distinct from all others. Such dynamic creations occur at a variety of levels, from the run-time behaviour of Lisp's *gensym*, to resolving questions of scope during program linking. For sound design and correct implementation, it is essential to develop an appropriate abstract understanding of what it means to be *new*.

The nu-calculus of Pitts and Stark [16] was devised to explore this common property of generativity, by adding *names* to the simply-typed lambda-calculus. Names may be created *locally*, passed around, and compared with one another, but that is all. Central to the nu-calculus is the use of *name abstraction*, as found in the π-calculus [10]: the expression $\boldsymbol{\nu} n.M$ represents the creation of a fresh name, which is then bound to n within the body of M. Functions may have local names that remain private and persist from one use of the function to the next; alternatively, names may also travel beyond their original scope and outlive their creator. It is this mobility of names that allows the nu-calculus to capture notions of scope, privacy and sharing.

Game semantics has emerged as a powerful paradigm for giving accurate semantics for a wide spectrum of programming languages. Fully abstract[1] game models have now been constructed for languages with references of various kinds, such as statically-scoped assignable variables [2] and general references [1]. Game models that have been constructed for these languages (with the notable exception of [8]) follow Reynolds in viewing a reference type $\mathtt{ref}[A]$ as a product of "read method" and "write method". As O'Hearn has pointed out, using this interpretation, functoriality of the $\mathtt{ref}[-]$ type constructor requires the presence of "bad variables" in the language, which has the unfortunate consequence of rendering testing for reference equality – a common programming idiom – meaningless. An alternative view is to interpret references as *names* (or locations) of storage cells, according to which testing for reference equality clearly makes good sense. As Stark has observed in [20]: "dynamically created names really do capture the difficult part of $\cdots$ references", our aim here is to give a game-semantic characterization of names as embodied in the nu-calculus, as a first (but we believe key) step towards the construction of a good game model of references as names.

Though designed in all respects to be as simple as possible, the nu-calculus has so far resisted all attempts to model it fully abstractly. The best effort so far, namely [8], still falls some way short of the criterion. In this paper, we present the first fully abstract model for the calculus, which is based on a version of Honda-Yoshida call-by-value games [5] but constructed in the universe of FM-sets [4]. We fix an infinite set $\mathcal{N}$ of names to be the "atoms" of the FM-theory, and interpret the type ν by (the flat arena of) $\mathcal{N}$. In our setting, a play is a justified sequence of moves-with-names (written m^S), satisfying certain conditions, but considered *up to appropriate renaming*. Intuitively the name set S in m^S comprises all names that have been introduced by P at moves that are P-*visible* (in the sense of [6]) at that point. This device, together with the Name Change Conditions which govern the way name sets evolve as the computation unfolds, records the *scope* of each freshly created

[1] A model of a programming language is *fully abstract* if observational equivalence coincides with the equational theory induced by the model.

name. We give a simple algorithm for the composition of strategies that processes name sets in accordance with the mobility of names in the nu-calculus. Semantic objects interpreting nu-calculus terms depend on names in an obvious way. Our approach is to leave the name dependence of the various game constructions (such as plays, strategies, view functions, etc.) *implicit* through the use of *finite support* [4], which in turn depends on the properties of arenas equipped with an action of name permutation.

Recently Laird, in work [8] to appear at FOSSACS'04, has also constructed a game model for the nu-calculus. He builds a category of dialogue games acted upon by the automorphism group of the natural numbers, and it allows certain properties of freshness and locality to be characterized semantically. However the model is not fully abstract for the nu-calculus. An open problem identified in [8, §6] is to characterize the denotations of the nu-calculus terms. As Laird has noted therein, the key to solving it is to find a way of keeping track of the scope of each new name. Our model gives just such an innovation.

2 Nu-Calculus

For a full description of the nu-calculus, see [16, 19, 20]; here we summarize briefly. The calculus has ground types o (booleans: t and f) and ν (names), with higher function types over these. Typing judgements have the form $S;\Gamma \vdash M : A$, where S is a finite set of names; it follows from the definition that all names that occur free in M are in S. It has a call-by-value operational semantics given in terms of a big-step relation $S \vdash M \Downarrow_A (S')C$, meaning that the closed term $S;\vdash M : A$ evaluates to a *canonical form* C and creates fresh names S'; the canonical forms are names, booleans and λ-abstractions. Evaluation can be shown to be deterministic and terminating. We say that terms-in-context $S;\Gamma \vdash M_i : A$ for $i = 1, 2$ are ***observationally equivalent***, written $S;\Gamma \vdash M_1 \approx_A M_2$, if for all contexts $C[\,]$ such that $S;\vdash C[M_i] : o$ for $i = 1, 2$, we have

$$\exists S_1.S;\vdash C[M_1] \Downarrow (S_1)\mathsf{t} \iff \exists S_2.S;\vdash C[M_2] \Downarrow (S_2)\mathsf{t}$$

Intuitively $S;\Gamma \vdash M_1 \approx_A M_2$ if they give the same result in any boolean context. E.g. at ground types:

$$\boldsymbol{\nu} n.(n = n) \approx_o \mathsf{t} \quad \text{and} \quad \boldsymbol{\nu} n.\boldsymbol{\nu} n'.(n = n') \approx_o \mathsf{f}.$$

Note that name generation is not itself observable, although names themselves may be, if revealed:

$$\boldsymbol{\nu} n.\boldsymbol{\lambda} x : o.n \not\approx_{o\to\nu} \boldsymbol{\lambda} x : o.(\boldsymbol{\nu} n.n). \tag{1}$$

Here the left hand term evaluates to a function with a local name n, that it returns when invoked; while the right hand function creates a new and different name each time it is applied. These are distinguished by passing them to the test function $(\boldsymbol{\lambda} f : o \to \nu.(f\,\mathsf{t} = f\,\mathsf{f}))$. Sometimes names remain private, as in the following:

$$\boldsymbol{\nu} n.(\boldsymbol{\lambda} x : \nu.(x = n)) \approx_{\nu\to o} \boldsymbol{\nu} n.(\boldsymbol{\lambda} x.\mathsf{f}) \tag{2}$$

$$\boldsymbol{\nu} n.\boldsymbol{\nu} n'.(\boldsymbol{\lambda} f : \nu \to o.(fn = fn')) \approx_{(\nu\to o)\to o} \boldsymbol{\lambda} f.\mathsf{t}\,. \tag{3}$$

Although the first two functions differ at the name n, no external context can provide it. Similarly for the second pair, any constructed function f must treat n and n' equally. For further examples, see [19, 17, 13].

Even without the classic difficulties of nontermination, issues like this make reasoning about the nu-calculus somewhat challenging, especially at higher types. Stark [18] presents an equational theory that is complete at ground types, and a relational theory that is complete for first-order functions, as in (2). There are also operational techniques [16] and a categorical model with logical relations, fully abstract to first order [20]. These show that observational equivalence is decidable up to first order; but with no general results at second or higher orders.

3 Call-by-value games

In this section we introduce Honda and Yoshida's call-by-value games [5]. The question-answer paradigm of the CBV games differs from the more familiar call-by-name games in an important way: initial moves of CBV arenas are P-answers. As we shall see these are the possible answers to a question posed by the environment (or O). There is an essentially equivalent version of CBV games due to Abramsky and McCusker [3]; our presentation here is a reformulation of Honda and Yoshida's games in the style of McCusker [9].

An ***arena*** is a triple $A = \langle M_A, \vdash_A, \lambda_A \rangle$ consisting of a set of moves M_A, a justification relation $\vdash_A \subseteq (M_A + \{\dagger\}) \times M_A$, where $\dagger$ is a dummy move, satisfying:

(f1) For each $m \in M_A$ there is a unique $x \in (M_A + \{\dagger\})$ such that $x \vdash_A m$; in case $\dagger \vdash_A m$, we call m an *initial move*.

(f2) $\vdash_A \restriction (M_A \times M_A)$ is well-founded;

a labelling function $\lambda_A : M_A \longrightarrow \{P, O\} \times \{Q, A\}$ that designates each move as one of four types: P-question, P-answer, O-question and O-answer, and satisfying:

(l1) Initial moves are P-answers.

(l2) If $m_1, m_2 \in M_A$ are such that $m_1 \vdash_A m_2$, then m_1 and m_2 are moves by different players.

(l3) Answers may only justify questions. I.e. if $m_1 \vdash_A m_2$ and m_1 is an answer then m_2 is a question.

It is useful to think of the arena A as a vertex-labelled directed graph with vertex-set M_A and edge-set $\vdash_A \upharpoonright (M_A \times M_A)$ such that the labels on vertices are given by λ_A satisfying (l1), (l2) and (l3). It follows from (f1) and (f2) that the graph so defined is a forest of trees in which the roots are the initial moves.

Example 3.1. We begin with the empty arena $\mathbf{0} = \langle \varnothing, \varnothing, \varnothing \rangle$, and the singleton arena

$$\mathbf{1} \;=\; \langle \{ * \}, \{ (*, PA) \}, \{ (\dagger, *) \} \rangle$$

The arenas representing the ground types of the nu-calculus, o and ν, (which we shall also write as o and ν respectively, by abuse of notation) are *flat* arenas. I.e. they are discrete graphs (forests), all of whose nodes are (necessarily) P-answers; $M_o = \{ * \} + \{ * \}$, but for convenience we name the elements $\mathtt{t}$ and $\mathtt{f}$, and $M_\nu = \mathcal{N}$, a distinguished, countably infinite set of *names*.

We introduce two arena constructors. For each construction, we shall first describe it rather informally as an operation on labelled forests, and then give a formal description (in terms of a triple). The former is probably easier to understand; the latter is important for the subsequent development of nominal games in Section 4. We shall use a simple notation for forest construction. We write $A = \sum_{i\in\mathcal{I}} T_i$ to mean the forest with constituent trees T_i's; and write $a \langle A \rangle$ for the *tree*, rooted at a, that is constructed from the forest A by adding, for each tree T_i of A, an edge that joins a to T_i's root. Thus every forest has a canonical form $\sum_{i\in\mathcal{I}} a_i \langle A_i \rangle$ (in which each forest A_i is also in canonical form). We write $A + B$ to mean the disjoint union of forests A and B.

Tensor product. Given arenas $A = \sum_{i\in\mathcal{I}} a_i \langle A_i \rangle$ and $B = \sum_{j\in\mathcal{J}} b_j \langle B_j \rangle$, we define

$$A \otimes B \;=\; \sum_{i\in\mathcal{I}, j\in\mathcal{J}} (a_i, b_j) \langle A_i + B_j \rangle$$

whose nodes inherit their labels from A or B as appropriate. Formally, writing I_A for the initial moves of A, and setting $\overline{I_A} = M_A - I_A$, we define:

$$\begin{array}{rcl} M_{A\otimes B} & = & I_A \times I_B \;+\; I_A \times \overline{I_B} \;+\; \overline{I_A} \times I_B \\ \lambda_{A\otimes B}(a,b) & = & \begin{cases} PA & \text{if } a \in I_A \text{ and } b \in I_B \\ \lambda_B(b) & \text{if } a \in I_A \text{ and } b \in \overline{I_B} \\ \lambda_A(a) & \text{if } a \in \overline{I_A} \text{ and } b \in I_B \end{cases} \end{array}$$

and $\vdash_{A\otimes B}$ is the least relation satisfying

$$\begin{array}{rcl} \dagger \vdash_{A\otimes B} (a,b) & \iff & a \in I_A \;\wedge b \;\in I_B \\ (a,b) \vdash_{A\otimes B} (a',b') & \iff & a = a' \in I_A \;\wedge\; b \vdash_B b' \\ & \vee & b = b' \in I_B \;\wedge\; a \vdash_A a' \end{array}$$

It is easy to see that $A \otimes B$ is an arena.

Function space (and prearenas). Given arenas $A = \sum_{i\in\mathcal{I}} a_i \langle A_i \rangle$ and $B = \sum_{j\in\mathcal{J}} b_j \langle B_j \rangle$, we define

$$\begin{array}{lrcl} \textit{prearena} & A \rightharpoonup B & = & \sum_{i\in\mathcal{I}} a_i^{OQ} \langle A_i^{\perp} + B \rangle \\ \textit{function space} & A \to B & = & * \langle A \rightharpoonup B \rangle \end{array}$$

where $*$ is a fresh P-answer, the forest $A_i^{\perp}$ is obtained from A_i by swapping all O/P labels therein, a_i^{OQ} fixes the label of a_i as OQ, and the labels of the rest of the nodes are inherited. Formally we define the ***function space arena*** $A \to B$ as:

$$\begin{array}{rcl} M_{A\to B} & = & \{ * \} \;+\; M_A \;+\; I_A \times M_B \\ \lambda_{A\to B}(m) & = & \begin{cases} PA & \text{if } m = * \\ OQ & \text{if } m \in I_A \\ \lambda_{A^\perp}(m) & \text{if } m \in M_A - I_A \\ \lambda_B(b) & \text{if } m = (a,b) \end{cases} \end{array}$$

and $\vdash_{A\to B}$ is the least relation satisfying

$$\begin{array}{rcl} \dagger \vdash_{A\to B} m & \iff & m = * \\ m \vdash_{A\to B} m' & \iff & m = * \;\wedge\; \dagger \vdash_A m' \\ & \vee & m, m' \in M_A \;\wedge\; m \vdash_A m' \\ a \vdash_{A\to B} (a',b') & \iff & a = a' \in I_A \;\wedge\; \dagger \vdash_B b' \\ (a,b) \vdash_{A\to B} (a',b') & \iff & a = a' \in I_A \;\wedge\; b \vdash_B b' \end{array}$$

It is straightforward to verify that $A \to B$ is an arena, but $A \rightharpoonup B$ is not (because its initial moves are O-questions, instead of P-answers). Structures of the form $A \rightharpoonup B$, where A and B are arenas, are called ***prearenas***, which will play an important part in the sequel: we shall define (plays and) strategies over prearenas, not arenas. Further, we shall interpret a term-in-context $x_1 : C_1, \cdots, x_n : C_n \vdash t : A$ as a strategy over the prearena $C_1 \otimes \cdots \otimes C_n \rightharpoonup A$ (writing A for the arena that denotes the type A).

We illustrate the tensor and function space constructions in Figure 1, wherein O-questions and P-answers are drawn as $\sqcap$ and $\sqcup$ respectively. Henceforth we fix a representation of disjoint union of sets:

$$X_0 + \cdots + X_n \;=\; \bigcup_{i=0}^{n} \{ (i,x) : x \in X_i \}.$$

Example 3.2. The arena $A = \nu \otimes (\nu \to \nu)$ is a forest of three levels. Formally the root nodes (initial moves), at level 1, are elements in M_A of the form $(0, (a, (0, *)))$, level-2 and level-3 moves are of the forms $(1, (a, (1, b)))$ and $(1, (a, (2, (b, c))))$ – call it m – respectively, where a, b and c range over $\mathcal{N}$.

Notation. The cumbersome disjoint-union representation of moves will come in useful in the definition of actions on FM-arenas (see Lemma 4.2 and Example 4.3). However, in practice (see Examples 4.4, 4.7, 4.9, etc.), it will be enough

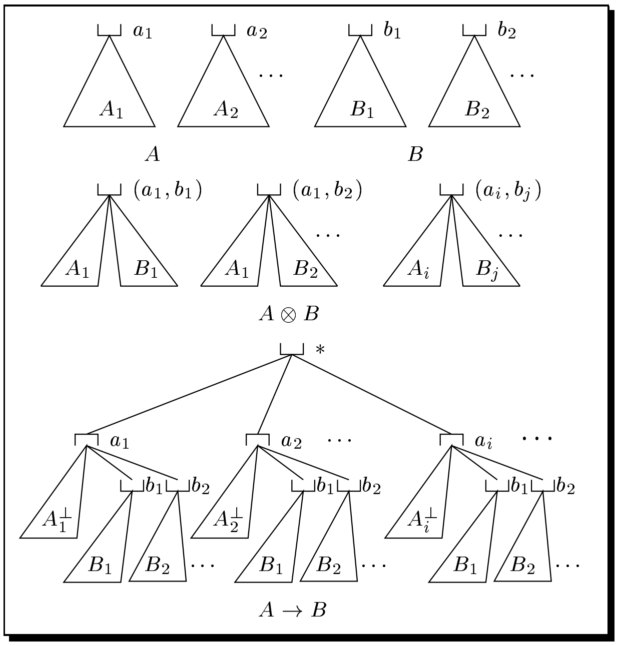

Figure 1. Arena constructors illustrated

to identify the moves informally. E.g. we say that m (from the preceding Example) is the level-3 name c whose parent and grandparent are names b and a respectively.

A ***justified sequence*** over a prearena A is a finite sequence of O/P-alternating moves such that, except for the opening move which is initial (and necessarily an O-question), every move m has a *justification pointer* (or simply *pointer*) to some earlier move m_0 such that $m_0 \vdash_A m$; we say that m is *explicitly justified* by m_0, or m_0 *explicitly justifies* m. It follows that there is exactly one occurrence of an initial move in a justified sequence, namely, the opening move. A question occurrence in a justified sequence s is said to be *pending* just in case no answer in s is explicitly justified by it. Recall the definition of the ***P-view*** [6] of a justified sequence s, written $\ulcorner s \urcorner$:

$$
\begin{array}{rcll}
\ulcorner s\, m \urcorner & = & \ulcorner s \urcorner m & \text{if } m \text{ is a P-move} \\
\ulcorner m \urcorner & = & m & \text{if } m \text{ is initial} \\
\ulcorner s\, m_0\, u\, m \urcorner & = & \ulcorner s \urcorner m_0\, m & \text{if O-move } m \text{ is explicitly} \\
 & & & \text{justified by } m_0
\end{array}
$$

In $\ulcorner s\, m_0\, u\, m \urcorner$ the pointer from m to m_0 is retained, similarly for the pointer from m in $\ulcorner s\, m \urcorner$ in case m is a P-move.

4 Nominal games

Pitts and Gabbay have argued cogently (in [4, 14]) for the use of Fraenkel-Mostowski (FM) permutation model of set theory as a syntax-independent mathematical model of fresh bindable names and α-conversion. Here we construct Honda-Yoshida style CBV arenas equipped with an action of name permutation that preserves justification and move-labels, and use it to make precise an implicit notion of dependence on names.

An overview of FM-sets. We fix a countably infinite set $\mathcal{N}$ of *names*, and write $\textsc{perm}(\mathcal{N})$ for the group of permutations of $\mathcal{N}$. In the following we shall be concerned with actions on certain (structured) sets X induced by elements of $\textsc{perm}(\mathcal{N})$: this is a function mapping pairs $(\pi, x) \in \textsc{perm}(\mathcal{N}) \times X$ to elements $\pi \cdot_X x$ of X and satisfying: for any $\pi, \pi' \in \textsc{perm}(\mathcal{N})$ and $x \in X$

$$
\begin{array}{rcl}
\pi' \cdot_X (\pi \cdot_X x) & = & (\pi' \circ \pi) \cdot_X x \\
\mathsf{id}_{\mathcal{N}} \cdot_X x & = & x
\end{array}
$$

where $\pi \circ \pi'$ is the composition of π' followed by π, and $\mathsf{id}_{\mathcal{N}}$ is the identity of the group $\textsc{perm}(\mathcal{N})$.

We think of an action of $\textsc{perm}(\mathcal{N})$ on a set X as an abstract device for regarding the elements x of X as somehow "involving names from $\mathcal{N}$ for their construction", in that x may be varied just by permuting certain names. Given such a set X, we say that a set of names $A \subseteq \mathcal{N}$ *supports* an element $x \in X$ just in case every permutation $\pi \in \textsc{perm}(\mathcal{N})$ that fixes A also fixes x: $\forall a \in A \,.\, \pi(a) = a \Longrightarrow \pi \cdot_X x = x$. We say that x is *finitely supported* if there is some finite subset $A \subseteq \mathcal{N}$ that supports it. It is an easy exercise to show that if x is finitely supported, then there is a least finite subset of $\mathcal{N}$ that supports it: we call this the ***support*** of x, and denote it by $\textsc{supp}_X x$, or simply $\textsc{supp}(x)$. We define an ***FM-set*** to be a set X equipped with an action of $\textsc{perm}(\mathcal{N})$ in which every element $x \in X$ is finitely supported. For example $\mathcal{N}$ is an FM-set: it has a *canonical*[2] $\textsc{perm}(\mathcal{N})$-action given by applying the permutation, as a function, to names i.e. $\pi \cdot_{\mathcal{N}} a = \pi(a)$. It follows that $\textsc{supp}_{\mathcal{N}} a = \{\, a \,\}$ for each $a \in \mathcal{N}$. Henceforth, we take $\mathcal{N}$ to mean the FM-set equipped with the canonical action. The empty set and the singleton set $\mathbf{1} = \{\, * \,\}$ are also FM-sets (both have the obvious, and unique, $\textsc{perm}(\mathcal{N})$-action).

Let X and Y be FM-sets. We say that $U \subseteq X$ is an *FM-subset* of X just in case for all $x \in X$, if $x \in U$ then for all $\pi \in \textsc{perm}(\mathcal{N})$, $\pi \cdot_X x \in U$. We say that $R \subseteq X \times Y$ is an *FM-relation* just if R is an FM-subset of $X \times Y$ i.e. for all $x \in X, y \in Y$, if $(x, y) \in R$ then for all $\pi \in \textsc{perm}(\mathcal{N})$, $(\pi \cdot_X x, \pi \cdot_Y y) \in R$. It is straightforward to see that an FM-subset of an FM-set is itself an FM-set, so is the *quotient* $X/\mathcal{R}$ of an FM-set by an equivalence FM-relation $\mathcal{R}$. In the same vein, an *FM-function* $f : X \longrightarrow Y$ is an FM-relation that is functional i.e. $f \subseteq X \times Y$ is single-valued and total. The FM-sets are the objects of a category (indeed topos), written $\mathbb{FM}$, whose maps $f : X \longrightarrow Y$ are FM-functions.

[2]There are other actions. E.g. *null action*: $\pi \cdot a = a$ for all π, and *conjugate action*: $\pi \cdot a = \pi_0^{-1}(\pi(\pi_0 a))$ for a fixed π_0.

FM-set constructions. Let X and Y be FM-sets.

(i) The disjoint union of X and Y, $X + Y$, is an FM-set. It inherits a $\textsc{perm}(\mathcal{N})$-action from X and Y by: $\pi \cdot_{X+Y} (0, x) = (0, \pi \cdot_X x)$ and $\pi \cdot_{X+Y} (1, y) = (1, \pi \cdot_Y y)$. It is straightforward to calculate that $\textsc{supp}_{X+Y}(0, x) = \textsc{supp}_X x$ and $\textsc{supp}_{X+Y}(1, y) = \textsc{supp}_Y y$.

(ii) The product of X and Y, $X \times Y$, is an FM-set. It can be endowed with a $\textsc{perm}(\mathcal{N})$-action by: $\pi \cdot_{X\times Y} (x, y) = (\pi \cdot_X x, \pi \cdot_Y y)$. We have $\textsc{supp}_{X\times Y}(x, y) = \textsc{supp}_X x \cup \textsc{supp}_Y y$.

(iii) The powerset $\mathcal{P}(X)$ inherits a $\textsc{perm}(\mathcal{N})$-action: $\pi \cdot_{\mathcal{P}(X)} U = \{ \pi \cdot_X x : x \in U \}$. In general, there is no guarantee that every element of $\mathcal{P}(X)$ has finite support. The set $\mathcal{P}_{\mathrm{fs}}(X)$, consisting of $U \in \mathcal{P}(X)$ that has finite support, is an FM-set. We define $\pi \cdot_{\mathcal{P}_{\mathrm{fs}}(X)} U = \{ \pi \cdot_X y : y \in U \}$, which has support $\pi \cdot_{\mathcal{P}(\mathcal{N})} (\textsc{supp}_{\mathcal{P}(X)} U)$ that is finite since $\textsc{supp}_{\mathcal{P}(X)} U$ is assumed to be finite.

(iv) The set "X restricted to Y", written $X \upharpoonright Y$, consisting of "elements of X abstracted until their support is that of a given element of Y" is an FM-set. We define $X \upharpoonright Y$ to be the quotient $(X \times Y)/\mathcal{R}$ where $(a, b)\ \mathcal{R}\ (a', b')$ iff $b = b'$ and there is some $\textsc{supp}_Y b$-invariant $\pi \in \textsc{perm}(\mathcal{N})$ such that $\pi \cdot_X a = a'$. Clearly $\mathcal{R}$ is an equivalence relation; and we write the equivalence class of (a, b) as $a \upharpoonright b$. Moreover $\mathcal{R}$ is an FM-equivalence relation, so $(X \times Y)/\mathcal{R}$ is an FM-set, with the action of permutations induced from that on $X \times Y$: we have $\pi \cdot_{X\upharpoonright Y} (a \upharpoonright b) = \pi \cdot_X a \upharpoonright \pi \cdot_Y b$, and $\textsc{supp}_{X\upharpoonright Y} a \upharpoonright b = \textsc{supp}_Y b$. For the application in the sequel, we shall take Y to be $\mathcal{P}_{\mathrm{fs}}(\mathcal{N})$: informally, $x \upharpoonright \{ a, b, c \}$ is the element x with all names other than a, b and c abstracted away.

Example 4.1. Given an FM-set X, the collection of lists of elements of X, $\mathsf{list}(X)$, is an FM-set which inherits a permutation action from X; see [14] for a justification. It follows that the "lists of names *up to renaming*" (where e.g. $[a, a, b]$ is equivalent to $[b, b, a]$ and to $[c, c, a]$, but not to $[a, a, a]$ nor to $[c, b, c]$) form an FM-set, namely, $\mathsf{list}(\mathcal{N}) \upharpoonright \mathbf{1}$. Note that every element of the FM-set has empty support.

A benefit of the FM-set approach we shall exploit is that if a particular (arena) construction can be shown to be an (FM-arena) FM-set construction then it has a canonical action of permutations.

FM-arenas and automorphisms. A *Fraenkel-Mostowski arena*, or simply ***FM-arena***, is defined to be an arena $A = \langle M_A, \vdash_A, \lambda_A \rangle$ with the additional requirement that M_A is an FM-set, $\vdash_A$ is an FM-relation, and λ_A is an FM-function. (Note that the label-set $\{ PQ, PA, OQ, OA \}$ is an FM-set on which each $\pi \in \textsc{perm}(\mathcal{N})$ acts as the identity function, and so, each label has empty support.)

Example 4.2. The arenas $\mathbf{0}, \mathbf{1}, o$ and ν (with the FM-set $\mathcal{N}$ as the underlying move-set) are FM-arenas. Let $\mathcal{N} = \{ a_1, a_2, a_3, \cdots \}$. On the other hand, the arena that has move-set $\mathcal{N}$ and justification relation

$$\{ (\dagger, a_{2n+1}) : n \geq 0 \} \cup \{ (a_{2n+1}, a_{2n+2}) : n \geq 0 \}$$

(where $\mathcal{N} = \{ a_1, a_2, a_3, \cdots \}$ and the labels are uniquely determined) is not an FM-arena because the justification relation is not an FM-relation.

The following result is straightforward to prove:

Lemma 4.1. *The tensor, function space and coproduct constructions defined in Section 3 are constructions of FM-arenas.* $\square$

We define an ***automorphism*** of an FM-arena $A = \langle M_A, \vdash_A, \lambda_A \rangle$ to be a bijective FM-function $f : M_A \longrightarrow M_A$ that

- preserves $\vdash_A$ i.e. for all $x, y \in (M_A + \{ \dagger \})$, if $x \vdash_A y$ then $f^\dagger(x) \vdash_A f^\dagger(y)$ where $f^\dagger$ is the extension of f to $M_A + \{ \dagger \}$ that fixes $\dagger$, and
- preserves λ_A i.e. for all $m \in M_A$, $\lambda_A(f(m)) = \lambda_A(m)$.

It follows that f maps a move in the arena forest to a move at the same level and with the same label. As a consequence of the definition of FM-arenas, we have:

Lemma 4.2. *Every $\pi \in \textsc{perm}(\mathcal{N})$ defines an automorphism of an FM-arena.* $\square$

Henceforth, by arenas and prearenas, we shall mean FM-arenas and FM-prearenas. We give an example of a $\textsc{perm}(\mathcal{N})$-action on an arena A. We shall write such actions as $\pi \cdot_A m$, instead of $\pi \cdot_{M_A} m$.

Example 4.3. We revisit the arena A in Example 3.2. The move m therein has support $\{ a, b, c \}$, and its parent and grandparent have support $\{ a, b \}$ and $\{ a \}$ respectively. Take $\pi \in \textsc{perm}(\mathcal{N})$ that cycles through a, b and c. We have $\pi \cdot_A (1, (a, (2, (b, c)))) = (1, (b, (2, (c, a))))$. Observe that the permuted move remains at level 3; it is the name a whose parent and grandparent are c and b respectively.

Name Change Conditions. We shall consider justified sequences of *moves-with-names*, and introduce new conditions of *Name Change*, which are reminiscent of the State Change conditions introduced in [12]. A ***name set*** (ranged over by S, S_i, T, etc.) is a finite subset of $\mathcal{N}$. A ***move-with-names*** of a (pre-)arena A is a pair, written m^S (or (m, S), or just m if S is understood), where m is a move of A and S is a name set. We often write $m^\varnothing$ simply as m.

We first give an informal account of the kind of justified sequences of moves-with-names that model nu-calculus computations. We say that a name a *occurs* in m^S if it is an element of either S or $\text{SUPP}_A m$ (or of both, in some cases). Suppose *a first* occurs at m^S (i.e. a is *fresh* at that point) in some justified sequence; we say that a is *introduced* by P (resp. O) just if m is a P-move (resp. O-move). Intuitively the name set of a move consists of the names that have been introduced by P at moves that are P-*visible* (i.e. occur in the P-view) at that point. Note the asymmetry: names introduced by O are not recorded in name sets.

Formally, given a prearena, we shall consider justified sequences of moves-with-names that satisfy *Visibility* [6], *Well-Bracketing*, and the following ***Name Change Conditions***:

(NC1) The name set of a P-move contains the name set of the preceding O-move, and possibly some other names, all of which are *fresh* at that point.

(NC2) Any name in the support of a P-move *that is fresh at that point* is a member of the name set of that move.

(NC3) The name set of a non-initial O-move coincides with that of the P-move that explicitly justifies it.

We shall call justified sequences satisfying these conditions ***plays*** (or *legal positions*). Let S be a name set. An S-***play*** is defined to be a play that opens with a move with name set S. Thus, as a consequence of Name Change, in any S-play, the name set of every move contains S.

Example 4.4. Here is a play[3] of $\mathbf{1} \rightharpoonup (\nu \to \nu)$:

$$\begin{array}{lccccc}
 & \mathbf{1} & \rightharpoonup & (\nu & \to & \nu) \\
O & * & & & & \\
P & & & & *^{\{a\}} & \\
O & & & c^{\{a\}} & & \\
P & & & & & b^{\{a,b\}}
\end{array}$$

In the figure we align the moves with the components of the prearena from which they arise; pointers are omitted as they are completely determined by the type of the arena in this case. Note that the support of the last move is $\{c, b\}$, but only b is fresh at that point. Thus, according to (NC2), b, but not c, is in the name set of that move. See also the play v in Example 4.9 and the explanation therein.

We state a simple but important consequence of the Name Change Conditions.

Lemma 4.3. *(i) Every name that occurs in the name set of a move is introduced by some P-move that appears in the P-view at that point.*

(ii) The name set of a P-move contains the name set of the O-move that explicitly justifies it. □

[3]as taken from the strategy denoting $\vdash \boldsymbol{\nu} a.\boldsymbol{\lambda} x^{\nu}.\boldsymbol{\nu} b.b : \nu \to \nu$.

Recall that every $\pi \in \text{PERM}(\mathcal{N})$ defines an automorphism on an arena A. The action extends (canonically) to an action on the FM-set $\mathsf{list}(M_A \times \mathcal{P}_{\text{fs}}(\mathcal{N}) \times \mathbb{N})$, the set of lists of justified moves-with-names (with pointers encoded as appropriate numeric offsets), as follows: writing such a list as $s = m_1^{S_1} \cdots m_l^{S_l}$, we have

$$s \mapsto \pi \cdot_A m_1^{\pi \cdot_{\mathcal{P}_{\text{fs}}(\mathcal{N})} S_1} \cdots \pi \cdot_A m_l^{\pi \cdot_{\mathcal{P}_{\text{fs}}(\mathcal{N})} S_l}$$

Note that the pointer structure (not displayed above) is preserved as the action of any π is the identity on $\mathbb{N}$.

Lemma 4.4. *The set of plays (over a prearena A), $\mathbf{P}_A$, is an FM-subset of* $\mathsf{list}(M_A \times \mathcal{P}_{\text{fs}}(\mathcal{N}) \times \mathbb{N})$*, and so, $\mathbf{P}_A$ is itself an FM-set.* □

In the following we shall consider "S-plays up to renaming of all names, *except those in S*", where S ranges over name sets. In FM parlance, these are (equivalence classes of) S-plays abstracted until their support is (that of) $S \in \mathcal{P}_{\text{fs}}(\mathcal{N})$ i.e. elements of the form $s \restriction S$ of the FM-set $\mathbf{P}_A \restriction \mathcal{P}_{\text{fs}}(\mathcal{N})$ where s ranges over S-plays.

Notation. We shall write the element $s \restriction S$ as $[s]^S$, or simply as $[s]$ whenever s is understood to be an S-play.

An S-***strategy*** σ of a prearena A is a set of equivalence classes of S-plays of A satisfying:

(i) *Prefix-closure*: If $[su] \in \sigma$ then $[s] \in \sigma$.

(ii) If even-length $[s] \in \sigma$ and sm^T is an S-play then $[sm^T] \in \sigma$.

(iii) *Determinacy*: If even-length $[s_1\, m_1^{S_1}]$ and $[s_2\, m_2^{S_2}]$ are in σ and $[s_1] = [s_2]$ then $[s_1\, m_1^{S_1}] = [s_2\, m_2^{S_2}]$.

Remark 4.5. Why consider S-strategies (or why S-plays rather than $|S|$-plays)? Take $\{a, b\};\ \vdash\ a \not\approx_\nu b$. The two terms are denoted by distinct strategies determined by maximal plays $*^{\{a,b\}} \cdot a^{\{a,b\}}$ and $*^{\{a,b\}} \cdot b^{\{a,b\}}$ respectively. (Though the swap-action (ab) does map one play to the other, it is *not* an S-invariant permutation.) However $; \vdash \boldsymbol{\nu} ab.a \approx_\nu \boldsymbol{\nu} ab.b$, and they have the same denotation as determined by the maximal play $* \cdot a^{\{a,b\}}$.

Example 4.6. Let $S = \{a_1, \cdots, a_n\}$. An S-strategy σ over $\mathbf{1} \rightharpoonup \nu$ can be identified by the unique maximal play in σ. Non-null plays over $\mathbf{1} \rightharpoonup \nu$ have the following shapes:

1. $*^S \cdot a_i^{S \oplus \{b_1, \cdots, b_m\}}$ where $m \geq 0$
2. $*^S \cdot b_j^{S \oplus \{b_1, \cdots, b_m\}}$ where $m \geq 1$.

(We write $S \oplus T$ to mean $S \cup T$, and it has the force that S and T are disjoint.) Strategies of type 1 denote $S;\ \vdash\ \boldsymbol{\nu} b_1 \cdots b_m.a_i : \nu$, and those of type 2 denote $S;\ \vdash\ \boldsymbol{\nu} b_1 \cdots b_m.b_j : \nu$.

Composing strategies. Let S be a name set. Suppose σ and τ are S-strategies of prearenas $A \rightharpoonup B$ and $B \rightharpoonup C$ respectively. Their composite $\sigma\,;\tau$, which will be shown to be a S-strategy of $A \rightharpoonup C$, is defined in the style of "parallel composition with hiding in CSP" (as is standard in game semantics) as far as the underlying justified moves are concerned. Roughly speaking the name sets of the composite strategy are obtained by an appropriate union of the respective name sets of the component strategies.

Example 4.7. To illustrate the idea, take A, B and C to be $\mathbf{1}, (o \to o)$ and o respectively, and consider the following pair of composable plays[4] from σ and τ – call them u and v respectively:

	$\mathbf{1}$ $\overset{\sigma}{\rightharpoonup}$	$(o \to o)$	$(o \to o)$	$\overset{\tau}{\rightharpoonup}$ o
O	$*$			
P		$*^{\{a\}}$	$*$	
O		$\mathsf{t}^{\{a\}}$	$\mathsf{t}^{\{c\}}$	
P		$\mathsf{f}^{\{a,b\}}$	$\mathsf{f}^{\{c\}}$	
				$\mathsf{f}^{\{c,d\}}$

Note that the last two moves of v are distinct elements of $M_{(o\to o)\rightharpoonup o}$. The resultant interaction sequence is

$$w \;=\; * \cdot *^{\{a\}} \cdot \mathsf{t}^{\{a,c\}} \cdot \mathsf{f}^{\{a,b,c\}} \cdot \mathsf{f}^{\{a,b,c,d\}}.$$

The corresponding play in $\sigma\,;\tau$ is then obtained from w by "hiding" the B-moves i.e. $* \cdot \mathsf{f}^{\{a,b,c,d\}}$.

Let u be a sequence of moves-with-names from A, B and C together with justification pointers from all moves except those initial in A. Define $u \upharpoonright (B, C)$ to be the subsequence of u consisting of all moves-with-names (with pointers) from $B \rightharpoonup C$; similarly for $u \upharpoonright (A, B)$. We say that u is an *interaction sequence* of (A, B, C) if $u \upharpoonright (B, C)$ is a justified sequence of moves-with-names of $B \rightharpoonup C$ satisfying Visibility and Well-Bracketing (but not necessarily Name Change); similarly for $u \upharpoonright (A, B)$. We shall call $u \upharpoonright (B, C)$ the (B, C)-*component* of u, and call $u \upharpoonright (A, B)$ the (A, B)-*component* of u. Note that any move that occurs in an interaction sequence u is either a P-move of $A \rightharpoonup C$, or it is a *generalized O-move* in the sense that it is an O-move in *exactly* one of the two components of u.

We define a binary relation over justified sequences $s \lessdot t$ (read "t is a *name accession* of s") by recursion as follows:

- $\epsilon \lessdot \epsilon$
- $m^S \lessdot m^T$ provided $S \subseteq T$
- $s\, m^S\, n^{S'} \lessdot t\, m^T\, n^{T'}$ provided $s\, m^S \lessdot t\, m^T$, $S' \subseteq T'$, and if n is a P-move then $T' - T = S' - S$.

[4]Here σ and τ are taken to be $[\![\, \vdash \boldsymbol{\nu} a.\boldsymbol{\lambda} x^o.\boldsymbol{\nu} b.\neg x : o \to o \,]\!]$ and $[\![\, g : o \to o \vdash \boldsymbol{\nu} d.g(\boldsymbol{\nu} c.\mathrm{t}) : o \,]\!]$ respectively.

We will shortly use the relation $\lessdot$ in the definition of the composition of strategies. Whenever we write $s \lessdot t$, s will be a play, t will often not be (because Name Change may not necessarily be satisfied), but they have the same underlying sequence of moves (with name sets erased). For example taking s to be each of the composable pair of plays in Example 4.7, we have

(i) $* \cdot *^{\{a\}} \cdot \mathsf{t}^{\{a\}} \cdot \mathsf{f}^{\{a,b\}} \lessdot w \upharpoonright (A, B)$

(ii) $* \cdot \mathsf{t}^{\{c\}} \cdot \mathsf{f}^{\{c\}} \cdot \mathsf{f}^{\{c,d\}} \lessdot w \upharpoonright (B, C)$.

Definition 4.8. Take σ and τ be S-strategies as before. We define $\mathbf{ISeq}^S(\sigma, \tau)$ to be the set of interaction sequences u of (A, B, C), whose first move has name set S, satisfying:

I1. There exists some $[t] \in \tau$ such that $t \lessdot u \upharpoonright (B, C)$.

I2. There exists some $[s] \in \sigma$ such that $s \lessdot u \upharpoonright (A, B)$.

I3. Suppose m^S is explicitly justified by $m_0^{S_0}$ in u, and m is an O-move in $A \rightharpoonup C$. Then $S = S_0$.

I4. $\mathcal{L}_{(A,B)} \cap \mathcal{L}_{(B,C)} = \varnothing$, where $\mathcal{L}_\Theta$ is the set of names introduced by P-moves of the component Θ of u.

For any $u \in \mathbf{ISeq}^S(\sigma, \tau)$ we define $u \upharpoonright (A, C)$ to be the justified sequence of moves-with-names of $A \rightharpoonup C$ that is obtained from u by first deleting all B-moves and then resetting pointers from initial C-moves to the opening A-move. We can now define the composite strategy $\sigma\,;\tau \;=\; \{\, [u \upharpoonright (A, C)] : u \in \mathbf{ISeq}^S(\sigma, \tau) \,\}$.

Lemma 4.5. *Let σ and τ be S-strategies as before. For any $u \in \mathbf{ISeq}^S(\sigma, \tau)$, there is a unique t where $[t] \in \tau$ such that $t \lessdot u \upharpoonright (B, C)$, and there is a unique s where $[s] \in \sigma$ such that $s \lessdot u \upharpoonright (A, B)$.* □

Theorem 4.6. *Composition is well-defined and associative. I.e. for any S-strategies σ, τ and ρ over $A \rightharpoonup B, B \rightharpoonup C$ and $C \rightharpoonup D$ respectively, the set $\sigma\,;\tau$ as defined is an S-strategy over $A \rightharpoonup C$; further $(\sigma\,;\tau)\,;\rho = \sigma\,;(\tau\,;\rho)$.* □

Example 4.9. Consider the following pair of composable plays[5] from σ and τ – call them u and v respectively:

	$\mathbf{1}$ $\overset{\sigma}{\rightharpoonup}$	$(\nu \to \nu) \to \nu$	$(\nu \to \nu) \to \nu$	$\overset{\tau}{\rightharpoonup}$ ν
O	$*$			
P		$*^{\{c\}}$	$*$	
O		$*^{\{c\}}$	$*$	
P		$a^{\{a,c\}}$	a	
O		$b^{\{a,c\}}$	$b^{\{b\}}$	
P		$b^{\{a,c\}}$	b	
				b

[5]We define σ and τ to be $[\![\, \vdash \boldsymbol{\nu} c.\boldsymbol{\lambda} f.f(\boldsymbol{\nu} a.a) : (\nu \to \nu) \to \nu \,]\!]$ and $[\![\, F : (\nu \to \nu) \to \nu \vdash F(\boldsymbol{\lambda} x^\nu.\boldsymbol{\nu} b.b) : \nu \,]\!]$ respectively. Note that $w \upharpoonright (A, C)$ is the maximal play of $\sigma\,;\tau = [\![\, \vdash \boldsymbol{\nu} abc.b : \nu \,]\!]$.

In v, since the 5th move points to the 2nd, the former's name set is empty by (NC3). The resultant interaction sequence in $\mathbf{ISeq}^{\varnothing}(\sigma, \tau)$ is

$$w \;=\; * \cdot *^{\{c\}} \cdot *^{\{c\}} \cdot a^{\{a,c\}} \cdot b^{S} \cdot b^{S} \cdot b^{S}$$

where $S = \{a, b, c\}$. The reader may wish to check that $u \lessdot w \upharpoonright (A, B)$ and $u \lessdot w \upharpoonright (B, C)$.

Innocent S-strategies. By a P-view, we mean a justified sequence that is the P-view of some play; similarly for O-view. Note that the P-view of an S-play is necessarily an S-play. We say that an S-strategy σ is ***innocent*** just in case whenever even-length $[s_1\, a_1^{S_1}\, b_1^{T_1}] \in \sigma$ and odd-length $[s_2\, a_2^{S_2}] \in \sigma$ such that $\ulcorner s_1\, a_1^{S_1} \urcorner = \ulcorner s_2\, a_2^{S_2} \urcorner$ (as elements of $\mathbf{P}_A \upharpoonright S$) then, for some $b_2^{T_2}$, we have $[s_2\, a_2^{S_2}\, b_2^{T_2}] \in \sigma$ and $\ulcorner s_1\, a_1^{S_1}\, b_1^{T_1} \urcorner = \ulcorner s_2\, a_2^{S_2}\, b_2^{T_2} \urcorner$. Innocent S-strategies are completely determined by ***view functions***, which are partial FM-functions from odd-length P-views p (which are S-plays) to *justified* P-moves (i.e. a P-move together with a pointer into p). Precisely it is a subset f of $\mathbf{P}_A \upharpoonright S$, consisting of (equivalence classes of) even-length S-plays that are P-views, satisfying

(i) *Even-prefix closure*: If $[p\, a^T] \in f$ and $q\, b^U$ is an even-length prefix of p then $[q\, b^U] \in f$.

(ii) *Single-valued*: If $[p_1\, a_1^{T_1}], [p_2\, a_2^{T_2}] \in f$ such that $[p_1] = [p_2]$ then $[p_1\, a_1^{T_1}] = [p_2\, a_2^{T_2}]$ (as elements of $\mathbf{P}_A \upharpoonright S$).

We say that σ is *generated* by a view function f, written $\sigma = \boldsymbol{strat}(f)$, just in case for any odd-length $[s\, a^T] \in \sigma$, we have $[s\, a^T\, b^{T'}] \in \sigma$ if and only if $[\ulcorner s\, a^T \urcorner\, b^{T'}] \in f$.

Example 4.10. Consider the inequivalence (1) in § 2. The innocent strategy $[\![\, \vdash \boldsymbol{\nu} n.\boldsymbol{\lambda} x.n : o \to \nu \,]\!]$ is generated by a view function determined by the (respective equivalence classes of the) P-views $* \cdot *^{\{n\}} \cdot \mathsf{t}^{\{n\}} \cdot n^{\{n\}}$ and $* \cdot *^{\{n\}} \cdot \mathsf{f}^{\{n\}} \cdot n^{\{n\}}$ over $\mathbf{1} \rightharpoonup (o \to \nu)$; the corresponding P-views in $[\![\, \vdash \boldsymbol{\lambda} x.\boldsymbol{\nu} n.n : o \to \nu \,]\!]$ are $* \cdot * \cdot \mathsf{t} \cdot n^{\{n\}}$ and $* \cdot * \cdot \mathsf{f} \cdot n^{\{n\}}$. Thus the terms are denoted by distinct innocent strategies.

The main result of the section is:

Theorem 4.7. *Suppose $\sigma : A \rightharpoonup B$ and $\tau : B \rightharpoonup C$ are innocent S-strategies, then the composite $\sigma ; \tau$ is an innocent S-strategy.* □

5 Several categories of nominal games

Henceforth, without further mention, by a strategy we mean an innocent strategy. Let S be a name set. We define the category $\mathbb{V}^S$ that has arenas as objects; its maps $A \longrightarrow B$ are given by S-strategies over the prearena $A \rightharpoonup B$. We write $\mathbb{V}^{\varnothing}$ simply as $\mathbb{V}$.

Interpreting ν-abstraction. The "type-2" strategy in Example 4.6, as determined by the unique (equivalence class of the) play $*^S \cdot a^{S \oplus \{a\}}$ – call it $\mathsf{new} : \mathbf{1} \longrightarrow \nu$, is the denotation of $S; \vdash \boldsymbol{\nu} b.b : \nu$. More generally take a $\mathbb{V}^{S \oplus \{a\}}$-map $f : A \longrightarrow B$ as given by a view function f, we define $\boldsymbol{\nu} a.f$ to be the $\mathbb{V}^S$-map $A \longrightarrow B$ as given by the view function consisting of (equivalence classes of) even-length P-views $[m_0^S\, m_1^{T \oplus \{a\}}\, u]$ provided $[m_0^{S \oplus \{a\}}\, m_1^{T \oplus \{a\}}\, u] \in f$ *and* $a \notin \textsc{supp}(m_0)$. This gives the interpretation of $S; \Gamma \vdash \boldsymbol{\nu} a.M : B$ in terms of that of $S \oplus \{a\}; \Gamma \vdash M : B$.

Total-map category $\mathbb{V}^S_{\mathsf{t}}$. Following [5], a $\mathbb{V}^S$-map $\sigma : A \longrightarrow B$ is said to be ***total*** just in case σ responds to every opening question with an answer that introduces *no* new name. I.e. the view function of σ is defined on each opening move a^S, and maps it to an answer b^S. Note that such an answer must be an initial move of the arena B (because in A, any move justified by an opening move – which must be an answer – is necessarily a question). It is easy to see that total maps compose. We denote the subcategory of arenas and total maps as $\mathbb{V}^S_{\mathsf{t}}$. It is straightforward to verify that $\mathbb{V}^S_{\mathsf{t}}$ has $\mathbf{0}$ as the initial object, $\mathbf{1}$ as the terminal object, and binary products given by $\otimes$. In contrast, $\mathbb{V}^S$ does not have binary products ($\otimes$ is not even bifunctorial); rather it is a premonoidal category in the sense of [15].

Lifted arenas $A_\bot$. Using our forest construction notation, we define the *lifted arena* $A_\bot$ of an arena A as

$$A_\bot \;=\; \bot^{PA} \langle\, \bot^{OQ} \langle\, A \,\rangle \rangle.$$

Note that $A_\bot$ is (graph)-isomorphic to $\mathbf{1} \to A$, and they are isomorphic in $\mathbb{V}^S$. There are two canonical maps: the $\mathbb{V}^S_{\mathsf{t}}$-map $\mathsf{up}_A : A \longrightarrow A_\bot$ (which is a mono), and the $\mathbb{V}^S$-map $\mathsf{dn}_A : A_\bot \longrightarrow A$.

Proposition 5.1. *The inclusion functor $I : \mathbb{V}^S_{\mathsf{t}} \longrightarrow \mathbb{V}^S$ has as right adjoint the lift functor $L(A) = A_\bot$, with unit up_A and counit dn_A. The associated monad on $\mathbb{V}^S_{\mathsf{t}}$, $\langle T, \mathsf{up}, L\, \mathsf{dn}\, I \rangle$, where $T = L\, I$ has a tensorial strength $t_{A,B} : A \otimes TB \longrightarrow T(A \otimes B)$.* □

Since the left adjoint is the inclusion functor, by a famous result of Kleisli, the associated Kleisli category $(\mathbb{V}^S_{\mathsf{t}})_T$ is equivalent to $\mathbb{V}^S$. Recall that a λ_c-*model* [11] over a category $\mathbb{C}$ with finite products is a strong monad (T, η, μ, t) together with a T-exponential for every pair (A, B) of $\mathbb{C}$-objects i.e. a bijection

$$\mathbb{C}(C \times A, TB) \;\cong\; \mathbb{C}(C, (TB)^A)$$

natural in A and B, for each $\mathbb{C}$-object C. We call $(TB)^A$ the *exponential of TB by A*.

Proposition 5.2 (λ_c-model). *For each S, $\mathbb{V}^S_{\mathsf{t}}$ is a λ_c-model: the exponential of TB by A is $A \to B$.* □

Thus it follows from [11] that we can interpret CBV λ-calculus in $\mathbb{V}^S_{\mathtt{t}}$.

Remark 5.1. One could define an interpretion of the CBV λ-calculus *directly* in the Kleisli category $(\mathbb{V}^S_{\mathtt{t}})_T \cong \mathbb{V}^S$, following [5]. Indeed it would seem natural to do so since (in the Honda-Yoshida approach to CBV games) the Kleisli category (or rather its equivalent) $\mathbb{V}^S$ is the prior construction, and the λ_c-model, $\mathbb{V}^S_{\mathtt{t}}$, is a derived notion. The downside is one would need to check that the partial pairing and projection maps of $\mathbb{V}^S$ are sufficiently well-behaved for the interpretation. The two interpretations are of course essentially equivalent, as they are related via the natural isomorphism $\mathbb{V}^S(A, B) \cong \mathbb{V}^S_{\mathtt{t}}(A, TB)$. However when considering examples, we prefer to give denotations in $\mathbb{V}^S$ because they are simpler.

We say that a $\mathbb{V}^S$-map $\sigma : A \longrightarrow o$ is ***truthful*** just in case σ responds to every opening question q^S with (the answer) $\mathtt{t}^{S'}$ i.e. "true" with name set S'; note that by (NC3), we have $S \subseteq S'$. For arenas A and B, we define a binary relation $\lesssim$ over $\mathbb{V}^S$-maps $A \longrightarrow B$ as follows. We say $\sigma_1 \lesssim \sigma_2$ just in case for any $\mathbb{V}^S$-maps $\rho : C \longrightarrow A$ and $\chi : B \longrightarrow o$, if $\rho ; \sigma_1 ; \chi$ is truthful, so is $\rho ; \sigma_2 ; \chi$.

Lemma 5.3. *For any $\mathbb{V}^S$-maps $\sigma_1, \sigma_2 : A \longrightarrow B$, $\sigma_1 \lesssim \sigma_2$ iff for any $\rho : \mathbf{1} \longrightarrow A$ and $\chi : B \longrightarrow o$, if $\rho ; \sigma_1 ; \chi$ is truthful, so is $\rho ; \sigma_2 ; \chi$.* □

Extensional category $\widehat{\mathbb{V}^S_{\mathtt{t}}}$. It is straightforward to verify that $\lesssim$ is a preorder. We write the equivalence relation induced by $\lesssim$ as $\approx$, and define the extensional category $\widehat{\mathbb{V}^S}$ with arenas as objects; and maps $A \longrightarrow B$ are given by $\approx$-equivalence classes of $\mathbb{V}^S$-maps. That composition in $\widehat{\mathbb{V}^S}$ is well-defined follows from the fact that it is monotone with respect to $\lesssim$. It turns out that $\widehat{\mathbb{V}^S_{\mathtt{t}}}$ is also a λ_c-model, using exactly the same constructions as $\mathbb{V}^S_{\mathtt{t}}$.

Example 5.2. We revisit the equivalence (3) in §2. Setting $M = \boldsymbol{\nu} nn'.\boldsymbol{\lambda} f.fn = fn'$ and $N = \boldsymbol{\lambda} f.\mathtt{t}$, we aim to show $[\![M]\!] = [\![N]\!]$ in $\widehat{\mathbb{V}}$. By Lemma 5.3 it suffices to show that for any $\mathbb{V}$-map $\rho : (\nu \to o) \to o \longrightarrow o$, $[\![M]\!] ; \rho$ is truthful iff $[\![N]\!] ; \rho$ is truthful. Consider the prearena $(\nu \to o_1) \to o_2 \rightharpoonup o$ (we label the copies of o, so that moves from them have the corresponding labels; similarly we denote the initial moves of the subarenas $\nu \to o_1$ and $(\nu \to o_1) \to o_2$ respectively as $*_1$ and $*_2$). Take $a \in \mathcal{N}$. If the even-length $[*_2 \cdot *_1 \cdot a \cdot m] \in \rho$, then for any $b \in \mathcal{N}$, we have $[*_2 \cdot *_1 \cdot b \cdot m] = [*_2 \cdot *_1 \cdot a \cdot m] \in \rho$. Consequently we have the following interaction sequence in $\mathbf{ISeq}^{\varnothing}([\![M]\!], \rho)$:

$$* \cdot *_2^{\{a,b\}} \cdot *_1^{\{a,b\}} \cdot a^{\{a,b\}} \cdot m_1^{\{a,b\}} \cdot b^{\{a,b\}} \cdot m_1^{\{a,b\}} \cdot \mathtt{t}_2^{\{a,b\}}.$$

Now the corresponding justified sequence in $\mathbf{ISeq}^{\varnothing}([\![N]\!], \rho)$ is $* \cdot *_2 \cdot *_1 \cdot \mathtt{t}_2$. When projected to $(\nu \to o) \to o \rightharpoonup o$, the two sequences have the same P-view. It follows from the innocence of ρ that $[\![M]\!] ; \rho$ is truthful iff $[\![N]\!] ; \rho$ is truthful. We know of no previous model of the nu-calculus that identifies M and N.

6 A fully abstract game model

We interpret a term-in-context $S; \Gamma \vdash M : B$, where $\Gamma = x_1 : A_1, \cdots, x_n : A_n$, as a map $A_1 \otimes \cdots \otimes A_n \longrightarrow B$ in $\widehat{\mathbb{V}^S_{\mathtt{t}}}$ (or equivalently in $\widehat{\mathbb{V}^S}$). It is straightforward to verify:

Lemma 6.1. *All rules and axioms in Figure 3 of [18] are validated by the extensional game model $\widehat{\mathbb{V}^S_{\mathtt{t}}}$.* □

Indeed we can show:

Lemma 6.2. *$\widehat{\mathbb{V}_{\mathtt{t}}}$ is a model of the nu-calculus in the sense of Stark in [20].*

Proof. (Sketch) Prop. 5.2 shows $\widehat{\mathbb{V}_{\mathtt{t}}}$ is a λ_c-model. (Note that the cartesian closure requirement in [20] is not necessary: it suffices for the category to have all T-exponentials.) The coproduct $\mathbf{1} + \mathbf{1}$ exists and is disjoint; the interpretation o of booleans is isomorphic to it. There is a distinguished, decidable object ν to interpret names. There is a distinguished $\mathbb{V}$-map $\mathsf{new} : \mathbf{1} \longrightarrow \nu$ satisfying the three axioms in [20, §4.1]. The first axiom requires an argument similar to that in Example 5.2, the second and third axioms hold in $\mathbb{V}_{\mathtt{t}}$ (and hence in $\widehat{\mathbb{V}_{\mathtt{t}}}$) and follow straightforwardly from the FM structure of plays and condition I4 respectively. □

Thanks to [20], this guarantees *adequacy* of our model: for any ground type b, $S \vdash M \Downarrow_b (S')C$ iff $[\![S \vdash M : b]\!] = [\![S \vdash \boldsymbol{\nu} S'.C : b]\!]$ in $\widehat{\mathbb{V}^S_{\mathtt{t}}}$. The main result of the paper is that our model is ***equationally fully abstract*** for the nu-calculus. I.e. for any $S; \Gamma \vdash M_i : A$ for $i = 1, 2$, we have $S; \Gamma \vdash M_1 \approx_A M_2$ iff $[\![S; \Gamma \vdash M_1 : A]\!] = [\![S; \Gamma \vdash M_2 : A]\!]$ as maps in $\widehat{\mathbb{V}^S_{\mathtt{t}}}$. One direction "⇒" follows from adequacy, the other is a straightforward consequence of the following definability result.

Theorem 6.3 (Definability). *For any denotable arenas $A_1, \cdots, A_n$ and B, for any totally-defined compact (i.e. generated by a finite view function) innocent S-strategy $\sigma : (\bigotimes_{i=1}^n A_i) \rightharpoonup B$, there is a term M_σ such that $[\![S; x_1 : A_1, \cdots, x_n : A_n \vdash M_\sigma : B]\!] = \sigma$ in $\mathbb{V}^S$.* □

Example 6.1. Consider the $\{a\}$-strategy σ over the prearena $A = (\nu \otimes \nu) \rightharpoonup \nu$ given by the following view function: (we list representatives of the respective equivalence classes in $\mathbf{P}_A \upharpoonright \{a\}$)

1. $(a, a)^{\{a\}} \mapsto a^{\{a\}}$
2. $(a, l_2)^{\{a\}} \mapsto l_2^{\{a\}}$
3. $(l_1, a)^{\{a\}} \mapsto l_1^{\{a,c\}}$
4. $(l_1, l_1)^{\{a\}} \mapsto l_1^{\{a\}}$
5. $(l_1, l_2)^{\{a\}} \mapsto l^{\{a,l\}}$

$$\begin{array}{l} \{\, a \,\}; x : \nu, y : \nu \\ \vdash \quad \mathsf{if}\ x = a\ \mathsf{then} \left\{ \mathsf{if}\ y = a\ \mathsf{then} \underbrace{a}_{1}\ \mathsf{else} \underbrace{y}_{2} \right\} \mathsf{else} \left\{ \mathsf{if}\ y = a\ \mathsf{then} \underbrace{\boldsymbol{\nu} c.x}_{3}\ \mathsf{else} \left[\mathsf{if}\ x = y\ \mathsf{then} \underbrace{x}_{4}\ \mathsf{else} \underbrace{\boldsymbol{\nu} l.l}_{5} \right] \right\} : \nu \end{array}$$

Table 1. The term M_σ in Example 6.1.

The term M_σ denoting σ is shown in Table 1. The section of the view function labelled i corresponds to the i-labelled path in the decision tree of nested conditionals which is M_σ.

7 Further directions

Pitts and Stark [16] have shown that observational equivalence of nu-calculus is decidable for first-order terms. We have a new proof using the fully abstract game model. Our aim is to use the game model to resolve the question of decidability for the second-order fragment.

What is a (categorical) model of the nu-calculus? The answer in [20] gives a way of understanding the nu-calculus as mediated by a computational metalanguage. We seek a neutral and direct analysis in a name-indexed setting, in which we would expect reindexing along the inclusion $S \longrightarrow S \oplus \{\, a \,\}$ to have a right adjoint of the shape $[\mathcal{N}](-)$ i.e. the *name abstraction* constructor in [4].

We are interested in a particular extension of the nu-calculus, where names serve as references to dynamically-allocated cells for storing integers. This is the *Reduced ML* of [19, §5], and similar to the languages treated in [17, 13]. The next step is to construct a fully abstract game model for *Reduced ML*. It would be interesting to clarify the connections with the fully abstract model of named reference in [8]. In a different direction, we also plan to use nominal games to model object-oriented languages in which names are used as the unique identities of objects.

Acknowledgements. We thank Andy Pitts for helpful advice on FM-sets, EPSRC for financial support (GR/R88861), and Merton College, Oxford for a Visiting Research Fellowship (for Stark).

References

[1] S. Abramsky, K. Honda and G. McCusker. Fully abstract game semantics for general reference. In *Proc. LICS '98*.

[2] S. Abramsky and G. McCusker. Linearity, sharing and state: a fully abstract game semantics for Idealized Algol with active expressions. In *Algol-like languages*. Birkhaüser, 1997.

[3] S. Abramsky and G. McCusker. Call-by-value games. In *Proc. CSL '97*. 1998. LNCS 1414.

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

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

[6] J. M. E. Hyland and C.-H. L. Ong. On Full Abstraction for PCF: I, II, III *Info. & Comp.*, 163:285–408, 2000.

[7] J. D. Laird. A fully abstract game semantics of local exception. In *Proc. LICS'01*. 2001.

[8] J. D. Laird. A game semantics of local names and good variables. In *Proc. FOSSACS'04*. 2004. LNCS Vol. 2987.

[9] G. McCusker. *Games for Recursive Types*. CUP, 1998.

[10] A. J. R. Milner. *Communicating and Mobile Systems: The Pi-Calculus*. CUP, 1999.

[11] E. Moggi. Notions of computation and monads. *Info. & Comp.*, 93:55–92, 1989.

[12] C.-H. L. Ong. Observational equivalence of third-order Idealized Algol is decidable. *Proc. LICS'02*, pp. 245–256, 2002.

[13] A. M. Pitts. Operational semantics and program equivalence. In *Applied Semantics*, pp. 378–412, 2002. LNCS 2395.

[14] A. M. Pitts and M. J. Gabbay. A metalanguage for programming with bound names modulo renaming. In *MPC 2000*, pp. 230–255, 2000. LNCS 1837.

[15] A. J. Power and E. P. Robinson. Premonoidal categories and notions of computation *MSCS*, 7:453–468, 1997.

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

[17] A. M. Pitts and I. D. B. Stark. Operational reasoning for functions with local state. In *Higher Order Operational Techniques in Semantics*, pp. 227-273. CUP, 1998.

[18] I. D. B. Stark. Names, equations, relations: Practical ways to reason about *new*. *Fund. Informaticae*, 33:369–396, 1998.

[19] I. D. B. Stark. *Names and Higher-Order Functions*. PhD thesis, University of Cambridge, December 1994.

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