Nominal Games and Full Abstraction for the Nu-Calculus

Ian Stark

School of Informatics
The University of Edinburgh

Samson Abramsky, Dan Ghica
Andrjez Murawski, Luke Ong

Computing Laboratory
Oxford University

Summary

We present nominal games, a game semantics in Gabbay and Pitts' world of FM-set theory, as a model for programming languages with dynamically generated local names.

This gives the first fully-abstract denotational semantics for the *nu-calculus*, a lambda-calculus with fresh name generation.

The FM-theory of *nominal sets* is a significant enabler, providing:

- General operations for freshness and privacy — $A * B$, $[A]B$
- Explicit mention of private names in custom constructions

. . . while keeping us honest about the proper anonymity of names.

Outline of talk

- Nu-calculus
- FM set theory and nominal sets
- Nominal game semantics
- Definability and full abstraction results

A calculus for functions and local names

The nu-calculus combines

- the simply-typed lambda-calculus $(A \to B)$; MN, $\lambda x{:}A.M$
- with names: $n, m : \nu$
- and name restriction: $\nu n.M$ (à la π-calculus).

A call-by-value operational semantics means that name restriction also serves as name creation.

- Functions may have private names, that persist from one use to the next: $\nu n.(\lambda x.\lambda y.\ ---)$
- Names may pass beyond their original scope and outlive their creator: $\nu n.n$

Observational equivalence with names

Terms in the nu-calculus are observationally equivalent if they give the same result in any boolean context $\mathcal{C}[-]$.

$$\nu n.(n = n) \approx \mathit{true} \qquad \nu n.\nu n'.(n = n') \approx \mathit{false}$$

$$\begin{aligned}
\nu n.(\lambda x.n) &\not\approx \lambda x.(\nu n.n) && : o \to \nu \\
\nu n.\lambda x.(x = n) &\approx \lambda x.\mathit{false} && : \nu \to o \\
\nu n.\nu n'.\lambda f.(fn = fn') &\approx \lambda f.\mathit{true} && : (\nu \to o) \to o
\end{aligned}$$

Methods based on logical relations show that observational equivalence is decidable up to first order.

No decidability results yet for 2nd or higher order.

No previous model fully abstract above first order.

Sets with names

Fix a countably infinite set of names $\mathcal{N}$. A nominal set X is a set with the following structure.

- An action of $\text{PERM}(\mathcal{N})$ on elements of X:

$$\forall \pi \in \text{PERM}(\mathcal{N})\ \forall x \in X\ .\ \pi \cdot x \in X$$

- For every $x \in X$, some finite support $A_x \subset \mathcal{N}$:

$$\forall \pi\ .\ \pi|_{A_x} = \mathrm{id}|_{A_x} \implies \pi \cdot x = x$$

Examples: $\mathcal{N}$ itself, $\mathcal{P}_{\mathrm{fin}}(\mathcal{N})$, any set with trivial action;
$X \times Y$, $X + Y$, $\mathrm{list}(X)$,. . . ; $X * Y$, $[X]Y$, $X \restriction Y \ldots$

Nominal sets are a Fraenkel-Mostowski permutation model of set theory with atoms [Gabbay and Pitts 2001, 2003]

More sets with names

A nominal subset $U \subseteq X$ is any subset of X that is closed under the permutation action:

$$x \in U \implies \pi \cdot x \in U \,.$$

A nominal relation $R \subseteq X \times Y$ is one preserved by the action:

$$x \mathrel{R} y \implies (\pi \cdot x) \mathrel{R} (\pi \cdot y) \,.$$

A nominal function $f : X \to Y$ is *equivariant* under permutation:

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

The first part of nominal game semantics is simply to use the nominal version of all constructions.

Game semantics

The structure of our games model is standard, with computation as interaction between a system and its environment.

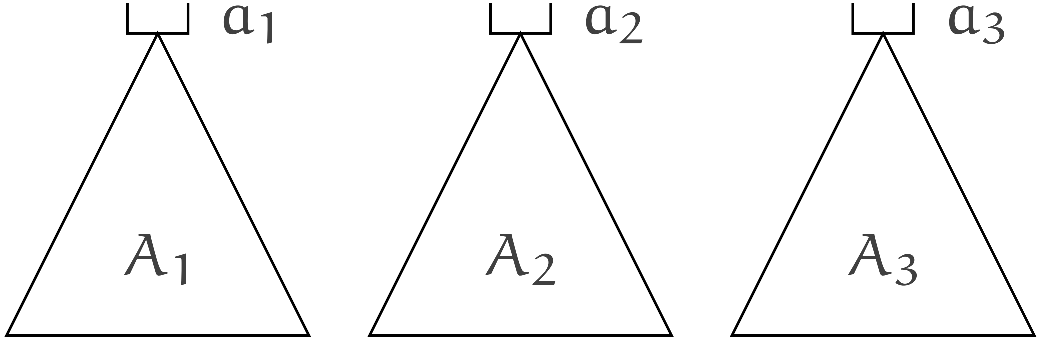
- We have a collection of concrete player-opponent games $A, B, \ldots,$ and some constructions $A \otimes B, A \to B, \ldots$
- Strategies are directions for the Player in such games.
- We model the nu-calculus with types as games, and terms as strategies: i.e. in the category of games, where an arrow from A to B is a strategy for playing the game $A \to B$.
- We can prove definability: that every strategy denotes some nu-calculus term.

An extensional collapse then gives the fully-abstract model.

Arenas for nominal games

Move set	Justification relation	Labelling function
M	$\vdash \subseteq M \times M$	$M \to \{\mathsf{O}, \mathsf{P}\} \times \{\mathsf{Q}, \mathsf{A}\}$

A sample call-by-value arena:

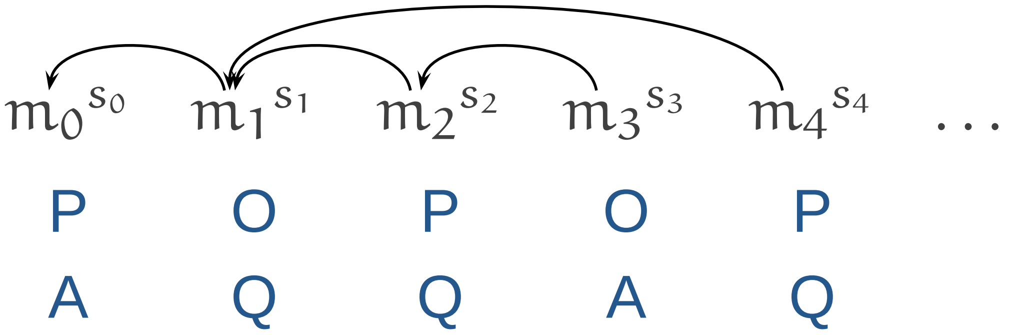


Making these nominal sets, relations and functions gives an automorphism action of $\mathrm{PERM}(\mathcal{N})$ on arenas themselves.

The flat arena with move set $\mathcal{N}$ interprets the type of names.

Nominal game play

A play in a nominal game over arena A is a sequence of moves with justification pointers and *name set annotations*



$m_0{}^{s_0} \quad m_1{}^{s_1} \quad m_2{}^{s_2} \quad m_3{}^{s_3} \quad m_4{}^{s_4} \quad \ldots$

P	O	P	O	P
A	Q	Q	A	Q

satisfying certain conditions:

- P/O alternation, Q/A justification, bracketing, visibility *etc.*
- Name Change: O-moves must preserve name sets; P-moves must add *at least* all names introduced by P.

Name sets denote the names generated by P, including those not (yet) exposed in moves.

Nominal plays and strategies

An $\mathcal{S}$-play is a nominal play $p^{\mathcal{S}}$ with name set $\mathcal{S} \in \mathcal{P}_{\mathrm{fin}}(\mathcal{N})$ on the initial move: $m_0^{\mathcal{S}} \cdots$

Take the equivalence classes $[p]^{\mathcal{S}}$ of these up to permutation of all names *except those in* $\mathcal{S}$. The $[p]^{\mathcal{S}}$ form a nominal set.

An $\mathcal{S}$-strategy $\sigma : A \to B$ is a prefix-closed set of equivalence classes of $\mathcal{S}$-plays on the game $A \to B$. Strategies compose by parallel composition of plays, with hiding.

Nominal games and (deterministic, innocent) $\mathcal{S}$-strategies form a category $\mathbb{V}^{\mathcal{S}}$.

Definability

There is a particular S-strategy for name creation:

$$new \overset{\text{def}}{=} \{*^{S}.a^{S\oplus\{a\}}\} \;:\; 1 \longrightarrow \mathcal{N}$$

We use this to interpret any nu-calculus term $S;\Gamma \vdash M : B$ as a map

$$[\![M]\!]_{S} : A_1 \otimes \cdots \otimes A_n \longrightarrow B\,.$$

in the category $\mathbb{V}^{S}$.

Thm. Every (total, finite) strategy between arenas interpreting nu-calculus types is the interpretation of some nu-calculus term.

Proof is by induction on the size of the strategy (as a view function).

Full abstraction

A strategy $\sigma : B \to \{\mathit{true}, \mathit{false}\}$ in $\mathbb{V}^S$ is truthful if for every opening question (in B) the response is true.

We define extensional equivalence between strategies $\sigma_1, \sigma_2 : A \to B$ by

$$\sigma_1 \approx \sigma_2 \quad \overset{\text{def}}{\Longleftrightarrow} \quad \begin{cases} \text{for all } \rho{:}C \to A \text{ and } \chi{:}B \to \{\mathit{true}, \mathit{false}\}, \\ \rho;\sigma_1;\chi \text{ is truthful iff } \rho;\sigma_2;\chi \text{ is truthful.} \end{cases}$$

Thm. The extensional collapse $\widehat{\mathbb{V}}$ which identifies (total) maps up to $\approx$ is:

(i) An adequate model of the nu-calculus. [Stark 96]

(ii) Fully abstract for observational equivalence. (By definability)

Review

We obtain a fully-abstract denotational semantics for the nu-calculus by adapting game models, using the following:

- Nominal sets as a general name-aware framework.
- Name-set annotations on moves to hold local state.
- Equivalence classes under name permutations to make that state private.

We observe that (yet again) games provide a powerful technique for precise semantics of programming language features.

Related and further work

Earlier game models of state treat local variables as free (reader,writer) pairs, including so-called “bad variables”. This does not support names, or testing for equality of references.

Laird [FoSSaCS 2004] uses names to give a game model for $\lambda\nu!$, an extension of the nu-calculus with name storage cells.

Next steps:

- Investigate decidability of nu-calculus observational equivalence at second order.
- Use nominal games to model the integer reference cells of *Reduced ML*.