NOMINAL GAME SEMANTICS

PART I

Andrzej Murawski
UNIVERSITY OF OXFORD

Winners of the 2019 Alonzo Church Award

By siglog April 17, 2019 0

The 2019 Alonzo Church Award for Outstanding Contributions to Logic and Computation is given jointly to Murdoch J. Gabbay and Andrew M. Pitts for their ground-breaking work introducing the theory of nominal representations.

The ACM Special Interest Group on Logic (SIGLOG), the European Association for Theoretical Computer Science (EATCS), the European Association for Computer Science Logic (EACSL), and the Kurt Gödel Society (KGS) are pleased to announce that Murdoch J. Gabbay (Heriot-Watt University, UK) and Andrew M. Pitts (Cambridge University, UK) have been selected as the winners of the 2019 Alonzo Church Award for Outstanding Contributions to Logic and Computation. The award recognizes their ground-breaking work introducing the theory of nominal representations, a powerful and elegant mathematical model for computing with data involving atomic names, described in the following papers:

Winners of the 2017 Alonzo Church Award

By siglog May 4, 2017 0

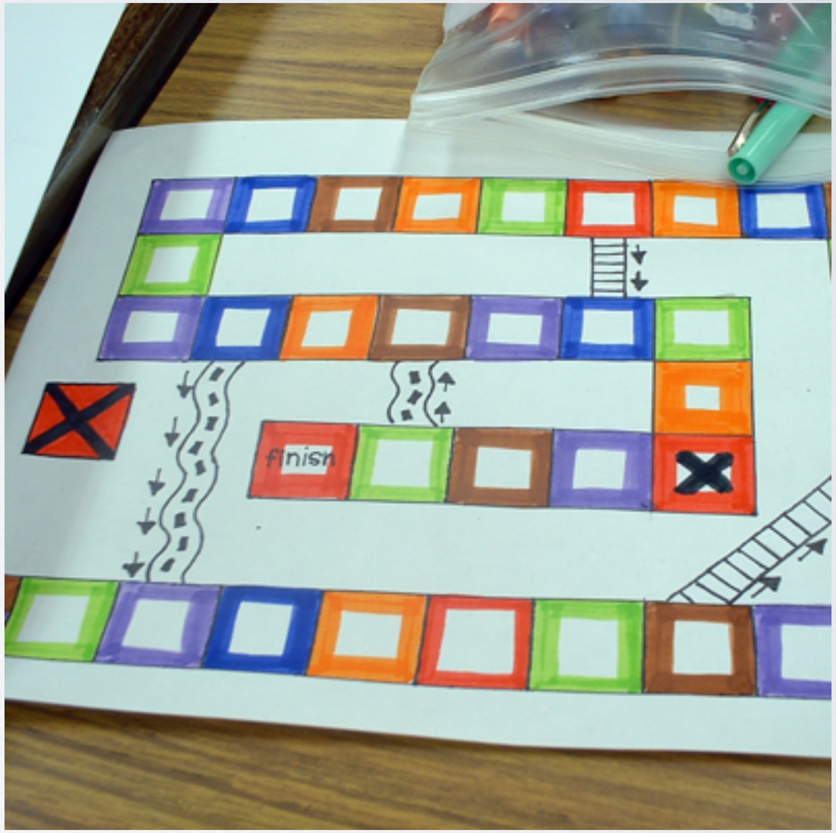
The 2017 Alonzo Church Award for Outstanding Contributions to Logic and Computation is given jointly to Samson Abramsky, Radha Jagadeesan, Pasquale Malacaria, Martin Hyland, Luke Ong, and Hanno Nickau for providing a fully-abstract semantics for higher-order computation through the introduction of game models, thereby fundamentally revolutionising the field of programming language semantics, and for the applied impact of these models.

Their contributions appeared in three papers:

DISCLAIMERS

- computer games
- game theory
- games logicians play
- parity games

OLYMPIC SPIRIT



NOMINAL GAME SEMANTICS

- Semantics
- Game semantics
- Nominal game semantics

TOWARD A MATHEMATICAL SEMANTICS FOR COMPUTER LANGUAGES

this passage. The purpose of a mathematical semantics is to give a correct and meaningful correspondence between programs and mathematical entities in a way that is entirely independent of an implementation. This plan is illustrated in a very elementary

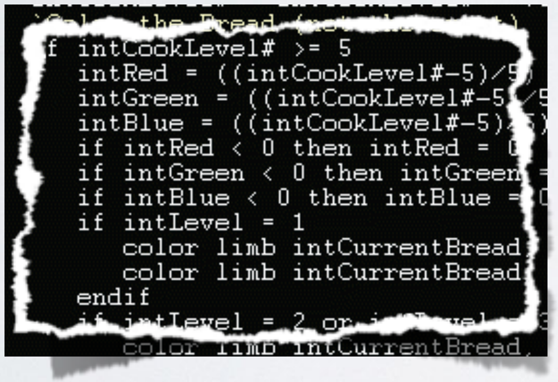
Dana Scott

and

Christopher Strachey

Oxford University
Computing Laboratory
Programming Research Group-Library
8-11 Keble Road
Oxford OX1 3QD
Oxford (0865) 54141

MATHEMATICAL SEMANTICS



$[\![M]\!]$ $\Rightarrow$

function

continuous function

strategy

PCF (SCOTT/MILNER/PLOTKIN)

- Programming Computable Functions
- Prototypical purely functional language
- Features integer arithmetic, higher-order functions and recursion
- Inspired early research on semantics

PCF TYPES

$$\theta \quad ::= \quad \mathsf{int} \mid \theta \to \theta$$

PCF TERMS

$$
\begin{aligned}
M \quad ::= \; & i \mid x \\
& \mid M \oplus M \mid \textsf{if } M \textsf{ then } M \textsf{ else } M \\
& \mid \lambda x^{\theta}.M \mid M\,M \mid \textsf{div}_{\theta}
\end{aligned}
$$

TYPING JUDGMENTS

$$x_1 : \theta_1, \cdots, x_n : \theta_n \vdash M : \theta$$

PCF TYPING JUDGMENTS

$$\frac{i \in \mathbb{Z}}{\Gamma \vdash i : \mathsf{int}} \qquad \frac{(x : \theta) \in \Gamma}{\Gamma \vdash x : \theta}$$

$$\frac{\Gamma \vdash M : \mathsf{int} \quad \Gamma \vdash N : \mathsf{int}}{\Gamma \vdash M \oplus N : \mathsf{int}} \qquad \frac{\Gamma \vdash M : \mathsf{int} \quad \Gamma \vdash N_0 : \theta \quad \Gamma \vdash N_1 : \theta}{\Gamma \vdash \mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_0 : \theta}$$

$$\frac{\Gamma \uplus \{x : \theta\} \vdash M : \theta'}{\Gamma \vdash \lambda x^{\theta}.M : \theta \to \theta'} \qquad \frac{\Gamma \vdash M : \theta \to \theta' \quad \Gamma \vdash N : \theta}{\Gamma \vdash MN : \theta'}$$

$$\frac{}{\Gamma \vdash \mathsf{div}_{\theta} : \theta}$$

TOWARDS MEANINGFUL CORRESPONDENCES

- Operational semantics

$$M \longrightarrow M'$$

- We shall focus on several meaningful correspondences between mathematical and operational semantics.

REDUCTION

$$\begin{array}{rcll} i \oplus j & \longrightarrow & k & (k = i \oplus j) \\ \textsf{if}\ 0\ \textsf{then}\ M\ \textsf{else}\ M' & \longrightarrow & M' & \\ \textsf{if}\ i\ \textsf{then}\ M\ \textsf{else}\ M' & \longrightarrow & M & (i \neq 0) \\ (\lambda x.M)N & \longrightarrow & M[N/x] & \end{array}$$

$$\frac{M \longrightarrow M'}{E[M] \longrightarrow E[M']}$$

$$E \ ::= \ [\,] \mid E \oplus M \mid i \oplus E \mid \textsf{if}\ E\ \textsf{then}\ M\ \textsf{else}\ M \mid EM$$

I. CORRECTNESS

If $M \longrightarrow M'$ then $[\![M]\!] = [\![M']\!]$.

In particular, if $\vdash M : \mathsf{int}$ and $M \longrightarrow i$ then $[\![M]\!] = [\![i]\!]$.

2. ADEQUACY

The following converse would be too strong:

$$\text{if } [\![M]\!] = [\![M']\!] \text{ then } M \longrightarrow M'.$$

Instead one aims for:

$$\text{Given } \vdash M : \mathsf{int}, \text{ if } [\![M]\!] = [\![i]\!] \text{ then } M \longrightarrow i.$$

3. DEFINABILITY (NO JUNK)

Suppose $[\![\theta]\!]$ is the mathematical object corresponding to θ, i.e. terms $[\![\vdash M : \theta]\!]$ can be thought of as elements of $[\![\theta]\!]$.

$$\forall_{x \in [\![\theta]\!]} \quad \exists_{\vdash M_x : \theta} \quad x = [\![\vdash M_x : \theta]\!]$$

DOMAIN-THEORETIC SEMANTICS

- Plotkin's domain-theoretic model of PCF uses the following partial order to model int.

$$\begin{tikzcd} \cdots & -1 & 0 & 1 & \cdots \\ & & \perp \arrow[ull, dash] \arrow[ul, dash] \arrow[u, dash] \arrow[ur, dash] \arrow[urr, dash] & & \end{tikzcd}$$

- Terms are interpreted by monotone functions.
- The model is correct and adequate, but does not have the definability property.

FAILURE OF DEFINABILITY

Consider the **parallel-or** function

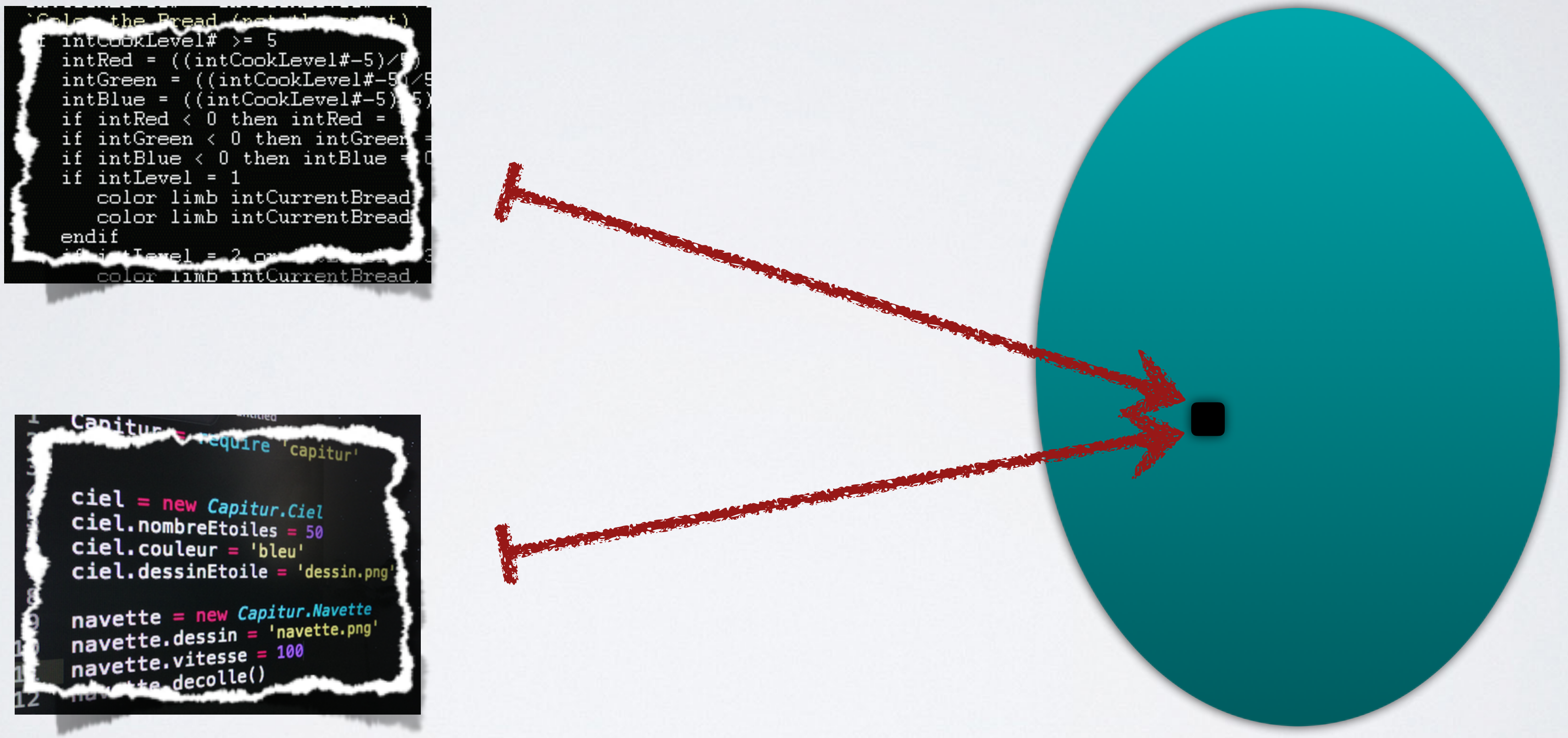
$$por\,x\,y = \begin{cases} 0 & x = 0 \text{ and } y = 0 \\ 1 & x \neq 0, \perp \text{ or } y \neq 0, \perp \\ \perp & \text{otherwise} \end{cases}$$

E.g. $por\,0\,0 = 0$ and $por\,1\,\perp = por\,\perp\,1 = 1$.

por turns out to be undefinable:
there is no PCF term M such that

$$M\ \mathsf{div}\ 1 \longrightarrow 1 \qquad M\ 1\ \mathsf{div} \longrightarrow 1 \qquad M\ 0\ 0 \longrightarrow 0$$

TOWARDS FULL ABSTRACTION



$$[\![M_1]\!] = [\![M_2]\!]?$$

4. FULL ABSTRACTION

$$[\![M_1]\!] = [\![M_2]\!] \quad \text{if and only if} \quad M_1 \cong M_2$$

Robin Milner (1977)

CONTEXTUAL TESTING

- Contexts

$$C \ ::= [\,] \mid C \oplus M \mid M \oplus C$$
$$\mid \text{if } C \text{ then } M \text{ else } M \mid \text{if } M \text{ then } C \text{ else } M \mid \text{if } M \text{ then } M \text{ else } C$$
$$\mid \lambda x^{\theta}.C \mid MC \mid CM$$

- Testing of $M : \theta$

$$C[M] : \text{int}$$

If there exists i such that $C[M] \longrightarrow^{*} i$, we write $C[M] \Downarrow$ (success!).

CONTEXTUAL EQUIVALENCE

Intuitively, two programs should be viewed as equivalent if they behave in the same way in any context, i.e. they can be used interchangeably.

- $\Gamma \vdash M_1 : \theta$ ***approximates*** $\Gamma \vdash M_2 : \theta$ if

  $$C[M_1]\Downarrow \text{ implies } C[M_2]\Downarrow$$

  for any context C such that $\vdash C[M_1], C[M_2] : \mathsf{int}$. Then we write $\Gamma \vdash M_1 \sqsubseteq M_2$.

- Two terms are ***equivalent*** if one approximates the other, written $\Gamma \vdash M_1 \cong M_2$.

SOUNDNESS

Correctness and adequacy turn out to imply:

$$\text{if } [\![M_1]\!] = [\![M_2]\!] \text{ then } M_1 \cong M_2.$$

Assume $[\![M_1]\!] = [\![M_2]\!]$ and suppose $M_1 \ncong M_2$, i.e. $C[M_1] \Downarrow$ and $C[M_2] \not\Downarrow$ for some context C (or $C[M_2] \Downarrow$ and $C[M_1] \not\Downarrow$).

- Correctness implies $[\![C[M_1]]\!] = [\![i]\!]$ for some i.
- Adequacy implies $[\![C[M_2]]\!] \neq [\![i]\!]$ for any i.

This is a contradiction, because $[\![M_1]\!] = [\![M_2]\!]$ implies $[\![C[M_1]]\!] = [\![C[M_2]]\!]$ by compositionality.

NO FULL ABSTRACTION

(FOR THE DOMAIN-THEORETIC MODEL)

$$
\begin{aligned}
M_1 \equiv\ & \lambda f^{\mathsf{int}\to\mathsf{int}\to\mathsf{int}}.\ \ \mathsf{if}\ (f\ 1\ \mathsf{div})\ \mathsf{then} \\
& \qquad\qquad (\mathsf{if}\ (f\ \mathsf{div}\ 1)\ \mathsf{then} \\
& \qquad\qquad\qquad (\mathsf{if}\ (f\ 0\ 0)\ \mathsf{then\ div\ else}\ 1) \\
& \qquad\qquad\qquad\qquad \mathsf{else\ div}) \\
& \qquad\qquad\quad \mathsf{else\ div}
\end{aligned}
$$

$$M_2 \equiv\ \lambda f^{\mathsf{int}\to\mathsf{int}\to\mathsf{int}}.\ \ \mathsf{div}$$

- Because por is not definable, we have $M_1 \cong M_2$.
- $[\![M_1]\!](por) \neq [\![M_2]\!](por)$, so $[\![M_1]\!] \neq [\![M_2]\!]$.

INTRINSIC QUOTIENT

In the presence of definability (as well as correctness and adequacy) one can construct fully abstract models by quotienting.

This boils down to recasting the idea of contextual testing inside the model.

Given $x_1, x_2 \in [\![\theta]\!]$,

$$x_1 \sim x_2 \iff \text{“}\forall_{y \in [\![\theta \to \mathsf{int}]\!]} y(x_1) = y(x_2)\text{”}.$$

Then $[\![\cdots]\!]/\sim$ is fully abstract.

This kind of quotienting may be an obstacle in reasoning about equivalence, so one should attempt to find more direct characterizations.

ELSEVIER

Theoretical Computer Science 266 (2001) 341–364

Theoretical
Computer Science

www.elsevier.com/locate/tcs

Finitary PCF is not decidable

Ralph Loader

Merton College, Oxford, UK

Received October 1996; revised March 2000; accepted April 2000

Communicated by G.D. Plotkin

Abstract

The question of the decidability of the observational ordering of finitary PCF was raised (Jung and Stoughton, in: M. Bezem, J.F. Groote (Eds.), Typed Lambda Calculi and Applications, Lecture Notes in Computer Science, vol. 664, Springer, Berlin, 1993, pp. 230–244) to give mathematical content to the full abstraction problem for PCF (Milner, Theoret. Comput. Sci. 4 (1977) 1–22). We show that the ordering is in fact undecidable. This result places limits on how explicit a representation of the fully abstract model can be. It also gives a slight strengthening of the author's earlier result on typed λ-definability (Loader, in: A. Anderson, M. Zeleny (Eds.), Church Memorial Volume, Kluwer Academic Press, Dordrecht, to appear).

The 2017 Alonzo Church Award

SIGLOG is delighted to announce that the 2017 Church Award goes to 6 people: Samson Abramsky, Martin Hyland, Radha Jagadeesan, Pasquale Malacaria, Hanno Nickau and Luke Ong for [Quoting from the official citation] "providing a fully-abstract semantics for higher-order computation through the introduction of games models, thereby fundamentally revolutionising the field of programming language semantics, and for the applied impact of these models."

FULL ABSTRACTION FOR PCF

- Abramsky, Jagadeesan, Malacaria
- Hyland, Ong
- Nickau

Follow-up work extended the techniques to state, control, concurrency, exceptions and more.

GAME SEMANTICS

- ★ Two players: **O** (System) and **P** (Program)
- ★ Types are interpreted by games.
- ★ Programs are interpreted as strategies for P.
- ★ No winners or losers.
- ★ The dialogue is the central object of study.

REFML

$$\theta, \theta' \;::=\; \mathsf{unit} \;\mid\; \mathsf{int} \;\mid\; \mathsf{ref}\,\theta \;\mid\; \theta \to \theta'$$

TYPING RULES I

$$\mathbb{A} = \biguplus_\theta \mathbb{A}_\theta$$

$$\frac{}{u, \Gamma \vdash () : \mathsf{unit}} \qquad \frac{i \in \mathbb{Z}}{u, \Gamma \vdash i : \mathsf{int}} \qquad \frac{a \in (u \cap \mathbb{A}_\theta)}{u, \Gamma \vdash a : \mathsf{ref}\,\theta}$$

$$\frac{(x : \theta) \in \Gamma}{u, \Gamma \vdash x : \theta} \qquad \frac{u, \Gamma \vdash M_1 : \mathsf{int} \qquad u, \Gamma \vdash M_2 : \mathsf{int}}{u, \Gamma \vdash M_1 \oplus M_2 : \mathsf{int}}$$

$$\frac{u, \Gamma \vdash M : \mathsf{int} \quad u, \Gamma \vdash N_0 : \theta \quad u, \Gamma \vdash N_1 : \theta}{u, \Gamma \vdash \mathsf{if}\ M\ \mathsf{then}\ N_1\ \mathsf{else}\ N_0 : \theta}$$

TYPING RULES II

$$\frac{u, \Gamma \vdash M : \mathsf{ref}\,\theta}{u, \Gamma \vdash\, !M : \theta} \qquad \frac{u, \Gamma \vdash M : \mathsf{ref}\,\theta \quad u, \Gamma \vdash N : \theta}{u, \Gamma \vdash M := N : \mathsf{unit}}$$

$$\frac{u, \Gamma \vdash M : \theta}{u, \Gamma \vdash \mathsf{ref}_\theta(M) : \mathsf{ref}\,\theta} \qquad \frac{u, \Gamma \vdash M : \mathsf{ref}\,\theta \quad u, \Gamma \vdash N : \mathsf{ref}\,\theta}{u, \Gamma \vdash M = N : \mathsf{int}}$$

$$\frac{u, \Gamma \vdash M : \theta \to \theta' \quad u, \Gamma \vdash N : \theta}{u, \Gamma \vdash MN : \theta'} \qquad \frac{u, \Gamma \cup \{x : \theta\} \vdash M : \theta'}{u, \Gamma \vdash \lambda x^\theta.M : \theta \to \theta'}$$

- A *store* is a function from a finite set of names to values such that the type of each name matches the type of its assigned value.

- We write $S[a \mapsto V]$ for the store obtained by updating S so that a is mapped to V (this may extend the domain of S).

- Given a store S and a term M we say that the pair (S, M) is *compatible* if all names occurring in M are from the domain of S.

- The small-step reduction rules are given as judgments of the shape $(S, M) \to (S', M')$, where (S, M), (S', M') are compatible and $\mathsf{dom}(S) \subseteq \mathsf{dom}(S')$.

VALUES AND EVALUATION CONTEXTS

$$E ::= (\lambda x.N)\,\_ \mid \_\,N \mid \_ \oplus N \mid i \oplus \_ \mid \_ = N \mid a = \_$$
$$\mid\ !\_ \mid \_ := N \mid a := \_ \mid \mathsf{ref}_\theta(\_) \mid \mathsf{if}\ \_\ \mathsf{then}\ N_1\ \mathsf{else}\ N_0.$$

$$\frac{(S, M) \to (S', M')}{(S, E[M]) \to (S', E[M'])}$$

$$V ::= () \mid i \mid a \mid x \mid \lambda x^\theta.M$$

OPERATIONAL SEMANTICS

$$(S, \mathsf{if}\ 0\ \mathsf{then}\ N_1\ \mathsf{else}\ N_0) \to (S, N_0)$$

$$(S, \mathsf{if}\ i\ \mathsf{then}\ N_1\ \mathsf{else}\ N_0) \to (S, N_1) \qquad i \neq 0$$

$$(S, a = b) \to (S, 0) \qquad a \neq b$$

$$(S, a = a) \to (S, 1)$$

OPERATIONAL SEMANTICS

$$(S, (\lambda x.M)V) \to (S, M[V/x])$$

$$(S, a := V) \to (S[a \mapsto V], ())$$

$$(S, !a) \to (S, S(a))$$

$$(S, \mathsf{ref}_\theta(V)) \to (S[a' \mapsto V], a') \qquad a' \notin \mathsf{dom}(S)$$

EVALUATION

We say that (S, M) *evaluates* to (S', V) if $(S, M) \twoheadrightarrow (S', V)$, with V a value. For $\vdash M$: unit we say that M *converges*, written $M \Downarrow$, if $(\emptyset, M)$ evaluates to some $(S', ())$.

CONTEXTUAL TESTING

$$C[M] \Downarrow ?$$

We say that $\Gamma \vdash M_1 : \theta$ ***approximates*** $\Gamma \vdash M_2 : \theta$ (written $\Gamma \vdash M_1 \sqsubsetsim M_2$) if

$$C[M_1] \Downarrow \text{ implies } C[M_2] \Downarrow$$

for any context $C[-]$ such that $\vdash C[M_1], C[M_2] : \mathsf{unit}$.

Two terms-in-context are ***equivalent*** if one approximates the other (written $\Gamma \vdash M_1 \cong M_2$).

FULL ABSTRACTION

$$[\![M_1]\!] = [\![M_2]\!] \quad \text{if and only if} \quad M_1 \cong M_2$$

SHORTHANDS

- $\mathsf{let}\, x = M \,\mathsf{in}\, N$ stands for

$$(\lambda x^{\theta}.N)M$$

- $M; N$ stands for

$$\mathsf{let}\, x = M \,\mathsf{in}\, N$$

where x does not occur in N.

EQUIVALENCE?

$$\mathsf{gen} \equiv \lambda z^{\mathsf{int}}.\, \mathsf{let}\, x = \mathsf{ref}(0)\, \mathsf{in}\, (x := z; x) \quad : \quad \mathsf{int} \to \mathsf{ref\ int}$$

$$\mathsf{gen}' \equiv \mathsf{let}\, x = \mathsf{ref}(0)\, \mathsf{in}\, \lambda z^{\mathsf{int}}.(x := z; x) \quad : \quad \mathsf{int} \to \mathsf{ref\ int}$$

$$C \;\equiv\; (\lambda f^{\mathsf{int} \to \mathsf{ref\ int}}.\, \mathsf{if}\, (f 0 = f 0)\, \mathsf{then}\, ()\, \mathsf{else\ div})\, [\,]$$

EQUIVALENCE?

$$M_1 \equiv \mathsf{let}\, x = \mathsf{ref}(0)\,\mathsf{in}\,\lambda y^{\mathsf{ref\,int}}.\, x = y \;:\; \mathsf{ref\,int} \to \mathsf{int},$$
$$M_2 \equiv \lambda y^{\mathsf{ref\,int}}.\, 0 \;:\; \mathsf{ref\,int} \to \mathsf{int}.$$

COMPOSITIONAL INTERPRETATION

- Types interpreted by games between O and P.
- Terms interpreted by strategies for P.
- Each syntactic construct interpreted through special strategies, constructions on strategies and composition.
- We start with a few concrete examples.

$$\vdash 2019 : \mathsf{int}$$

O What is the result?

P 2019.

$$\star \quad 2019$$

$$O \quad P$$

$$\begin{array}{lll} & \vdash & \mathsf{int}_0 \\ O & \star & \\ P & & 2019_0 \end{array}$$

$$\vdash \lambda x^{\mathsf{int}}.x + 1 : \mathsf{int} \to \mathsf{int}$$

O	What is the result?
P	A function.
O	What is the result if the argument is 3?
P	4.
O	What is the result if the argument is 1?
P	2.

$$\begin{array}{cccccc} \star & \dagger & 3_1 & 4_0 & 1_1 & 2_0 \\ O & P & O & P & O & P \end{array}$$

$$\vdash \lambda x^{\mathsf{int}}.x + 1 : \mathsf{int} \to \mathsf{int}$$

$$\begin{array}{cccccc}
\star & \dagger & 3_1 & 4_0 & 1_1 & 2_0 \\
O & P & O & P & O & P
\end{array}$$

$$\begin{array}{ccccc}
 & \vdash & \mathsf{int}_1 & \to & \mathsf{int}_0 \\
O & \star & & & \\
P & & & \dagger & \\
O & & 3_1 & & \\
P & & & & 4_0 \\
O & & 1_1 & & \\
P & & & & 2_0
\end{array}$$

$$\vdash \lambda x^{\mathsf{int}}.\lambda y^{\mathsf{int}}.x + y + 1 : \mathsf{int} \to \mathsf{int} \to \mathsf{int}$$

$$\begin{array}{cccc} \star & \dagger & 3_2 & \dagger' \\ O & P & O & P \end{array}$$

$$\begin{array}{lccccc} & \vdash & \mathsf{int}_2 & \to & \mathsf{int}_1 & \to & \mathsf{int}_0 \\ O & \star & & & & & \\ P & & & & \dagger & & \\ O & & 3_2 & & & & \\ P & & & & & & \dagger' \end{array}$$

$$\vdash \lambda x^{\mathsf{int}}.\lambda y^{\mathsf{int}}.x + y + 1 : \mathsf{int} \to \mathsf{int} \to \mathsf{int}$$

$$\begin{array}{cccccc} \star & \dagger & 3_2 & \dagger' & 5_1 & 9_0 \\ O & P & O & P & O & P \end{array}$$

$$\begin{array}{ccccccc}
 & \vdash & \mathsf{int}_2 & \to & \mathsf{int}_1 & \to & \mathsf{int}_0 \\
O & \star & & & & & \\
P & & & \dagger & & & \\
O & & 3_2 & & & & \\
P & & & & & \dagger' & \\
O & & & & 5_1 & & \\
P & & & & & & 9_0
\end{array}$$

$$\vdash \lambda x^{\mathsf{int}}.\lambda y^{\mathsf{int}}.x + y + 1 : \mathsf{int} \to \mathsf{int} \to \mathsf{int}$$

$$\begin{array}{cccccccccc}
\star & \dagger & 3_2 & \dagger' & 4_2 & \dagger' & 5_1 & 9_0 & 5_1 & 10_0 \\
O & P & O & P & O & P & O & P & O & P
\end{array}$$

$$\vdash \lambda f^{\mathsf{int}\to\mathsf{int}}.f(0)+1 : (\mathsf{int} \to \mathsf{int}) \to \mathsf{int}$$

$$\begin{array}{cccc} \star & \dagger & \dagger' & 0_2 \\ O & P & O & P \end{array}$$

	$\vdash$	$(\mathsf{int}_2$	$\to$	$\mathsf{int}_1)$	$\to$	int_0
O	$\star$					
P						$\dagger$
O				$\dagger'$		
P		0_2				

$$\vdash \lambda f^{\mathsf{int}\to\mathsf{int}}.f(0)+1 : (\mathsf{int}\to\mathsf{int})\to\mathsf{int}$$

$$\begin{array}{cccccc} \star & \dagger & \dagger' & 0_2 & 3_1 & 4_0 \\ O & P & O & P & O & P \end{array}$$

		$\vdash$	$(\mathsf{int}_2$	$\to$	$\mathsf{int}_1)$	$\to$	int_0
O	$\star$						
P						$\dagger$	
O				$\dagger'$			
P			0_2				
O					3_1		
P							4_0

$$\mathsf{let}\ g = [\ ]\ \mathsf{in}\ g(\lambda x^{\mathsf{int}}.x+3)$$

$$\vdash \lambda f^{\mathsf{int}\to\mathsf{int}}.f(0)+1 : (\mathsf{int}\to\mathsf{int})\to\mathsf{int}$$

$$\begin{array}{cccccccccc} \star & \dagger & \dagger' & 0_2 & \dagger' & 0_2 & 3_1 & 4_0 & 8_1 & 9_0 \\ O & P & O & P & O & P & O & P & O & P \end{array}$$

$$\mathsf{let}\; g = [\;]\;\mathsf{in}\; g(\lambda x^{\mathsf{int}}.g(\lambda y^{\mathsf{int}}.x+y+3)+4)$$