Imperative Programs as Proofs (via Game Semantics)

Martin Churchill, Jim Laird, Guy McCusker
University of Bath

University of Birmingham, 8th July 2011

Motivation 1: Curry-Howard Correspondence

The Curry-Howard isomorphism notes a striking correspondence between *proofs* and *functional programs*:

Types	Propositions
Programs	Proofs
Evaluation	Proof normalisation

- We can extend our notion of programs to include those with imperative effects...
- What are the corresponding proofs?

Motivation 2: A Simple Games Model

- Modelling programs/proofs as *strategies* is a compelling metaphor and has yielded strong technical results.
- $\Rightarrow$ the games themselves are important mathematical entities.
- Curien-Lamarche sequential games are a strikingly simple formulation
  - Rich mathematical structure, can model many languages and logics
- Can we find a logic where each strategy interprets a proof?

Overview

- We will develop a logic WS1 where formulas correspond to games and proofs to history-sensitive strategies
  - Proofs with imperative computational content
- The system is expressive:
  - This logic contains **first-order intuitionistic linear logic**
  - We can embed a **total imperative programming language**
  - $\Rightarrow$ We can use it to reason about imperative programs
- This logic admits a strong *full completeness* result with respect to the game model

Formulas of WS1

- Fix a first-order language $\mathcal{L}$ with pairs of predicates $(\phi, \overline{\phi})$ and a variable set $\mathcal{V}$ $(=\in \phi)$
- For formulas of the logic are as follows:

$$
\begin{array}{llllllll}
M, N := & \mathbf{1} & | & \bot & | & \phi(\overrightarrow{x}) & | \\
& M \otimes N & | & M \oslash N & | & N \lhd P & | \\
& \forall x.P & | & M \& N & | & !N & \\
P, Q := & \mathbf{0} & | & \top & | & \overline{\phi}(\overrightarrow{x}) & | \\
& P \parr Q & | & P \lhd Q & | & P \oslash N & | \\
& \exists x.P & | & P \oplus Q & | & ?P &
\end{array}
$$

- We have an involutive $(-)^{\perp}$ operation switching polarity
- We can encode implication $M \multimap N = N \lhd M^{\perp}$

Formulas as Games

- Formulas denote (families of) two-player games
  - (indexed over $\mathcal{L}$-structures and valuations)
  - **Opponent** and **Player** alternately play moves according to a tree of valid plays
  - In negative formulas Opponent starts, in positive formulas Player starts
- Proofs of a formula denote (families of) winning P-strategies on the interpretation of that formula.
  - Player must always respond to an Opponent-move
  - There is a winning condition for infinite plays

Units and Atoms

$$
\begin{array}{rcccccccl}
M, N := & \mathbf{1} & \mid & \bot & \mid & \phi(\overrightarrow{x}) & \mid & \ldots \\
P, Q := & \mathbf{0} & \mid & \top & \mid & \overline{\phi}(\overrightarrow{x}) & \mid & \ldots
\end{array}
$$

- $\mathbf{1}$ represents the empty negative game (no moves) ($\vdash \mathbf{1}$)
- $\bot$ represents the game with a single Opponent move ($\not\vdash \bot$)
- $\phi(\overrightarrow{x})$ is interpreted as $\mathbf{1}$ if the model validates $\phi(\overrightarrow{x})$, $\bot$ if it does not

Units and Atoms

$$
\begin{array}{rcccccccl}
M, N := & \mathbf{1} & \mid & \bot & \mid & \phi(\overrightarrow{x}) & \mid \ldots \\
P, Q := & \mathbf{0} & \mid & \top & \mid & \overline{\phi}(\overrightarrow{x}) & \mid \ldots
\end{array}
$$

- $\mathbf{0}$ represents the empty positive game (no moves) ($\nvdash \mathbf{0}$)
- $\top$ represents the game with a single Player move ($\vdash \top$)
- $\overline{\phi}(\overrightarrow{x})$ is interpreted as $\mathbf{0}$ if the model validates $\phi(\overrightarrow{x})$, $\top$ if it does not

Additives and Quantifiers

$$\begin{array}{rcccccc} M,\, N := & M \& N & \mid & \forall x.P & \mid & \ldots \\ P,\, Q := & P \oplus Q & \mid & \exists x.P & \mid & \ldots \end{array}$$

- In $M \& N$, Opponent may chose to start a play in M or in N
  - So a strategy $\vdash M \& N$ is a pair $(\vdash M, \vdash N)$
- In $\forall x.M(x)$, Opponent may chose a value v for x in the model and start a play in $M(v)$

Additives and Quantifiers

$$\begin{array}{rcccccc} M, N := & M \& N & \mid & \forall x.P & \mid & \ldots \\ P, Q := & P \oplus Q & \mid & \exists x.P & \mid & \ldots \end{array}$$

- In $P \oplus Q$, Player may chose to start a play in P or in Q
  - So a strategy $\vdash P \oplus Q$ is a strategy $\vdash P$ or a strategy $\vdash Q$
- In $\exists x.P(x)$, Player may chose a value v for x in the model and start a play in $P(v)$

Multiplicatives

$$\begin{array}{rcccc} M, N := & M \otimes N & \mid \; M \oslash N & \mid \; N \lhd P & \mid \; \ldots \\ P, Q := & P ⅋ Q & \mid \; P \lhd Q & \mid \; P \oslash N & \mid \; \ldots \end{array}$$

- A play in $M \otimes N$ is an **interleaving** of a play in M with a play in N
  - Opponent may start in either component, and then switch between components
- A play in $M \oslash N$ is a play in $M \otimes N$ that must start in M
  - So we have $M \otimes N \cong M \oslash N \& N \oslash M$
- In the game $M \lhd P$, it is Player that can switch between the two components.

Multiplicatives

$$
\begin{array}{rcccccccc}
M, N := & M \otimes N & | & M \oslash N & | & N \lhd P & | & \ldots \\
P, Q := & P \parr Q & | & P \lhd Q & | & P \oslash N & | & \ldots
\end{array}
$$

- A play in $P \parr Q$ is an **interleaving** of a play in P with a play in Q
  - Player may start in either component, and then switch between components
- A play in $P \lhd Q$ is a play in $P \parr Q$ that must start in P
  - So we have $P \parr Q \cong P \lhd Q \oplus Q \lhd P$
- In the game $P \oslash M$, it is Opponent that can switch between the two components.

Exponentials

$$\begin{array}{rcl} M, N := & !M & \ldots \\ P, Q := & ?P & \ldots \end{array}$$

- $!M$ denotes an (ordered) interleaving of infinitely many copies of M
  - Opponent may spawn new copies of M and switch between copies he has opened ($!M \cong M \oslash !M$)
- $?P$ denotes an (ordered) interleaving of infinitely many copies of P
  - Player may spawn new copies of P and switch between copies he has opened ($?P \cong P \lhd ?P$)
- These are the only operators yielding infinite games

Example

- The game of Booleans can be given by $\mathbf{B} = \bot \lhd (\top \oplus \top)$
  - A play consists of an O-move (q) followed by one of two P-moves (t or f)
  - Two winning strategies correspond to True or False values
- We can represent 'functions' Bool $\rightarrow$ Bool by $\mathbf{B} \multimap \mathbf{B} = \mathbf{B} \lhd \mathbf{B}^{\bot} = (\bot \lhd (\top \oplus \top)) \lhd (\top \oslash (\bot \& \bot))$

Example

- The game of Booleans can be given by $\mathbf{B} = \bot \lhd (\top \oplus \top)$
  - A play consists of an O-move (q) followed by one of two P-moves (t or f)
  - Two winning strategies correspond to True or False values
- We can represent 'functions' Bool $\rightarrow$ Bool by $\mathbf{B} \multimap \mathbf{B} = \mathbf{B} \lhd \mathbf{B}^{\perp} = (\underline{\bot} \lhd (\top \oplus \top)) \lhd (\top \oslash (\bot \& \bot))$

Opponent asks for output

Example

- The game of Booleans can be given by $\mathbf{B} = \bot \lhd (\top \oplus \top)$
  - A play consists of an O-move (q) followed by one of two P-moves (t or f)
  - Two winning strategies correspond to True or False values
- We can represent 'functions' Bool $\rightarrow$ Bool by $\mathbf{B} \multimap \mathbf{B} = \mathbf{B} \lhd \mathbf{B}^{\perp} = (\bot \lhd (\underline{\top} \oplus \underline{\top})) \lhd (\top \oslash (\bot \& \bot))$

Player gives output

Example

- The game of Booleans can be given by $\mathbf{B} = \bot \lhd (\top \oplus \top)$
  - A play consists of an O-move (q) followed by one of two P-moves (t or f)
  - Two winning strategies correspond to True or False values
- We can represent 'functions' Bool $\rightarrow$ Bool by $\mathbf{B} \multimap \mathbf{B} = \mathbf{B} \lhd \mathbf{B}^{\perp} = (\bot \lhd (\top \oplus \top)) \lhd (\underline{\top} \oslash (\bot \& \bot))$

or *Player asks for input*

Example

- The game of Booleans can be given by $\mathbf{B} = \bot \lhd (\top \oplus \top)$
  - A play consists of an O-move (q) followed by one of two P-moves (t or f)
  - Two winning strategies correspond to True or False values
- We can represent 'functions' Bool $\to$ Bool by $\mathbf{B} \multimap \mathbf{B} = \mathbf{B} \lhd \mathbf{B}^{\bot} = (\bot \lhd (\top \oplus \top)) \lhd (\top \oslash (\underline{\bot} \& \underline{\bot}))$

Opponent gives input

Example

- The game of Booleans can be given by $\mathbf{B} = \bot \lhd (\top \oplus \top)$
  - A play consists of an O-move (q) followed by one of two P-moves (t or f)
  - Two winning strategies correspond to True or False values
- We can represent 'functions' Bool $\rightarrow$ Bool by $\mathbf{B} \multimap \mathbf{B} = \mathbf{B} \lhd \mathbf{B}^{\perp} = (\bot \lhd (\underline{\top} \oplus \underline{\top})) \lhd (\top \oslash (\bot \& \bot))$

Player gives output

Sequents

A *sequent* of WS1 is $\Phi \vdash \Gamma$ where:

- $\Phi = X; \Theta$ where X is variables in scope, Θ is atomic assumptions on those variables.
- Γ is a nonempty list of formulas, of either polarity.

$$\Phi \vdash M, P, Q, N$$

Comma is to be read as a left-associative $\oslash$ or $\lhd$:

$$\Phi \vdash ((M \lhd P) \lhd Q) \oslash N$$

$\Rightarrow$ First move must occur in first formula.

Core Rules

$$\frac{}{\Phi \vdash \mathbf{1}, \Gamma} \qquad \frac{}{\Phi \vdash \top}$$

$$\frac{\Phi \vdash M, N, \Gamma \qquad \Phi \vdash N, M, \Gamma}{\Phi \vdash M \otimes N, \Gamma} \qquad \frac{\Phi \vdash Q, P, \Gamma}{\Phi \vdash P \invamp Q, \Gamma} \qquad \frac{\Phi \vdash P, Q, \Gamma}{\Phi \vdash P \invamp Q, \Gamma}$$

$$\frac{\Phi \vdash M, \Gamma \qquad \Phi \vdash N, \Gamma}{\Phi \vdash M \& N, \Gamma} \qquad \frac{\Phi \vdash P, \Gamma}{\Phi \vdash P \oplus Q, \Gamma} \qquad \frac{\Phi \vdash Q, \Gamma}{\Phi \vdash P \oplus Q, \Gamma}$$

$$\frac{\Phi \vdash P}{\Phi \vdash \bot, P} \qquad \frac{\Phi \vdash \bot, P \invamp Q, \Gamma}{\Phi \vdash \bot, P, Q, \Gamma} \qquad \frac{\Phi \vdash \bot, P \oslash N, \Gamma}{\Phi \vdash \bot, P, N, \Gamma}$$

$$\frac{\Phi \vdash \bot, \Gamma}{\Phi \vdash \bot, N, \Gamma} \qquad \frac{\Phi \vdash N}{\Phi \vdash \top, N} \qquad \frac{\Phi \vdash \top, M \otimes N, \Gamma}{\Phi \vdash \top, M, N, \Gamma}$$

$$\frac{\Phi \vdash \top, N \lhd P, \Gamma}{\Phi \vdash \top, N, P, \Gamma} \qquad \frac{\Phi \vdash \top, \Gamma}{\Phi \vdash \top, P, \Gamma} \qquad \frac{\Phi \vdash A, N, \Gamma}{\Phi \vdash A \oslash N, \Gamma}$$

$$\frac{\Phi \vdash A, P, \Gamma}{\Phi \vdash A \lhd P, \Gamma} \qquad \frac{\Phi \vdash N, !N, \Gamma}{\Phi \vdash !N, \Gamma} \qquad \frac{\Phi \vdash P, ?P, \Gamma}{\Phi \vdash ?P, \Gamma}$$

Core Rules (atoms and quantifiers)

A proof of $X; \Theta \vdash \Gamma$ is interpreted as a strategy on $[\![\Gamma]\!](L)$ for each Θ-satisfying $\mathcal{L}$-model over X

$$\frac{\Theta, \overline{\phi}(\overrightarrow{x}) \vdash \bot, \Gamma}{\Theta \vdash \phi(\overrightarrow{x}), \Gamma} \qquad \frac{\Theta, \overline{\phi}(\overrightarrow{x}) \vdash \top, \Gamma}{\Theta, \overline{\phi}(\overrightarrow{x}) \vdash \overline{\phi}(\overrightarrow{x}), \Gamma}$$

$$\frac{X \uplus \{x\}; \Theta \vdash N, \Gamma}{X; \Theta \vdash \forall x.N, \Gamma}\ x \notin FV(\Theta, \Gamma) \qquad \frac{X \uplus \{y\}; \Theta \vdash P[y/x], \Gamma}{X \uplus \{y\}; \Theta \vdash \exists x.P, \Gamma}$$

$$\frac{(X; \Theta \vdash \Gamma)[\frac{z}{x}, \frac{z}{y}] \qquad X; \Theta, x \neq y \vdash \Gamma}{X; \Theta \vdash \Gamma} \qquad \frac{}{\Theta, x \neq x \vdash \Gamma}$$

Other Rules

$$\frac{\vdash \Gamma^*, \Delta}{\vdash \Gamma^*, \mathbf{1}, \Delta} \qquad \frac{\vdash \Gamma^*, M, N, \Delta}{\vdash \Gamma^*, M \otimes N, \Delta} \qquad \frac{\vdash \Gamma^*, M, N, \Delta}{\vdash \Gamma^*, N, M, \Delta} \qquad \frac{\vdash \Gamma^*, M, \Delta}{\vdash \Gamma^*, \Delta}$$

$$\frac{\vdash \Gamma^*, \Delta}{\vdash \Gamma^*, \mathbf{0}, \Delta} \qquad \frac{\vdash \Gamma^*, P, Q, \Delta}{\vdash \Gamma^*, P \parr Q, \Delta} \qquad \frac{\vdash \Gamma^*, P, Q, \Delta}{\vdash \Gamma^*, Q, P, \Delta} \qquad \frac{\vdash \Gamma^*, \Delta}{\vdash \Gamma^*, P, \Delta}$$

$$\frac{\vdash M, \Gamma, \Delta^+ \qquad \vdash N, \Delta_1^+}{\vdash M, \Gamma, N, \Delta^+, \Delta_1^+} \qquad \frac{}{\vdash N, N^\perp} \qquad \frac{\vdash \Gamma, !M, \Delta}{\vdash \Gamma, M, \Delta}$$

$$\frac{\vdash \Gamma^*, N^\perp, \Gamma_1 \qquad \vdash N, \Delta^+}{\vdash \Gamma^*, \Delta^+, \Gamma_1} \qquad \frac{\vdash \Gamma, !M, \Delta}{\vdash \Gamma, !M, !M, \Delta} \qquad \frac{\vdash M, P^\perp, P}{\vdash !M, P}$$

Interpretation of Proofs

- We can interpret proofs as (families of) strategies using the ideas described above
- Semantics of the ‘other’ rules use the *categorical structure* of the games model:
  - One may *compose* strategies $M \multimap N$ and $N \multimap L$, take the tensor of maps $M \otimes N \multimap M' \otimes N'$ and so on
- We distinguish them from the ‘core’ rules due to full completeness result...

Full Completeness

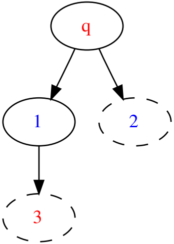
- We can show that any uniform family of winning finitary strategies is the denotation of a unique analytic proof
- We define a **semantics-guided proof search procedure**:
  - Choice of rule to prove $\vdash A, \Gamma$ determined by A in most cases
  - There is a choice in $\otimes$,$\oplus$,$\exists$ cases; determined by move played by the strategy
    - But what if there is a different choice in different components?

Uniformity of Strategies

- The interpretations of proofs are *uniform* families of strategies.
  - If $(L, v) \models \overline{\phi}(\overrightarrow{x})$ whenever $(L', v') \models \overline{\phi}(\overrightarrow{x})$ then $[\![\Gamma]\!](L', v')$ is a subgame of $[\![\Gamma]\!](L, v)$
  - Uniformity means that the strategy on (L', v') is the restriction of the strategy on (L, v)

Non-example

Consider $\bot \lhd (\overline{\phi} \oplus (\top \oslash \phi))$ ("excluded middle")



There is a family of winning strategies, but it is not uniform.

Categorical Formalisation

Uniformity is formalised using tools from category theory...

- A sequent $X; \Theta \vdash \Gamma$ is interpreted as a functor $\mathcal{M}_X^\Theta \to \mathcal{G}$
  - $\mathcal{M}_X^\Theta$ is the category where objects are Θ-satisfying $\mathcal{L}$-structures and X-valuations, and morphisms are functions that preserve positive predicates and valuation ($\Rightarrow$ injective)
  - $\mathcal{G}$ is the category of games and strategies
- A proof of $X; \Theta \vdash \Gamma$ is interpreted as a uniform winning strategy on $[\![X; \Theta \vdash \Gamma]\!]$
  - A *lax-natural transformation* $I \Rightarrow [\![X; \Theta \vdash \Gamma]\!]$ that is *pointwise winning*

Uniformity Results

Proposition

Provided Θ is "lean" (contains $x \neq y$ for all distinct $x, y \in X$)

- *A uniform winning strategy on $P \oplus Q$ is a uniform winning strategy on P or a uniform strategy on Q*
- *A uniform winning strategy on $P \otimes Q$ is a uniform winning strategy on $P \lhd Q$ or a uniform winning strategy on $Q \lhd P$*
- *A uniform winning strategy on $\exists x.P(x)$ corresponds to a choice of a* **unique variable** *y (in scope) and uniform winning strategy on $P(y)$.*

Reification of Strategies

We can hence define our proof search procedure for bounded strategies:

- Apply match rule to ensure Θ is lean
- Decompose the head formula using core introduction rules until it is a unit
  - Choices for $\otimes$,$\oplus$,$\exists$ determined by strategy
- Consolidate the tail into a single formula using the elimination rules
- Strictly decrease the size of the strategy using the rules that remove the head unit

Some Core Rules (reminder)

$$\frac{(\Phi \vdash \Gamma)[\frac{z}{x}, \frac{z}{y}] \qquad \Phi, x \neq y \vdash \Gamma}{\Phi \vdash \Gamma} \qquad \frac{}{\Phi \vdash \mathbf{1}, \Gamma} \qquad \frac{}{\Phi \vdash \top}$$

$$\frac{\Phi \vdash M, N, \Gamma \qquad \Phi \vdash N, M, \Gamma}{\Phi \vdash M \otimes N, \Gamma} \qquad \frac{\Phi \vdash Q, P, \Gamma}{\Phi \vdash P \parr Q, \Gamma} \qquad \frac{\Phi \vdash P, Q, \Gamma}{\Phi \vdash P \parr Q, \Gamma}$$

$$\frac{\Phi \vdash M, \Gamma \qquad \Phi \vdash N, \Gamma}{\Phi \vdash M \& N, \Gamma} \qquad \frac{\Phi \vdash P, \Gamma}{\Phi \vdash P \oplus Q, \Gamma} \qquad \frac{\Phi \vdash Q, \Gamma}{\Phi \vdash P \oplus Q, \Gamma}$$

$$\frac{\Phi \vdash P}{\Phi \vdash \bot, P} \qquad \frac{\Phi \vdash \bot, P \parr Q, \Gamma}{\Phi \vdash \bot, P, Q, \Gamma} \qquad \frac{\Phi \vdash \bot, P \oslash N, \Gamma}{\Phi \vdash \bot, P, N, \Gamma}$$

$$\frac{\Phi \vdash \bot, \Gamma}{\Phi \vdash \bot, N, \Gamma} \qquad \frac{\Phi \vdash N}{\Phi \vdash \top, N} \qquad \frac{\Phi \vdash \top, M \otimes N, \Gamma}{\Phi \vdash \top, M, N, \Gamma}$$

$$\frac{\Phi \vdash \top, N \lhd P, \Gamma}{\Phi \vdash \top, N, P, \Gamma} \qquad \frac{\Phi \vdash \top, \Gamma}{\Phi \vdash \top, P, \Gamma} \qquad \frac{\Phi \vdash A, N, \Gamma}{\Phi \vdash A \oslash N, \Gamma}$$

$$\frac{\Phi \vdash A, P, \Gamma}{\Phi \vdash A \lhd P, \Gamma} \qquad \frac{\Phi \vdash N, !N, \Gamma}{\Phi \vdash !N, \Gamma} \qquad \frac{\Phi \vdash P, ?P, \Gamma}{\Phi \vdash ?P, \Gamma}$$

Full Completeness

Thus, each finitary winning uniform strategy is the denotation of a unique analytic proof.

- In the exponential-free subsystem, the interpretation of any proof is finitary.
- We can ‘normalise’ proofs to analytic proofs via the semantics
  - (unique analytic proof with same semantics)
- $\Rightarrow$ all of the ‘other’ rules (e.g. cut) are admissible

This also works for the full system, if we allow normal forms to be *infinitary analytic proofs*.

Analytic Theorems

- In e.g. ILL, we can reduce any proof to an analytic (cut-free) finite proof, even in the presence of exponentials
- In WS1, the analytic proof may be infinite. Why the weaker situation?
  - In ILL proofs $\cong$ *innocent* strategies — a strategy on $!N$ must act the same way in each thread.
  - In WS1 proofs are history-sensitive — so ! really introduces infinite (possibly non-computable) behaviour
- But we can write proofs which denote infinite (computable, total) behaviour...

Non-core rules for Exponential

To generate a finite proof on a type involving the exponentials, we can use the following proof rule:

$$\frac{\vdash M, P^{\perp}, P}{\vdash !M, P}$$

This represents the fact that:

Proposition

In $\mathcal{G}$, $!M$ is the final coalgebra of the functor $M \oslash \_$.

Intuitionistic Linear Logic

- We can use this (with contraction) to derive promotion
  - $\Rightarrow$ **Embedding of Intuitionistic Linear Logic in WS1**
- There are formulas that are not provable in ILL but are provable in WS1 e.g. medial:

$$\vdash ((\alpha \otimes \beta \multimap \bot) \otimes (\gamma \otimes \delta \multimap \bot) \multimap \bot) \multimap$$
$$((\alpha \multimap \bot) \otimes (\gamma \multimap \bot) \multimap \bot) \otimes ((\beta \multimap \bot) \otimes (\delta \multimap \bot) \multimap \bot)$$

Boolean Variables

- Let $\mathbf{Bi} = (\bot \& \bot) \lhd \top$ (input Boolean)
- $!\mathbf{var} = !(\mathbf{B} \& \mathbf{Bi})$ is a type of reusable Boolean variables (read method and write method)
- We can define a reusable Boolean cell $\vdash \mathbf{B} \multimap !\mathbf{var}$ using the anamorphism rule and a proof $p \vdash \mathbf{var}, \mathbf{B}, \mathbf{B}^{\perp}$

Boolean Cell — p

$\mathbf{B}$	$\multimap$	$(\mathbf{B}$	$\&$	$\mathbf{Bi})$	$\oslash$	$\mathbf{B}$
		r				
r						
b						
		b				
						r
						b

				w_b		
				ok		
						r
						b

Boolean Cell — ana(p)

$\mathbf{B}$	$\multimap$	$(\mathbf{B}$	$\&$	$\mathbf{Bi})$	$\oslash$	$\mathbf{B}$	$\multimap$	$(\mathbf{B}$	$\&$	$\mathbf{Bi})$	$\oslash$	$((\mathbf{B}$	$\&$	$\mathbf{Bi})$	$\oslash$
										w_b					
				w_b											
				ok											
										ok					
												r			
						r									
						b									
												b			
												$\vdots$			

- We can extend this example to define a Boolean *Stack* in WS1 ($\mathbf{B} \cong \mathtt{pop}$, $\mathbf{Bi} \cong \mathtt{push}$. For the “state” we use $!\mathbf{B}$)

Algol-style Total Programming Language

We can embed a total programming language (TotLang).

- Simply typed lambda calculus
- Ground types: com, nat, var
- Constants: skip, sequencing, ifzero, repeat, 0, suc, assignment, deref, newvar, coroutines, encaps, mkvar

```
     add   =  λ m n .  newvar x := n in
              repeat m (x := succ !x) ; !x
newstack   =  encaps (λ g .  newvar x := 0 in g a b) 0
   where      a = λ n .  mkvar n (λ m .  x := suc m)
              b = λ n .  ifzero !x then n else
                 (let z = !x - 1 in x := 0 ; z)
```

Naturals in WS1

- To embed TotLang into WS1 we must add natural numbers to WS1...

$$N := \omega \mid \ldots \qquad P := \overline{\omega} \mid \ldots$$

- ω (resp. $\overline{\omega}$) denotes the game $\perp^{\omega}$ (resp. $\top^{\omega}$)
- Proof rules:

$$0\,\frac{}{\vdash \overline{\omega}} \quad \text{suc}\,\frac{}{\vdash \omega, \overline{\omega}} \quad \text{ind}\,\frac{\vdash P \qquad \vdash P^{\perp}, P}{\vdash \omega, P}$$

- Full completeness, normalisation etc extends to this setting

Language Embedding

- We can map types to negative formulas: com $\mapsto \bot \lhd \top$, nat $\mapsto \bot \lhd \overline{\omega}$, var $\mapsto$ **B&Bi**, $A \to B \mapsto B \lhd ?A^{\bot}, \ldots$
- The lambda calculus part uses the structure of the ILL embedding
- Constants can be mapped to proofs in WS1

The games model of TotLang is fully abstract, resultantly:

- Two programs M and N are observationally equivalent if and only if their representations in WS1 have the same (infinitary) normal form

(we can also embed a call-by-value language with these features)

Formulas as Specifications

Formulas of WS1 are much finer than the programming language types, we can use them to represent specifications on programs.

- Evalation order of arguments
- Number of times an argument is interrogated
- Predicates on ground values

Example:

- Identity specification on nat $\to$ nat given by
  $\perp \lhd \top \oslash \forall x.\perp \lhd \exists y.y = x$

Adding function symbols increases expressivity further.

Uniformity for Controlling Imperative Flow

We can use uniformity of the underlying strategies to give refinements on imperative beavhiour. E.g...

- Define $\mathbf{B}' = \perp \lhd (\overline{\alpha} \oplus \overline{\beta})$, $\mathbf{Bi}' = (\alpha \& \beta) \lhd \top$.
- If α and β are false, $\mathbf{B}' = \mathbf{B}$, $\mathbf{Bi}' = \mathbf{Bi}$
- ... in which case $\mathbf{worm} = \mathbf{Bi}' \oslash ! \mathbf{B}'$ represents the type of a "write-once-read-many" Boolean cell.
- But since any proof must be a uniform strategy on *all* models, any proof of **worm** must act as a well-behaved Boolean cell.

Data-independence

We can use the first-order structure in a different way to model a data-independent language:

- Interpretation of ground type depends on model ($\mathsf{val} = \bot \lhd \exists x.\top$)
- Cells of ground type, only operation is equality
- Example program: data-independent set

Further Directions

- Enhancing the logic to be able to specify more interesting properties of more interesting programs
- Introducing propositional variables
  - Ranging over arbitrary games — polymorphism
- Recursive types
  - a la Clairambault — e.g. $\mathrm{list}(\mathbf{B}) = \mu X.\bot \lhd (\top \oplus (\top \oslash (\mathbf{B} \otimes X)))$
- Universality results
  - $!\mathbf{N}$ is a universal type in the games model... are the embeddings/retractions definable in the logic?

Thank You

Any questions?