Game Semantics and Block-Structured State

Andrzej Murawski and Nikos Tzevelekos

Oxford University Computing Laboratory

GaLoP, York, March 2009
Evaluation strategies vs scoping.

- Call-by-value and mobility: RML [AM98], Reduced ML [MT09], etc.
- Call-by-name and blocks: Idealized Algol [AM97].
What this talk is about

Evaluation strategies vs scoping.

- Call-by-value and mobility: RML [AM98], Reduced ML [MT09], etc.

- Call-by-name and blocks: Idealized Algol [AM97].

Call-by-value and (base-type) blocks?

- There is a gap.

- Name-mobility has been be described as a semantical intricacy of ML-like languages (e.g. [PS98]).
Simply-typed \( \lambda \)-calculus with integers and references.

- **Types:**
  \[
  \theta ::= \text{unit} \mid \text{int} \mid \text{int ref} \mid \theta \rightarrow \theta
  \]

- **Terms:**
  \[
  M ::= x \mid \lambda x.M \mid MM \mid () \mid \Omega \\
  \mid n \mid M \odot M \mid \text{if } M \text{ then } M \text{ else } M \\
  \mid a \mid \text{ref } M \mid \text{! } M \mid M ::= M
  \]

\[
V ::= n \mid () \mid a \mid x \mid \lambda x.M
\]
Games for Reduced ML [MT09]

At FoSSaCS: Fully abstract game model for Reduced ML, where full abstraction is achieved with very special plays.
At FoSSaCS: Fully abstract game model for Reduced ML, where full abstraction is achieved with very special plays.

- Participants can only use names available to them:

\[ s = m_1^{S_1} \ldots m_n^{S_n} , \quad s = m_1 \ldots m_n \]

a name is available to \( X \) at \( s \) if it first occurs in an \( X \)-move or it is present in \( X \)-view of \( s \).
At FoSSaCS: Fully abstract game model for Reduced ML, where full abstraction is achieved with very special plays.

- Participants can only use names available to them:

\[ s = m_1^{S_1} \ldots m_n^{S_n}, \quad \underline{s} = m_1 \ldots m_n \]

a name is available to \( X \) at \( s \) if it first occurs in an \( X \)-move or it is present in \( X \)-view of \( \underline{s} \).

- Moves carry carefully selective stores: to be included in the store a name has to be available to both participants.
Games for Reduced ML [MT09]

At FoSSaCS: Fully abstract game model for Reduced ML, where full abstraction is achieved with very special plays.

- Participants can only use names available to them:
  \[ s = m_1^{S_1} \ldots m_n^{S_n}, \quad s = m_1 \ldots m_n \]
  
a name is available to \( X \) at \( s \) if it first occurs in an \( X \)-move or it is present in \( X \)-view of \( s \).

- Moves carry carefully selective stores: to be included in the store a name has to be available to both participants.

- Names adhere to freshness conditions: an \( X \)-name which is fresh in the view of the other participant must be fresh.

These plays give full abstraction: see paper for details!
RML = Reduced ML

- $a$
+ mkvar
What this talk is about
Reduced ML
Games for Reduced ML [MT09]

Example
RML with blocks: bRML
Observations
Binocence
A problem
More intensional

Terms:

\[ M ::= x \mid \lambda x. M \mid MM \mid () \mid \Omega \]
\[ \mid n \mid M \odot M \mid \text{if } M \text{ then } M \text{ else } M \]
\[ \mid \text{mkvar } MM \mid \text{ref } M \mid ! M \mid M := M \]

\[ V ::= n \mid () \mid \text{mkvar } MM \mid x \mid \lambda x. M \]
Terms:

\[ M :: = x \mid \lambda x. M \mid MM \mid () \mid \Omega \]
\[ \mid n \mid M \odot M \mid \text{if } M \text{ then } M \text{ else } M \]
\[ \mid \text{mkvar } M M \mid \text{ref } M \mid ! M \mid M := M \]

\[ V :: = n \mid () \mid \text{mkvar } M M \mid x \mid \lambda x. M \]

- No names
- \[ \text{new } x \text{ in } M \triangleq (\lambda x. M)(\text{ref0}) \]
Games for RML [AM98]

- Games for PCF$_v$ [HY99], without Innocence
  \[
  \llbracket \text{int ref} \rrbracket = (\llbracket \text{unit} \rrbracket \Rightarrow \llbracket \text{int} \rrbracket) \times (\llbracket \text{int} \rrbracket \Rightarrow \llbracket \text{unit} \rrbracket)
  \]
- \text{cell} : \llbracket \text{unit} \rrbracket \rightarrow \llbracket \text{int ref} \rrbracket
Games for RML [AM98]

- Games for PCF\textsubscript{v} [HY99], without Innocence
  \[ \text{int ref} = ([\text{unit}] \Rightarrow [\text{int}]) \times ([\text{int}] \Rightarrow [\text{unit}]) \]

- cell : [\text{unit}] \rightarrow [\text{int ref}]

- Go simpler: RML

Example RML with blocks: bRML

Observations
Binocence
A problem
More intensional
Taking $M \triangleq \lambda y. (x := !x+1; !x)$, $\llbracket \text{new } x \text{ in } M : \text{unit } \rightarrow \text{int} \rrbracket =$

$\llbracket \text{unit} \rrbracket \xrightarrow{\text{cell}} (\llbracket \text{unit} \rrbracket \Rightarrow \llbracket \text{int} \rrbracket) \times (\llbracket \text{int} \rrbracket \Rightarrow \llbracket \text{unit} \rrbracket)$ \xrightarrow{[M]} $\llbracket \text{unit} \rrbracket \Rightarrow \llbracket \text{int} \rrbracket$

$\bullet \text{OQ}$

$\text{rd}$

$0$

$\text{rd}$

$1$

$\text{w}(1)$

$\text{ok}$

$\ast \text{PA}$

$\ast \text{OQ}$

$1_{\text{PA}}$
Taking $M \triangleq \lambda y. (x := !x+1; !x)$, $\llbracket\text{new } x \text{ in } M : \text{unit } \to \text{int}\rrbracket =$

$\llbracket\text{unit}\rrbracket \xrightarrow{\text{cell}} (\llbracket\text{unit}\rrbracket \Rightarrow \llbracket\text{int}\rrbracket) \times (\llbracket\text{int}\rrbracket \Rightarrow \llbracket\text{unit}\rrbracket) \xrightarrow{[M]} \llbracket\text{unit}\rrbracket \Rightarrow \llbracket\text{int}\rrbracket$

$\bullet OQ$

$(\ast, \ast)$
Non-innocence in this case crucially depends on
new _ in _ : unit ⇒ int

What if we use blocks? Take $\beta = \{\text{unit, int}\}$ and:

$$bRML = RML - \text{ref} + \text{new } _ \text{in } _ : \text{int ref} \rightarrow \beta \rightarrow \beta$$
Non-innocence in this case crucially depends on new_in_: unit ⇒ int

What if we use blocks? Take \( \beta = \{\text{unit, int}\} \) and:

\[
bRML = RML - \text{ref} + \text{new_in_}: \text{int ref} \rightarrow \beta \rightarrow \beta
\]

Explicitly, bRML terms are:

\[
M ::= x | \lambda x.M | MM | () | \Omega \\
| n | M \odot M | \text{if } M \text{ then } M \text{ else } M \\
| \text{mkvar } MM | \text{new } x \text{ in } M | !M | M ::= M
\]
The model of RML models (soundly) bRML too.

bRML is less expressive than RML.
Observations

- The model of RML models (soundly) bRML too.
- bRML is less expressive than RML.
- All RML terms of type unit are expressible in bRML.
- RML is a conservative extension of bRML, plus more.
Observations

- The model of RML models (soundly) bRML too.
- bRML is less expressive than RML.
- All RML terms of type unit are expressible in bRML.
- RML is a conservative extension of bRML, plus more.

\[
\begin{array}{ccc}
\text{PCF}_v & \downarrow & \text{bRML} \\
\uparrow & & \downarrow \\
\text{Innocence} & & ?? \\
& & \downarrow \\
& & \text{Knowledge}
\end{array}
\]

What this talk is about
Reduced ML Games for Reduced ML [MT09]
Go simpler: RML Games for RML [AM98]
Example
RML with blocks: bRML
Observations
Binnocence
A problem
More intensional
bRML strategies exhibit a particular kind of uniformity:

\[ A \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_n \rightarrow \beta \]

\( \bullet OQ \)

\[ \vdots \]

\( \ast PA \)
bRML strategies exhibit a particular kind of uniformity:

\[ A \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_n \rightarrow \beta \]
bRML strategies exhibit a particular kind of uniformity:

\[ A \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_n \rightarrow \beta \]

\[ \bullet OQ \]

\[ \vdots \]

\[ \bullet OQ \]

\[ \vdots \]

\[ \bullet OQ \]

\[ \vdots \]
bRML strategies exhibit a particular kind of uniformity:

\[ A \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_n \rightarrow \beta \]

What this talk is about
Reduced ML Games for Reduced ML [MT09]
Go simpler: RML Games for RML [AM98]
Example
RML with blocks: bRML
Observations
Binocence
A problem
More intensional
bRML strategies exhibit a particular kind of uniformity:

\[ A \rightarrow B_1 \rightarrow B_2 \rightarrow \cdots \rightarrow B_n \rightarrow \beta \]

\[ \text{bRML strategies exhibit a particular kind of uniformity:} \]

Binoccence can be described as a “recursive version” of 3rd-move-binoccence (thread-independence).
A problem

What this talk is about
Reduced ML Games for Reduced ML [MT09]
Go simpler: RML Games for RML [AM98]
Example
RML with blocks: bRML
Observations
Binocence

A problem

A problem

More intensional
A problem

What this talk is about
Reduced ML
Games for Reduced ML [MT09]
Go simpler: RML
Games for RML [AM98]
Example
RML with blocks: bRML
Observations
Binnocence
A problem
More intensional

If the play on the left is non-uniform then $P$ cannot play $\bullet PQ$
A problem

If the play on the left is non-uniform then $P$ cannot play $\bullet_{PQ}$

In fact, $P$ cannot play $\bullet_{PQ}$ if it is justified by a move in an open block
Annotate explicitly blocks in plays.
Annotate explicitly blocks in plays.

Annotations give us a means to express blocks, but binnocence has become very complicated.
Annotate explicitly blocks in plays.

Annotations give us a means to express blocks, but binnocence has become very complicated.

Use stores as annotations, and go innocent.