Games and Full Abstraction for Java programs

Andrzej Murawski
University of Warwick

Nikos Tzevelekos
Queen Mary Uni. of London

PLSeminar 2013

What this talk is about

Semantics of programs:

- Operational (abstract machines)
- Denotational (abstract domains)

and the notion of Full Abstraction

Full abstraction: storyline, sequentiality problems, solution with Game Semantics

A fully abstract game semantics for Java programs (Interface Middleweight Java)

Operational Semantics

Interpret programs in abstract machines:

$$(s,\ \ x = 1; P)\ \rightarrow\ (s[x \mapsto 1],\ \ P)$$

- concrete step-wise description
- 'intuitive' semantics

Operational Equivalence

- Closed programs: $P \cong P'$ if

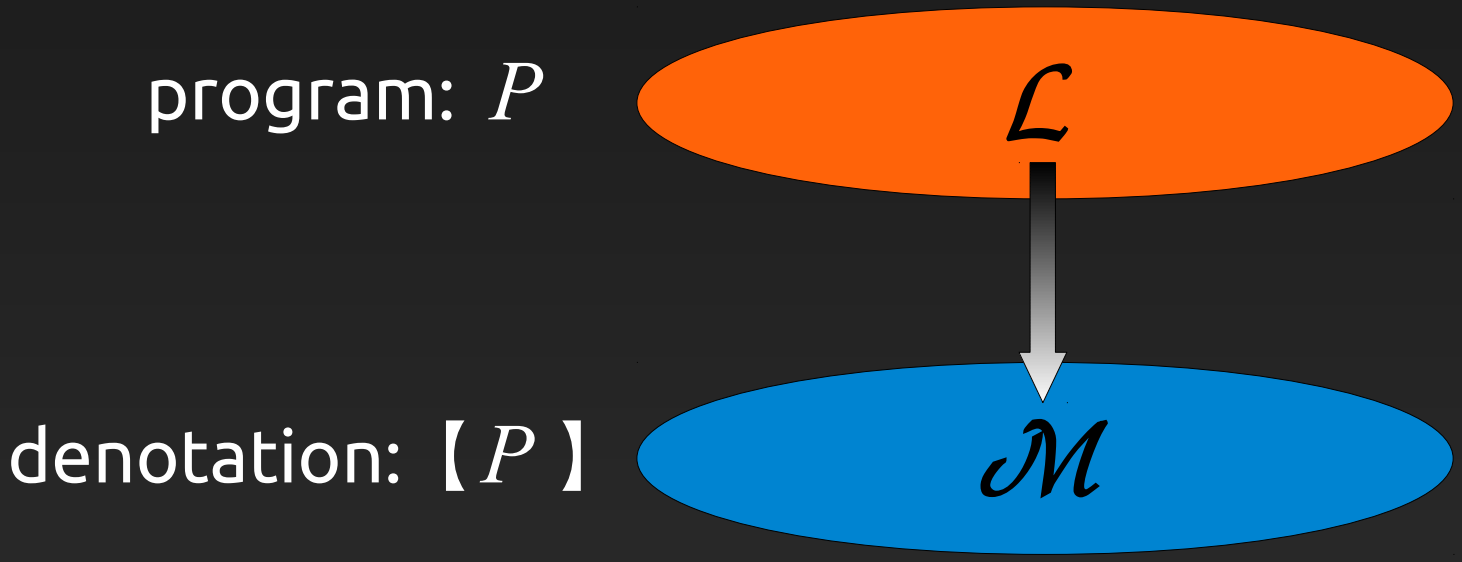
$$(s_0, P) \rightarrow^* (s, V) \iff (s_0, P') \rightarrow^* (s', V)$$

- Open programs: $P \cong P'$ if

$$(s_0, C[P]) \rightarrow^* (s, V) \iff (s_0, C[P']) \rightarrow^* (s', V)$$

Denotational Semantics

Translate programs into a domain of 'functions':



- abstract mathematical description
- 'high-level' semantics

Power of denotations

- Abstract away from implementation details
- Compositional translation:
  - “divide-and-conquer” modular approach
  - modelling of components in isolation

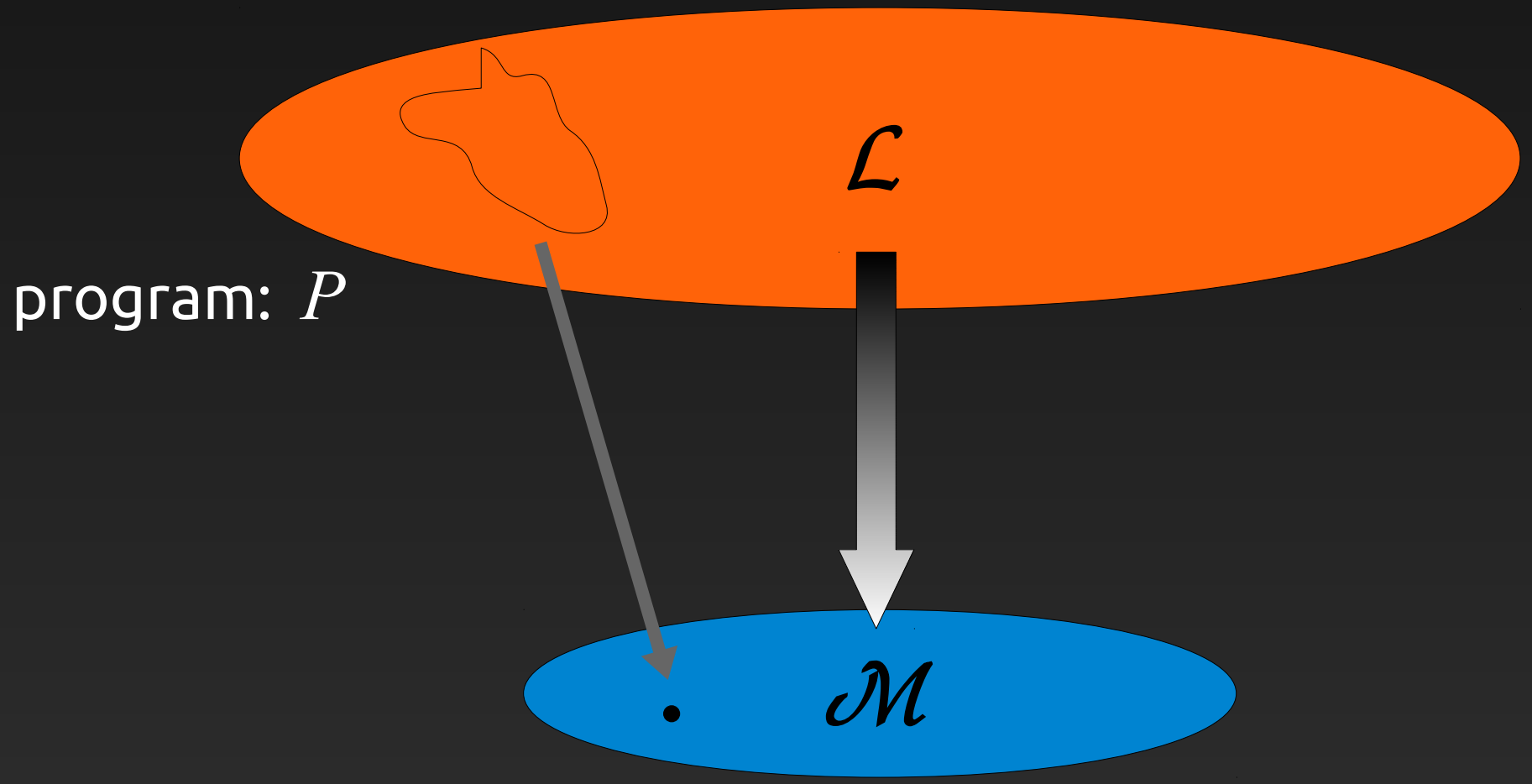
Useful for understanding & analysing programs

Full Abstraction

Desired properties of the translation:

- Correctness: $P \longrightarrow P' \Rightarrow \llbracket P \rrbracket = \llbracket P' \rrbracket$
- Soundness: $P \cong P' \Leftarrow \llbracket P \rrbracket = \llbracket P' \rrbracket$
- Full abstraction: $P \cong P' \Leftrightarrow \llbracket P \rrbracket = \llbracket P' \rrbracket$

Full abstraction pictorially



The quest for full abstraction

1977 [Milner, Plotkin]:

- Formulation of the problem (λ-calculus, PCF)
- Functions cannot capture sequentiality

1980-90's:

- Function stability [Berry, Bucciarelli, Erhard]
- Sequential algorithms [Berry, Currien]

1993 [AJM, HO/N *]: Game semantics (PCF)

- 'Functions' with operational content (*games*)

* Abramsky, Jagadeesan, Malacaria; Hyland, Ong; Nickau

From PCF to realistic languages

Full Abstraction for PCF (early 90's)

- Two groups in the UK, one in Germany
- Roots in Mathematical Logic

First stage (1993-2004)

- Models for various programming features
- Program analysis

} resources?

Nominal game semantics (2004-)

- Fragments of ML, now Java (IMJ)

Interface Middleweight Java (IMJ)

Object calculus based on MJ [Bierman, Parkinson, Pitts]

- Objects, inheritance, subtyping, casting, interfaces

Terms

$$M ::= \mathtt{x} \mid \mathsf{skip} \mid \mathsf{null} \mid x \mid i \mid M \oplus M \mid \mathsf{let}\, x = M \,\mathsf{in}\, M$$
$$\mid M = M \mid \mathsf{if}\, M \,\mathsf{then}\, M \,\mathsf{else}\, M \mid (\mathcal{I})M \mid \mathsf{new}(x : \mathcal{I}; \mathcal{M})$$
$$\mid M.\mathsf{f} \mid M.\mathsf{f} := M \mid M.\mathsf{m}(\overrightarrow{M})$$

Method implementations

$$\mathcal{M} ::= \emptyset \mid (\mathsf{m} : \lambda\vec{x}.M), \mathcal{M}$$

Game Semantics

Computation is modelled as a 2-player game between:

- *Opponent* (the environment), aka O
- *Proponent* (the program), aka P

Qualitative games (≠ Game Theory)

Computations = *plays* of a specified game

Programs = *strategies* for P

Families (i.e. *categories*) of games

Examples

$$x : \mathtt{Var} \vdash \; x.\mathtt{val} + 1 : \mathtt{int}$$

$$\left(\begin{array}{l} \mathtt{Var} \;\; : \{\; \mathtt{val} : \mathtt{int} \;\} \\ \mathtt{Fun} \;\; : \{\; \mathtt{val} : \mathtt{int} \to \mathtt{int} \;\} \end{array} \right)$$

Examples

$$x : \mathrm{Var} \vdash \; x.\mathrm{val} + 1 : \mathrm{int}$$

$$\begin{pmatrix} \mathrm{Var} & : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} & : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{pmatrix}$$

$$o : \mathtt{x}^{(\mathtt{x.val} = 5)}$$

Examples

$$x : \mathrm{Var} \vdash \; x.\mathrm{val} + 1 : \mathrm{int}$$

$$\begin{pmatrix} \mathrm{Var} & : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} & : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{pmatrix}$$

$$O : \mathrm{x}^{(\mathrm{x.val} = 5)}$$

$$P : 6^{(\mathrm{x.val} = 5)}$$

Examples

$$x : \mathrm{Var} \vdash x.\mathrm{val} + 1 : \mathrm{int}$$

$$\begin{pmatrix} \mathrm{Var} & : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} & : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{pmatrix}$$

$$\begin{array}{l} O : \mathtt{x}^{(\mathtt{x.val} = 5)} \\ P : 6^{(\mathtt{x.val} = 5)} \end{array} \quad \Big| \quad O : \mathtt{x}^{(\mathtt{x.val} = 8)}$$

Examples

$$x : \mathrm{Var} \vdash x.\mathrm{val} + 1 : \mathrm{int}$$

$$\begin{pmatrix} \mathrm{Var} & : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} & : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{pmatrix}$$

$$\begin{array}{l} \boldsymbol{O} : \mathtt{x}^{(\mathtt{x.val} = 5)} \\ \boldsymbol{P} : 6^{(\mathtt{x.val} = 5)} \end{array} \quad \Big| \quad \begin{array}{l} \boldsymbol{O} : \mathtt{x}^{(\mathtt{x.val} = 8)} \\ \boldsymbol{P} : 9^{(\mathtt{x.val} = 8)} \end{array}$$

Examples

$$x : \mathrm{Var} \vdash x.\mathrm{val} + 1 : \mathrm{int}$$

$$\begin{pmatrix} \mathrm{Var} & : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} & : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{pmatrix}$$

$$\begin{array}{l|l|ll} O : \mathrm{x}^{(\mathrm{x.val} = 5)} & O : \mathrm{x}^{(\mathrm{x.val} = 8)} & O : \mathrm{x}^{(\mathrm{x.val} = 3)} & \ldots \\ P : 6^{(\mathrm{x.val} = 5)} & P : 9^{(\mathrm{x.val} = 8)} & P : 4^{(\mathrm{x.val} = 3)} & \ldots \end{array}$$

Examples

$$x : \mathrm{Var} \vdash \; x.\mathrm{val} + 1 : \mathrm{int} \qquad \left(\begin{array}{ll} \mathrm{Var} & : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} & : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{array} \right)$$

$$\begin{array}{l|l|l} \boldsymbol{O} : \mathtt{x}^{(\mathtt{x.val}\,=\,5)} & \boldsymbol{O} : \mathtt{x}^{(\mathtt{x.val}\,=\,8)} & \boldsymbol{O} : \mathtt{x}^{(\mathtt{x.val}\,=\,3)} \ \ldots \\ \boldsymbol{P} : 6^{(\mathtt{x.val}\,=\,5)} & \boldsymbol{P} : 9^{(\mathtt{x.val}\,=\,8)} & \boldsymbol{P} : 4^{(\mathtt{x.val}\,=\,3)} \ \ldots \end{array}$$

$$\llbracket x : \mathrm{Var} \vdash \; x.\mathrm{val} + 1 : \mathrm{int} \rrbracket = \{\ \mathtt{x}^{(\mathtt{x.val}\,=\,i)} (i+1)^{(\mathtt{x.val}\,=\,i)}\ \}$$

Examples

$$x : \mathtt{Var}, f : \mathtt{Fun} \vdash$$
$$f.\,\mathrm{val}\,(x.\mathrm{val}) + 1 : \mathtt{int}$$

$$\begin{pmatrix} \mathtt{Var} : \{\ \mathrm{val} : \mathtt{int}\ \} \\ \mathtt{Fun} : \{\ \mathrm{val} : \mathtt{int} \to \mathtt{int}\ \} \end{pmatrix}$$

Examples

$$x : \mathrm{Var}, f : \mathrm{Fun} \vdash f.\,\mathrm{val}\,(x.\mathrm{val}) + 1 : \mathrm{int}$$

$$\begin{pmatrix} \mathrm{Var} : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{pmatrix}$$

O : (x,f)$^{(\mathrm{x.val} = 5)}$

P : call f.val(5)$^{(\mathrm{x.val} = 5)}$

O : ret f.val(8)$^{(\mathrm{x.val} = 42)}$

P : 9$^{(\mathrm{x.val} = 42)}$

Examples

$$x : \mathrm{Var}, f : \mathrm{Fun} \vdash f.\,\mathrm{val}\,(x.\mathrm{val}) + 1 : \mathrm{int}$$

$$\begin{pmatrix} \mathrm{Var} : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{pmatrix}$$

O : (x,f)$^{(\mathrm{x.val} = 5)}$
P : call f.val(5)$^{(\mathrm{x.val} = 5)}$
O : ret f.val(8)$^{(\mathrm{x.val} = 42)}$
P : 9$^{(\mathrm{x.val} = 42)}$

O : (x,f)$^{(\mathrm{x.val} = 7)}$
P : call f.val(7)$^{(\mathrm{x.val} = 7)}$
O : ret f.val(13)$^{(\mathrm{x.val} = 7)}$
P : 14$^{(\mathrm{x.val} = 7)}$

Examples

$$x : \mathrm{Var}, f : \mathrm{Fun} \vdash f.\,\mathrm{val}\,(x.\mathrm{val}) + 1 : \mathrm{int}$$

$$\begin{pmatrix} \mathrm{Var} : \{\ \mathrm{val} : \mathrm{int}\ \} \\ \mathrm{Fun} : \{\ \mathrm{val} : \mathrm{int} \to \mathrm{int}\ \} \end{pmatrix}$$

$\boldsymbol{O}$: $(\mathrm{x},\mathrm{f})^{(\mathrm{x.val}\,=\,5)}$
$\boldsymbol{P}$: $\mathrm{call}\ \mathrm{f.val}(5)^{(\mathrm{x.val}\,=\,5)}$
$\boldsymbol{O}$: $\mathrm{ret}\ \mathrm{f.val}(8)^{(\mathrm{x.val}\,=\,42)}$
$\boldsymbol{P}$: $9^{(\mathrm{x.val}\,=\,42)}$

$\boldsymbol{O}$: $(\mathrm{x},\mathrm{f})^{(\mathrm{x.val}\,=\,7)}$
$\boldsymbol{P}$: $\mathrm{call}\ \mathrm{f.val}(7)^{(\mathrm{x.val}\,=\,7)}$
$\boldsymbol{O}$: $\mathrm{ret}\ \mathrm{f.val}(13)^{(\mathrm{x.val}\,=\,7)}$
$\boldsymbol{P}$: $14^{(\mathrm{x.val}\,=\,7)}$

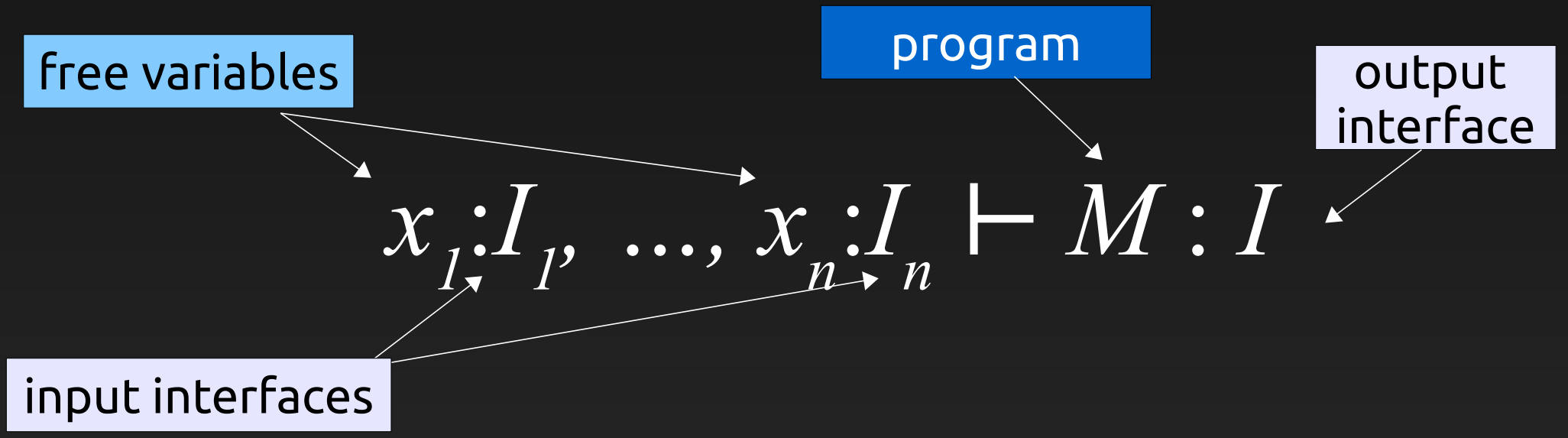
$$\llbracket x : \mathrm{Var}, f : \mathrm{Fun} \vdash f.\,\mathrm{val}\,(x.\mathrm{val}) + 1 : \mathrm{int} \rrbracket =$$
$$= \{\ (\mathrm{x},\mathrm{f})^{(\mathrm{x.val}\,=\,i)}\ \mathrm{call}\ \mathrm{f.val}(i)^{(\mathrm{x.val}\,=\,i)}\ \mathrm{ret}\ \mathrm{f.val}(j)^{(\mathrm{x.val}\,=\,i')}\ (j{+}1)^{(\mathrm{x.val}\,=\,i')}\ \}$$

Games in detail

$$x_1 : I_1, \ \ldots, \ x_n : I_n \vdash M : I$$

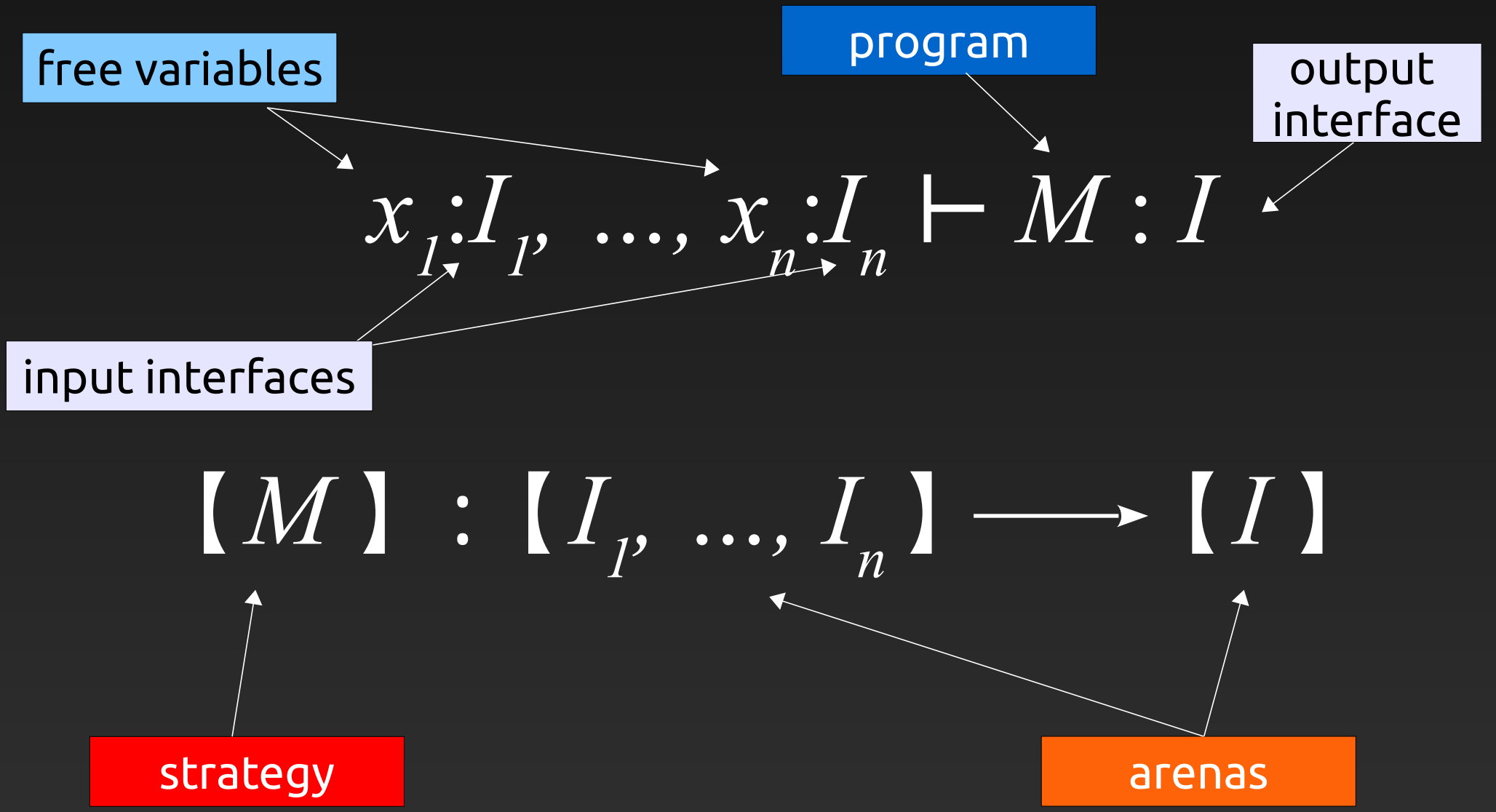
$$\llbracket M \rrbracket : \llbracket I_1, \ \ldots, \ I_n \rrbracket \longrightarrow \llbracket I \rrbracket$$

Games in detail

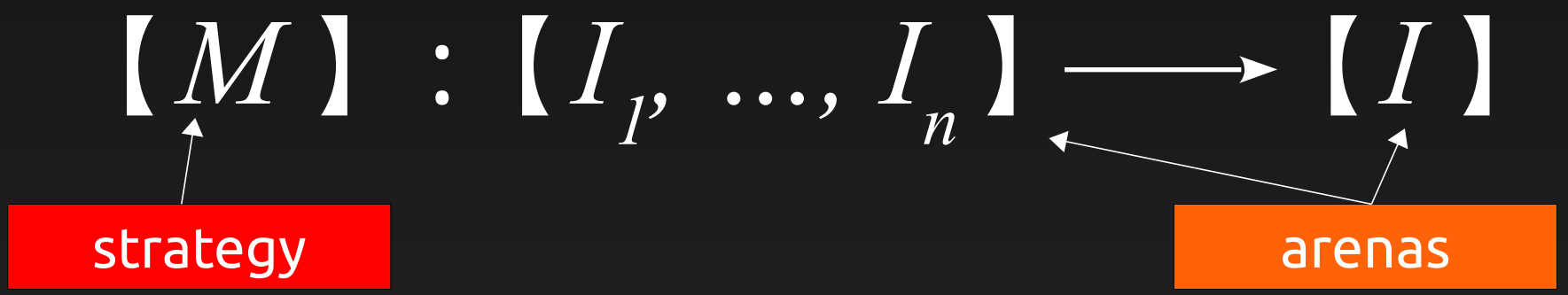


$$\llbracket M \rrbracket : \llbracket I_1, \ldots, I_n \rrbracket \longrightarrow \llbracket I \rrbracket$$

Games in detail



Arenas, moves



Arenas : sets of *names* with assigned interfaces

Moves :

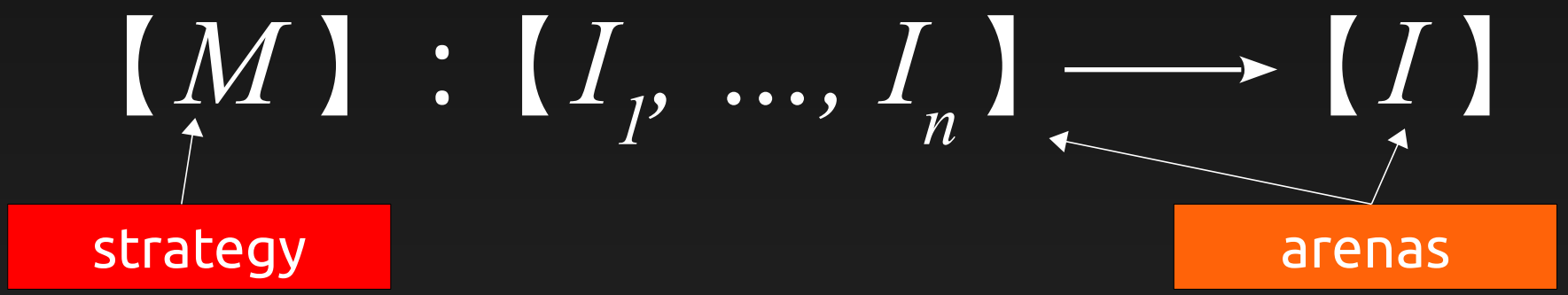
- *value* moves are tuples of names, integers, etc.)

```
42, (5,*,x,f), ...
```

- *method* moves are *calls* and *returns* (of methods, using values)

```
call f.val(42),
ret p.set(5,4), ...
```

Plays, strategies



Plays : sequences of *moves-with-store*

call f.val(5)$^{(x.val=5)}$, ...

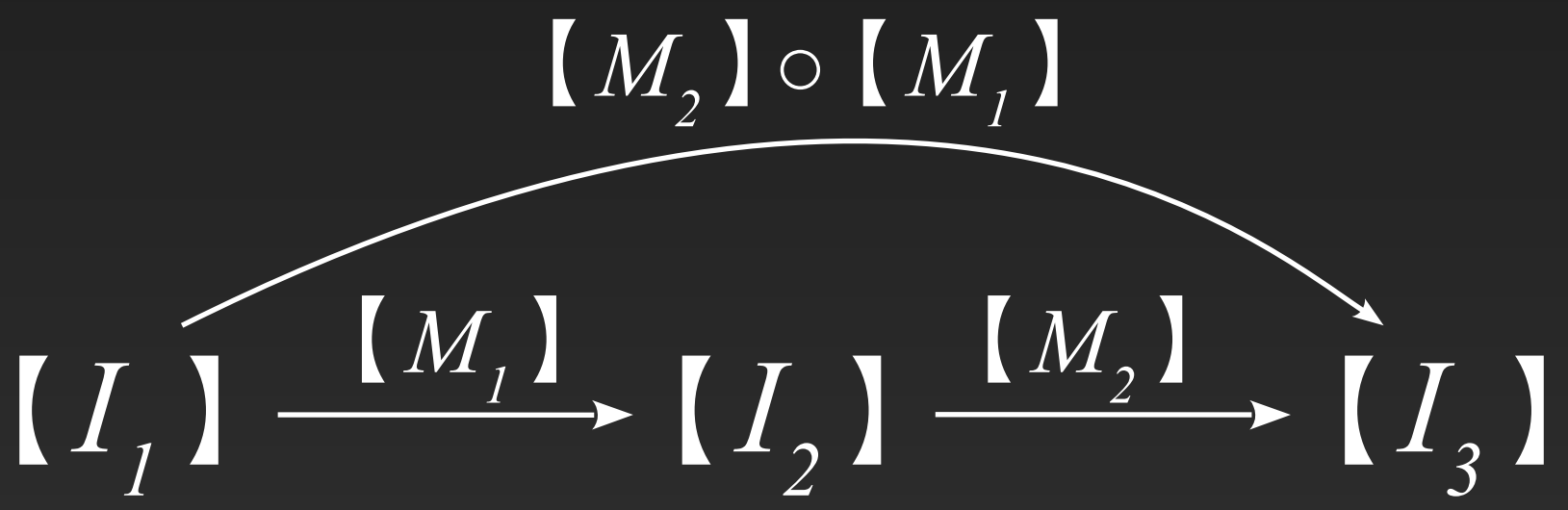
Strategies : sets of plays

A series of sanity conditions is used, e.g.
- moves have polarities (O/P), which alternate
- P calls methods of O, and viceversa
- dually for returns
- calls and returns adhere to interfaces of names
- strategies are closed wrt to O-subtyping
- …

Composition

Compound programs translated compositionally

- Strategy composition: play one strategy against the other



Full abstraction for IMJ

Theorem. The game translation is correct & sound

Lemma. Every finitary strategy is IMJ-definable

Theorem. The game model is fully abstract

Further on

Program analysis for IMJ

- Algorithmic representations
- Automata – over infinite alphabets!
- Model checking

Further effects

- Exceptions (cf. FoSSaCS'14)
- Concurrency (multi-threading, cf. Laird'06)

Further on

thanks!

Program analysis for IMJ

- Algorithmic representations
- Automata – over infinite alphabets!
- Model checking

Further effects

- Exceptions (cf. FoSSaCS'14)
- Concurrency (multi-threading, cf. Laird'06)