GAME SEMANTICS FOR GOOD GENERAL REFERENCES

Andrzej S. Murawski
University of Leicester

Nikos Tzevelekos
University of Oxford

ML-LIKE REFERENCES

- Creation

$$\mathsf{ref}(M)$$

- Assignment

$$M := N$$

- Dereferencing

$$!M$$

- Equality

$$M = N$$

OCAML SESSION

Ground storage

```
# let r0=ref(());;
val r0 : unit ref = {contents = ()}

# let r1=ref(3);;
val r1 : int ref = {contents = 3}
```

Reference storage

```
# let r2=ref(r1);;
val r2 : int ref ref = {contents = {contents = 3}}
```

MORE OCAML

Higher-order storage

```
# let r3=ref(fun(x:int ref) -> x==r1);;
val r3 : (int ref -> bool) ref = {contents = <fun>}

# (!r3)(r1);;
- : bool = true

# (!r3)(ref(3));;
- : bool = false
```

TYPES

$$\theta, \theta' \quad ::= \quad \mathsf{unit} \quad | \quad \mathsf{int} \quad | \quad \mathsf{ref}\,\theta \quad | \quad \theta \to \theta'$$

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

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

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

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

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

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

Fig. 1. Syntax of RefML.

EXPRESSIVITY

- Divergence

$$\text{let } y = \text{ref}_{\text{unit}\to\text{unit}}(\cdots)\text{ in}$$
$$y := (\lambda z^{\text{unit}}.(!y)z);\ \ !y$$

- Recursion

$$\lambda f^{(\theta_1\to\theta_2)\to(\theta_1\to\theta_2)}.\text{let } y = \text{ref}_{\theta_1\to\theta_2}(\cdots)\text{ in}$$
$$y := (\lambda z^{\theta_1}.f(!y)z);\ \ !y$$

- Objects, aspects, ...

SEMANTICS

$\Gamma \vdash M_1 : \theta$ and $\Gamma \vdash M_2 : \theta$ are **equivalent**
if and only if
they are not distinguishable by any context.

For all contexts $C[-]$ such that $\vdash C[M_1], C[M_2]$
$C[M_1] \Downarrow$ if and only if $C[M_2] \Downarrow$.

Full Abstraction

$\Gamma \vdash M_1 \cong M_2$ if and only if $[\![\Gamma \vdash M_1]\!] = [\![\Gamma \vdash M_2]\!]$

REYNOLDS' APPROACH

$$\mathsf{ref}\,\theta = (\mathsf{unit} \to \theta) \times (\theta \to \mathsf{unit})$$

Issues

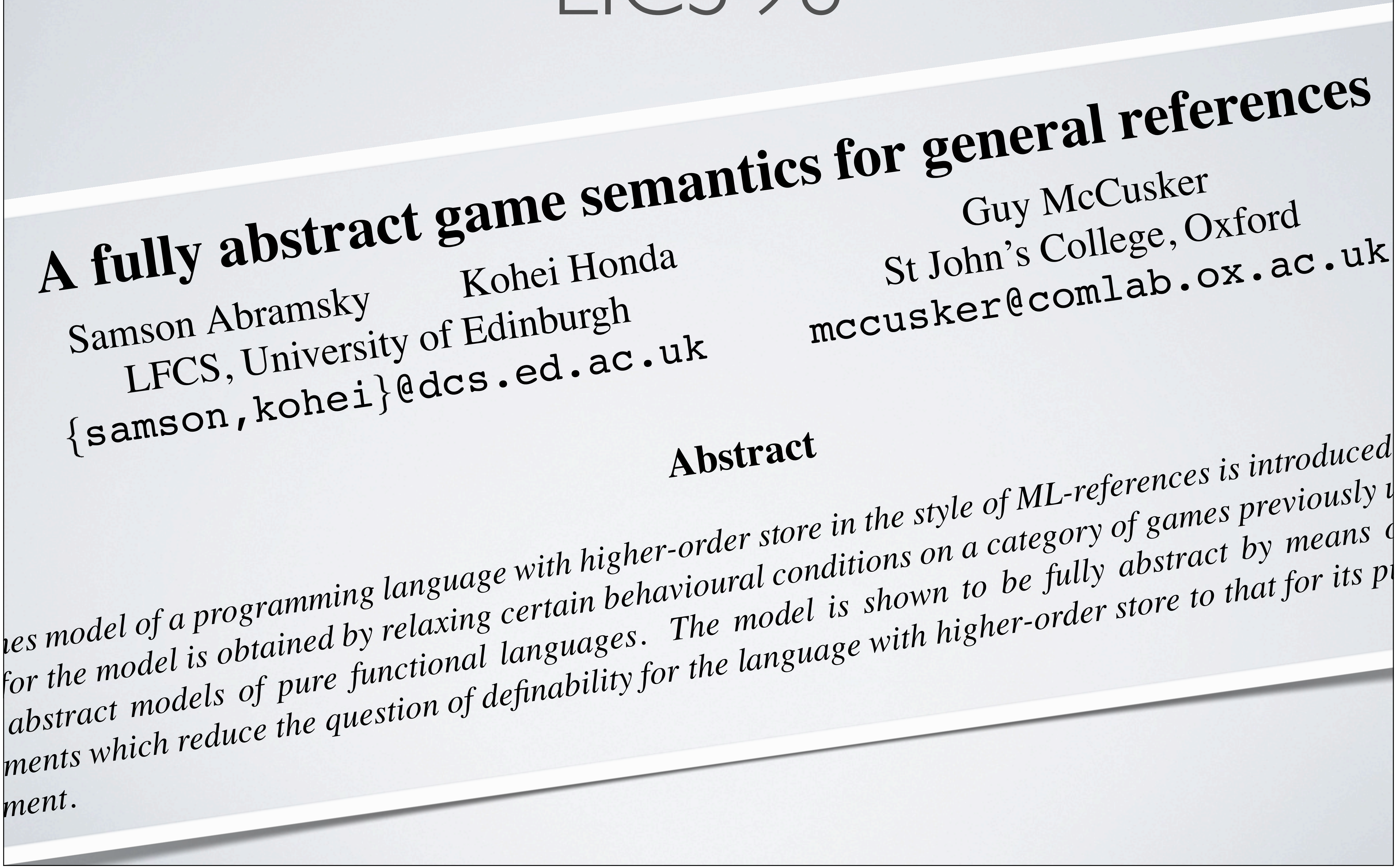
- Disconnect between reads and writes.
- Reading and writing can produce unrestricted side-effects.

LACK OF FULL ABSTRACTION

Some basic equivalences are not validated by the model.

$$x : \mathsf{ref}\,\theta \quad \vdash \qquad \begin{array}{rcl} x := !x & \cong & () \\ x := V; !x & \cong & x := V; V \\ x := V; x := W & \cong & x := W \end{array}$$

LICS'98



A fully abstract game semantics for general references

Samson Abramsky Kohei Honda
LFCS, University of Edinburgh
{samson,kohei}@dcs.ed.ac.uk

Guy McCusker
St John's College, Oxford
mccusker@comlab.ox.ac.uk

Abstract

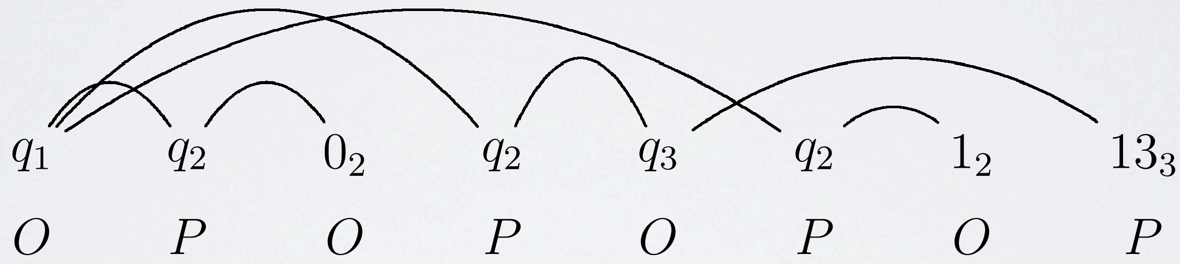
...es model of a programming language with higher-order store in the style of ML-references is introduced ...for the model is obtained by relaxing certain behavioural conditions on a category of games previously u... abstract models of pure functional languages. The model is shown to be fully abstract by means o... ...ments which reduce the question of definability for the language with higher-order store to that for its pu... ...ment.

NOMINAL GAME SEMANTICS

$$[\![\mathsf{ref}\,\theta]\!] = \mathbb{A}_\theta$$

GAME SEMANTICS

- Models programs as strategies in games between programs and potential contexts.



- The plays can be viewed as an abstracted account of interaction ("only observable behaviour is revealed").

SOME NOMINAL PLAYS

$$f : \mathsf{ref\,int} \to \mathsf{unit} \vdash \quad \mathsf{let}\, n = \mathsf{ref}_{\mathsf{int}}(0)\, \mathsf{in}\, (f n); n : \mathsf{ref\,int}$$

$$\begin{aligned} f : \mathsf{ref\,int} \to \mathsf{unit} \vdash \quad & \mathsf{let}\, n_1 = \mathsf{ref}_{\mathsf{int}}(0)\, \mathsf{in} \\ & \mathsf{let}\, n_2 = \mathsf{ref}_{\mathsf{int}}(0)\, \mathsf{in} \\ & \quad (f n_1); (n_2 := !n_1); n_2 : \mathsf{ref\,int} \end{aligned}$$

$$* \quad n^{(n,0)} - *^{(n,k)} \quad n^{(n,k)}$$

$$* \quad n_1^{(n_1,0)} - *^{(n_1,k)} \quad n_2^{(n_1,k),(n_2,k)}$$

MOVES WITH STORES

$$m^{\Sigma}$$

What is the content of Σ? Pairs $(\ell, \cdots)$ of four kinds.

$$(n_0, \star) \qquad (n_1, 3) \qquad (n_2, n_1) \qquad (n_3, \star)$$

cf. OCAML interpreter

```
# let r=ref(fun (x:int) -> x);;
val r : (int -> int) ref = {contents = <fun>}
```

HIGHER-ORDER STORE

- We cannot reveal higher-order values in the store. This would jeopardize full abstraction!
- The properties of stored values will be revealed during play thanks to the use of special pointers to the store (in previous game models, pointers could only point at other moves).

$$m^{(a,\star)} \quad \ldots \quad n^{(\cdots)}$$

HIGHER-ORDER STORE

$$x : \mathsf{ref}\,(\mathsf{int} \to \mathsf{int}) \vdash\, !x : \mathsf{int} \to \mathsf{int}$$

$$a^{(a,\star)}\ \star^{(a,\star)}\ 1^{(a,\star)}\ 1^{(a,\star)}\ 3^{(a,\star)}\ 3^{(a,\star)}$$

$$x : \mathsf{ref}\,(\mathsf{int} \to \mathsf{int}) \vdash \lambda h^{\mathsf{int}}.(!x)h : \mathsf{int} \to \mathsf{int}$$

$$a^{(a,\star)}\ \star^{(a,\star)}\ 1^{(a,\star)}\ 1^{(a,\star)}\ 3^{(a,\star)}\ 3^{(a,\star)}$$

STANDARD COMPOSITION

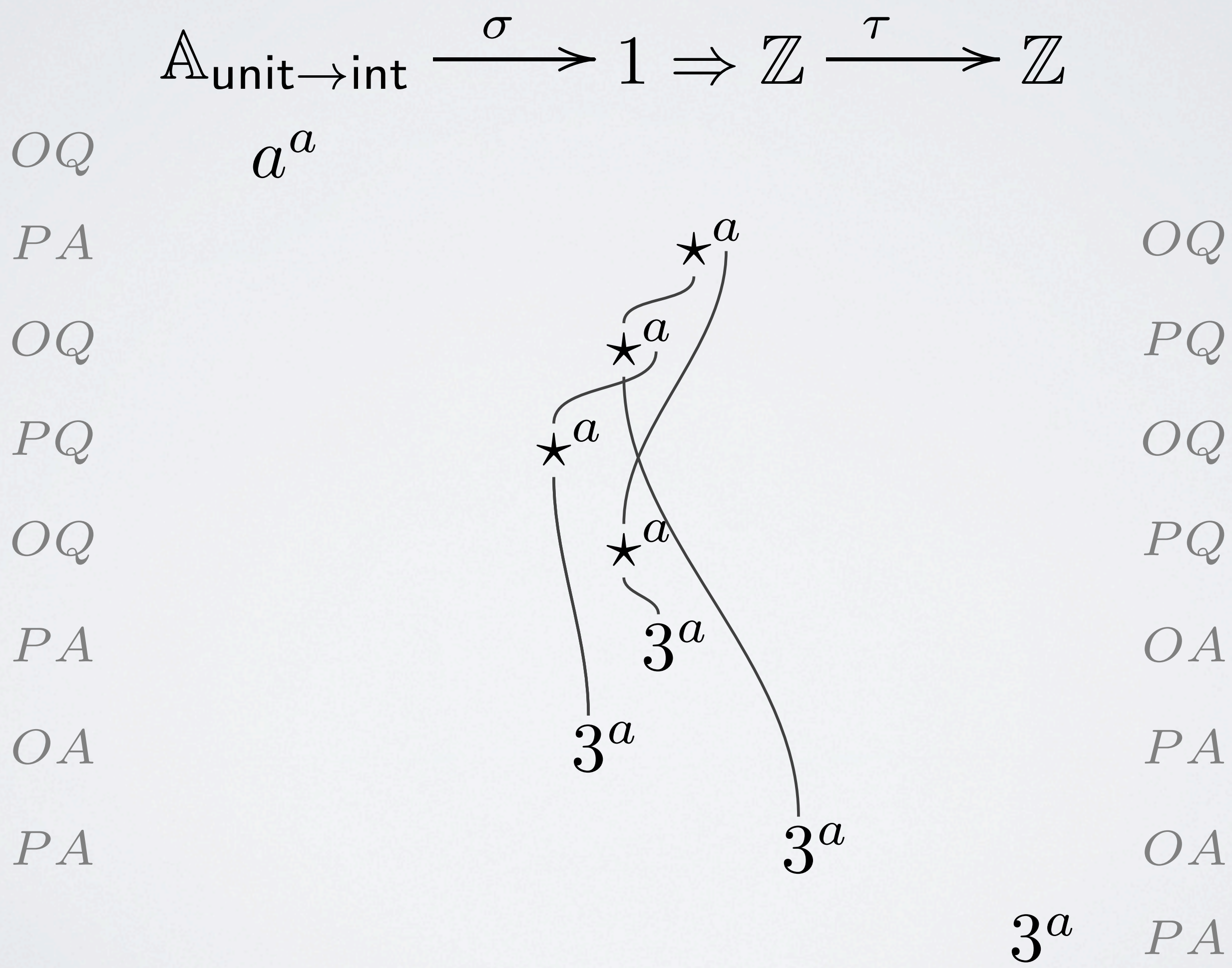
Given $\sigma : A \to B$ and $\tau : B \to C$,
the strategy $\sigma;\tau : A \to C$ is defined to be

$$(\sigma \parallel_B \tau) \setminus B.$$

The two strategies play each other in B.

REFINED COMPOSITION

What if one strategy accesses a location that the other has no access to? Copycat!

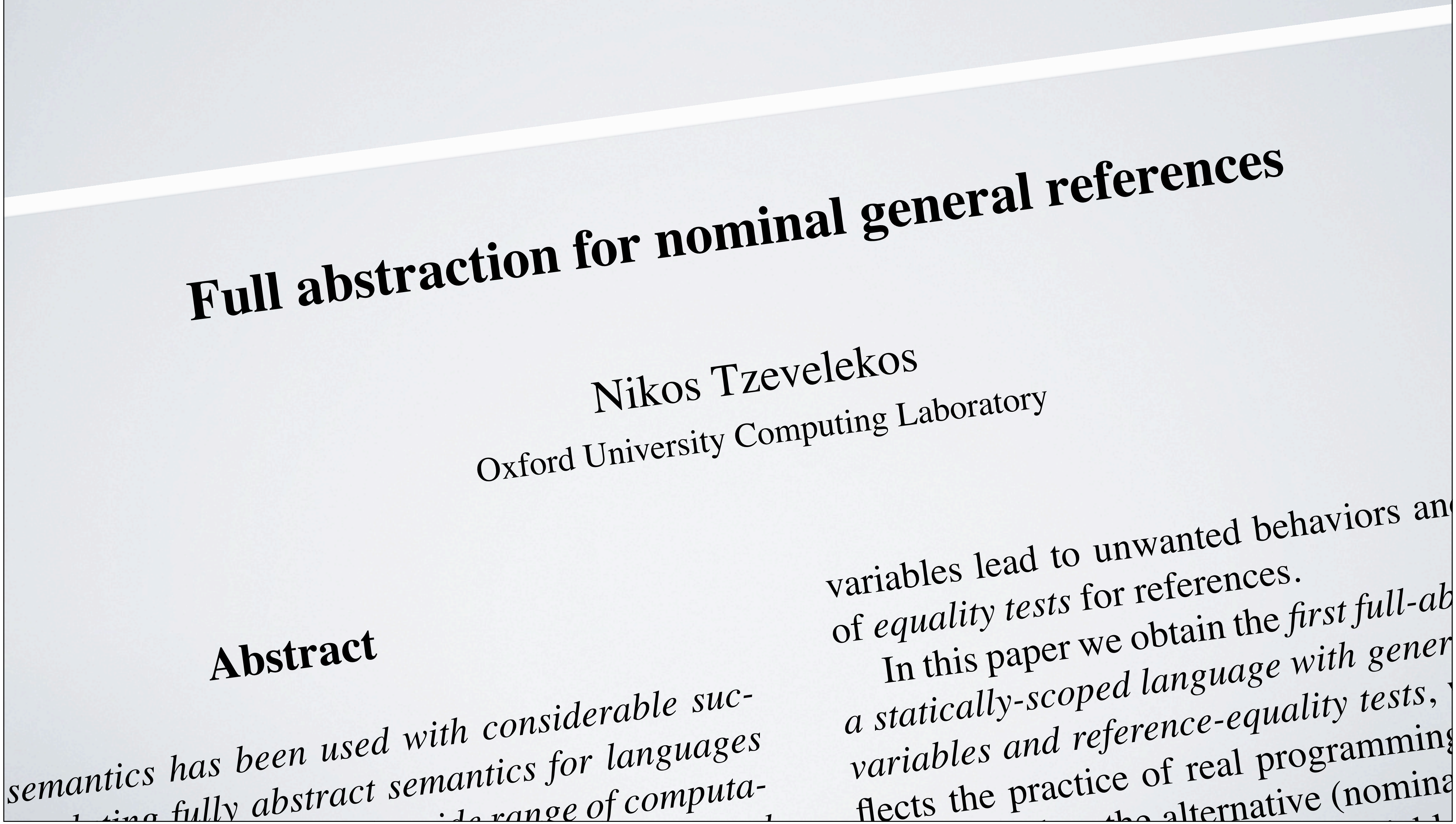


NEW INGREDIENTS

- Sequences of moves with store that point to other moves or stores of other moves.
- Composition is parallel composition with copying plus hiding.

$$\Gamma \vdash M_1 \cong M_2 \qquad \text{if and only if} \qquad [\![\Gamma \vdash M_1]\!] = [\![\Gamma \vdash M_2]\!]$$

LICS'07



Full abstraction for nominal general references

Nikos Tzevelekos

Oxford University Computing Laboratory

Abstract

semantics has been used with considerable suc-
fully abstract semantics for languages
range of computa-

variables lead to unwanted behaviors an
of *equality tests* for references.

In this paper we obtain the *first full-ab*
a statically-scoped language with gener
variables and reference-equality tests,
flects the practice of real programming
alternative (nomina

FUTURE WORK

- ★ Connections with LTS semantics

  [Jeffrey & Rathke, LICS'99]
  [Laird, ICALP'07]

- ★ Polymorphism and recursive types

- ★ Algorithmic game semantics