Normal form bisimulation
aka Open bisimulation
aka Operational game semantics
for polymorphism

SOREN LASSEN, Google Sydney

PAUL BLAIN LEVY, Birmingham

Part 1 Normal form bisimulation

Part 2 Polymorphism

RELATED WORK

Denotational game semantics

- Hyland, Ong (PCF)
- Nickau (PCF)
- Abramsky, McCusker (Idealized Algol)
- Abramsky, Honda, McCusker (General refs)

Operational game semantics

- Laird (ICALP '07, traces)
- Jagadeesan, Riely, Pitcher (Aspects)

Ultimate patterns

- Abramsky, McCusker (call-by-value games)
- Harper, Licata, Zeilberger

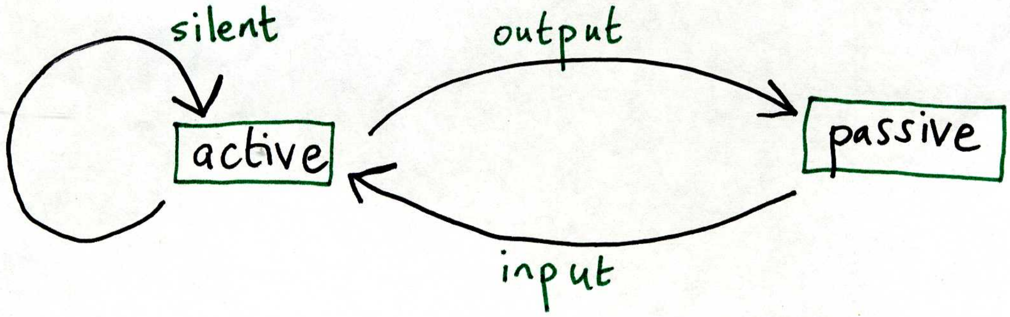
Generic polymorphism

- Abramsky, Jagadeesan

NORMAL FORM BISIMULATION – THE IDEA
(Sangiorgi)

- A labelled transition system on open terms of λ-calculus

- More precisely, a bi-labelled transition system with $\boxed{\text{active}}$ and $\boxed{\text{passive}}$ nodes



- Deterministic, so
  bisimilarity = trace equivalence

JUMP-WITH-ARGUMENT

- JWA is the target of CPS transforms
- Functions are called, but they do not return.

Types $A ::= \sum_{i \in I} A_i \mid 1 \mid A \times A \mid \neg A \mid \mathrm{rec}\ X.\ A$

Judgements $\Gamma \vdash^{v} V : A$ values
$\Gamma \vdash^{n} M$ nonreturning commands

Terms $V ::= x \mid \langle i, V \rangle \mid \langle\rangle \mid \langle V, V' \rangle \mid \lambda x.M \mid \mathrm{fold}\ V$

$M ::= \mathrm{pm}\ V\ \mathrm{as}\ \{\langle i, x \rangle.\ M_i\}_{i \in I} \mid$
$\mathrm{pm}\ V\ \mathrm{as}\ \langle x, y \rangle.\ M \mid \mathrm{pm}\ V\ \mathrm{as}\ \langle\rangle.\ M \mid$
$V\ V \mid \mathrm{pm}\ V\ \mathrm{as}\ \mathrm{fold}\ x.\ M$

$$\frac{\Gamma, x : A \vdash^{n} M}{\Gamma \vdash^{v} \lambda x.M : \neg A} \qquad \frac{\Gamma \vdash^{v} V : \neg A \quad \Gamma \vdash^{v} W : A}{\Gamma \vdash^{n} V W}$$

OPERATIONAL SEMANTICS: THE C-MACHINE

We evaluate commands in a fixed context Γ

Transitions (β-reductions)

$$\mathsf{pm}\ \langle \hat{\imath}, V\rangle\ \mathsf{as}\ \{\langle i, x\rangle . M_i\}_{i\in I} \rightsquigarrow M_{\hat{\imath}}[V/x]$$

$$\mathsf{pm}\ \langle V, V'\rangle\ \mathsf{as}\ \langle x, y\rangle . M \rightsquigarrow M[V/x, V'/y]$$

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

$$\mathsf{pm}\ \mathsf{fold}\ V\ \mathsf{as}\ \mathsf{fold}\ x . M \rightsquigarrow M[V/x]$$

Terminal commands

$$\left.\begin{array}{l} \mathsf{pm}\ z\ \mathsf{as}\ \{\langle i, x\rangle . M_i\}_{i\in I} \\ \mathsf{pm}\ z\ \mathsf{as}\ \langle x, y\rangle .\ M \\ z\, V \\ \mathsf{pm}\ z\ \mathsf{as}\ \mathsf{fold}\ x . M \end{array}\right\} \text{where } z \in \Gamma$$

Observational equivalence

With respect to commands $x : \neg \sum_{i\in I} 1 \vdash^{\mathsf{c}} M$

ULTIMATE PATTERN MATCHING

Theorem

function context (all ¬ types)

A value $\Gamma \vDash V : A$

is uniquely of the form $p\,[w]$

Ultimate pattern (the tags)

Filling (the functions)

$\langle i, \langle j, \langle \lambda x.M, y \rangle\rangle\rangle$

Ultimate pattern $\langle i, \langle j, \langle -, - \rangle\rangle\rangle$

Filling $\lambda x.M \quad y$

THE ULTIMATE PATTERNS

$$p ::= \_ \mid \langle i, p \rangle \mid \langle p, p \rangle \mid \mathsf{fold}\ p$$

upatt(A) the set of ultimate pattens of type A

H(p) the list of types of holes in p (all ¬ types)

THE NODES

- When I pass you a function, for you it's a fresh identifier

- So we have two contexts $\overrightarrow{f:\neg A} \,\|\, \overrightarrow{g:\neg B}$

functions that have been input

functions that have been output

- And we have a substitution $\overrightarrow{g \mapsto V}$

Passive node $\quad \overrightarrow{f:\neg A} \,\|\, \overrightarrow{g:\neg B} \vdash \overrightarrow{g \mapsto V}$

Active node $\quad \overrightarrow{f:\neg A} \,\|\, \overrightarrow{g:\neg B} \vdash \overrightarrow{g \mapsto V} ; M$

$\overrightarrow{f:\neg A} \vdash \vec{V}, M$

Silent transition

$$\overrightarrow{f:\neg A} \,\|\, \overrightarrow{g:\neg B} \vdash \overrightarrow{g\mapsto V} \,;\, M$$
$$\rightsquigarrow \overrightarrow{f:\neg A} \,\|\, \overrightarrow{g:\neg B} \vdash \overrightarrow{g\mapsto V} \,;\, M'$$

$M \rightsquigarrow M'$

Output transition

$$\overrightarrow{f:\neg A} \,\|\, \overrightarrow{g:\neg B} \vdash \overrightarrow{g\mapsto V} \,;\, f\,p[\vec{W}]$$
$$\overset{fp}{\rightsquigarrow} \overrightarrow{f:\neg A} \,\|\, \overrightarrow{g:\neg B}, \vec{h}:H(p) \vdash \overrightarrow{g\mapsto V}, \overrightarrow{h\mapsto W}$$

$p \in \mathrm{upatt}(A)$

Input transition

$$\overrightarrow{f:\neg A} \,\|\, \overrightarrow{g:\neg B} \vdash \overrightarrow{g\mapsto V}$$
$$:gq = \overrightarrow{f:\neg A}, \vec{h}:H(p) \,\|\, \overrightarrow{g:\neg B} \vdash \overrightarrow{g\mapsto V} \,;\, V\,q[\vec{h}]$$

$q \in \mathrm{upatt}(B)$

BISIMILARITY

$$
\begin{tikzcd}
\Gamma_i \| \Gamma_o \vdash & c \arrow[r, dash, "\mathcal{R}"] \arrow[d, "*"] & c' \arrow[d, dashed, "*"] \\
& \cdot \arrow[d, "xp"] & \cdot \arrow[d, dashed, "xp"] \\
\Gamma_i \| \Gamma_o, \vec{y}:H(p) \vdash & d \arrow[r, dash, dashed, "\mathcal{R}"] & d'
\end{tikzcd}
\qquad
\begin{array}{l}
(x:\neg A) \in \Gamma_i \\
p \in \mathrm{upatt}(A)
\end{array}
$$

$$
\begin{array}{lcl}
\Gamma_i \| \Gamma_o \vdash & d \overset{\mathcal{R}}{\text{———}} d' & \text{implies} \\
\Gamma_i, \vec{x}:H(q) \| \Gamma_o \vdash & d:yq \overset{\mathcal{R}}{\text{———}} d':yq &
\end{array}
\qquad
\begin{array}{l}
(y:\neg B) \in \Gamma_o \\
q \in \mathrm{upatt}(B)
\end{array}
$$

function context

- $\Gamma \mathrel{\hat{\vdash}} M \approx M'$ when $\Gamma \| \varepsilon \vdash \varepsilon; M \approx \varepsilon; M'$
- Use ultimate pattern matching for other contexts, and for values.
- This is a congruence.

Cumulative style - the Opponent accumulates values.

JUMP-WITH-ARGUMENT + POLYMORPHISM

Types $\vec{X} \vdash A \text{ type}$

$$A ::= \sum_{i \in I} A_i \mid 1 \mid A \times A \mid \neg A \mid \text{rec } X . A \mid X \mid \sum X . A$$

Term Judgements $\vec{X}, \Gamma \vdash^{v} V : A \qquad \vec{X}, \Gamma \vdash^{n} M$

Terms $V ::= \langle \hat{i}, V \rangle \mid \langle\rangle \mid \langle V, V \rangle \mid \lambda x . M \mid \text{fold } V \mid \langle A, V \rangle$

$$\begin{aligned} M ::= {} & \text{pm } V \text{ as } \{\langle i, x \rangle . M_i\}_{i \in I} \mid \\ & \text{pm } V \text{ as } \langle x, y \rangle . M \mid \text{pm } V \text{ as } \langle\rangle . M \mid \\ & V\,V \mid \quad \text{pm } V \text{ as fold } x . M \\ & \mid \text{pm } V \text{ as } \langle X, x \rangle . M \end{aligned}$$

C-MACHINE + POLYMORPHISM

We evaluate commands in fixed context $\vec{X}, \Gamma$

Transitions

$$\text{pm } \langle \hat{\imath}, V\rangle \text{ as } \{\langle i, x\rangle. M_i\}_{i\in I} \rightsquigarrow M_{\hat{\imath}}[V/x]$$

$$\text{pm } \langle V, V'\rangle \text{ as } \langle x, y\rangle. M \rightsquigarrow M[V/x, V'/x]$$

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

$$\text{pm fold } V \text{ as fold } x. M \rightsquigarrow M[V/x]$$

$$\text{pm } \langle A, V\rangle \text{ as } \langle X, x\rangle. M \rightsquigarrow M[A/X, V/x]$$

Terminal commands

$$\left.\begin{array}{l} \text{pm } z \text{ as } \{\langle i, x\rangle. M_i\}_{i\in I} \\ \text{pm } z \text{ as } \langle x, y\rangle. M \\ z\, V \\ \text{pm } z \text{ as fold } x. M \\ \text{pm } z \text{ as } \langle X, x\rangle. M \end{array}\right\} \text{ where } z \in \Gamma$$

ULTIMATE PATTERN / FILLING

A value V that I pass to you contains

- Tags (that you know are tags) Ultimate pattern
- Functions (that you know are functions) Filling
- Types (that you know are types) Filling
- Opaque values, whose type is an identifier X

Two kinds of opaque values

- Sender type, where I either have sent or am sending within V the type X Filling
- Recipient type, where you've sent me the type X Ultimate pattern

EXAMPLE COMMAND

$$X,\ x:X,\ x':X,\quad f:\neg(1+\Sigma Y.(X\times Y\times\neg\Sigma Z.(X\times Y\times Z\times\neg(Y\times Z))))$$

$$\begin{aligned}
\vdash^{n}\ f\ \mathrm{inr}\ \langle X\times\mathrm{bool},\ \langle x,\ \langle x,\ \mathrm{true}\rangle,\ & \\
&\lambda u.\ \mathrm{pm}\ u\ \mathrm{as}\ \langle Z,\ \langle x'',\ \langle x''',\ v\rangle,\ z,\ f'\rangle\rangle. \\
&\qquad \mathrm{pm}\ v\ \mathrm{as}\ \{\ \mathrm{true}.\ f'\langle\langle x,\ \mathrm{false}\rangle,\ z\ \rangle \\
&\qquad\qquad\qquad \mathrm{false}.\ \mathrm{diverge}\ \ \}\ \rangle\rangle
\end{aligned}$$

INITIAL CONFIGURATION

$$\begin{array}{l|l}
X,\ x:X,\ x':X, & \\
f:\neg(1+\Sigma Y.(X\times Y\times\neg\Sigma Z.(X\times Y\times Z\times\neg(Y\times Z)))) & \varepsilon\quad\vdash
\end{array}$$

$$\begin{aligned}
\varepsilon;\ f\ \mathrm{inr}\ \langle X\times\mathrm{bool},\ \langle x,\ \langle x,\ \mathrm{true}\rangle,\ & \\
&\lambda u.\ \mathrm{pm}\ u\ \mathrm{as}\ \langle Z,\ \langle x'',\ \langle x''',\ v\rangle,\ z,\ f'\rangle\rangle. \\
&\qquad \mathrm{pm}\ v\ \mathrm{as}\ \{\ \mathrm{true}.\ f'\langle\langle x,\ \mathrm{false}\rangle,\ z\rangle \\
&\qquad\qquad\qquad \mathrm{false}.\ \mathrm{diverge}\ \}\ \rangle\rangle
\end{aligned}$$

$$\rightsquigarrow^{*}$$

$$X,\ x : X,\ x' : X,\ f : \neg(1 + \Sigma Y.(X \times Y \times \neg \Sigma Z.(X \times Y \times Z \times \neg(Y \times Z)))) \ \| \ \varepsilon \ \vdash$$

$$\varepsilon;\ f\ \mathrm{inr} < X \times \mathrm{bool},\ \langle x,\ \langle x, \mathrm{true}\rangle,$$
$$\lambda u.\ \mathrm{pm}\ u\ \mathrm{as}\ \langle Z,\ \langle x'', \langle x''', v\rangle, z, f'\rangle\rangle.$$
$$\mathrm{pm}\ v\ \mathrm{as}\ \{\mathrm{true}.\ f'\langle\langle x, \mathrm{false}\rangle, z\rangle$$
$$\mathrm{false}.\ \mathrm{diverge}\}\ \rangle\rangle$$

$$\xrightarrow{f\ \mathrm{inr}\ \langle -,\ \langle x,\ -,\ -\rangle\rangle}$$

$$X,\ x : X,\ x' : X,\ f : \neg(1 + \Sigma Y.(X \times Y \times \neg \Sigma Z.(X \times Y \times Z \times \neg(Y \times Z)))) \ \| \ Y,\ y : Y,\ g : \neg \Sigma Z.(X \times Y \times Z \times \neg(Y \times Z))$$

$$\vdash\ Y \mapsto X \times \mathrm{bool},\ y \mapsto \langle x, \mathrm{true}\rangle,$$
$$g \mapsto \lambda u.\ \mathrm{pm}\ u\ \mathrm{as}\ \langle Z,\ \langle x'', \langle x''', v\rangle, z, f'\rangle\rangle.$$
$$\mathrm{pm}\ v\ \mathrm{as}\ \{\mathrm{true}.\ f'\langle\langle x, \mathrm{false}\rangle, z\rangle$$
$$\mathrm{false}.\ \mathrm{divege}\}$$

$X, x : X, x' : X,$

$f : \neg(1 + \Sigma Y.(X \times Y \times \neg\Sigma Z.(X \times Y \times Z \times \neg(Y \times Z))))$ $\Vert$ $Y, y : Y,$

$g : \neg\Sigma Z.(X \times Y \times Z \times \neg(Y \times Z))$

$\vdash$ $Y \mapsto X \times \mathrm{bool}, \; y \mapsto \langle x, \mathrm{true}\rangle$

$g \mapsto \lambda u.\ \mathrm{pm}\ u\ \mathrm{as}\ \{\ \langle Z, \langle x'', \langle x''', v\rangle, z, f'\rangle\rangle.$

$\mathrm{pm}\ v\ \mathrm{as}\ \{\ \mathrm{true}.\ f'\langle\langle x, \mathrm{false}\rangle, z\rangle$

$\mathrm{false}.\ \mathrm{diverge}\ \}$

$: \; g\langle -, \langle -, y, -, -\rangle\rangle \; =$

$X, x : X, x' : X, x'' : X,$

$Z, z : Z,$

$f : \neg(1 + \Sigma Y.(X \times Y \times \neg\Sigma Z.(X \times Y \times Z \times \neg(Y \times Z))))$

$h : \neg(Y \times Z)$

$\Vert$ $Y, y : Y$

$g : \neg\Sigma Z.(X \times Y \times Z \times \neg(Y \times Z))$ $\vdash$

$Y \mapsto X \times \mathrm{bool}, \; y \mapsto \langle x, \mathrm{true}\rangle, \; g \mapsto \lambda u.\ \mathrm{pm}\ u\ \mathrm{as}\ \langle Z, \langle x'', \langle x''', v\rangle, z, f'\rangle\rangle.$

$\mathrm{pm}\ v\ \mathrm{as}\ \{\mathrm{true}.\ f'\langle\langle x, \mathrm{false}\rangle, z\rangle$

$\mathrm{false}.\ \mathrm{diverge}$

$; \; \left(\lambda u.\ \mathrm{pm}\ u\ \mathrm{as}\ \langle Z, \langle x'', \langle x''', v\rangle, z, f'\rangle\rangle.\ \mathrm{pm}\ v\ \mathrm{as}\ \{\ \mathrm{true}.\ f'\langle\langle x, \mathrm{false}\rangle, z\rangle\ \ \mathrm{false}.\ \mathrm{diverge}\ \}\right)\langle Z, \langle x'', \langle x, \mathrm{true}\rangle, z, h\rangle\rangle$

ULTIMATE PATTERN MATCHING THEOREM

Given $\vec{X}, \overrightarrow{x:\Xi}, \overrightarrow{f:\neg A} \parallel \vec{Y} \vdash D$ and $\overrightarrow{Y \mapsto B}$

For any value $\vec{X}, \overrightarrow{x:\Xi}, \overrightarrow{f:\neg A[\overrightarrow{B/Y}]} \vdash^{v} V : D[\overrightarrow{B/Y}]$

there's a unique decomposition

$$V = p[\overrightarrow{B/Y}, w]$$

where $p \in \mathrm{upatt}(\vec{X}, \overrightarrow{x:\Xi} \parallel \vec{Y} \vdash D)$

and w is a filling.

THEOREMS

- It's a congruence
- The nat type is generic (Longo, Abramsky·Jagadeesan)

If $\vec{X}, X, \Gamma \vdash M, M'$

then $M[nat/X] \simeq M'[nat/X]$

implies $M \simeq M'$

- Type isomorphisms

Many examples

① $\Sigma X.\left(\neg(B \times X) \times A\right) \cong A[\neg_v X.\, B/X]$

A covariant, B contravariant in X

② $\Sigma X.\left(X^n \times A\right) \cong A[n/X]$

A contravariant in X

FURTHER WORK

- Full abstraction in the presence of state
- Characterize those strategies definable without state
- Hence obtain a fully complete game semantics

Very speculative

- Similar story for applicative bisimulation?