Characterizing recursive programs up to bisimilarity

Paul Blain Levy

University of Birmingham

August 21, 2010

Imperative Language, Deterministic

Syntax

$$M ::= \ \mathtt{print}\ c.\ M \mid \mathtt{x} \mid \mathtt{rec}\ \mathtt{x}.\ M \qquad\qquad c \in \mathcal{A}$$

We write $\Gamma \vdash M$, where Γ is a list of identifiers.

Imperative Language, Deterministic

Syntax

$$M ::= \texttt{print}\ c.\ M \mid \texttt{x} \mid \texttt{rec x.}\ M \qquad c \in \mathcal{A}$$

We write $\Gamma \vdash M$, where Γ is a list of identifiers.

Small-step semantics

$$\begin{array}{rcl} \texttt{print}\ c.\ M & \overset{c}{\leadsto} & M \\ \texttt{rec x.}\ M & \leadsto & M[\texttt{rec x.}\ M/\texttt{x}] \end{array}$$

Imperative Language, Deterministic

Syntax

$$M ::= \quad \texttt{print}\ c.\ M \mid \texttt{x} \mid \texttt{rec x.}\ M \qquad\qquad c \in \mathcal{A}$$

We write $\Gamma \vdash M$, where Γ is a list of identifiers.

Small-step semantics

$$\begin{array}{rcl} \texttt{print}\ c.\ M & \overset{c}{\rightsquigarrow} & M \\ \texttt{rec x.}\ M & \rightsquigarrow & M[\texttt{rec x.}\ M/\texttt{x}] \end{array}$$

A program either

- prints a finite string, then diverges
- or prints an infinite string.

Medium step semantics

Convergence

Define $M \stackrel{c}{\Rightarrow} N$ inductively:

$$\frac{}{\texttt{print}\ c.\ M \stackrel{c}{\Rightarrow} M} \qquad \frac{M[\texttt{rec x.}\ M/\texttt{x}] \stackrel{c}{\Rightarrow} N}{\texttt{rec x.}\ M \stackrel{c}{\Rightarrow} N}$$

Divergence

Define $M \Uparrow$ coinductively:

$$\frac{M[\texttt{rec x.}\ M/\texttt{x}] \Uparrow}{\texttt{rec x.}\ M \Uparrow}$$

Medium step semantics

Convergence

Define $M \overset{c}{\Rightarrow} N$ inductively:

$$\frac{}{\texttt{print}\ c.\ M \overset{c}{\Rightarrow} M} \qquad \frac{M[\texttt{rec x.}\ M/\texttt{x}] \overset{c}{\Rightarrow} N}{\texttt{rec x.}\ M \overset{c}{\Rightarrow} N}$$

Divergence

Define $M \Uparrow$ coinductively:

$$\frac{M[\texttt{rec x.}\ M/\texttt{x}] \Uparrow}{\texttt{rec x.}\ M \Uparrow}$$

We have

- $M \overset{c}{\Rightarrow} N$ iff $M \rightsquigarrow^* \overset{c}{\rightsquigarrow} N$
- $M \Uparrow$ iff $M \rightsquigarrow^\omega$

Denotational Semantics

Let $\mathrm{Streams}$ be the domain of finite and infinite streams of characters.

Then a term $\mathtt{x}, \mathtt{y}, \mathtt{z} \vdash M$ denotes a continuous function

$$[\![M]\!] : \mathrm{Streams}^3 \longrightarrow \mathrm{Streams}$$

Recursion is interpreted as least pre-fixed point.

Adding Nondeterminism

$$M ::= \ \mathtt{print}\ c.\ M \mid \mathtt{x} \mid \mathtt{rec\ x}.\ M \mid \mathtt{choose}\ \{M_n\}_{n\in\mathbb{N}}$$

$\mathtt{choose}\ \{M_n\}_{n\in\mathbb{N}}$ means: choose a number n, then execute M_n.

Adding Nondeterminism

$$M ::= \ \texttt{print}\ c.\ M \mid \texttt{x} \mid \texttt{rec}\ \texttt{x}.\ M \mid \texttt{choose}\ \{M_n\}_{n \in \mathbb{N}}$$

$\texttt{choose}\ \{M_n\}_{n \in \mathbb{N}}$ means: choose a number n, then execute M_n.

Denotational semantics?

Equivalence relations on closed terms P and Q

Infinite trace equivalence

$P \equiv Q$ when they have the same set of behaviours (divergences and infinite traces).
This implies they have the same finite traces.

Lower bisimilarity

Let $\mathcal{R}$ be a binary relation on closed terms.

It is a lower simulation when $M \; \mathcal{R} \; M'$ and $M \overset{c}{\Rightarrow} N$ implies $\exists N'$ such that $M' \overset{c}{\Rightarrow} N'$ and $N \; \mathcal{R} \; N'$.

It is a lower bisimulation when $\mathcal{R}$ and $\mathcal{R}^{\mathrm{op}}$ are lower simulations.

The greatest lower bisimulation is called $\eqsim$.

Other definitions of $\precsim$

Two programs are lower bisimilar

Other definitions of $\precsim$

Two programs are lower bisimilar

- iff they satisfy the same formulas in Hennessy-Milner logic

Other definitions of $\precsim$

Two programs are lower bisimilar

- iff they satisfy the same formulas in Hennessy-Milner logic
- iff there is a strategy for the bisimilarity game between them

Other definitions of $\precsim$

Two programs are lower bisimilar

- iff they satisfy the same formulas in Hennessy-Milner logic
- iff there is a strategy for the bisimilarity game between them
- iff they have the same anamorphic image.

Open extension of Infinite Trace Equivalence

Two terms $\mathrm{x}, \mathrm{y}, \mathrm{z} \vdash P \;\; \equiv^o \;\; Q$ when

Definition via substitution

$$P[M/\mathrm{x}, M'/\mathrm{y}, M''/\mathrm{z}] \equiv Q[M/\mathrm{x}, M'/\mathrm{y}, M''/\mathrm{z}]$$

for any closed terms M, M', M''.

Definition via operational meaning

they give the same function

$$(\mathcal{P}^{>0}\mathrm{Streams})^3 \longrightarrow (\mathcal{P}^{>0}\mathrm{Streams})$$

Open extension of lower bisimilarity

Two terms $\mathrm{x}, \mathrm{y}, \mathrm{z} \vdash P \ \precsim^o \ Q$ when

Definition via substitution

$$P[M/\mathrm{x}, M'/\mathrm{y}, M''/\mathrm{z}] \precsim Q[M/\mathrm{x}, M'/\mathrm{y}, M''/\mathrm{z}]$$

for any closed terms M, M', M''.

Definition via operational meaning

they give the same function

$$\mathrm{Proc}^3 \longrightarrow \mathrm{Proc}$$

where Proc is the set of programs modulo lower bisimilarity

i.e. a final coalgebra for $X \mapsto (\mathcal{P}^{(0,\aleph_0]}X)^{\mathcal{A}}$.

The Dream

Infinite trace equivalence

Can we give a denotational semantics for $\equiv^o$?
A term $\mathtt{x}, \mathtt{y}, \mathtt{z} \vdash P$ would denote a [...] function

$$(\mathcal{P}^{>0}\mathrm{Streams})^3 \longrightarrow (\mathcal{P}^{>0}\mathrm{Streams})$$

Recursion $\mathtt{rec}\ \mathtt{x}.M$ would denote the [...] fixpoint of $[\![M]\!]$.

Lower bisimilarity

Can we give a denotational semantics for $\precsim^o$?
A term $\mathtt{x}, \mathtt{y}, \mathtt{z} \vdash P$ would denote a [...] function

$$\mathrm{Proc}^3 \longrightarrow \mathrm{Proc}$$

Recursion $\mathtt{rec}\ \mathtt{x}.M$ would denote the [...] fixpoint of $[\![M]\!]$.

Nightmare on $\equiv$ street

Here are two terms with a free identifier x.

$$\begin{array}{rcl} N & = & \mathtt{choose}^{\perp}\ n \in \mathbb{N}.\ \checkmark^{n}.\ \perp \ \mathtt{or}\ \mathtt{x} \\ N' & = & \mathtt{choose}^{\perp}\ n \in \mathbb{N}.\ \checkmark^{n}.\ \perp \ \mathtt{or}\ \mathtt{x}\ \mathtt{or}\ \checkmark.\mathtt{x} \end{array}$$

	$\checkmark^{n}$, then diverge	$\checkmark^{\omega}$
N	yes	iff x can
N'	yes	iff x can
rec x. N	yes	no
rec x. N'	yes	yes

Same endofunction, different fixpoint.

Encouraging situation on $\precapprox$ street

We know that $\precapprox^o$ is a congruence.

Encouraging situation on $\approx$ street

We know that $\approx^o$ is a congruence.

Proof by Howe's method

Encouraging situation on $\precapprox$ street

We know that $\precapprox^o$ is a congruence.

Proof by Howe's method i.e. magic.

Encouraging situation on $\eqsim$ street

We know that $\eqsim^o$ is a congruence.

Proof by Howe's method i.e. magic.

For a term $\mathtt{x} \vdash N$, the endofunction determines the fixpoint $\mathtt{rec}\ \mathtt{x}.N$.

Encouraging situation on $\eqsim$ street

We know that $\eqsim^o$ is a congruence.

Proof by Howe's method i.e. magic.

For a term $\mathtt{x} \vdash N$, the endofunction determines the fixpoint $\mathtt{rec}\ \mathtt{x}.N$.

But how is that fixpoint obtained?

What is the structure of Proc?

Nested simulation

A 2-nested lower simulation is a simulation contained in mutual similarity.

Nested simulation

A 2-nested lower simulation is a simulation contained in mutual similarity.

Characterized by

- Hennessy-Milner logic with one alternation
- Bisimulation game with one change of side
- Final coalgebra for suitable endofunctor.

A 3-nested lower simulation is a simulation contained in mutual 2-nested similarity. And so through all countable ordinals.

The intersection of n-nested similarity for $n < \omega_1$ is bisimilarity.

An ω_1-nested preordered set is a set X with a sequence of preorders $(\leqslant_\alpha)_{\alpha \leqslant \omega_1}$ where

- $(\leqslant_{\alpha+1}$ is contained in the symmetrization of $(\leqslant_\alpha)$
- at a limit ordinal, it's the intersection of the previous ones (hence symmetric).

Example: programs, ordered by α-nested simulation

An ω_1-nested preordered set is a set X with a sequence of preorders $(\leqslant_\alpha)_{\alpha \leqslant \omega_1}$ where

- $(\leqslant_{\alpha+1}$ is contained in the symmetrization of $(\leqslant_\alpha)$
- at a limit ordinal, it's the intersection of the previous ones (hence symmetric).

Example: programs, ordered by α-nested simulation

It's an ω_1-nested poset when $\leqslant_{\omega_1}$ is discrete.

Example: Proc

A function between these is monotone when it preserves all the preorders.

Calculating the nesting fixpoint

Suppose f is a monotone endofunction on an ω_1-nested poset $(X, \leqslant)$.

Calculating the nesting fixpoint

Suppose f is a monotone endofunction on an ω_1-nested poset $(X, \leqslant)$.

Obtain a decreasing sequence of sets $(U_\alpha)_{\alpha \leqslant \omega_1}$.

We put $U_0 = X$.

Calculating the nesting fixpoint

Suppose f is a monotone endofunction on an ω_1-nested poset $(X, \leqslant)$.

Obtain a decreasing sequence of sets $(U_\alpha)_{\alpha \leqslant \omega_1}$.

We put $U_0 = X$.

Then U_1 is the set of least pre-fixed points of f wrt $(\leqslant_1)$
—Might be empty

Calculating the nesting fixpoint

Suppose f is a monotone endofunction on an ω_1-nested poset $(X, \leqslant)$.

Obtain a decreasing sequence of sets $(U_\alpha)_{\alpha \leqslant \omega_1}$.

We put $U_0 = X$.

Then U_1 is the set of least pre-fixed points of f wrt $(\leqslant_1)$
—Might be empty

f restricts to an endofunction on U_1.
So U_2 is the set of least pre-fixed points of $f \upharpoonright U_1$ wrt $(\leqslant_2)$
—Might be empty

Calculating the nesting fixpoint

Suppose f is a monotone endofunction on an ω_1-nested poset $(X, \leqslant)$.

Obtain a decreasing sequence of sets $(U_\alpha)_{\alpha \leqslant \omega_1}$.

We put $U_0 = X$.

Then U_1 is the set of least pre-fixed points of f wrt $(\leqslant_1)$
—Might be empty

f restricts to an endofunction on U_1.
So U_2 is the set of least pre-fixed points of $f \upharpoonright U_1$ wrt $(\leqslant_2)$
—Might be empty

At a limit ordinal we take the intersection of the previous sets.
—Might be empty

Calculating the nesting fixpoint

Suppose f is a monotone endofunction on an ω_1-nested poset $(X, \leqslant)$.

Obtain a decreasing sequence of sets $(U_\alpha)_{\alpha \leqslant \omega_1}$.

We put $U_0 = X$.

Then U_1 is the set of least pre-fixed points of f wrt $(\leqslant_1)$
—Might be empty

f restricts to an endofunction on U_1.
So U_2 is the set of least pre-fixed points of $f \upharpoonright U_1$ wrt $(\leqslant_2)$
—Might be empty

At a limit ordinal we take the intersection of the previous sets.
—Might be empty

Then U_{ω_1} is a singleton set —Or empty
That's the nesting fixpoint.

Towards denotational semantics

$\texttt{rec x}$. M is the nesting fixpoint of $N \mapsto M[N/\texttt{x}]$.

But not every monotone endofunction has a nesting fixpoint.

Can we restrict to a class of functions to guarantee existence?

Maybe exploratory functions (Levy and Weldemariam, MFPS 2009)?

Functional language: call-by-name FPC

Types

$$A ::= \quad A \to A \mid \sum_{i \in I} A_i \mid \prod_{i \in I} A_i \mid \mathtt{X} \mid \mathtt{rec}\ \mathtt{X}.\ A \quad (I \text{ countable})$$

Terms

$$\begin{array}{rcl} M & ::= & \mathtt{x} \mid \langle i, M \rangle \mid \mathtt{case}\ M\ \mathtt{of}\ \{\langle i, \mathtt{x} \rangle . M_i\}_{i \in I} \\ & & \mid \lambda \mathtt{x}.M \mid MM \mid Mi \mid \lambda\{i.M_i\}_{i \in I} \\ & & \mid \mathtt{rec}\ \mathtt{x}.\ M \mid \mathtt{fold}\ M \mid \mathtt{unfold}\ M \\ & & \mid \mathtt{choose}\ \{M_n\}_{n \in \mathbb{N}} \end{array}$$

Big-step semantics

$$\begin{array}{ll} M \Downarrow T & \text{inductively defined} \\ M \Uparrow & \text{coinductively defined} \end{array}$$

Applicative bisimilarity [Abramsky]

A binary relation $\mathcal{R}$ on closed terms is a lower applicative simulation when $M \mathrel{\mathcal{R}} M' : A$ implies

- (if $A = B \to C$) for all closed $N : B$ we have $MN \mathrel{\mathcal{R}} M'N$
- (if $A = \prod_{i \in I} B_i$) for all $i \in I$ we have $Mi \mathrel{\mathcal{R}} M'i$
- (if $A = \sum_{i \in I} A_i$) if $M \Downarrow \langle i, N \rangle$ then $\exists N'$ such that $M' \Downarrow \langle i, N' \rangle$ and $N \mathrel{\mathcal{R}} N'$.

The largest is applicative bisimilarity.

Summary of results

For both languages, Howe's method shows that $\precapprox^o$ is a congruence.

Imperative language

- $\mathtt{rec}\ \mathtt{x}.M$ is nesting fixpoint of $N \mapsto M[N/\mathtt{x}]$
- This implies $\precapprox^o$ is a congruence

Functional language

- $\mathtt{rec}\ \mathtt{x}.M$ is a nesting fixpoint of $N \mapsto M[N/\mathtt{x}]$
- This does not imply $\precapprox^o$ is a congruence.
- We would also need to show application preserves $\precapprox$.