Towards Probabilistic Testing of Lambda Terms

Simona Kašterović[1]
joint work with: Michele Pagani[2]

[1] Faculty of Technical Sciences, University of Novi Sad
[2] Institut de Recherche en Informatique Fondamentale, Universite Paris Diderot - Paris 7

LAP, September 2018

Probabilistic lambda calculus $\Lambda_{\oplus}$

$$M, N ::= x \mid \lambda x.M \mid MN \mid M \oplus N$$

Call-by-Name

$$(\lambda x.M)N \to M\{N/x\}$$

Context Equivalence

Testing Equivalence

Bisimilarity

Call-by-Value

$$(\lambda x.M)V \to M\{V/x\}$$

Context Equivalence

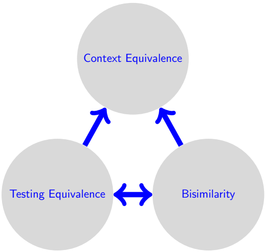
Testing Equivalence

Bisimilarity

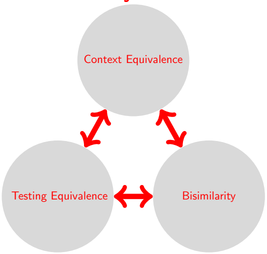
Probabilistic lambda calculus $\Lambda_\oplus$

$$M, N ::= x \mid \lambda x.M \mid MN \mid M \oplus N$$

Call-by-Name



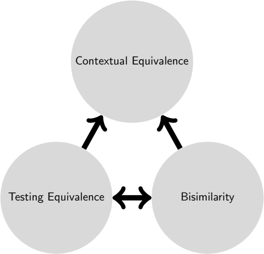
Call-by-Value



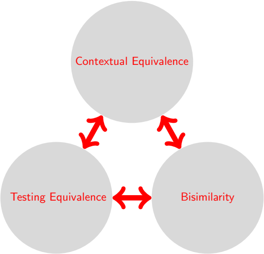
U. Dal Lago, D. Sangiorgi M. Alberti, On Coinductive Equivalences for Higher-Order Probabilistic Functional Programs, *In 41st International Symposium on Principles of Programming Languages*, Proceedings, pages 297-308. 2014.

R. Crubill, U. Dal Lago, On Probabilistic Applicative Bisimulation and Call-by-Value Lambda-Calculi, *In Programming Languages and Systems, 23rd European Symposium on Programming*, Proceedings, volume 8410 of Lecture Notes in Computer Science, pages 209-228. Springer, 2014.

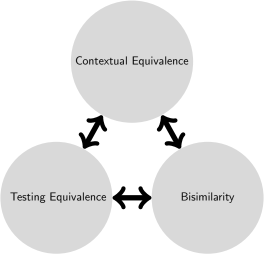
Call-by-Name



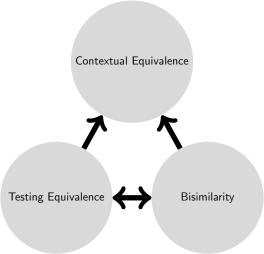
Call-by-Name + ?



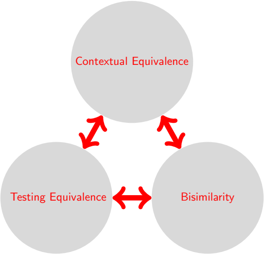
Call-by-Value



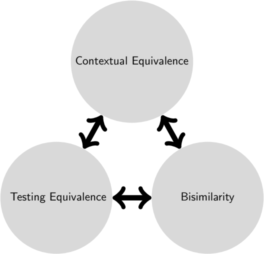
Call-by-Name



Call-by-Name + "let ... in ..."



Call-by-Value



1. Probabilistic lambda calculus $\Lambda_{\oplus,\mathsf{let}}$
2. Probabilistic lambda calculus as LMC
3. Testing
4. Full Abstraction

Probabilistic Lambda Calculus $\Lambda_{\oplus,\mathsf{let}}$

Syntax

$$M, N ::= x \mid \lambda x.M \mid MN \mid M \oplus N \mid \mathsf{let}\ x = M\ \mathsf{in}\ N$$

Values ($V\Lambda_{\oplus,\mathsf{let}}$):
- variables
- abstractions

Reduction

$$\begin{aligned}
(\lambda x.M)N &\to M\{N/x\} \\
\mathsf{let}\ x = V\ \mathsf{in}\ M &\to M\{V/x\} \\
M \oplus N &\to M, N \\
MN &\to LN, \text{ if } M \to L \\
\mathsf{let}\ x = M\ \mathsf{in}\ N &\to \mathsf{let}\ x = M'\ \mathsf{in}\ N, \text{ if } M \to M'
\end{aligned}$$

Operational semantics

- value distribution $\mathcal{D} : V\Lambda_{\oplus,\mathsf{let}} \to \mathbb{R}_{[0,1]}$, $\quad \sum_V \mathcal{D}(V) \leqslant 1$

$$\frac{}{M \Downarrow \emptyset} \qquad \frac{}{V \Downarrow \{V^1\}}$$

$$\frac{N \Downarrow \mathcal{G} \qquad \{M\{V/x\} \Downarrow \mathcal{H}_V\}_{V \in \mathcal{S}(\mathcal{G})}}{\mathsf{let}\ x = N\ \mathsf{in}\ M \Downarrow \sum_{V \in \mathcal{S}(\mathcal{G})} \mathcal{G}(V)\mathcal{H}_V} \qquad \frac{M \Downarrow \mathcal{D} \qquad N \Downarrow \mathcal{E}}{M \oplus N \Downarrow \frac{1}{2}\mathcal{D} + \frac{1}{2}\mathcal{E}}$$

$$\frac{M \Downarrow \mathcal{D} \qquad \{P\{N/x\} \Downarrow \mathcal{E}_{P,N}\}_{\lambda x.P \in \mathcal{S}(\mathcal{D})}}{MN \Downarrow \sum_{\lambda x.P \in \mathcal{S}(\mathcal{D})} \mathcal{D}(\lambda x.P)\mathcal{E}_{P,N}}$$

$$[\![M]\!] = \sup_{M \Downarrow \mathcal{D}} \mathcal{D}$$

$$\sum [\![M]\!] = \sum_{V \in V\Lambda_{\oplus,\mathsf{let}}} [\![M]\!](V)$$

Context Equivalence

Context

$C ::= [\cdot] \mid \lambda x.C \mid CM \mid MC \mid C \oplus M \mid M \oplus C \mid \text{let } x = C \text{ in } M$
$\mid \text{let } x = M \text{ in } C.$

Definition (Context Equivalence)

Terms M and N are context equivalent ($M \simeq N$) if for every context C, it holds $\sum[\![C[M]]\!] = \sum[\![C[N]]\!]$.

Example

$$\lambda x.\lambda y.(x \oplus y) \not\simeq (\lambda x.\lambda y.x) \oplus (\lambda x.\lambda y.y)$$

- $\lambda x.\lambda y.(x \oplus y)$ and $(\lambda x.\lambda y.x) \oplus (\lambda x.\lambda y.y)$ **are context equivalent in** $\Lambda_\oplus$ (probabilistic lambda calculus without let ... in operator).

Probabilistic lambda calculus as LMC

Labelled Markov Chain: $(\Lambda(\emptyset) \uplus V\Lambda(\emptyset), \{\tau\} \cup \Lambda(\emptyset), P)$

- states: $\Lambda(\emptyset) \uplus V\Lambda(\emptyset)$
- actions: $\{\tau\} \cup \Lambda(\emptyset)$
- to distinguish a value $\lambda x.M$, we indicate it with $\nu x.M$

Transition probability matrix P

- for term M and distinguished value $\nu x.N$,

$$P(M, \tau, \nu x.N) = [\![M]\!](\lambda x.N),$$

- for term M and distinguished value $\nu x.N$,

$$P(\nu x.N, M, N\{M/x\}) = 1,$$

- in all other cases, P returns 0.

Example

$$\lambda x.\lambda y.x \oplus y \xrightarrow{\tau} \nu x.\lambda y.x \oplus y \xrightarrow{I} \lambda y.I \oplus y \xrightarrow{\tau} \nu y.I \oplus y \xrightarrow{\Omega} I \oplus \Omega \xrightarrow[\frac{1}{2}]{\tau} \nu x.x$$

$$\lambda x.\lambda y.x \oplus \lambda x.\lambda y.y \xrightarrow[\frac{1}{2}]{\tau} \nu x.\lambda y.x \xrightarrow{I} \lambda y.I \xrightarrow{\tau} \nu y.I \xrightarrow{\Omega} I \xrightarrow{\tau} \nu x.x$$

$$\lambda x.\lambda y.x \oplus \lambda x.\lambda y.y \xrightarrow[\frac{1}{2}]{\tau} \nu x.\lambda y.y \xrightarrow{I} \lambda y.y \xrightarrow{\tau} \nu y.y \xrightarrow{\Omega} \Omega$$

Bisimilarity

- let $(\mathcal{S}, \mathcal{L}, P)$ be a labelled Markov Chain;
- let $\mathcal{R}$ be a preorder relation on $\mathcal{S}$;

Probabilistic simulation

$\mathcal{R}$ is a *probabilistic simulation* if:
$(s, t) \in \mathcal{R}$ implies $P(s, l, X) \leqslant P(t, l, \mathcal{R}(X))$, for every $l \in \mathcal{L}$ and $X \subseteq \mathcal{S}$.

Similarity: $M \lesssim N$

Probabilistic bisimulation

$\mathcal{R}$ is *a probabilistic bisimulation* if $\mathcal{R}$ and $\mathcal{R}^{-1}$ are probabilistic simulations .

Bisimilarity: $M \approx N$ $\approx \, = \, \lesssim \cap (\lesssim)^{-1}$

Example

$$\begin{tikzcd}[row sep=small]
\lambda x.\lambda y.x \oplus y \arrow[d, "\tau"] \\
\nu x.\lambda y.x \oplus y \arrow[d, "I"] \\
\lambda y.I \oplus y \arrow[d, "\tau"] \\
\nu y.I \oplus y \arrow[d, "\Omega"] \\
I \oplus \Omega \arrow[d, "\frac{1}{2}" ', "\tau"] \arrow[dr, dashed] & \\
\nu x.x &
\end{tikzcd}
\qquad
\begin{tikzcd}[row sep=small]
& \lambda x.\lambda y.x \oplus \lambda x.\lambda y.y \arrow[dl, "\frac{1}{2}" ', "\tau"] \arrow[dr, "\frac{1}{2}" ', "\tau"] & \\
\nu x.\lambda y.x \arrow[d, "I"] & & \nu x.\lambda y.y \arrow[d, "I"] \\
\lambda y.I \arrow[d, "\tau"] & & \lambda y.y \arrow[d, "\tau"] \\
\nu y.I \arrow[d, "\Omega"] & & \nu y.y \arrow[d, "\Omega"] \\
I \arrow[d, "\tau"] & & \Omega \\
\nu x.x & &
\end{tikzcd}$$

$$(\lambda x.\lambda y.(x \oplus y)) \not\approx ((\lambda x.\lambda y.x) \oplus (\lambda x.\lambda y.y))$$

Testing

- let $(\mathcal{S}, \mathcal{L}, \mathcal{P})$ be a labelled Markov Chain

Testing language

$$t ::= \omega \mid a.t \mid (t, t)$$

- ω- a symbol for termination
- $a \in \mathcal{L}$

Definition ($P_t(\cdot) : \mathcal{S} \to \mathbb{R}_{[0,1]}$)

$$\begin{aligned}
P_\omega(M) &= 1 \\
P_{a.t}(M) &= \sum_{M'} P(M, a, M') P_t(M') \\
P_{(t,s)}(M) &= P_t(M) \cdot P_s(M)
\end{aligned}$$

Example

$$\lambda x.\lambda y.x \oplus y \xrightarrow{\tau} \nu x.\lambda y.x \oplus y \xrightarrow{I} \lambda y.I \oplus y \xrightarrow{\tau} \nu y.I \oplus y \xrightarrow{\Omega} I \oplus \Omega \xrightarrow{\frac{1}{2}\,\tau} \nu x.x$$

$$\lambda x.\lambda y.x \oplus \lambda x.\lambda y.y \xrightarrow{\frac{1}{2}\,\tau} \nu x.\lambda y.x \xrightarrow{I} \lambda y.I \xrightarrow{\tau} \nu y.I \xrightarrow{\Omega} I \xrightarrow{\tau} \nu x.x$$

$$\lambda x.\lambda y.x \oplus \lambda x.\lambda y.y \xrightarrow{\frac{1}{2}\,\tau} \nu x.\lambda y.y \xrightarrow{I} \lambda y.y \xrightarrow{\tau} \nu y.y \xrightarrow{\Omega} \Omega$$

Test: $t = \tau.(I.\tau.\Omega.\tau.\omega, I.\tau.\Omega.\tau.\omega)$

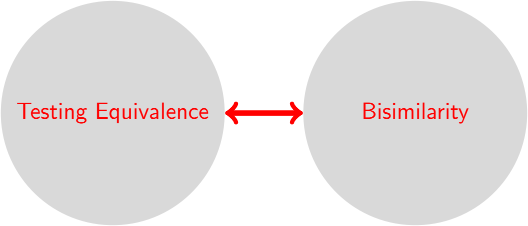
$P_t(\lambda x.\lambda y.x \oplus y) = \frac{1}{4}$;

$P_t(\lambda x.\lambda y.x \oplus \lambda x.\lambda y.y) = \frac{1}{2}$.

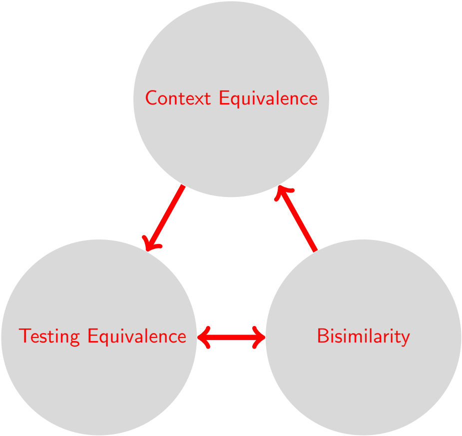
$\Lambda_{\oplus,\text{let}}$ as labelled Markov chain: $(\Lambda(\emptyset) \uplus V\Lambda(\emptyset), \{\tau\} \cup \Lambda(\emptyset), P)$

Theorem

$M \approx N$ if and only if $P_t(M) = P_t(N)$, for every test t.



F. van Breugel, M. W. Mislove, J. Ouaknine, and J. Worrell. Domain theory, testing and simulation for labelled markov processes. *Theor. Comput. Sci.*, 333(1-2):171197, 2005.


Context Equivalence
Testing Equivalence
Bisimilarity

Bisimilarity implies Context Equivalence

- Howe's Method;
- $\lesssim \dashrightarrow (\lesssim)^H$ Howe's lifting;
- $(\lesssim)^H \dashrightarrow ((\lesssim)^H)^+$ reflexive and transitive closure;
- $((\lesssim)^H)^+$ is precongruence and $((\lesssim)^H)^+ = \lesssim$;
- $\approx$ is a congruence.

Theorem

Let M and N be terms in $\Lambda_{\oplus,\mathsf{let}}$. If $M \approx N$, then $\sum[\![C[M]]\!] = \sum[\![C[N]]\!]$, for every context C.

Context Equivalence implies Testing Equivalence

- every test has an equivalent context

Lemma

Let $t \in \mathcal{T}$ be a test. There are contexts C and D, such that for every term M and value V, $P_t(M) = \sum[\![C[M]]\!]$ and $P_t(V) = \sum[\![D[V]]\!]$.

- context equivalence implies testing equivalence

Theorem

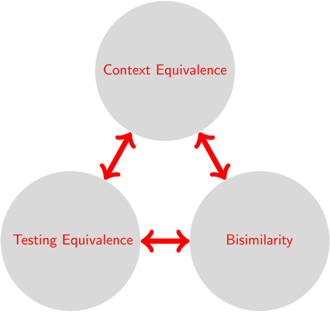
Let M and N be terms in $\Lambda_{\oplus,\mathrm{let}}$. If $\sum[\![C[M]]\!] = \sum[\![C[N]]\!]$, for every context C, then $P_t(M) = P_t(N)$ for every test $t \in \mathcal{T}$.

To conclude

- $\Lambda_{\oplus,\mathsf{let}}$:

  call-by-name probabilistic lambda calculus
  + "let … in …",

- full abstraction

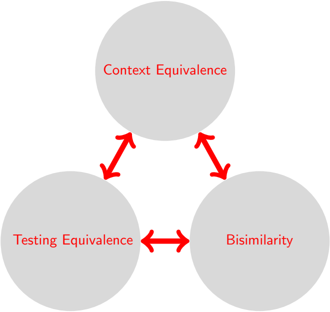


To conclude

- $\Lambda_{\oplus,\mathsf{let}}$:

  call-by-name probabilistic lambda calculus
  + "let ... in ...",

- full abstraction



Thank you!