Contextual Equivalence in a Probabilistic Call-by-Need Lambda-Calculus

David Sabel   Manfred Schmidt-Schauß   Luca Maio

PPDP 2022
22nd September 2022, Tiblisi, Georgia

Motivation and Goals

Probabilistic Programming + **Functional Programming** + **Call-by-Need Evaluation**

Probabilistic Programming

- programs express probabilistic models
- evaluation results in (multi-)distributions
- apply correct program transformations

Functional Programming

- declarative, high-level and generic programming
- clean (mathematical) definition
- equational reasoning

Call-by-Need Evaluation

- declarative: only needed bindings are evaluated
- efficient implementation of lazy evaluation
- in the probabilistic setting: different from call-by-name and call-by-value

A lot of related work on probabilistic lambda calculi with call-by-name or call-by-value evaluation
(see Ugo Dal Lago: *On Probabilistic Lambda-Calculi*, 2020)

→ **Investigate the semantics of a probabilistic call-by-need functional language**

Probabilistic Calculi and Call-by-Name, Call-by-Value, Call-By-Need

possible evaluation results

	call-by-name	**call-by-need**	**call-by-value**
$(\lambda y.1)\ \bot$	1	1	diverges
$(\lambda x.x + x)\ (1\ \oplus\ 2)$	2,3,4	2 and 4	2 and 4

where **probabilistic choice** $(1\ \oplus\ 2)$ means:
randomly choose between 1 and 2

The Calculus $L_{need,\oplus}$

Expressions and environments:

$s, t, r \in Exp ::= x \mid \lambda x.s \mid (s\ t) \mid (s \oplus t) \mid \texttt{let}\ env\ \texttt{in}\ s \qquad env ::= x = s \mid x = s, env$

Reduction contexts:

$A \in \mathbb{A} ::= [\cdot] \mid (A\ s)$

$R \in \mathbb{R} ::= A \mid \texttt{let}\ env\ \texttt{in}\ A \mid \texttt{let}\ env, x_1 = A_1[x_2], \ldots, x_n = A_n[y], y = A\ \texttt{in}\ A[x_1]$

Small-step operational semantics: standard reduction relation $\xrightarrow{sr}$ defined by

$(sr,lbeta)\ \ R[((\lambda x.s)\ t)] \to R[\texttt{let}\ x = t\ \texttt{in}\ s]$

$(sr,cp\text{-}in)\ \ \texttt{let}\ x_1 = x_2, \ldots, x_{n-1} = x_n, x_n = \lambda y.s, env\ \texttt{in}\ A[x_1]$
$\quad \to \texttt{let}\ x_1 = x_2, \ldots, x_{n-1} = x_n, x_n = \lambda y.s, env\ \texttt{in}\ A[\lambda y.s]$

$$\left.\begin{array}{ll}(sr,probl) & R[s \oplus t] \to R[s] \\ (sr,probr) & R[s \oplus t] \to R[t]\end{array}\right\}\ \textit{prob-reductions}$$

$\ldots$

Evaluation results: weak head normal forms (WHNFs) $\lambda x.s$, $\texttt{let}\ env\ \texttt{in}\ \lambda x.s$

Tracking Probabilities

Weighted expression (p, s) with rational number $p \in (0, 1]$ and expression s

Weighted standard reduction step $\xrightarrow{wsr}$

$$(p, s) \xrightarrow{wsr,a} \begin{cases} (p, t) & \text{iff } s \xrightarrow{sr,a} t \text{ and } a \notin \{probl, probr\} \\ \left(\frac{p}{2}, t\right) & \text{iff } s \xrightarrow{sr,a} t \text{ and } a \in \{probl, probr\} \end{cases}$$

$\xrightarrow{wsr,*}$ denotes the reflexive-transitive closure of $\xrightarrow{wsr}$

Evaluation

An **evaluation** of (p, s) is a sequence $(p, s) \xrightarrow{wsr,*} (q, t)$ where t is a WHNF.
Eval(p, s) = set of all evaluations starting with (p, s)

Notation: $(p, s) \wr_L (q, t) \in \mathit{Eval}(p, s)$ where L = sequence of labels of prob-reductions

Expected Convergence

Expected convergence

$$\mathrm{ExCv}(s) = \sum_{(1,s) \wr_L (q,t) \in \mathit{Eval}(1,s)} q.$$

"$=$ probability that evaluation of s ends with a WHNF"

Examples

$$\begin{aligned}
&\mathrm{ExCv}(\Omega) = 0\\
&\mathrm{ExCv}(\Omega \oplus K) = 0.5\\
&\mathrm{ExCv}(\texttt{let}\ x = (x \oplus K)\ \texttt{in}\ x) = 0.5\\
&\mathrm{ExCv}(\texttt{let}\ x = (\lambda y.(x\ I) \oplus K)\ \texttt{in}\ (x\ I)) = \tfrac{1}{2} + \tfrac{1}{4} + \tfrac{1}{8} + \ldots = 1
\end{aligned}$$

where

$$\Omega := (\lambda x.x\ x)\ (\lambda x.x\ x) \qquad K := \lambda x.\lambda y.x \qquad I := \lambda x.x$$

Contextual Equivalence

Contexts: $C ::= [\cdot] \mid \lambda x.C \mid (C\ t) \mid (t\ C) \mid (C \oplus t) \mid (t \oplus C) \mid \texttt{let}\ env\ \texttt{in}\ C \mid \texttt{let}\ env, y = C\ \texttt{in}\ t$

Contextual Preorder and Equivalence

- **contextual preorder** $s \leq_c t$ iff $\forall C$: $\mathrm{ExCv}(C[s]) \leq \mathrm{ExCv}(C[t])$

  "in any context: t converges at least as often as s"

- **contextual equivalence** $s \sim_c t$ iff $s \leq_c t \wedge t \leq_c s$

Refuting equivalences requires one context acting as counter-example

Example:

$K \oplus I \not\sim_c K$ since for for $C = [\cdot]\ (\lambda z.z)\ \Omega$:

- $\mathrm{ExCv}(C[K]) = 1$, but
- $\mathrm{ExCv}(C[K \oplus I]) = 0.5$

Proving equivalences is **harder** due to the quantification over all contexts.

Program Transformations

A program transformation T (=binary relation on expressions) is **correct** iff $\xrightarrow{T} \subseteq \sim_c$

Some Correct Program Transformations

$(lbeta)$ $((\lambda x.s)\ t) \rightarrow \texttt{let}\ x = t\ \texttt{in}\ s$
$(lapp)$ $((\texttt{let}\ env\ \texttt{in}\ s)\ t) \rightarrow \texttt{let}\ env\ \texttt{in}\ (s\ t)$
$(llet)$ $\texttt{let}\ env_1\ \ldots\ \texttt{let}\ env_2\ \ldots \rightarrow \texttt{let}\ env_1, env_2\ \ldots$
(cp) $\texttt{let}\ x = \lambda y.s, \ldots C[x] \ldots \rightarrow \texttt{let}\ x = \lambda y.s, \ldots C[\lambda y.s] \ldots$
(ucp) $\texttt{let}\ x = t\ \ldots\ S[x]\ \ldots \rightarrow \texttt{let}\ \ldots\ S[t]\ \ldots$, if x occurs only in $S[x]$
(gc) $\texttt{let}\ env\ \texttt{in}\ s \rightarrow s$, if bindings of env are not used in env', s
$(probid)$ $s \oplus s \rightarrow s$ $\quad$ $(probdistr)$ $r \oplus (s \oplus t) \rightarrow (r \oplus s) \oplus (r \oplus t)$
$(probcomm)$ $s \oplus t \rightarrow t \oplus s$ $\quad$ $(probreorder)$ $(s_1 \oplus s_2) \oplus (t_1 \oplus t_2) \rightarrow (s_1 \oplus t_1) \oplus (s_2 \oplus t_2)$

- Proving correctness requires proof tools and techniques:
  we provide a context lemma, a diagram technique, a "same distribution"-criterion

Context Lemma

Context Lemma

If $\forall k \geq 0$, for all reduction contexts R, $\exists d \geq 0 : \mathrm{ExCv}(R[s], k) \leq \mathrm{ExCv}(R[t], k + d)$, then $s \leq_c t$.

"$\leq_c$ holds if expected convergence (with bounded number of prob-reduction) is never decreased in any ***reduction context*** *(and for any bound) "*

- where $\mathrm{ExCv}(r, k) = \sum_{(1,r) \rightsquigarrow_L (q,r') \in \mathit{Eval}(1,r),\ |L| \leq k} q$

  (probability to converge using not more than k prob-reductions)
- In the paper: a more general context lemma using multiple expressions and multi-contexts.

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.

If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \wr_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \wr_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.

If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \rightsquigarrow_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \rightsquigarrow_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Preservation of evaluations and prob-sequences can be shown by the diagram method:

$$\begin{tikzcd} s \arrow[r, "X{,}T"] \arrow[d, "sr{,}*"'] & t \\ \text{WHNF} & \end{tikzcd}$$

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.

If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \rightsquigarrow_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \rightsquigarrow_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Preservation of evaluations and prob-sequences can be shown by the diagram method:

$$\begin{tikzcd} s \arrow[r, "X{,}T"] \arrow[d, "sr{,}k"'] & t \arrow[d, dashed, "sr{,}k'"] \\ \mathrm{WHNF} & \mathrm{WHNF} \end{tikzcd}$$

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.
If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \rightsquigarrow_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \rightsquigarrow_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Preservation of evaluations and prob-sequences can be shown by the diagram method:

$$\begin{tikzcd}
s \arrow[r, "X{,}T"] & t \arrow[d, dashed, "sr{,}k'"] \\
\text{WHNF} & \text{WHNF}
\end{tikzcd}$$

base case: s is a WHNF

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.

If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \rightsquigarrow_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \rightsquigarrow_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Preservation of evaluations and prob-sequences can be shown by the diagram method:

$$\begin{tikzcd} s \arrow[r, "{X,T}"] \arrow[d, "sr"'] & t \arrow[dd, dashed, "{sr,k'}"] \\ s' \arrow[d, "{sr,k-1}"'] & \\ \mathrm{WHNF} & \mathrm{WHNF} \end{tikzcd}$$

step: evaluation has length > 0

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.
If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \wr_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \wr_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Preservation of evaluations and prob-sequences can be shown by the diagram method:

$$\begin{tikzcd}
s \arrow[r, "X{,}T"] \arrow[d, "sr"'] & t \arrow[d, dashed, "sr{,}*"] \\
s' \arrow[r, dashed, "X{,}T{,}*"'] \arrow[d, "sr{,}k-1"'] & t' \\
\text{WHNF} & \text{WHNF}
\end{tikzcd}$$

apply a forking diagram to the first step

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.

If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \rightsquigarrow_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \rightsquigarrow_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Preservation of evaluations and prob-sequences can be shown by the diagram method:

$$\begin{tikzcd}
s \arrow[r, "X{,}T"] \arrow[d, "sr"'] & t \arrow[d, dashed, "sr{,}*"] \\
s' \arrow[r, dashed, "X{,}T{,}*"'] \arrow[d, "sr{,}k-1"'] & t' \arrow[d, squiggly, "sr{,}*"] \\
\text{WHNF} & \text{WHNF}
\end{tikzcd}$$

use induction

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.

If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \leadsto_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \leadsto_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Preservation of evaluations and prob-sequences can be shown by the diagram method:

$$\begin{tikzcd}
s \arrow[r, "X{,}T"] \arrow[d, "sr"'] & t \arrow[d, dashed, "sr{,}*"] \\
s' \arrow[r, dashed, "X{,}T{,}*"'] \arrow[d, "sr{,}k-1"'] & t' \arrow[d, squiggly, "sr{,}*"] \\
\text{WHNF} & \text{WHNF}
\end{tikzcd}$$

reverse direction:

symmetric using commuting diagrams:

$$\begin{tikzcd}
s \arrow[r, "X{,}T"] \arrow[d, dashed, "sr{,}*"'] & t \arrow[d, "sr"] \\
s' \arrow[r, dashed, "X{,}T{,}*"'] \arrow[d, squiggly, "sr{,}*"'] & t' \arrow[d, "sr{,}k-1"] \\
\text{WHNF} & \text{WHNF}
\end{tikzcd}$$

Correctness by Diagrams and Same Prob-Sequences

Correctness Criterion: Same Prob-Sequences

Let $\xrightarrow{X,T}$ the closure of $\xrightarrow{T}$ by reduction or surface contexts.

If for all $s \xrightarrow{X,T} t$:

for all evaluations $s \rightsquigarrow_L s' \in \mathit{Eval}(s)$ there exists an evaluation $t \rightsquigarrow_L t' \in \mathit{Eval}(t)$

then $\xrightarrow{T} \subseteq \leq_c$ holds.

Preservation of evaluations and prob-sequences can be shown by the diagram method:

$$\begin{tikzcd}
s \arrow[r, "X{,}T"] \arrow[d, "sr"'] & t \arrow[d, dashed, "sr{,}*"] \\
s' \arrow[r, dashed, "X{,}T{,}*"'] \arrow[d, "sr{,}k-1"'] & t' \arrow[d, squiggly, "sr{,}*"] \\
\mathrm{WHNF} & \mathrm{WHNF}
\end{tikzcd}$$

same prob-sequences:
check the base case and the diagrams

$$\begin{tikzcd}
s \arrow[r, "X{,}T"] \arrow[d, dashed, "sr{,}*"'] & t \arrow[d, "sr"] \\
s' \arrow[r, dashed, "X{,}T{,}*"'] \arrow[d, squiggly, "sr{,}*"'] & t' \arrow[d, "sr{,}k-1"] \\
\mathrm{WHNF} & \mathrm{WHNF}
\end{tikzcd}$$

Diagram Method: Automation

diagram method requires ...	**can be automated by ...**
computing overlaps between transformation steps and standard reductions	unification of lhs/rhs of transformations and reductions [Schmidt-Schauß, S. 2015]
joining the overlaps	symbolic reduction and α-renaming [S. 2017]
treating the base cases	similar to diagram computation, unification of lhs/rhs of transformations with WHNF
inductive proof using the diagrams	encode diagrams as term rewrite system and prove (innermost) termination [Rau, S., Schmidt-Schauß 2012]
preservation of prob-reductions	easy inspection of base cases and diagrams

- our LRSX-tool [S. 2018] can do all these steps (using external termination provers AProVE and TTT2 and certifier CeTA)
- we obtained correctness for (lapp),(llet),(cp),(ucp),(gc) using this technique

Example: Correctness of Copy (cp), One Direction

Base case: If $s \xrightarrow{\mathbb{S},cp} t$ and s is a WHNF, then t is a WHNF.

Forking diagrams:

$$
\begin{tikzcd}
\cdot \arrow[r, bend left, "{\mathbb{S},cp}"] \arrow[r, bend right, "{sr,cp}"'] & \cdot
\end{tikzcd}
\qquad
\begin{tikzcd}
\cdot \arrow[r, "{\mathbb{S},cp}"] \arrow[d, "{sr,a}"'] & \cdot \arrow[d, dashed, "{sr,a}"] \\
\cdot \arrow[r, dashed, "{\mathbb{S},cp}"'] & \cdot
\end{tikzcd}
\qquad
\begin{tikzcd}
\cdot \arrow[r, "{\mathbb{S},cp}"] \arrow[d, "{sr,a}"'] & \cdot \arrow[dl, dashed, "{sr,a}"] \\
\cdot &
\end{tikzcd}
\qquad
\begin{tikzcd}
\cdot \arrow[rr, "{\mathbb{S},cp}"] \arrow[d, "{sr,cp}"'] & & \cdot \arrow[d, dashed, "{sr,cp}"] \\
\cdot \arrow[r, dashed, "{\mathbb{S},cp}"'] & \cdot \arrow[r, dashed, "{\mathbb{S},cp}"'] & \cdot
\end{tikzcd}
$$

$a \in \{probl, probr\}$

Term rewrite system for forking diagrams:

$\mathsf{Scp}(\mathsf{SR}(x)) \to x \quad \mathsf{Scp}(\mathsf{SR}(x)) \to \mathsf{SR}(\mathsf{Scp}(x)) \quad \mathsf{Scp}(\mathsf{SR}(x)) \to \mathsf{SR}(x) \quad \mathsf{Scp}(\mathsf{SR}(x)) \to \mathsf{SR}(\mathsf{Scp}(\mathsf{Scp}(x)))$

Since the the base case and diagrams preserve the prob-reductions, and the TRS is terminating, $\xrightarrow{cp} \subseteq \leq_c$ follows.

Correctness by Comparing Distributions

A frontier evaluation result of s is a multiset with entries (q, s_i) constructed as:

- only $\xrightarrow{wsr,probl}$ or $\xrightarrow{wsr,probr}$ reductions are used, starting with $(1, s)$
- take any finite cut of the whole evaluation tree starting with $(1, s)$ and applying prob-reductions, where for branches, both branches or no branch are included in the cut
- a frontier evaluation result contains exactly all (q, s_i) at leaves of the cut.
- the sum over all q in the multiset is 1

Example

$$\begin{tikzcd}
 & (1, (s_1 \oplus (s_2 \oplus ((s_3 \oplus s_4) \oplus s_5)))) \arrow[dl] \arrow[dr] & & & \\
(0.5, s_1) & & (0.5, (s_2 \oplus ((s_3 \oplus s_4) \oplus s_5))) \arrow[dl] \arrow[dr] & & \\
 & (0.25, s_2) & & (0.25, (((s_3 \oplus s_4) \oplus s_5))) \arrow[dl] \arrow[dr] & \\
 & & (0.125, (s_3 \oplus s_4)) \arrow[dl] \arrow[dr] & & (0.125, s_5) \\
 & (0.0625, s_3) & & (0.0625, s_4) &
\end{tikzcd}$$

Correctness by Comparing Distributions

A frontier evaluation result of s is a multiset with entries (q, s_i) constructed as:

- only $\xrightarrow{wsr,probl}$ or $\xrightarrow{wsr,probr}$ reductions are used, starting with $(1, s)$
- take any finite cut of the whole evaluation tree starting with $(1, s)$ and applying prob-reductions, where for branches, both branches or no branch are included in the cut
- a frontier evaluation result contains exactly all (q, s_i) at leaves of the cut.
- the sum over all q in the multiset is 1

Example

$$\begin{tikzcd}
 & \boxed{(1, (s_1 \oplus (s_2 \oplus ((s_3 \oplus s_4) \oplus s_5))))} \arrow[dl] \arrow[dr] & & & \\
(0.5, s_1) & & (0.5, (s_2 \oplus ((s_3 \oplus s_4) \oplus s_5))) \arrow[dl] \arrow[dr] & & \\
 & (0.25, s_2) & & (0.25, (((s_3 \oplus s_4) \oplus s_5))) \arrow[dl] \arrow[dr] & \\
 & & (0.125, (s_3 \oplus s_4)) \arrow[dl] \arrow[dr] & & (0.125, s_5) \\
 & (0.0625, s_3) & & (0.0625, s_4) &
\end{tikzcd}$$

Correctness by Comparing Distributions

A frontier evaluation result of s is a multiset with entries (q, s_i) constructed as:

- only $\xrightarrow{wsr,probl}$ or $\xrightarrow{wsr,probr}$ reductions are used, starting with $(1, s)$
- take any finite cut of the whole evaluation tree starting with $(1, s)$ and applying prob-reductions, where for branches, both branches or no branch are included in the cut
- a frontier evaluation result contains exactly all (q, s_i) at leaves of the cut.
- the sum over all q in the multiset is 1

Example

$(1, (s_1 \oplus (s_2 \oplus ((s_3 \oplus s_4) \oplus s_5))))$

$\boxed{(0.5, s_1)}$ $\quad$ $\boxed{(0.5, (s_2 \oplus ((s_3 \oplus s_4) \oplus s_5)))}$

$(0.25, s_2)$ $\quad$ $(0.25, (((s_3 \oplus s_4) \oplus s_5)))$

$(0.125, (s_3 \oplus s_4))$ $\quad$ $(0.125, s_5)$

$(0.0625, s_3)$ $\quad$ $(0.0625, s_4)$

Correctness by Comparing Distributions

A frontier evaluation result of s is a multiset with entries (q, s_i) constructed as:

- only $\xrightarrow{wsr,probl}$ or $\xrightarrow{wsr,probr}$ reductions are used, starting with $(1, s)$
- take any finite cut of the whole evaluation tree starting with $(1, s)$ and applying prob-reductions, where for branches, both branches or no branch are included in the cut
- a frontier evaluation result contains exactly all (q, s_i) at leaves of the cut.
- the sum over all q in the multiset is 1

Example

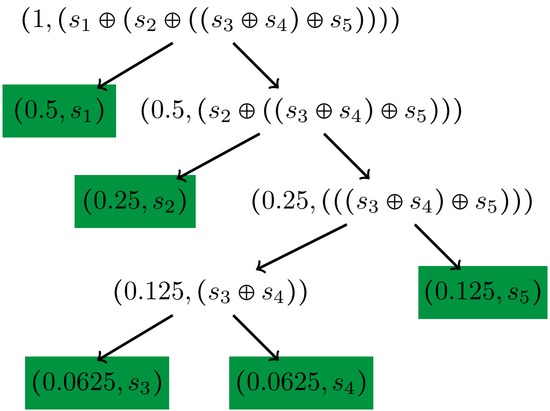
$$\begin{tikzcd}
 & (1, (s_1 \oplus (s_2 \oplus ((s_3 \oplus s_4) \oplus s_5)))) \arrow[dl] \arrow[dr] & & & \\
\boxed{(0.5, s_1)} & & (0.5, (s_2 \oplus ((s_3 \oplus s_4) \oplus s_5))) \arrow[dl] \arrow[dr] & & \\
 & (0.25, s_2) & & (0.25, (((s_3 \oplus s_4) \oplus s_5))) \arrow[dl] \arrow[dr] & \\
 & & (0.125, (s_3 \oplus s_4)) \arrow[dl] \arrow[dr] & & \boxed{(0.125, s_5)} \\
 & \boxed{(0.0625, s_3)} & & \boxed{(0.0625, s_4)} &
\end{tikzcd}$$

Correctness by Comparing Distributions

A frontier evaluation result of s is a multiset with entries (q, s_i) constructed as:

- only $\xrightarrow{wsr,probl}$ or $\xrightarrow{wsr,probr}$ reductions are used, starting with $(1, s)$
- take any finite cut of the whole evaluation tree starting with $(1, s)$ and applying prob-reductions, where for branches, both branches or no branch are included in the cut
- a frontier evaluation result contains exactly all (q, s_i) at leaves of the cut.
- the sum over all q in the multiset is 1

Example



Correctness by Comparing Distributions

Correctness Criterion: Same distribution after prob-reduction

If for all $s \xrightarrow{\mathbb{R},T} t$, criterion EqCr1, EqCr2, or EqCr3 holds for frontier-evaluation results A of s and B of t, then $\xrightarrow{T} \ \subseteq \ \leq_c$.

- **EqCr1** For every $(q,s) \in A$ there is some $(q',s) \in B$ with $q \leq q'$.
- **EqCr2** For every $(q,s) \in A$: $q_{s,A} \leq q_{s,B}$ where $q_{s,X} = \sum_{(p,s)\in X} p$
- **EqCr3** For every $(q,s) \in A$, with $s \neq \Omega$: $q_{s,A} \leq q_{s,B}$ where $q_{s,X} = \sum_{(p,s)\in X} p$

Examples:

- $(r \oplus r) \leq_c r$: EqCr1 holds for $A = \{(0.5, R[r]), (0.5, R[r])\}$ and $B = \{(1, R[r])\}$
- $r \leq_c (r \oplus r)$: EqCr2 holds for $A = \{(1, R[r])\}$ and $B = \{(0.5, R[r]), (0.5, R[r])\}$
- $(\Omega \oplus r) \leq_c r$: EqCr3 holds for $A = \{(0.5, R[\Omega]), (0.5.R[r])\}$ and $B = \{(1, R[r])\}$

Correctness by Comparing Distributions

Correctness of prob-transformations

$$
\begin{array}{ll}
(\mathit{probid}) & s \oplus s \to s \\
(\mathit{probcomm}) & s \oplus t \to t \oplus s \\
(\mathit{probdistr}) & r \oplus (s \oplus t) \to (r \oplus s) \oplus (r \oplus t) \\
(\mathit{probreorder}) & (s_1 \oplus s_2) \oplus (t_1 \oplus t_2) \to (s_1 \oplus t_1) \oplus (s_2 \oplus t_2)
\end{array}
$$

is shown by the criterion on comparing distributions

Extensions by Data Constructors, Case, and Seq: $L_{need,\oplus}^{case,seq}$

Calculus $L_{need,\oplus}^{case,seq}$ extends $L_{need,\oplus}$ by data constructors, case and seq:

$$
\begin{aligned}
s,t,r \in \mathit{Exp} &::= \ldots \mid \mathtt{seq}\ s\ t \mid c_{T,i}\ s_1 \ldots s_{ar(c_{T,i})} \mid \mathtt{case}_T\ s\ \mathtt{of}\ alts_T \\
alts_T &::= \{alt_{T,1}; \ldots ; alt_{T,n_T}\} \\
alt_{T,i} &::= c_{T,i}\ x_1 \ldots x_{ar(c_{T,i})}\ \texttt{->}\ s
\end{aligned}
$$

Example (with lists and booleans):

$$
\begin{aligned}
&\mathtt{let}\ map = \lambda f.\lambda xs.\mathtt{case}_{List}\, xs\, \mathtt{of}\ \{c_{\mathtt{Nil}}\ \texttt{->}\ c_{\mathtt{Nil}}; c_{\mathtt{Cons}}\ x\ xs\ \texttt{->}\ c_{\mathtt{Cons}}\ (f\ x)\ (map\, f\, xs)\}, \\
&\qquad not = \lambda x.\mathtt{case}_{Bool}\, x\, \mathtt{of}\ \{c_{\mathtt{False}}\ \texttt{->}\ c_{\mathtt{True}}; c_{\mathtt{True}}\ \texttt{->}\ c_{\mathtt{False}}\} \\
&\mathtt{in}\ map\, not\, (c_{\mathtt{Cons}}\ c_{\mathtt{True}}\ (c_{\mathtt{Cons}}\ c_{\mathtt{False}}\ c_{\mathtt{Nil}}))
\end{aligned}
$$

In the paper:

- extension of the operational semantics
- sketch that the context lemma etc. still hold for the extended calculus
- correctness of program transformations via diagrams (automated computation)
- non-extensionality of $L_{need,\oplus}^{case,seq}$

Conclusion & Further Work

Conclusions

- We introduced a probabilistic call-by-need lambda calculus
- We analysed contextual equivalence and provided several techniques to show equivalences
- We added extensions to realistic models of probabilistic programming languages
- Our previously developed methods are adaptable to the probabilistic setting

Further Work

- add (polymorphic) typing to the calculus
- compare the contextual semantics with mathematical probabilistic models
- add other probabilistic constructs

Thank You!

Backup-Slide: Counterexample to Extensionality

We provide an example (similar to [Schmidt-Schauß,S.,Machkasova 2011]):

- there are closed abstractions s_1, s_2 such that: $s_1\ r \sim_c s_2\ r$ for all values or diverging expressions r, but $s_1 \not\sim_c s_2$
- Thus $L_{need,\oplus}^{case,seq}$ is not extensional even for a weak form of extensionality
- Hence usual definitions of applicative bisimilarity are unsound for $L_{need,\oplus}^{case,seq}$

$$
\begin{array}{llll}
s_1 & := \lambda x.p_1 \oplus p_2 & s_2 & := \lambda x.(p_1 \oplus p_3) \oplus (p_2 \oplus p_4) \\
p_1 & := (\mathit{False}, \mathit{seqp}\ x\ \mathit{False}) & p_3 & := (\mathit{False}, \mathit{seqp}\ x\ \mathit{True}) \\
p_2 & := \mathit{seqp}\ x\ (\mathit{False}, \mathit{True}) & p_4 & := \mathit{seqp}\ x\ (\mathit{False}, \mathit{False}) \\
\mathit{seqp} & := \lambda x.\lambda y.\mathtt{case}_{\mathit{Pair}}\ x\ \mathtt{of}\ \{(z_1, z_2)\ \mathtt{->}\ y\} & &
\end{array}
$$

For $C = \mathtt{let}\ y = ([\cdot]\ y)\ \mathtt{in}\ \mathtt{if}\ (\mathit{snd}\ y)\ \mathtt{then}\ \mathit{True}\ \mathtt{else}\ \Omega$

- $C[s_1]$ diverges ($\mathrm{ExCv}(C[s_1]) = 0$)
- $C[s_2]$ can evaluate to True with a positive probability ($\mathrm{ExCv}(C[s_2]) > 0$)

Backup-Slide: Expected Convergence with Bound

Expected convergence of s with bound k = number prob-reductions

$$\mathrm{ExCv}(s,k) = \sum_{\substack{(1,s) \wr_L (q,t) \in \mathit{Eval}(1,s), \\ |L| \leq k}} q$$

→ allows inductive proofs and constructions on the number k,
and in the limit, differences in k do not matter:

Lemma

Let s, t be expressions and such that $\forall k \geq 0 : \exists d \geq 0 : \mathrm{ExCv}(s,k) \leq \mathrm{ExCv}(t, k+d)$.
Then $\mathrm{ExCv}(s) \leq \mathrm{ExCv}(t)$.

Backup-Slide: Nondeterminism vs. Probability (1)

May-convergence:

- Definition: s may-converges if it can be reduced to a WHNF
- Properties: s may-converges iff $\text{ExCv}(s) > 0$

Must-Convergence:

- Definition: s must-converges if s has no infinite reduction and all reductions end with a WHNF.
- Properties:
  - s must-converges $\implies \text{ExCv}(s) = 1$
  - $\text{ExCv}(s) = 1 \not\Longrightarrow s$ must-converges:

    $\text{ExCv}(\texttt{let}\ x = (\lambda y.(x\ id) \oplus K)\ \texttt{in}\ (x\ id)) = 1$,

    but the expression has an infinite evaluation

Backup-Slide: Nondeterminism vs. Probability (2)

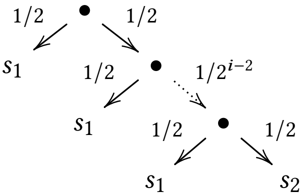
Should-Convergence:

- Definition: s should-converges if s is not reducible to a must-divergent expression (equivalently: for all $s' : s \xrightarrow{sr,*} s' \implies s'$ is may-convergent.)
- Properties: • $\mathrm{ExCv}(s) = 1 \implies s$ should-converges
  - s should-converges $\not\Longrightarrow \mathrm{ExCv}(s) = 1$:
    expression s should-converge, but $\mathrm{ExCv}(s) = 5/12$

$$
\begin{array}{l}
s := \texttt{let}\ cprob = \\
\qquad\qquad \lambda i.if\ i = 0\ then\ K \\
\qquad\qquad\quad else\ \lambda x, y.(cprob\ (i{-}1)\ x\ y) \oplus y, \\
\qquad\quad gen = \lambda i.cprob\ i\ K\ (gen\ (i{+}1)) \\
\quad \texttt{in}\ gen\ 2
\end{array}
$$

  - s should-converges $\implies EC(s) > 0$

$cprob\ i\ s_1\ s_2$:



$gen\ 2$: