Improvement Theory for Probabilistic Call-by-Need

David Sabel

Hochschule RheinMain Wiesbaden

joint work with Manfred Schmidt-Schauß

42. Workshop der GI-Fachgruppe
Programmiersprachen und Rechenkonzepte
Bad Honnef, 07/May/2026

Motivation

hsrm*

Probabilistic Programming

- probabilistic programs represent stochastic models
- program execution is performing a probabilistic experiment
- reasoning on program semantics is reasoning on the models

+

Call-by-Need Functional Programming Languages

- declarative, high-level programming allowing equational reasoning
- efficient implementation of lazy evaluation
- semantics is different from call-by-name and call-by-value

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

Previous Work

hsrm*

Probablistic call-by-need lambda-calculi with binary choice $s \oplus t$

- correctness of program transformations w.r.t. contextual equivalence in an untyped calculus with recursive `let` [PPDP 2022]
- a PCF-like calculus, distribution equivalence coincides with contextual equivalence on programs of type nat [WPTE 2022, JLAMP 2023]
- encoding unfair, biased choice $\overset{p}{\oplus}$ with real-valued probability p by fair choice $\overset{1/2}{\oplus}$ [WPTE 2025]

Improvement Theory

hsrm*

Improvement Theory was invented by Dave Sands in the 1990s, e.g.

- *D.Sands: Total Correctness by Local Improvement in Program Transformation, POPL 95*
- *A.Moran & D.Sands, Improvement in a Lazy Context: An Operational Theory for Call-by-Need, POPL 99*
- *J.Gustavsson & D.Sands: Possibilities and Limitations of Call-by-Need Space Improvement, ICFP 01*

Improvement

Program p_2 improves program p_1 iff

- p_1 and p_2 are semantically equivalent
- Replacing p_1 by p_2 in any context, does not increase the cost:

$$\text{for all contexts } C\text{: } Cost(C[p_1]) \geq Cost(C[p_2])$$

$Cost(\cdot)$: runtime (length of evaluations) or space for space-improvements etc.

Our Goal: Develop an **improvement theory for probabilistic call-by-need**

Improvement Theory & Probabilism: Challenges

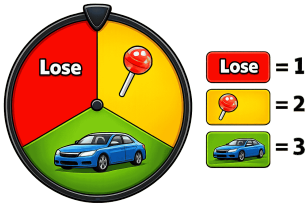
hsrm

p_2 improves p_1 iff $p_1 \sim p_2$ and $\forall C : Cost(C[p_1]) \geq Cost(C[p_2])$

- since a probabilistic program stands for a stochastic model, an improvement computes the same model more efficiently.
- while in deterministic PLs programs converge or diverge, and converging programs have a unique evaluation length,
  in probabilistic PLs expectation values have to be used (expected convergence, expected evaluation length)
- find alternative characterisations of the improvement-property that avoid the universal quantification over all contexts

Running Example: Wheel of Fortune

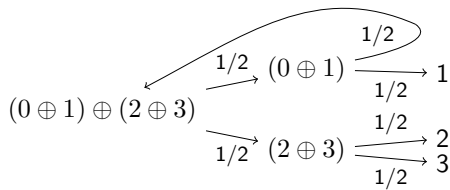
hsrm*



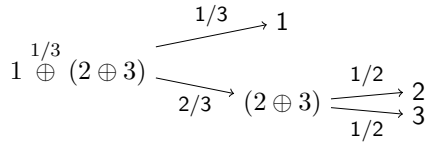
Stochastic model:
uniform distribution over the outcomes $1, 2, 3$.

Two probabilistic programs

- s_1: uniformly draw from $\{0,1,2,3\}$, restart on 0:



- s_2: use biased choice:



Programs are semantically equivalent. Replacing s_1 by s_2 should be an improvement

Probabilistic Lazy PCF $ProbPCF^{need}_{\mathbb{R}}$

hsrm

Expressions:

$$s,t \in \mathit{Exp} \quad ::= x \mid \lambda x.s \mid (s\ t) \mid \textbf{fix}\ s \mid \texttt{if}\ s\ \texttt{then}\ t_1\ \texttt{else}\ t_2 \mid \texttt{let}\ x = s\ \texttt{in}\ t$$
$$n \mid \texttt{pred}\ s \mid \texttt{succ}\ s \mid (s \overset{p}{\oplus} t) \text{ where } n \in \mathbb{N} \text{ and } p \in (0,1)$$

Types: $\tau, \sigma \in \mathit{Typ} ::= \mathsf{nat} \mid \tau \to \sigma$

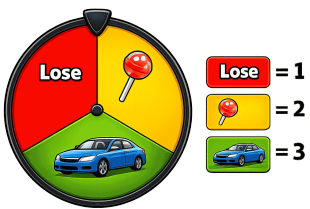
Small-Step Reduction $\xrightarrow{sr}$:

(*sr,lbeta*)	$R[(\lambda x.s)\ t] \xrightarrow{sr} R[\texttt{let}\ x = t\ \texttt{in}\ s]$	
(*sr,if-0*)	$R[\texttt{if}\ 0\ \texttt{then}\ s\ \texttt{else}\ t] \xrightarrow{sr} R[s]$	
(*sr,if-not-0*)	$R[\texttt{if}\ n\ \texttt{then}\ s\ \texttt{else}\ t] \xrightarrow{sr} R[t]$ if $n > 0$	probability
(*sr,probl*)	$R[s \overset{p}{\oplus} t] \xrightarrow{sr} R[s]$	p
(*sr,probr*)	$R[s \overset{p}{\oplus} t] \xrightarrow{sr} R[t]$	$1-p$
...	...	

where $R ::= \mathrm{LR}[A] \mid \mathrm{LR}[\texttt{let}\ x = A\ \texttt{in}\ R[x]]$ $\quad \mathrm{LR} ::= [\cdot] \mid \texttt{let}\ x = s\ \texttt{in}\ \mathrm{LR}$

$A ::= [\cdot] \mid (A\ s) \mid \texttt{if}\ A\ \texttt{then}\ s\ \texttt{else}\ t \mid \texttt{pred}\ A \mid \texttt{succ}\ A \mid \texttt{fix}\ A$

Running Example: Wheel of Fortune



s_1: uniformly draw from $\{0,1,2,3\}$, restart on 0:

$$s_1 := \textbf{fix}\ (\lambda f.\ \texttt{let}\ r = ((0 \oplus 1) \oplus (2 \oplus 3))\ \texttt{in if}\ r\ \texttt{then}\ f\ \texttt{else}\ r)$$

s_2: use biased choice

$$s_2 := 1 \overset{1/3}{\oplus} (2 \overset{1/2}{\oplus} 3)$$

hsrm*

Contextual Cost Improvement

$$s \preceq_{ci} t \text{ iff } s \sim_c t \text{ and } \forall C[\cdot] : \text{nat} : \textsc{ExLen}(C[s]) \geq \textsc{ExLen}(C[t])$$

- **Contextual equivalence** $s \sim_c t$ iff $\forall C[\cdot] : \text{nat} : \textsc{ExCv}(C[s]) = \textsc{ExCv}(C[t])$
- **Expected convergence** $\textsc{ExCv}(s) =$ sum all probabilities of evaluations $s \xrightarrow{sr,*} \text{LR}[v]$
- **(Partial) expected evaluation length** $\textsc{ExLen}(s) \in [0, \infty]$ is the expected length (counting essential reduction steps) of all evaluations $s \xrightarrow{sr,*} \text{LR}[v]$

For our running example:

$$\textsc{ExVCv}(s_1) = \textsc{ExVCv}(s_2) = 1 \qquad \textsc{ExLen}(s_1) \geq 3 \sum_{k \geq 1} \frac{2k}{4^k} = \frac{8}{3} \text{ and } \textsc{ExLen}(s_2) = \frac{5}{3}$$

This shows $s_2 \not\preceq_{ci} s_1$, but for $s_1 \preceq_{ci} s_2$ we have to reason about $C[s_1]$ and $C[s_2]$ for all C!

Counting Expectations per Number

Distribution-Cost Improvement

For closed s, t : nat: $s \preceq_{di} t$ iff $s \sim_d t$ and $\forall n \in \mathbb{N} : \mathrm{CExLen}(s, n) \geq \mathrm{CExLen}(t, n)$

- **Distribution equivalence** $s \sim_d t$ iff $\forall n \in \mathbb{N} : \mathrm{ExVCv}(s, n) = \mathrm{ExVCv}(t, n)$
- **Expected value convergence** $\mathrm{ExVCv}(s, \boldsymbol{n})$
  $=$ sum all probabilities of evaluations $s \xrightarrow{sr,*} \mathrm{LR}[\boldsymbol{n}]$
- **Expected reduction length** by evaluations **ending in** $n \in \mathbb{N}$: $\mathrm{ExLen}(s, \boldsymbol{n})$.
- **Conditional expected length** $\mathrm{CExLen}(s, n) := \frac{\mathrm{ExLen}(s,n)}{\mathrm{ExVCv}(s,n)}$ if $\mathrm{ExVCv}(s, n) > 0$

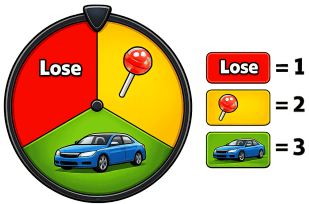
Intuition:

- $\mathrm{ExVCv}(s, n)$ says how likely result n is,
- $\mathrm{CExLen}(s, n)$ says how expensive it is, conditional on obtaining n.

Advantage: $\preceq_{di}$ is defined without context quantification!

The Example with Distribution-Cost

hsrm*



$s_1 := \textbf{fix}\ (\lambda f.\ \texttt{let}\ r = ((0 \oplus 1) \oplus (2 \oplus 3))\ \texttt{in if}\ r\ \texttt{then}\ f\ \texttt{else}\ r)$

$s_2 := 1 \overset{1/3}{\oplus} (2 \overset{1/2}{\oplus} 3)$

$\textsc{ExVCv}(s_1, n) = \frac{1}{3}$ for $n \in \{1,2,3\}$ $\quad$ $\textsc{ExVCv}(s_2, n) = \frac{1}{3}$ for $n \in \{1,2,3\}$

Hence $s_1 \sim_d s_2$

$\textsc{ExLen}(s_1, n) \geq \sum_{k \geq 1} \frac{2k}{4^k} = \frac{8}{9}$ for $n \in \{1,2,3\}$ $\quad$ $\textsc{ExLen}(s_2, 1) = \frac{1}{3}$, $\textsc{ExLen}(s_2, n) = \frac{2}{3}$ for $n \in \{2,3\}$

$\textsc{CExLen}(s_1, n) \geq \frac{8}{3}$ for $n \in \{1,2,3\}$ $\quad$ $\textsc{CExLen}(s_2, 1) = 1$, $\textsc{CExLen}(s_2, n) = 2$ for $n \in \{2,3\}$

Hence $s_1 \preceq_{di} s_2$

hsrm

Main Theorem

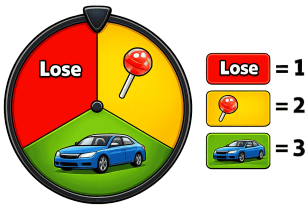
For stochastic programs s, t : nat, the contextual cost improvement coincides with the distribution-cost improvement, i.e.

$$s \preceq_{ci} t \iff s \preceq_{di} t.$$

- From previous work $\sim_d = \sim_c$, so the proof mainly has to show the length parts
- $s \preceq_{ci} t \Rightarrow s \preceq_{di} t$ is the easier part
- $s \preceq_{di} t \Rightarrow s \preceq_{ci} t$ is the hard part.
  Show $s \preceq_{di} t \implies \forall C : C[s] \preceq_{di} C[t]$ as follows
  1. Show $s \preceq_{di} t \implies \forall R : R[s] \preceq_{di} R[t]$ using a so-called s-first strategy
  2. Use a context lemma to lift this to arbitrary contexts.
     Proof uses multi-contexts, bounded measures (bound on the number of prob-steps) and an asymmetric induction

Wheel of Fortune Revisited

hsrm



$$s_1 := \textbf{fix}\ (\lambda f.\ \texttt{let}\ r = ((0 \oplus 1) \oplus (2 \oplus 3))\ \texttt{in if}\ r\ \texttt{then}\ f\ \texttt{else}\ r)$$

$$s_2 := 1 \overset{1/3}{\oplus} (2 \overset{1/2}{\oplus} 3)$$

Since $s_1 \preceq_{di} s_2$ by the main theorem also $s_1 \preceq_{ci} s_2$

So s_2 computes the same model more efficiently than s_1.

Further Transformations

hsrm*

For expressions of type nat:

Commutativity: $s \overset{p}{\oplus} t \;\asymp_{di}\; t \overset{1-p}{\oplus} s$

Idempotency: $s \overset{p}{\oplus} s \;\preceq_{di}\; s$

Distributivity: $(r \overset{p}{\oplus} s) \overset{q}{\oplus} (r \overset{p}{\oplus} t) \;\preceq_{di}\; r \overset{p}{\oplus} (s \overset{q}{\oplus} t)$

Garbage collection: $\texttt{let}\ x{=}s\ \texttt{in}\ t \;\asymp_{di}\; t$ if $x \notin FV(t)$

Let-float: $A^1[\texttt{let}\ x{=}s\ \texttt{in}\ t] \;\asymp_{di}\; \texttt{let}\ x{=}s\ \texttt{in}\ A^1[t]$

where $A^1 ::= ([\cdot]\ s) \mid \texttt{if}\ [\cdot]\ \texttt{then}\ s\ \texttt{else}\ t \mid \texttt{pred}\ [\cdot] \mid \texttt{succ}\ [\cdot] \mid \textbf{fix}\ [\cdot]$

Surface unique copying: $\texttt{let}\ x{=}t\ \texttt{in}\ C[x] \;\asymp_{di}\; C[t]$

if the hole of C is not in an abstraction and x is fresh for C, t

where $s \asymp_{di} t$ iff $s \preceq_{di} t$ and $t \preceq_{di} s$

Introducing Sharing can be Unsound

hsrm*

A crucial probabilistic phenomenon is that sharing can change the distribution.

Independent sampling:

$$r_1 = \texttt{if } (0 \oplus 1) \texttt{ then } (0 \oplus 1) \texttt{ else } 2$$

Shared sampling:

$$r_2 = \texttt{let } x = 0 \oplus 1 \texttt{ in if } x \texttt{ then } x \texttt{ else } 2$$

The expressions are not distribution equivalent.

Transformations such as common subexpression elimination are sound only under suitable restrictions, for example on deterministic subexpressions.

Conclusion

hsrm

- Probabilistic call-by-need is a distinctive semantic setting because laziness and sharing interact non-trivially with random choice
- Improvement theory captures the optimisation problem for stochastic models: preserve the distribution, reduce expected cost
- The main theorem gives an intrinsic characterisation:

$$s \preceq_{ci} t \iff s \preceq_{di} t$$

Future Work

hsrm*

- Move expressions of number-type to all programs including higher-order types
- Introduce work-decorations, to enable equational reasoning
- Explore further transformations
- Investigate common subexpression elimination for deterministic subterms
- Study further cost models, like e.g. space consumption
- Study different / extended languages

Thank You!