



Formalising Semantics for Expected Running Time of Probabilistic Programs

(Rough Diamond)

Johannes Hölzl

TU München, Germany

- Probabilistic programs (pGCL) + expected running time.

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016]

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★
- Two semantics:

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★
- Two semantics:

$$\text{Denotational: } \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0})$$

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★
- Two semantics:

Denotational: $\sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$

Operational: $(\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ stream measure}$

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★

- Two semantics:

Denotational: $\sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0})$

Operational: $(\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ stream measure}$

Correspondence: Denotational \Leftrightarrow Operational

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★

- Two semantics:

Denotational: $\sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$

Operational: $(\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma)$ stream measure

Correspondence: Denotational \Leftrightarrow Operational

- Examples:

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★

- Two semantics:

Denotational: $\sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0})$

Operational: $(\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ stream measure}$

Correspondence: Denotational \Leftrightarrow Operational

- Examples:
 - Simple Random Walk ⚡

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★

- Two semantics:

Denotational: $\sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0})$

Operational: $(\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma)$ stream measure

Correspondence: Denotational \Leftrightarrow Operational

- Examples:
 - Simple Random Walk ⚡
 - Coupon Collector

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★

- Two semantics:

Denotational: $\sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$

Operational: $(\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ stream measure}$

Correspondence: Denotational \Leftrightarrow Operational

- Examples:
 - Simple Random Walk ⚡
 - Coupon Collector

Our work:

– clarified semantics –

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★

- Two semantics:

Denotational: $\sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0})$

Operational: $(\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ stream measure}$

Correspondence: Denotational \Leftrightarrow Operational

- Examples:
 - Simple Random Walk ⚡
 - Coupon Collector

Our work:

- clarified semantics – different proofs –

This Talk

- Probabilistic programs (pGCL) + expected running time.
Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ★

- Two semantics:

Denotational: $\sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \bar{\mathbb{R}}_{\geq 0})$

Operational: $(\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ stream measure}$

Correspondence: Denotational \Leftrightarrow Operational

- Examples:
 - Simple Random Walk ⚡
 - Coupon Collector

Our work:

- clarified semantics – different proofs – correct proofs –

pGCL

Probabilistic Guarded Command Language (pGCL)

$$\sigma \text{ pgcl} = \perp$$

Probabilistic Guarded Command Language (pGCL)

σ pgcl = \perp | ϵ

Probabilistic Guarded Command Language (pGCL)

σ pgcl = \perp | ϵ | ⚡

Probabilistic Guarded Command Language (pGCL)

σ pgcl = \perp | ϵ | !
| $x \sim \mathcal{D}$ or $x := \text{expr}$ "Assign ($\sigma \Rightarrow \sigma$ pmf)"

Probabilistic Guarded Command Language (pGCL)

σ pgcl = \perp | ϵ | !
| $x \sim \mathcal{D}$ or $x := \text{expr}$ "Assign ($\sigma \Rightarrow \sigma$ pmf)"
| $p_1 ; p_2$

Probabilistic Guarded Command Language (pGCL)

σ pgcl = \perp | ϵ | !
| $x \sim \mathcal{D}$ or $x := \text{expr}$ "Assign ($\sigma \Rightarrow \sigma$ pmf)"
| $p_1 ; p_2$
| $p_1 \mid p_2$

Probabilistic Guarded Command Language (pGCL)

σ pgcl = \perp | € | ⚡
| $x \sim \mathcal{D}$ or $x := \text{expr}$ "Assign ($\sigma \Rightarrow \sigma$ pmf)"
| $p_1 ; p_2$
| $p_1 \mid p_2$
| ITE g p_1 p_2 $g :: \sigma \Rightarrow \text{bool}$

Probabilistic Guarded Command Language (pGCL)

σ pgcl = \perp | € | ⚡
| $x \sim \mathcal{D}$ or $x := \text{expr}$ "Assign ($\sigma \Rightarrow \sigma$ pmf)"
| $p_1 ; p_2$
| $p_1 \mid p_2$
| ITE g p_1 p_2 $g :: \sigma \Rightarrow \text{bool}$
| WHILE g DO p

Denotational Semantics (Expected Running Time)

$$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$$

Denotational Semantics (Expected Running Time)

$$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$$

Values we want assigned to a *terminal state*

Denotational Semantics (Expected Running Time)

Values computed for the a *starting state*

$$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$$

Values we want assigned to a *terminal state*

Denotational Semantics (Expected Running Time)

$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$

$\text{ert} \perp \quad \quad \quad c = c$

Denotational Semantics (Expected Running Time)

$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$

$\text{ert } \perp \quad \quad \quad c = c$

$\text{ert } \epsilon \quad \quad \quad c = 1 + c$

Denotational Semantics (Expected Running Time)

$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$

$\text{ert} \perp \quad c = c$

$\text{ert} \text{€} \quad c = 1 + c$

$\text{ert} \text{⚡} \quad c = 0$

Denotational Semantics (Expected Running Time)

$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$

$\text{ert } \perp \quad c = c$

$\text{ert } \epsilon \quad c = 1 + c$

$\text{ert } \text{!} \quad c = 0$

$\text{ert } (\text{Assign } u) \quad c = 1 + \left(\lambda x. \int_y c y d(ux) \right)$

Denotational Semantics (Expected Running Time)

$$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$$

$$\text{ert } \perp \quad c = c$$

$$\text{ert } \epsilon \quad c = 1 + c$$

$$\text{ert } \text{!} \quad c = 0$$

$$\text{ert } (\text{Assign } u) \quad c = 1 + \left(\lambda x. \int_y c y d(ux) \right)$$

$$\text{ert } (p_1; p_2) \quad c = \text{ert } p_1 (\text{ert } p_2 c)$$

Denotational Semantics (Expected Running Time)

$$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$$

$$\text{ert } \perp \quad c = c$$

$$\text{ert } \epsilon \quad c = 1 + c$$

$$\text{ert } \text{!} \quad c = 0$$

$$\text{ert } (\text{Assign } u) \quad c = 1 + \left(\lambda x. \int_y c y d(ux) \right)$$

$$\text{ert } (p_1; p_2) \quad c = \text{ert } p_1 (\text{ert } p_2 c)$$

$$\text{ert } (p_1 \mid p_2) \quad c = \text{ert } p_1 c \sqcup \text{ert } p_2 c$$

Denotational Semantics (Expected Running Time)

$$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$$

$$\text{ert } \perp \quad c = c$$

$$\text{ert } \epsilon \quad c = 1 + c$$

$$\text{ert } \text{!} \quad c = 0$$

$$\text{ert } (\text{Assign } u) \quad c = 1 + \left(\lambda x. \int_y c y d(ux) \right)$$

$$\text{ert } (p_1; p_2) \quad c = \text{ert } p_1 (\text{ert } p_2 c)$$

$$\text{ert } (p_1 \mid p_2) \quad c = \text{ert } p_1 c \sqcup \text{ert } p_2 c$$

$$\text{ert } (\text{ITE } g p_1 p_2) \quad c = 1 + (\lambda x. \text{if } g x \text{ then } \text{ert } p_1 c x \text{ else } \text{ert } p_2 c x)$$

Denotational Semantics (Expected Running Time)

$$\text{ert} :: \sigma \text{ pgcl} \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}) \Rightarrow (\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0})$$

$$\text{ert } \perp \quad c = c$$

$$\text{ert } \epsilon \quad c = 1 + c$$

$$\text{ert } \text{!} \quad c = 0$$

$$\text{ert } (\text{Assign } u) \quad c = 1 + \left(\lambda x. \int_y c y d(ux) \right)$$

$$\text{ert } (p_1; p_2) \quad c = \text{ert } p_1 (\text{ert } p_2 c)$$

$$\text{ert } (p_1 \mid p_2) \quad c = \text{ert } p_1 c \sqcup \text{ert } p_2 c$$

$$\text{ert } (\text{ITE } g p_1 p_2) \quad c = 1 + (\lambda x. \text{if } g x \text{ then } \text{ert } p_1 c x \text{ else } \text{ert } p_2 c x)$$

$$\text{ert } (\text{WHILE } g \text{ DO } p) \quad c = \text{lfp } (\lambda Wx. 1 + \text{if } g x \text{ then } \text{ert } p Wx \text{ else } c x)$$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

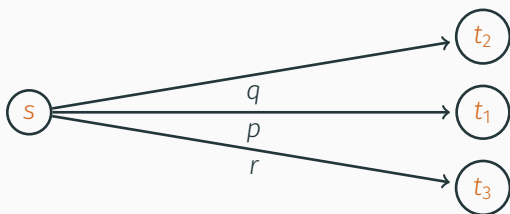
$K :: \sigma \Rightarrow \sigma$ pmf set $Ks \neq \emptyset$



Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

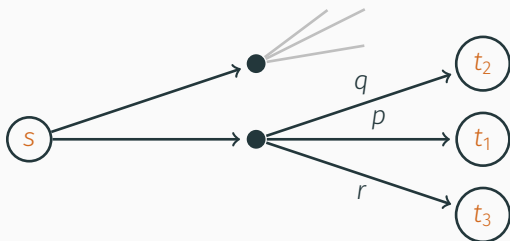
$K :: \sigma \Rightarrow \sigma$ pmf set $Ks \neq \emptyset$



Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

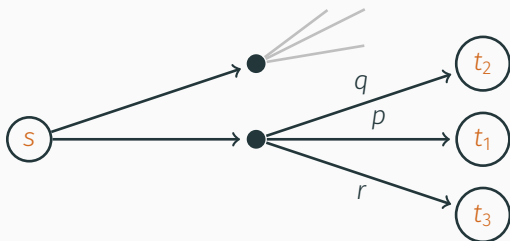
$K :: \sigma \Rightarrow \sigma$ pmf set $Ks \neq \emptyset$



Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

$K :: \sigma \Rightarrow \sigma$ pmf set $Ks \neq \emptyset$

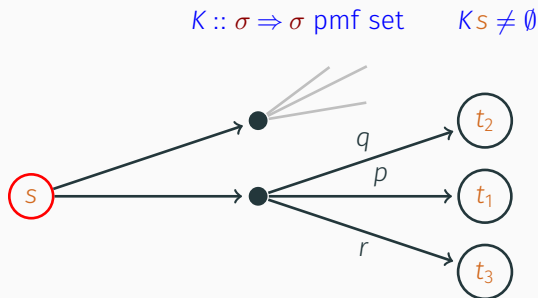


Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f :

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in K_s} \int_t \hat{\mathbb{E}}_t[\lambda \omega. f(t \cdot \omega)] \, dD$$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice



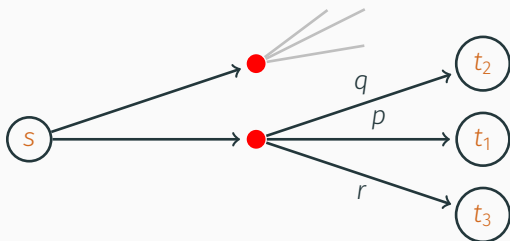
Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f :

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in Ks} \int_t \hat{\mathbb{E}}_t[\lambda \omega. f(t \cdot \omega)] dD$$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

$K :: \sigma \Rightarrow \sigma$ pmf set $Ks \neq \emptyset$

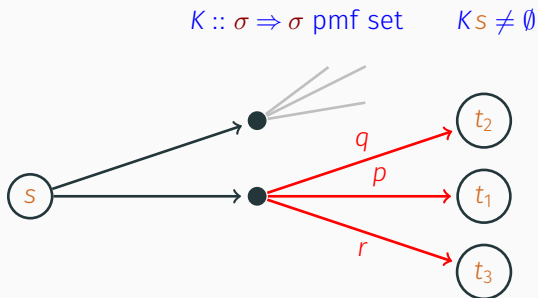


Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f :

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in Ks} \int_t \hat{\mathbb{E}}_t[\lambda \omega. f(t \cdot \omega)] \, dD$$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

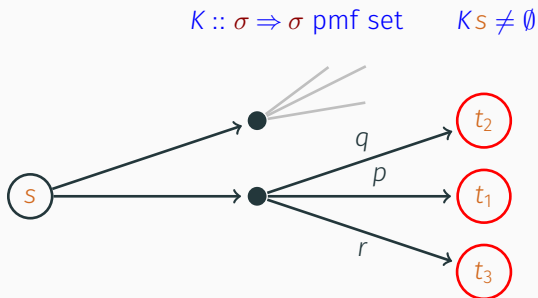


Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f :

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in K_s} \int_t \hat{\mathbb{E}}_t[\lambda \omega. f(t \cdot \omega)] \, dD$$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice



Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f :

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in Ks} \int_t \hat{\mathbb{E}}_t[\lambda \omega. f(t \cdot \omega)] \, dD$$

Operational Semantics

$K :: (\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ pmf set}$

Operational Semantics

$K :: (\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ pmf set}$

$K(\perp, s) = \ll \perp, s \gg$

$K(\epsilon, s) = \ll \perp, s \gg$

$K(\text{!}, s) = \ll \text{!}, s \gg$

Operational Semantics

$K :: (\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ pmf set}$

$K(\perp, s) = \ll \perp, s \gg$

$K(\epsilon, s) = \ll \perp, s \gg$

$K(\text{!}, s) = \ll \text{!}, s \gg$

$K(\text{Assign } u, s) = [\lambda s'. (\perp, s')] \{u\ s\}$

Operational Semantics

$K :: (\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ pmf set}$

$$K(\perp, s) = \ll \perp, s \gg$$

$$K(\epsilon, s) = \ll \perp, s \gg$$

$$K(\text{!}, s) = \ll \text{!}, s \gg$$

$$K(\text{Assign } u, s) = [\lambda s'. (\perp, s')] \{u\ s\}$$

$$K(p_1 \mid p_2, s) = \ll p_1, s \gg \cup \ll p_2, s \gg$$

Operational Semantics

$K :: (\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ pmf set}$

$K(\perp, s) = \ll \perp, s \gg$

$K(\epsilon, s) = \ll \perp, s \gg$

$K(\text{!}, s) = \ll \text{!}, s \gg$

$K(\text{Assign } u, s) = [\lambda s'. (\perp, s')] \{u\ s\}$

$K(p_1 \mid p_2, s) = \ll p_1, s \gg \cup \ll p_2, s \gg$

$K(\text{ITE } g \ p_1 \ p_2, s) = \text{if } g \ s \text{ then } \ll p_1, s \gg \text{ else } \ll p_2, s \gg$

$K(\text{WHILE } g \ \text{DO } p, s) = \text{if } g \ s \text{ then } \ll p; \text{ WHILE } g \ \text{DO } p, s \gg \text{ else } \ll \perp, s \gg$

Operational Semantics

$K :: (\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ pmf set}$

$K(\perp, s) = \ll \perp, s \gg$

$K(\epsilon, s) = \ll \perp, s \gg$

$K(\text{!}, s) = \ll \text{!}, s \gg$

$K(\text{Assign } u, s) = [\lambda s'. (\perp, s')] \{u\} s$

$K(p_1 \mid p_2, s) = \ll p_1, s \gg \cup \ll p_2, s \gg$

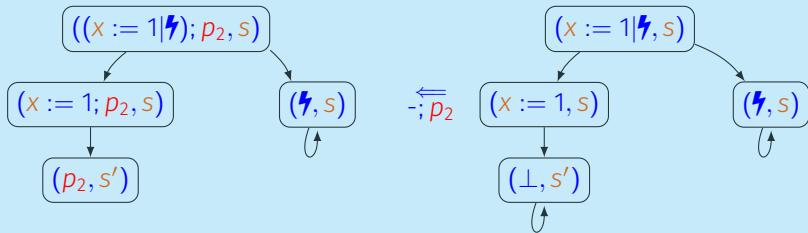
$K(\text{ITE } g \ p_1 \ p_2, s) = \text{if } g \ s \ \text{then } \ll p_1, s \gg \ \text{else } \ll p_2, s \gg$

$K(\text{WHILE } g \ \text{DO } p, s) = \text{if } g \ s \ \text{then } \ll p; \text{ WHILE } g \ \text{DO } p, s \gg \ \text{else } \ll \perp, s \gg$

$K(p_1; p_2, s) = \left[\begin{array}{l} (\perp, s'). \ (p_2, s') \\ \lambda (\text{!}, s'). \ (\text{!}, s') \\ (p', s'). \ (p'; p_2, s') \end{array} \right] K(p_1, s)$

Operational Semantics

$K :: (\sigma \text{ pgcl} \times \sigma) \Rightarrow (\sigma \text{ pgcl} \times \sigma) \text{ pmf set}$



$K(\text{IF } g \text{ DO } p_1; p_2, s) = \text{if } g \text{ s then } \ll p_1, s \gg \text{ else } \ll p_2, s \gg$

$K(\text{WHILE } g \text{ DO } p, s) = \text{if } g \text{ s then } \ll p; \text{WHILE } g \text{ DO } p, s \gg \text{ else } \ll \perp, s \gg$

$K(p_1; p_2, s) = \left[\begin{array}{l} (\perp, s'). \quad (p_2, s') \\ \lambda (\text{⚡}, s'). \quad (\text{⚡}, s') \\ (p', s'). \quad (p'; p_2, s') \end{array} \right] K(p_1, s)$

Traces

Trace

$\text{COST}_{\text{stream}}$

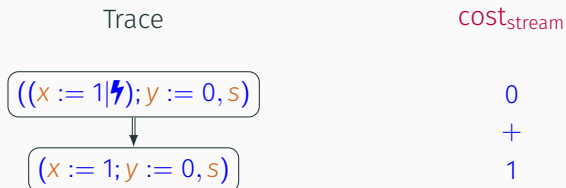
$((x := 1 \mid \text{!}); y := 0, s)$

0

Cost per stream:

$$\text{COST}_{\text{stream}} f (s \cdot \omega) \stackrel{\text{lfp}}{=} \text{COST} f s (\text{COST}_{\text{stream}} f \omega)$$

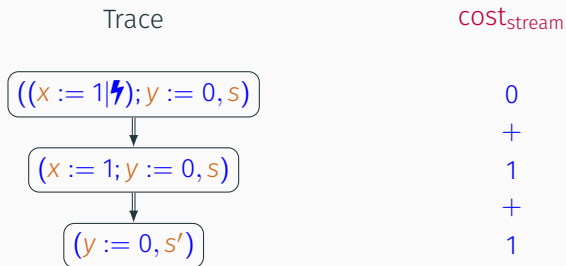
Traces



Cost per stream:

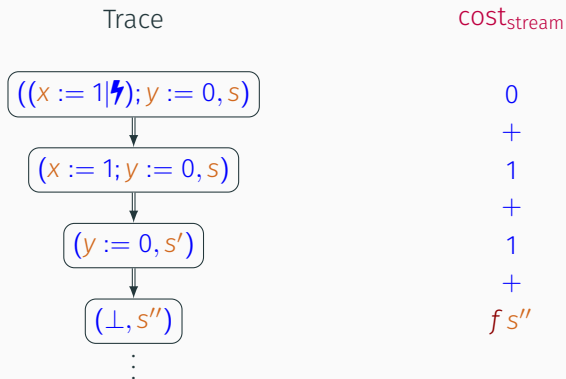
$$\text{COST}_{\text{stream}} f (s \cdot \omega) \stackrel{\text{lfp}}{=} \text{COST} f s (\text{COST}_{\text{stream}} f \omega)$$

Traces



Cost per stream:

$$\text{COST}_{\text{stream}} f (s \cdot \omega) \stackrel{\text{lfp}}{=} \text{COST } f s (\text{COST}_{\text{stream}} f \omega)$$



Cost per stream:

$$\text{COST}_{\text{stream}} f (s \cdot \omega) \stackrel{\text{lfp}}{=} \text{COST } f s (\text{COST}_{\text{stream}} f \omega)$$

Interjection: Least Fixed Points

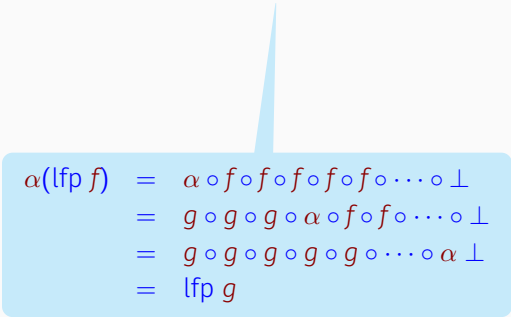
Theorem (Transfer rule for least fixed points)

$$\frac{\sqcup\text{-continuous } \alpha, f, g \quad \alpha \perp = \perp \quad \alpha \circ f = g \circ \alpha}{\alpha(\text{lfp } f) = \text{lfp } g}$$

Interjection: Least Fixed Points

Theorem (Transfer rule for least fixed points)

$$\frac{\sqcup\text{-continuous } \alpha, f, g \quad \alpha \perp = \perp \quad \alpha \circ f = g \circ \alpha}{\alpha(\text{lfp } f) = \text{lfp } g}$$


$$\begin{aligned}\alpha(\text{lfp } f) &= \alpha \circ f \circ f \circ f \circ f \circ f \circ \dots \circ \perp \\ &= g \circ g \circ g \circ \alpha \circ f \circ f \circ \dots \circ \perp \\ &= g \circ g \circ g \circ g \circ g \circ \dots \circ \alpha \perp \\ &= \text{lfp } g\end{aligned}$$

Interjection: Least Fixed Points

Theorem (Transfer rule for least fixed points)

$$\frac{\sqcup\text{-continuous } \alpha, f, g \quad \alpha \perp = \perp \quad \alpha \circ f = g \circ \alpha}{\alpha(\text{lfp } f) = \text{lfp } g}$$


$$\alpha f = \hat{\mathbb{E}}_s[f] \quad \text{for } f \text{ Borel-measurable}$$

Interjection: Least Fixed Points

Theorem (Transfer rule for least fixed points)

$$\frac{\sqcup\text{-continuous } \alpha, f, g \quad \alpha \perp = \perp \quad \alpha \circ f = g \circ \alpha}{\alpha(\text{lfp } f) = \text{lfp } g}$$

$$\hat{\mathbb{E}}_{(p,s)}[\text{COST}_{\text{stream}} f] = \text{lfp} \left(\bigsqcup_{K \text{ s}} \int \text{cost} \right) p \text{ s}$$

Correspondence Proof

Theorem

$$\hat{\mathbb{E}}_{(p,s)}[\text{COST}_{\text{stream}} f] = \text{ert } p f s$$

Correspondence Proof

Theorem

$$\hat{\mathbb{E}}_{(p,s)}[\text{COST}_{\text{stream}} f] = \text{ert } p f s$$

Proof by induction on p :

Correspondence Proof

Theorem

$$\hat{\mathbb{E}}_{(p,s)}[\text{COST}_{\text{stream}} f] = \text{ert } p f s$$

Proof by induction on p :

$p_1; p_2$ – Antisymmetry then fixed point induction

$$\begin{aligned} & \hat{\mathbb{E}}_{(p_1,s)}[\text{COST}_{\text{stream}} (\hat{\mathbb{E}}_{(p_2,\cdot)}[\text{COST}_{\text{stream}} f])] = \\ & \hat{\mathbb{E}}_{(p_1;p_2,s)}[\text{COST}_{\text{stream}} f] \end{aligned}$$

Correspondence Proof

Theorem

$$\hat{\mathbb{E}}_{(p,s)}[\text{COST}_{\text{stream}} f] = \text{ert } p f s$$

Proof by induction on p :

$p_1; p_2$ – Antisymmetry then fixed point induction

$$\hat{\mathbb{E}}_{(p_1,s)}[\text{COST}_{\text{stream}} (\hat{\mathbb{E}}_{(p_2,\cdot)}[\text{COST}_{\text{stream}} f])] =$$
$$\hat{\mathbb{E}}_{(p_1;p_2,s)}[\text{COST}_{\text{stream}} f]$$

WHILE g DO p_1 – Fixed point massaging

Correspondence Proof

Theorem

$$\hat{\mathbb{E}}_{(p,s)}[\text{COST}_{\text{stream}} f] = \text{ert } p f s$$

Proof by induction on p :

$p_1; p_2$ – Antisymmetry then fixed point induction

$$\hat{\mathbb{E}}_{(p_1,s)}[\text{COST}_{\text{stream}} (\hat{\mathbb{E}}_{(p_2,\cdot)}[\text{COST}_{\text{stream}} f])] = \\ \hat{\mathbb{E}}_{(p_1;p_2,s)}[\text{COST}_{\text{stream}} f]$$

WHILE g DO p_1 – Fixed point massaging

□

- Operational semantics & Correspondence ~ 330 lines of theory

Correspondence Proof

Theorem

$$\hat{\mathbb{E}}_{(p,s)}[\text{COST}_{\text{stream}} f] = \text{ert } p f s$$

Proof by induction on p :

$p_1; p_2$ – Antisymmetry then fixed point induction

$$\hat{\mathbb{E}}_{(p_1,s)}[\text{COST}_{\text{stream}} (\hat{\mathbb{E}}_{(p_2,\cdot)}[\text{COST}_{\text{stream}} f])] = \\ \hat{\mathbb{E}}_{(p_1;p_2,s)}[\text{COST}_{\text{stream}} f]$$

WHILE g DO p_1 – Fixed point massaging

□

- Operational semantics & Correspondence ~ 330 lines of theory
- Central to our proof: Expectation as fixed point

Correspondence Proof

Theorem

$$\hat{\mathbb{E}}_{(p,s)}[\text{COST}_{\text{stream}} f] = \text{ert } p f s$$

Proof by induction on p :

$p_1; p_2$ – Antisymmetry then fixed point induction

$$\begin{aligned} & \hat{\mathbb{E}}_{(p_1,s)}[\text{COST}_{\text{stream}} (\hat{\mathbb{E}}_{(p_2,\cdot)}[\text{COST}_{\text{stream}} f])] = \\ & \hat{\mathbb{E}}_{(p_1;p_2,s)}[\text{COST}_{\text{stream}} f] \end{aligned}$$

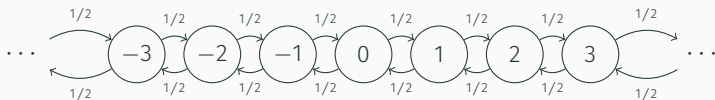
WHILE g DO p_1 – Fixed point massaging

□

- Operational semantics & Correspondence ~ 330 lines of theory
- Central to our proof: Expectation as fixed point
- [KKMO 2016]: Expectation as sums over all paths

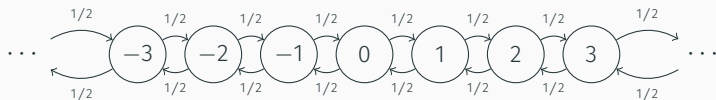
Examples

Simple Symmetric Random Walk (ssrw)



- $\Pr_i(\diamond j) = 1$
- $H_{ij} := \text{ert}(\text{ssrw } j) \perp i$
for $i \neq j$: $H_{ij} =$

Simple Symmetric Random Walk (ssrw)



- $\Pr_i(\diamond j) = 1$
- $H_{ij} := \text{ert}(\text{ssrw } j) \perp i$
for $i \neq j$: $H_{ij} = \infty$

Simple Symmetric Random Walk – [KKMO]

How do Kaminski, Katoen, Matheja, and Olmedo prove it?

Simple Symmetric Random Walk – [KKMO]

How do Kaminski, Katoen, Matheja, and Olmedo prove it?

- $\text{ert}(\text{ssrw } j) \perp i$ has lower ω -invariant $I_{n+1} \leq \text{ert STEP } I_n$:

$$I_n = 1 + \llbracket 0 < x \leq n \rrbracket \cdot \infty$$

How do Kaminski, Katoen, Matheja, and Olmedo prove it?

- $\text{ert}(\text{ssrw } j) \perp i$ has lower ω -invariant $I_{n+1} \leq \text{ert STEP } I_n$:

$$I_n = 1 + \llbracket 0 < x \leq n \rrbracket \cdot \infty$$

- They need to prove this equation:

$$1 + \llbracket x > 0 \rrbracket \cdot 2 + \llbracket 1 < x \leq n + 1 \rrbracket \cdot \infty + \llbracket 0 < x \leq n - 1 \rrbracket \cdot \infty = \\ 1 + \llbracket x > 0 \rrbracket \cdot 2 + \llbracket 0 < x \leq n + 1 \rrbracket \cdot \infty$$

Simple Symmetric Random Walk – [KKMO]

How do Kaminski, Katoen, Matheja, and Olmedo prove it?

- $\text{ert}(\text{ssrw } j) \perp i$ has lower ω -invariant $I_{n+1} \leq \text{ert STEP } I_n$:

$$I_n = 1 + \llbracket 0 < x \leq n \rrbracket \cdot \infty$$

- They need to prove this equation:

$$1 + \llbracket x > 0 \rrbracket \cdot 2 + \llbracket 1 < x \leq i + 1 \rrbracket \cdot \infty + \llbracket 0 < x \leq n - 1 \rrbracket \cdot \infty = \\ 1 + \llbracket x > 0 \rrbracket \cdot 2 + \llbracket 0 < x \leq n + 1 \rrbracket \cdot \infty$$

ERROR

Simple Symmetric Random Walk – [KKMO]

How do Kaminski, Katoen, Matheja, and Olmedo prove it?

- $\text{ert}(\text{ssrw } j) \perp i$ has lower ω -invariant $I_{n+1} \leq \text{ert STEP } I_n$:

$$I_n = 1 + \llbracket 0 < x \leq n \rrbracket \cdot \infty$$

- They need to prove this equation:

$$1 + \llbracket x > 0 \rrbracket \cdot 2 + \llbracket 1 < x \leq n + 1 \rrbracket \cdot \infty + \llbracket 0 < x \leq n - 1 \rrbracket \cdot \infty = \\ 1 + \llbracket x > 0 \rrbracket \cdot 2 + \llbracket 0 < x \leq n + 1 \rrbracket \cdot \infty$$

Fails for $x = 1$ and $n = 0$.

Simple Symmetric Random Walk – our Solution

Operational semantics – trace representation

Use $Hij = \hat{\mathbb{E}}_{(ssrw_{j,i})}[\text{COST}_{\text{stream}}]$ to prove

$$Hij = Hik + Hkj \quad \text{for } i \leq k \leq j$$

Simple Symmetric Random Walk – our Solution

Operational semantics – trace representation

Use $H_{ij} = \hat{\mathbb{E}}_{(ssrw_{j,i})}[\text{COST}_{\text{stream}}]$ to prove

$$H_{ij} = H_{ik} + H_{kj} \quad \text{for } i \leq k \leq j$$

Then derive $H_{ij} = \infty$ for $i \neq j$

Simple Symmetric Random Walk – our Solution

Operational semantics – trace representation

Use $H_{ij} = \hat{\mathbb{E}}_{(ssrw_{j,i})}[\text{COST}_{\text{stream}}]$ to prove

$$H_{ij} = H_{ik} + H_{kj} \quad \text{for } i \leq k \leq j$$

Then derive $H_{ij} = \infty$ for $i \neq j$ ✓

Coupon Collector

```
 $x := 0, i := 0, cp := \overbrace{[F, \dots, F]}^{N \text{ times}};$   
WHILE  $x < N$  DO  
  WHILE  $cp[i]$  DO  
     $i := \mathcal{U}(\{1, \dots, N\});$   
     $cp[i] := T, x := x + 1$ 
```

$$\text{ert } CC_N \text{ } 0 \text{ } s = 2 + N \cdot \left(4 + 2 \sum_{i=1}^N \frac{1}{i} \right).$$

Coupon Collector

```
 $x := 0, i := 0, cp := \overbrace{[F, \dots, F]}^{N \text{ times}};$   
WHILE  $x < N$  DO  
  WHILE  $cp[i]$  DO  
     $i := \mathcal{U}(\{1, \dots, N\});$   
     $cp[i] := T, x := x + 1$ 
```

```
 $c := 0, b := F;$   
WHILE  $c < N$  DO  
  WHILE  $b$  DO  
     $b := \mathcal{B}(x/N);$   
     $b := T, c := c + 1$ 
```

$$\text{ert } CC_N \text{ } 0 \text{ } s = 2 + N \cdot \left(4 + 2 \sum_{i=1}^N \frac{1}{i} \right).$$

Coupon Collector

$x := 0, i := 0, cp := \overbrace{[F, \dots, F]}^{N \text{ times}};$
WHILE $x < N$ DO
 WHILE $cp[i]$ DO
 $i := \mathcal{U}(\{1, \dots, N\});$
 $cp[i] := T, x := x + 1$

$x \rightarrow c$
 $cp[i] \rightarrow b$
 $|cp| = x$

$c := 0, b := F;$
WHILE $c < N$ DO
 WHILE b DO
 $b := \mathcal{B}(x/N);$
 $b := T, c := c + 1$

$$\text{ert } CC_N = 2 + N \cdot \left(4 + 2 \sum_{i=1}^N \frac{1}{i} \right).$$

Coupon Collector

$x := 0, i := 0, cp := \overbrace{[F, \dots, F]}^{N \text{ times}};$
WHILE $x < N$ DO
 WHILE $cp[i]$ DO
 $i := \mathcal{U}(\{1, \dots, N\});$
 $cp[i] := T, x := x + 1$

$x \rightarrow c$
 $cp[i] \rightarrow b$
 $|cp| = x$

$c := 0, b := F;$
WHILE $c < N$ DO
 WHILE b DO
 $b := \mathcal{B}(x/N);$
 $b := T, c := c + 1$

$$\text{ert } CC_N \text{ } 0 \text{ } s = 2 + N \cdot \left(4 + 2 \sum_{i=1}^N \frac{1}{i}\right).$$

KKMO use loop invariants to prove running time

Conclusion

Different pGCL formalizations already exist:

Different pGCL formalizations already exist:

Hurd *et al.* [QAPL 2003] VCG in HOL, deep

Different pGCL formalizations already exist:

Hurd *et al.* [QAPL 2003] VCG in HOL, deep

Celiku & McIver [N. J. C. 2004] Expected running time

- Allow cost analysis for pGCL
- Refinement only for upper bounds
- No relation to an operational model

Different pGCL formalizations already exist:

Hurd *et al.* [QAPL 2003] VCG in HOL, deep

Celiku & McIver [N. J. C. 2004] Expected running time

- Allow cost analysis for pGCL
- Refinement only for upper bounds
- No relation to an operational model

Cock [SSV 2012] VCG in Isabelle/HOL, shallow

Different pGCL formalizations already exist:

Hurd *et al.* [QAPL 2003] VCG in HOL, deep

Celiku & McIver [N. J. C. 2004] Expected running time

- Allow cost analysis for pGCL
- Refinement only for upper bounds
- No relation to an operational model

Cock [SSV 2012] VCG in Isabelle/HOL, shallow

Various other formalizations of purely probabilistic programs...

- Relating denotational and operational semantics

- Relating denotational and operational semantics
- Explore use cases where this relation simplifies proofs

- Relating denotational and operational semantics
- Explore use cases where this relation simplifies proofs
Loop invariants are not always enough

- Relating denotational and operational semantics
- Explore use cases where this relation simplifies proofs
Loop invariants are not always enough
- Explore: Probabilistic relational Hoare logic (pRHL) + non-determinism