Formalising Semantics for Expected Running Time of Probabilistic Programs (Rough Diamond)

Johannes Hölzl

TU München, Germany

This Talk

- Probabilistic programs (pGCL) + expected running time.

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] ⭐

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

This Talk

- Probabilistic programs (pGCL) + expected running time.
  Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ⭐
- Two semantics:
  Denotational: $\sigma\ \text{pgcl} \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right) \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right)$
  Operational: $(\sigma\ \text{pgcl} \times \sigma) \Rightarrow (\sigma\ \text{pgcl} \times \sigma)$ stream measure

This Talk

- Probabilistic programs (pGCL) + expected running time.
  Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ⭐
- Two semantics:
  Denotational: σ pgcl $\Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right) \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right)$
  Operational: $(\sigma$ pgcl $\times\ \sigma) \Rightarrow (\sigma$ pgcl $\times\ \sigma)$ stream measure
  Correspondence: Denotational $\Leftrightarrow$ Operational

This Talk

- Probabilistic programs (pGCL) + expected running time.
  Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ⭐
- Two semantics:
  Denotational: $\sigma\ \text{pgcl} \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right) \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right)$
  Operational: $(\sigma\ \text{pgcl} \times \sigma) \Rightarrow (\sigma\ \text{pgcl} \times \sigma)\ \text{stream measure}$
  Correspondence: Denotational $\Leftrightarrow$ Operational
- Examples:

This Talk

- Probabilistic programs (pGCL) + expected running time.
  Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ⭐
- Two semantics:
  Denotational: σ pgcl $\Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right) \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right)$
  Operational: $(\sigma$ pgcl $\times\ \sigma) \Rightarrow (\sigma$ pgcl $\times\ \sigma)$ stream measure
  Correspondence: Denotational $\Leftrightarrow$ Operational
- Examples:
  - Simple Random Walk ϟ

This Talk

- Probabilistic programs (pGCL) + expected running time.
  Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ⭐
- Two semantics:
  Denotational: $\sigma\ \text{pgcl} \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right) \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right)$
  Operational: $(\sigma\ \text{pgcl} \times \sigma) \Rightarrow (\sigma\ \text{pgcl} \times \sigma)$ stream measure
  Correspondence: Denotational $\Leftrightarrow$ Operational
- Examples:
  - Simple Random Walk ↯
  - Coupon Collector

clarified semantics

This Talk

- Probabilistic programs (pGCL) + expected running time.
  Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ⭐
- Two semantics:
  Denotational: $\sigma\ \text{pgcl} \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right) \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right)$
  Operational: $(\sigma\ \text{pgcl} \times \sigma) \Rightarrow (\sigma\ \text{pgcl} \times \sigma)$ stream measure
  Correspondence: Denotational $\Leftrightarrow$ Operational
- Examples:
  - Simple Random Walk ↯
  - Coupon Collector

clarified semantics – different proofs

This Talk

- Probabilistic programs (pGCL) + expected running time.
  Kaminski, Katoen, Matheja, and Olmedo [ESOP 2016] ⭐
- Two semantics:
  Denotational: σ pgcl $\Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right) \Rightarrow \left(\sigma \Rightarrow \overline{\mathbb{R}}_{\geq 0}\right)$
  Operational: $(\sigma$ pgcl $\times\, \sigma) \Rightarrow (\sigma$ pgcl $\times\, \sigma)$ stream measure
  Correspondence: Denotational $\Leftrightarrow$ Operational
- Examples:
  - Simple Random Walk ϟ
  - Coupon Collector

clarified semantics – different proofs – fixed proofs

Probabilistic Guarded Command Language (pGCL)

$$\sigma \; \text{pgcl} \quad = \quad \bot$$

Probabilistic Guarded Command Language (pGCL)

$$\sigma \ \mathrm{pgcl} \quad = \quad \bot \ | \ \text{€}$$

Probabilistic Guarded Command Language (pGCL)

$$\sigma \; \text{pgcl} \quad = \quad \bot \mid € \mid ϟ$$

Probabilistic Guarded Command Language (pGCL)

$$\sigma \text{ pgcl} \quad = \quad \bot \mid € \mid ϟ$$
$$\mid \quad x :\sim \mathcal{D} \quad \text{or} \quad x := \text{expr} \qquad \text{"Assign } (\sigma \Rightarrow \sigma \text{ pmf)"}$$

Probabilistic Guarded Command Language (pGCL)

$$
\begin{aligned}
\sigma\ \text{pgcl} \quad &= \quad \bot \mid \text{€} \mid \text{ϟ} \\
&\mid \quad x :\sim \mathcal{D} \quad \text{or} \quad x := \text{expr} \qquad \text{"Assign } (\sigma \Rightarrow \sigma \text{ pmf)"} \\
&\mid \quad p_1 \,;\, p_2
\end{aligned}
$$

Probabilistic Guarded Command Language (pGCL)

$$
\begin{array}{rcll}
\sigma \text{ pgcl} & = & \bot \mid \text{€} \mid \text{ϟ} & \\
& \mid & x :\sim \mathcal{D} \quad \text{or} \quad x := \text{expr} & \text{"Assign } (\sigma \Rightarrow \sigma \text{ pmf)"} \\
& \mid & p_1 \; ; \; p_2 & \\
& \mid & p_1 \mid p_2 &
\end{array}
$$

Probabilistic Guarded Command Language (pGCL)

$$
\begin{array}{lcll}
\sigma \text{ pgcl} & = & \bot \mid € \mid ϟ & \\
 & \mid & x :\sim \mathcal{D} \quad \text{or} \quad x := \text{expr} & \text{"Assign } (\sigma \Rightarrow \sigma \text{ pmf})\text{"} \\
 & \mid & p_1 \, ; \, p_2 & \\
 & \mid & p_1 \mid p_2 & \\
 & \mid & \text{ITE } g \; p_1 \; p_2 & g :: \sigma \Rightarrow \text{bool}
\end{array}
$$

Probabilistic Guarded Command Language (pGCL)

$$
\begin{array}{rcll}
\sigma\ \text{pgcl} & = & \bot \mid \text{€} \mid \text{ϟ} & \\
 & \mid & x :\sim \mathcal{D} \quad \text{or} \quad x := \text{expr} & \text{"Assign } (\sigma \Rightarrow \sigma \text{ pmf)"} \\
 & \mid & p_1 \ ; \ p_2 & \\
 & \mid & p_1 \mid p_2 & \\
 & \mid & \text{ITE } g\ p_1\ p_2 & g :: \sigma \Rightarrow \text{bool} \\
 & \mid & \text{WHILE } g \text{ DO } p &
\end{array}
$$

Denotational Semantics (Expected Running Time)

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

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)

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

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

Denotational Semantics (Expected Running Time)

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

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

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

Denotational Semantics (Expected Running Time)

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

$$\begin{aligned}
\mathrm{ert}\ \bot\ c &= c \\
\mathrm{ert}\ \text{€}\ c &= 1 + c \\
\mathrm{ert}\ \lightning\ c &= 0
\end{aligned}$$

Denotational Semantics (Expected Running Time)

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

$$\begin{aligned}
\text{ert}\ \bot\ & c &=&\ c \\
\text{ert}\ €\ & c &=&\ 1 + c \\
\text{ert}\ ϟ\ & c &=&\ 0 \\
\text{ert}\ (\text{Assign}\ u)\ & c &=&\ 1 + \left(\lambda x.\ \int_y c\ y\ d(u\ x)\right)
\end{aligned}$$

Denotational Semantics (Expected Running Time)

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

$$\begin{aligned}
\text{ert}\ \bot\ & c = c \\
\text{ert}\ €\ & c = 1 + c \\
\text{ert}\ ↯\ & c = 0 \\
\text{ert}\ (\text{Assign}\ u)\ & c = 1 + \left(\lambda x.\ \int_y c\ y\ d(u\ x)\right) \\
\text{ert}\ (p_1;\ p_2)\ & c = \text{ert}\ p_1\ (\text{ert}\ p_2\ c)
\end{aligned}$$

Denotational Semantics (Expected Running Time)

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

$$\begin{aligned}
\text{ert}\ \bot\ && c &= c \\
\text{ert}\ \text{€}\ && c &= 1 + c \\
\text{ert}\ \text{ϟ}\ && c &= 0 \\
\text{ert}\ (\text{Assign}\ u)\ && c &= 1 + \left(\lambda x.\ \int_y c\ y\ d(u\ x)\right) \\
\text{ert}\ (p_1;\ p_2)\ && c &= \text{ert}\ p_1\ (\text{ert}\ p_2\ c) \\
\text{ert}\ (p_1 \mid p_2)\ && c &= \text{ert}\ p_1\ c \sqcup \text{ert}\ p_2\ c
\end{aligned}$$

Denotational Semantics (Expected Running Time)

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

$$\begin{aligned}
&\text{ert}\ \bot && c &&= && c \\
&\text{ert}\ \text{€} && c &&= && 1 + c \\
&\text{ert}\ \text{↯} && c &&= && 0 \\
&\text{ert}\ (\text{Assign}\ u) && c &&= && 1 + \left(\lambda x.\ \int_y c\ y\ d(u\ x)\right) \\
&\text{ert}\ (p_1;\ p_2) && c &&= && \text{ert}\ p_1\ (\text{ert}\ p_2\ c) \\
&\text{ert}\ (p_1 \mid p_2) && c &&= && \text{ert}\ p_1\ c \sqcup \text{ert}\ p_2\ c \\
&\text{ert}\ (\text{ITE}\ g\ p_1\ p_2) && c &&= && 1 + (\lambda x.\ \text{if}\ g\ x\ \text{then}\ \text{ert}\ p_1\ c\ x\ \text{else}\ \text{ert}\ p_2\ c\ x)
\end{aligned}$$

Denotational Semantics (Expected Running Time)

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

$$\begin{array}{lcl}
\text{ert}\ \bot & c = & c \\
\text{ert}\ € & c = & 1 + c \\
\text{ert}\ ϟ & c = & 0 \\
\text{ert}\ (\text{Assign}\ u) & c = & 1 + \left(\lambda x.\ \int_y c\ y\ d(u\ x)\right) \\
\text{ert}\ (p_1;\ p_2) & c = & \text{ert}\ p_1\ (\text{ert}\ p_2\ c) \\
\text{ert}\ (p_1 \mid p_2) & c = & \text{ert}\ p_1\ c \sqcup \text{ert}\ p_2\ c \\
\text{ert}\ (\text{ITE}\ g\ p_1\ p_2) & c = & 1 + (\lambda x.\ \text{if}\ g\ x\ \text{then}\ \text{ert}\ p_1\ c\ x\ \text{else}\ \text{ert}\ p_2\ c\ x) \\
\text{ert}\ (\text{WHILE}\ g\ \text{DO}\ p) & c = & \text{lfp}\ (\lambda W x. 1 + \text{if}\ g\ x\ \text{then}\ \text{ert}\ p\ W x\ \text{else}\ c\ x)
\end{array}$$

Interjection: Markov decision processes

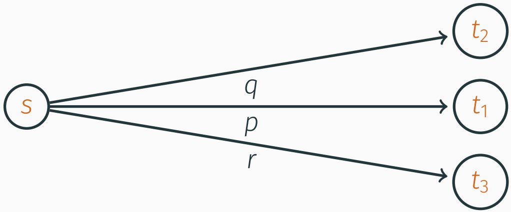
Automata with probabilistic and non-deterministic choice

$K :: \sigma \Rightarrow \sigma$ pmf set $\qquad K\,s \neq \emptyset$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

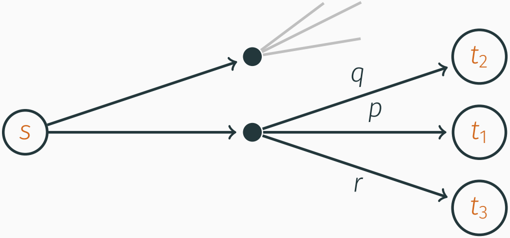
$K :: \sigma \Rightarrow \sigma$ pmf set $\qquad K\, s \neq \emptyset$



Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

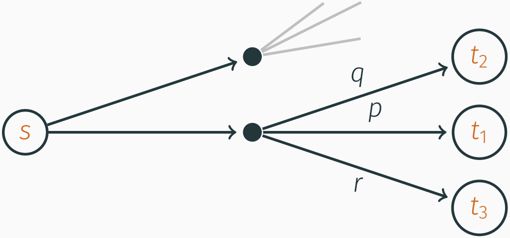
$K :: \sigma \Rightarrow \sigma \text{ pmf set} \qquad K\, s \neq \emptyset$



Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

$K :: \sigma \Rightarrow \sigma \text{ pmf set} \qquad K\,s \neq \emptyset$



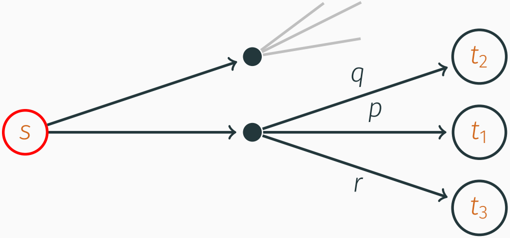
Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f:

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

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

$K :: \sigma \Rightarrow \sigma \text{ pmf set} \qquad K\, s \neq \emptyset$



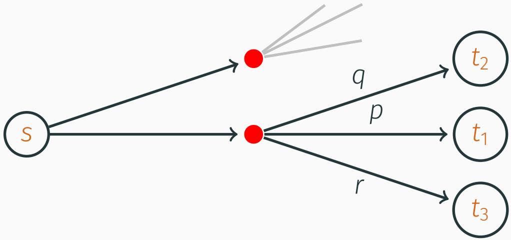
Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f:

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in K\, s} \int_t \hat{\mathbb{E}}_t[\lambda\omega.\, f\,(t \cdot \omega)] \; \mathrm{d}D$$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

$K :: \sigma \Rightarrow \sigma \text{ pmf set} \qquad K\, s \neq \emptyset$



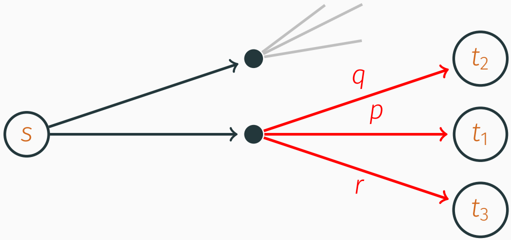
Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f:

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in K\, s} \int_t \hat{\mathbb{E}}_t[\lambda\omega.\, f\,(t \cdot \omega)] \; \mathrm{d}D$$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

$K :: \sigma \Rightarrow \sigma \text{ pmf set} \qquad K\,s \neq \emptyset$



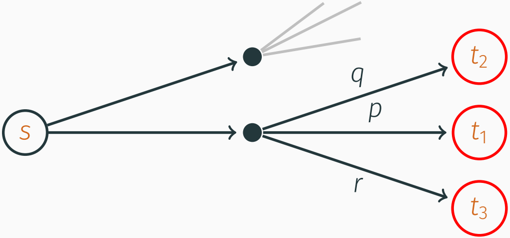
Construct maximal expectation $\hat{\mathbb{E}}_s[f]$ of a (cost) function f:

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in K\,s} \int_t \hat{\mathbb{E}}_t[\lambda\omega.\, f\,(t \cdot \omega)] \;\mathrm{d}D$$

Interjection: Markov decision processes

Automata with probabilistic and non-deterministic choice

$K :: \sigma \Rightarrow \sigma$ pmf set $\qquad K\,s \neq \emptyset$



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

$$\hat{\mathbb{E}}_s[f] = \bigsqcup_{D \in K\,s} \int_t \hat{\mathbb{E}}_t[\lambda\omega.\, f\,(t \cdot \omega)] \;\mathrm{d}D$$

Operational Semantics

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

Operational Semantics

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

$K(\bot, \quad s) = \ll \bot, s \gg$

$K(\text{€}, \quad s) = \ll \bot, s \gg$

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

Operational Semantics

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

$$\mathsf{K}(\bot, \quad s) = \ll \bot, s \gg$$

$$\mathsf{K}(\boldsymbol{€}, \quad s) = \ll \bot, s \gg$$

$$\mathsf{K}(\text{↯}, \quad s) = \ll \text{↯}, s \gg$$

$$\mathsf{K}(\mathsf{Assign}\ u, \quad s) = \left[\lambda s'.\ (\bot, s')\right]\ \{u\ s\}$$

Operational Semantics

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

$$K(\bot, \qquad s) = \ll \bot, s \gg$$

$$K(€, \qquad s) = \ll \bot, s \gg$$

$$K(ϟ, \qquad s) = \ll ϟ, s \gg$$

$$K(\text{Assign } u, \qquad s) = \left[\lambda s'.\ (\bot, s')\right] \{u\ s\}$$

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

Operational Semantics

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

$$\begin{aligned}
\mathsf{K}(\bot,\quad & s) = \ll \bot, s \gg \\
\mathsf{K}(\text{€},\quad & s) = \ll \bot, s \gg \\
\mathsf{K}(\text{ϟ},\quad & s) = \ll \text{ϟ}, s \gg \\
\mathsf{K}(\mathsf{Assign}\ u,\quad & s) = \left[\lambda s'.\ (\bot, s')\right]\ \{u\ s\} \\
\mathsf{K}(p_1 \mid p_2,\quad & s) = \ll p_1, s \gg \cup \ll p_2, s \gg \\
\mathsf{K}(\mathsf{ITE}\ g\ p_1\ p_2,\quad & s) = \mathsf{if}\ g\ s\ \mathsf{then}\ \ll p_1, s \gg\ \mathsf{else}\ \ll p_2, s \gg \\
\mathsf{K}(\mathsf{WHILE}\ g\ \mathsf{DO}\ p,\ & s) = \mathsf{if}\ g\ s\ \mathsf{then}\ \ll p;\ \mathsf{WHILE}\ g\ \mathsf{DO}\ p, s \gg\ \mathsf{else}\ \ll \bot, s \gg
\end{aligned}$$

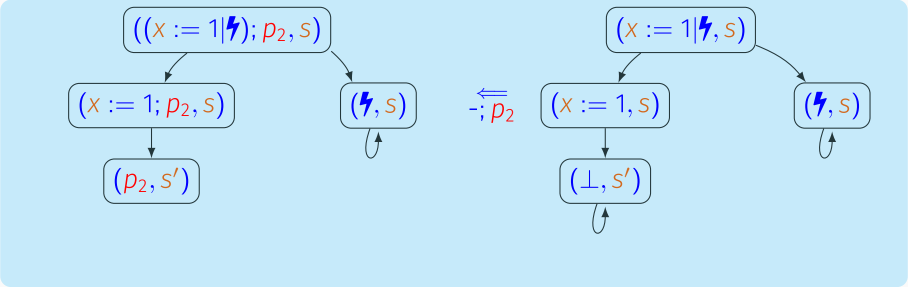
Operational Semantics

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

$$\begin{aligned}
\mathsf{K}(\bot, \quad & s) = \ll \bot, s \gg \\
\mathsf{K}(\text{€}, \quad & s) = \ll \bot, s \gg \\
\mathsf{K}(\lightning, \quad & s) = \ll \lightning, s \gg \\
\mathsf{K}(\mathsf{Assign}\ u, \quad & s) = \left[\lambda s'.\ (\bot, s')\right] \{u\ s\} \\
\mathsf{K}(p_1 \mid p_2, \quad & s) = \ll p_1, s \gg \cup \ll p_2, s \gg \\
\mathsf{K}(\mathsf{ITE}\ g\ p_1\ p_2, \quad & s) = \mathsf{if}\ g\ s\ \mathsf{then}\ \ll p_1, s \gg\ \mathsf{else}\ \ll p_2, s \gg \\
\mathsf{K}(\mathsf{WHILE}\ g\ \mathsf{DO}\ p, \ & s) = \mathsf{if}\ g\ s\ \mathsf{then}\ \ll p;\ \mathsf{WHILE}\ g\ \mathsf{DO}\ p, s \gg\ \mathsf{else}\ \ll \bot, s \gg \\
\mathsf{K}(p_1;\ p_2, \quad & s) = \left[\lambda \begin{array}{ll} (\bot, s'). & (p_2, s') \\ (\lightning, s'). & (\lightning, s') \\ (p', s'). & (p';\ p_2, s') \end{array}\right] \mathsf{K}(p_1, s)
\end{aligned}$$

Operational Semantics

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



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

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

$$\mathsf{K}(p_1;\ p_2,\ s) = \left[\lambda \begin{array}{ll} (\bot, s'). & (p_2, s') \\ (\lightning, s'). & (\lightning, s') \\ (p', s'). & (p';\ p_2, s') \end{array}\right] \mathsf{K}(p_1, s)$$

Traces

Trace	$\mathsf{cost}_{\mathsf{stream}}$
$((x := 1 \mid \lightning); y := 0, s)$	0

Cost per stream:

$$\mathsf{cost}_{\mathsf{stream}}\ f\ (s \cdot \omega) \stackrel{\mathsf{lfp}}{=} \mathsf{cost}\ f\ s\ (\mathsf{cost}_{\mathsf{stream}}\ f\ \omega)$$

Traces

Trace	$\mathsf{cost}_{\mathsf{stream}}$
$((x := 1 \mid ⚡); y := 0, s)$	0
$\Downarrow$	$+$
$(x := 1; y := 0, s)$	1

Cost per stream:

$$\mathsf{cost}_{\mathsf{stream}}\ f\ (s \cdot \omega) \stackrel{\mathrm{lfp}}{=} \mathsf{cost}\ f\ s\ (\mathsf{cost}_{\mathsf{stream}}\ f\ \omega)$$

Traces

Trace	$\mathrm{cost}_{\mathrm{stream}}$
$((x := 1 \mid \lightning); y := 0, s)$	0
$\Downarrow$	$+$
$(x := 1; y := 0, s)$	1
$\Downarrow$	$+$
$(y := 0, s')$	1

Cost per stream:

$$\mathrm{cost}_{\mathrm{stream}}\ f\ (s \cdot \omega) \overset{\mathrm{lfp}}{=} \mathrm{cost}\ f\ s\ (\mathrm{cost}_{\mathrm{stream}}\ f\ \omega)$$

Traces

Trace	$\mathsf{cost}_{\mathsf{stream}}$
$((x := 1 \mid ⚡); y := 0, s)$	0
$\Downarrow$	$+$
$(x := 1; y := 0, s)$	1
$\Downarrow$	$+$
$(y := 0, s')$	1
$\Downarrow$	$+$
$(\bot, s'')$	$f\, s''$
$\vdots$	

Cost per stream:

$$\mathsf{cost}_{\mathsf{stream}}\; f\; (s \cdot \omega) \overset{\mathsf{lfp}}{=} \mathsf{cost}\; f\; s\; (\mathsf{cost}_{\mathsf{stream}}\; f\; \omega)$$

Interjection: Least Fixed Points

Theorem (Transfer rule for least fixed points)

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

Theorem (Transfer rule for least fixed points)

$$\frac{\sqcup\text{–continuous } \alpha, f, g \qquad \alpha\ \bot = \bot \qquad \alpha \circ f = g \circ \alpha}{\alpha(\mathsf{lfp}\ f) = \mathsf{lfp}\ g}$$

$$\begin{aligned}
\alpha(\mathsf{lfp}\ f) &= \alpha \circ f \circ f \circ f \circ f \circ f \circ \cdots \circ \bot \\
&= g \circ g \circ g \circ \alpha \circ f \circ f \circ \cdots \circ \bot \\
&= g \circ g \circ g \circ g \circ g \circ \cdots \circ \alpha\ \bot \\
&= \mathsf{lfp}\ g
\end{aligned}$$

Interjection: Least Fixed Points

Theorem (Transfer rule for least fixed points)

$$\frac{\sqcup\text{-continuous } \alpha, f, g \qquad \alpha\,\bot = \bot \qquad \alpha \circ f = g \circ \alpha}{\alpha(\mathrm{lfp}\ f) = \mathrm{lfp}\ g}$$

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

Interjection: Least Fixed Points

Theorem (Transfer rule for least fixed points)

$$\frac{\sqcup\text{-continuous } \alpha, f, g \qquad \alpha\ \bot = \bot \qquad \alpha \circ f = g \circ \alpha}{\alpha(\mathsf{lfp}\ f) = \mathsf{lfp}\ g}$$

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

Correspondence Proof

Theorem

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

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

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

Correspondence Proof

Theorem

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

Proof by induction on p:

$p_1; p_2$ – Antisymmetry then fixed point induction

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

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 $\sim$ 330 lines of theory

Correspondence Proof

Theorem

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

Proof by induction on p:

$p_1; p_2$ – Antisymmetry then fixed point induction

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

WHILE g DO p_1 – Fixed point massaging

$\square$

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

Correspondence Proof

Theorem

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

Proof by induction on p:

$p_1; p_2$ – Antisymmetry then fixed point induction

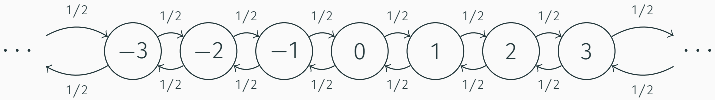
$$\hat{\mathbb{E}}_{(p_1,s)}[\mathrm{cost}_{\mathrm{stream}}\ (\hat{\mathbb{E}}_{(p_2,\cdot)}[\mathrm{cost}_{\mathrm{stream}}\ f])] = \hat{\mathbb{E}}_{(p_1;p_2,s)}[\mathrm{cost}_{\mathrm{stream}}\ f]$$

WHILE g DO p_1 – Fixed point massaging

□

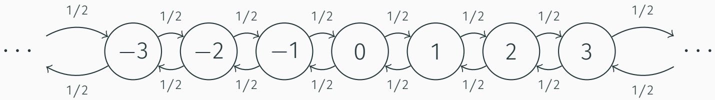
- Operational semantics & Correspondence ∼ 330 lines of theory
- Central to our proof: Expectation as fixed point
- [KKMO 2016]: Expectation as sums over all paths

Simple Symmetric Random Walk (ssrw)



- $\Pr_i(\Diamond j) = 1$
- $H\ i\ j := \mathrm{ert}\ (\mathrm{ssrw}\ j)\ \bot\ i$
  for $i \neq j$: $H\ i\ j =$

Simple Symmetric Random Walk (ssrw)



- $\Pr_i(\Diamond j) = 1$
- $H\ i\ j := \mathrm{ert}\ (\mathrm{ssrw}\ j)\ \bot\ i$
  for $i \neq j$: $H\ i\ j = \infty$

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?

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

$$I_n = 1 + [\![0 < x \leq n]\!] \cdot \infty$$

Simple Symmetric Random Walk – [KKMO]

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

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

$$I_n = 1 + [\![0 < x \leq n]\!] \cdot \infty$$

- They need to prove this equation:
$1 + [\![x > 0]\!] \cdot 2 + [\![1 < x \leq n + 1]\!] \cdot \infty + [\![0 < x \leq n - 1]\!] \cdot \infty =$
$1 + [\![x > 0]\!] \cdot 2 + [\![0 < x \leq n + 1]\!] \cdot \infty$

Simple Symmetric Random Walk – [KKMO]

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

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

$$I_n = 1 + [\![0 < x \leq n]\!] \cdot \infty$$

- They need to prove this equation:
  $1 + [\![x > 0]\!] \cdot 2 + [\![1 < x \leq n+1]\!] \cdot \infty + [\![0 < x \leq n-1]\!] \cdot \infty =$
  $1 + [\![x > 0]\!] \cdot 2 + [\![0 < x \leq n+1]\!] \cdot \infty$

ERROR

Simple Symmetric Random Walk – [KKMO]

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

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

$$I_n = 1 + [\![0 < x \leq n]\!] \cdot \infty$$

- They need to prove this equation:
$1 + [\![x > 0]\!] \cdot 2 + [\![1 < x \leq n + 1]\!] \cdot \infty + [\![0 < x \leq n - 1]\!] \cdot \infty =$
$1 + [\![x > 0]\!] \cdot 2 + [\![0 < x \leq n + 1]\!] \cdot \infty$

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

Simple Symmetric Random Walk – our Solution

Operational semantics – trace representation

Use $H\ i\ j = \hat{\mathbb{E}}_{(\text{ssrw}\ j, i)}[\text{cost}_{\text{stream}}]$ to prove

$$H\ i\ j = H\ i\ k + H\ k\ j \quad \text{for} \quad i \leq k \leq j$$

Simple Symmetric Random Walk – our Solution

Operational semantics – trace representation

Use $H\ i\ j = \hat{\mathbb{E}}_{(\text{ssrw}\ j, i)}[\text{cost}_{\text{stream}}]$ to prove

$$H\ i\ j = H\ i\ k + H\ k\ j \quad \text{for} \quad i \leq k \leq j$$

Then derive $H\ i\ j = \infty$ for $i \neq j$

Simple Symmetric Random Walk – our Solution

Operational semantics – trace representation

Use $H\ i\ j = \hat{\mathbb{E}}_{(\mathrm{ssrw}\ j, i)}[\mathrm{cost}_{\mathrm{stream}}]$ to prove

$$H\ i\ j = H\ i\ k + H\ k\ j \quad \text{for} \quad i \leq k \leq j$$

Then derive $H\ i\ j = \infty$ for $i \neq j$ ✓

Coupon Collector

$$
\begin{aligned}
&x := 0, i := 0, cp := [\overbrace{F, \ldots, F}^{N \text{ times}}];\\
&\text{WHILE } x < N \text{ DO}\\
&\quad \text{WHILE } cp[i] \text{ DO}\\
&\quad\quad i :\sim \mathcal{U}(\{1, \ldots, N\});\\
&\quad cp[i] := T, x := x + 1
\end{aligned}
$$

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

Coupon Collector

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

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

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

Coupon Collector

$$
\begin{array}{lcl}
x := 0, i := 0, cp := [\overbrace{F, \ldots, F}^{N \text{ times}}]; & & c := 0, b := F; \\
\text{WHILE } x < N \text{ DO} & x \to c & \text{WHILE } c < N \text{ DO} \\
\quad \text{WHILE } cp[i] \text{ DO} & cp[i] \to b & \quad \text{WHILE } b \text{ DO} \\
\quad\quad i :\sim \mathcal{U}(\{1, \ldots, N\}); & |cp| = x & \quad\quad b :\sim \mathcal{B}(x/N); \\
\quad cp[i] := T, x := x + 1 & & \quad b := T, c := c + 1
\end{array}
$$

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

Coupon Collector

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

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

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

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

KKMO use loop invariants to prove running time

Related Work

Different pGCL formalizations already exist:

Related Work

Different pGCL formalizations already exist:

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

Related Work

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

Related Work

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

Conclusion & Future Work

- Relating denotational and operational semantics

Conclusion & Future Work

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

Conclusion & Future Work

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

Conclusion & Future Work

- 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