Conditioning in Probabilistic Programming

MFPS XXXI 2015

Friedrich Gretz Nils Jansen Benjamin Kaminski
Joost-Pieter Katoen Annabelle McIver Federico Olmedo

Lehrstuhl für Informatik 2 Softwaremodellierung und Verifikation | RWTH AACHEN UNIVERSITY

23.6.2015

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad \longrightarrow \quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\ [p]\ \{P\}$$
$$\mid \{P\} \,\square\, \{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad \longrightarrow \quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\ [p]\ \{P\}$$
$$\mid \{P\} \,\Box\, \{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad \longrightarrow \quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\;[p]\;\{P\}$$
$$\mid \{P\} \,\square\, \{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad \longrightarrow \quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\ [p]\ \{P\}$$
$$\mid \{P\} \,\square\, \{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad \longrightarrow \quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\ [p]\ \{P\}$$
$$\mid \{P\} \,\square\, \{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad\longrightarrow\quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\;[p]\;\{P\}$$
$$\mid \{P\}\,\Box\,\{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad \longrightarrow \quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\ [p]\ \{P\}$$
$$\mid \{P\} \,\square\, \{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Syntax of cpGCL Programs

$$P \quad \longrightarrow \quad \text{“see pGCL”} \mid \texttt{observe}\ G$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad \longrightarrow \quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\ [p]\ \{P\}$$
$$\mid \{P\}\,\square\,\{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Syntax of cpGCL Programs

$$P \quad \longrightarrow \quad \text{“see pGCL”} \mid \texttt{observe}\ G$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \quad \longrightarrow \quad x \coloneqq E \mid \texttt{if}\,(G)\,\{P\}\,\texttt{else}\,\{P\} \mid \{P\}\;[p]\;\{P\}$$
$$\mid \{P\}\,\square\,\{P\} \mid \texttt{while}\,(G)\,\{P\}$$

Syntax of cpGCL Programs

$$P \quad \longrightarrow \quad \text{“see pGCL”} \mid \texttt{observe}\ G$$

Given a probabilistic program P and a random variable f: What is the conditional expected value of f after termination of P given that no observation is violated while executing P?

Expectations

Unbounded and Bounded Expectations

$\mathbb{S} = \{\sigma \mid \sigma \colon \mathcal{V}ars \to \mathbb{R}\}$ denotes the set of program states.

Expectations

Unbounded and Bounded Expectations

$\mathbb{S} = \{\sigma \mid \sigma \colon \mathcal{V}ars \to \mathbb{R}\}$ denotes the set of program states. The set of expectations $\mathbb{E}$ and the set of bounded expectations $\mathbb{E}_{\leq 1}$ are

$$\mathbb{E} \;=\; \left\{f \;\middle|\; f \colon \mathbb{S} \to \mathbb{R}_{\geq 0}^{\infty}\right\} \qquad\qquad \mathbb{E}_{\leq 1} \;=\; \{g \mid g \colon \mathbb{S} \to [0,\, 1]\}\ .$$

Expectations

Unbounded and Bounded Expectations

$\mathbb{S} = \{\sigma \mid \sigma \colon \mathcal{V}ars \to \mathbb{R}\}$ denotes the set of program states. The set of expectations $\mathbb{E}$ and the set of bounded expectations $\mathbb{E}_{\leq 1}$ are

$$\mathbb{E} \;=\; \left\{f \;\middle|\; f \colon \mathbb{S} \to \mathbb{R}_{\geq 0}^{\infty}\right\} \qquad \mathbb{E}_{\leq 1} \;=\; \{g \mid g \colon \mathbb{S} \to [0,\, 1]\}\ .$$

random variable = expectation

Expectations

Unbounded and Bounded Expectations

$\mathbb{S} = \{\sigma \mid \sigma \colon \mathcal{V}ars \to \mathbb{R}\}$ denotes the set of program states. The set of expectations $\mathbb{E}$ and the set of bounded expectations $\mathbb{E}_{\leq 1}$ are

$$\mathbb{E} \;=\; \left\{f \;\middle|\; f \colon \mathbb{S} \to \mathbb{R}_{\geq 0}^{\infty}\right\} \qquad \mathbb{E}_{\leq 1} \;=\; \{g \mid g \colon \mathbb{S} \to [0,\, 1]\}\ .$$

random variable = expectation ≠ expected value

Operational Semantics for cpGCL

Operational Reward Markov Decision Process (RMDP)

Operational Reward Markov Decision Process (RMDP)

- Given:

Operational Reward Markov Decision Process (RMDP)

- Given: $P \in$ cpGCL,

Operational Reward Markov Decision Process (RMDP)

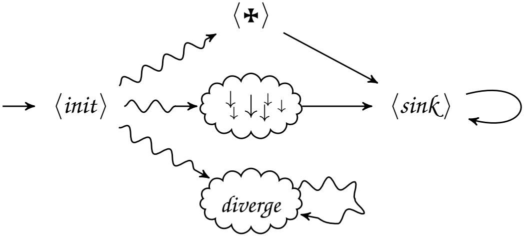
- Given: $P \in$ cpGCL, $f \in \mathbb{E}$.

Operational Reward Markov Decision Process (RMDP)

- Given: $P \in$ cpGCL, $f \in \mathbb{E}$.
- Construct an operational RMDP à la Gretz et al. and define conditional expected rewards under a minimizing scheduler.

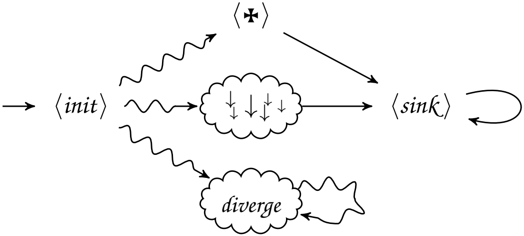
Operational Reward Markov Decision Process (RMDP)

- Given: $P \in$ cpGCL, $f \in \mathbb{E}$.
- Construct an operational RMDP à la Gretz et al. and define conditional expected rewards under a minimizing scheduler.
- Schematically such RMDPs look as follows:



Operational Reward Markov Decision Process (RMDP)

- Given: $P \in$ cpGCL, $f \in \mathbb{E}$.
- Construct an operational RMDP à la Gretz et al. and define conditional expected rewards under a minimizing scheduler.
- Schematically such RMDPs look as follows:



Only terminal states can contribute positive non–zero reward!

Denotational Semantics

Denotational Semantics for (unconditioned) pGCL

Programs as Unconditional Expectation Transformers

Programs as Unconditional Expectation Transformers

Think of a pGCL program P as a state–to–distribution transformer

$$\sigma \mapsto [\![P]\!](\sigma).$$

Programs as Unconditional Expectation Transformers

Think of a pGCL program P as a state–to–distribution transformer

$$\sigma \mapsto [\![P]\!](\sigma).$$

An expectation $\mathrm{wp}[P](f)$ is called the weakest pre–expectation of P with respect to post–expectation f

Programs as Unconditional Expectation Transformers

Think of a pGCL program P as a state–to–distribution transformer

$$\sigma \mapsto [\![P]\!](\sigma).$$

An expectation $\mathsf{wp}[P](f)$ is called the weakest pre–expectation of P with respect to post–expectation f, if

$$\mathbf{E}_{[\![P]\!](\sigma)}(f) \;=\; \mathbf{E}_{\delta_\sigma}\big(\mathsf{wp}[P](f)\big)$$

Programs as Unconditional Expectation Transformers

Think of a pGCL program P as a state–to–distribution transformer

$$\sigma \mapsto [\![P]\!](\sigma).$$

An expectation $\mathsf{wp}[P](f)$ is called the weakest pre–expectation of P with respect to post–expectation f, if

$$\mathbf{E}_{[\![P]\!](\sigma)}(f) \;=\; \mathbf{E}_{\delta_\sigma}\big(\mathsf{wp}[P](f)\big) \;=\; \mathsf{wp}[P](f)(\sigma).$$

Programs as Unconditional Expectation Transformers

Think of a pGCL program P as a state–to–distribution transformer

$$\sigma \mapsto \llbracket P \rrbracket(\sigma).$$

An expectation $\mathsf{wp}[P](f)$ is called the weakest pre–expectation of P with respect to post–expectation f, if

$$\mathbf{E}_{\llbracket P \rrbracket(\sigma)}(f) \;=\; \mathbf{E}_{\delta_\sigma}\big(\mathsf{wp}[P](f)\big) \;=\; \mathsf{wp}[P](f)(\sigma).$$

For $g \in \mathbb{E}_{\leq 1}$, an expectation $\mathsf{wlp}[P](g)$ is called the weakest liberal pre–expectation of P with respect to post–expectation g

Programs as Unconditional Expectation Transformers

Think of a pGCL program P as a state–to–distribution transformer

$$\sigma \mapsto [\![P]\!](\sigma).$$

An expectation $\mathsf{wp}[P](f)$ is called the weakest pre–expectation of P with respect to post–expectation f, if

$$\mathbf{E}_{[\![P]\!](\sigma)}(f) \;=\; \mathbf{E}_{\delta_\sigma}\big(\mathsf{wp}[P](f)\big) \;=\; \mathsf{wp}[P](f)(\sigma).$$

For $g \in \mathbb{E}_{\leq 1}$, an expectation $\mathsf{wlp}[P](g)$ is called the weakest liberal pre–expectation of P with respect to post–expectation g, if

$$\mathsf{wlp}[P](g) \;=\; \mathsf{wp}[P](g) + \mathsf{Pr}(\text{“}P\text{ diverges”})$$

Denotational Semantics for Fully Probabilistic cpGCL

Programs as Conditional Expectation Transformers

Programs as Conditional Expectation Transformers

Let $P \in \text{cpGCL}^{\boxtimes}$ (i.e. P contains no non–deterministic choices) and let $f \in \mathbb{E}$.

Programs as Conditional Expectation Transformers

Let $P \in \text{cpGCL}^{\boxtimes}$ (i.e. P contains no non–deterministic choices) and let $f \in \mathbb{E}$.

The conditional weakest pre–expectation of P with respect to post–expectation f is a pair (f', g'), such that

Programs as Conditional Expectation Transformers

Let $P \in \text{cpGCL}^{\boxtimes}$ (i.e. P contains no non–deterministic choices) and let $f \in \mathbb{E}$.

The conditional weakest pre–expectation of P with respect to post–expectation f is a pair (f', g'), such that

- given an initial state σ

Programs as Conditional Expectation Transformers

Let $P \in \mathrm{cpGCL}^{\boxtimes}$ (i.e. P contains no non–deterministic choices) and let $f \in \mathbb{E}$.

The conditional weakest pre–expectation of P with respect to post–expectation f is a pair (f', g'), such that

- given an initial state σ
- ${}^{f'(\sigma)}/_{g'(\sigma)}$ is the conditional expected value of f

Programs as Conditional Expectation Transformers

Let $P \in \text{cpGCL}^{\boxtimes}$ (i.e. P contains no non–deterministic choices) and let $f \in \mathbb{E}$.

The conditional weakest pre–expectation of P with respect to post–expectation f is a pair (f', g'), such that

- given an initial state σ
- ${}^{f'(\sigma)}/_{g'(\sigma)}$ is the conditional expected value of f
- after termination of P on σ

Programs as Conditional Expectation Transformers

Let $P \in \text{cpGCL}^{\boxtimes}$ (i.e. P contains no non–deterministic choices) and let $f \in \mathbb{E}$.

The conditional weakest pre–expectation of P with respect to post–expectation f is a pair (f', g'), such that

- given an initial state σ
- ${}^{f'(\sigma)}/_{g'(\sigma)}$ is the conditional expected value of f
- after termination of P on σ
- given that no observation is violated while executing P.

Programs as Conditional Expectation Transformers

Let $P \in \text{cpGCL}^{\boxtimes}$ (i.e. P contains no non–deterministic choices) and let $f \in \mathbb{E}$.

The conditional weakest pre–expectation of P with respect to post–expectation f is a pair (f', g'), such that

- given an initial state σ
- ${}^{f'(\sigma)}/_{g'(\sigma)}$ is the conditional expected value of f
- after termination of P on σ
- given that no observation is violated while executing P.

We provide a transformer that satisfies $\text{cwp}[P](f, \mathbf{1}) = (f', g')$.

Programs as Conditional Expectation Transformers

Let $P \in \mathsf{cpGCL}^{\boxtimes}$ (i.e. P contains no non–deterministic choices) and let $f \in \mathbb{E}$.

The conditional weakest pre–expectation of P with respect to post–expectation f is a pair (f', g'), such that

- given an initial state σ
- ${}^{f'(\sigma)}/_{g'(\sigma)}$ is the conditional expected value of f
- after termination of P on σ
- given that no observation is violated while executing P.

We provide a transformer that satisfies $\mathsf{cwp}[P](f, \mathbf{1}) = (f', g')$. Furthermore:

$$\mathbf{cwp}[P](f,\, g) \;=\; \big(\mathbf{wp}[P](f),\, \mathbf{wlp}[P](g)\big)$$

Conditional Expectations

Conditional Expectations

Let G be some event and χ_G be the characteristic function of G.

Conditional Expectations

Let G be some event and χ_G be the characteristic function of G. The conditional expected value of a random variable r given event G is then given by

$$\frac{\mathsf{E}(r \cdot \chi_G)}{\mathsf{Pr}(G)}$$

Conditional Expectations

Let G be some event and χ_G be the characteristic function of G. The conditional expected value of a random variable r given event G is then given by

$$\frac{\mathbf{E}(r \cdot \chi_G)}{\Pr(G)} = \frac{\mathbf{E}(r \cdot \chi_G)}{\mathbf{E}(\chi_G)} .$$

Conditional Expectations

Let G be some event and χ_G be the characteristic function of G. The conditional expected value of a random variable r given event G is then given by

$$\frac{\mathbf{E}(r \cdot \chi_G)}{\Pr(G)} \;=\; \frac{\mathbf{E}(r \cdot \chi_G)}{\mathbf{E}(\chi_G)}\;.$$

Conditional Expectations

Conditional Expectations

Let G be some event and χ_G be the characteristic function of G. The conditional expected value of a random variable r given event G is then given by

$$\frac{\mathbf{E}(r \cdot \chi_G)}{\Pr(G)} \;=\; \frac{\mathbf{E}(r \cdot \chi_G)}{\mathbf{E}(\chi_G)}\;.$$

Conditional Expectations

A conditional expectation is a pair $(f,\, g) \in \mathbb{E} \times \mathbb{E}_{\leq 1}$.

Conditional Expectations

Let G be some event and χ_G be the characteristic function of G. The conditional expected value of a random variable r given event G is then given by

$$\frac{\mathbf{E}(r \cdot \chi_G)}{\mathsf{Pr}(G)} = \frac{\mathbf{E}(r \cdot \chi_G)}{\mathbf{E}(\chi_G)} .$$

Conditional Expectations

A conditional expectation is a pair $(f, g) \in \mathbb{E} \times \mathbb{E}_{\leq 1}$. Intuitively, f represents the numerator and g represents the denominator above.

The Four Possibilities

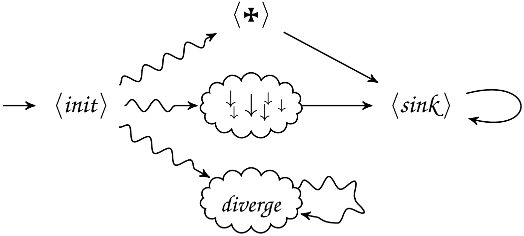
$$\frac{\textbf{wp}}{\textbf{wlp}}$$

The Four Possibilities

$$\frac{\mathbf{wp}}{\mathbf{wlp}} \sim \mathbf{cwp}$$

The Four Possibilities

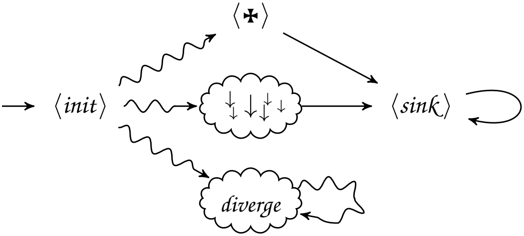
$$\frac{\textbf{wp}}{\textbf{wlp}} \sim \textbf{cwp}$$



The Four Possibilities

$$\frac{\textbf{wp}}{\textbf{wlp}} \sim \textbf{cwp} \qquad\qquad \frac{\textbf{wp}}{\textbf{wp}}$$

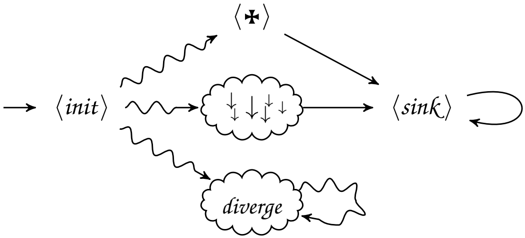
$$\frac{\textbf{wlp}}{\textbf{wlp}} \qquad\qquad \frac{\textbf{wlp}}{\textbf{wp}}$$



The Four Possibilities

$$\frac{\textbf{wp}}{\textbf{wlp}} \sim \textbf{cwp} \qquad\qquad \frac{\textbf{wp}}{\textbf{wp}} = \text{Nori et al.}$$

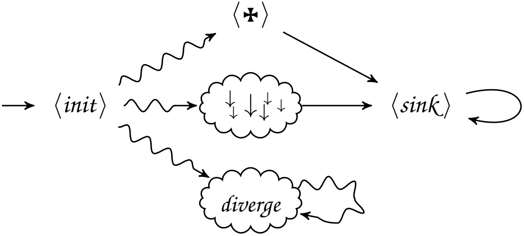
$$\frac{\textbf{wlp}}{\textbf{wlp}} \qquad\qquad \frac{\textbf{wlp}}{\textbf{wp}}$$



The Four Possibilities

$$\frac{\textbf{wp}}{\textbf{wlp}} \sim \textbf{cwp} \qquad\qquad \frac{\textbf{wp}}{\textbf{wp}} = \text{Nori et al.}$$

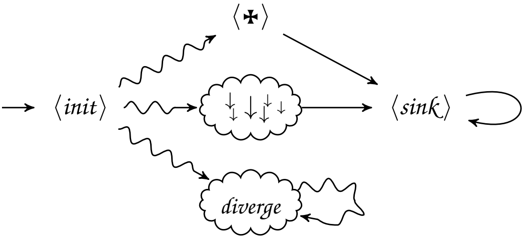
$$\frac{\textbf{wlp}}{\textbf{wlp}} \sim \textbf{cwlp} \qquad\qquad \frac{\textbf{wlp}}{\textbf{wp}}$$



The Four Possibilities

$$\frac{\textbf{wp}}{\textbf{wlp}} \sim \textbf{cwp} \qquad \frac{\textbf{wp}}{\textbf{wp}} = \text{Nori et al.}$$

$$\frac{\textbf{wlp}}{\textbf{wlp}} \sim \textbf{cwlp} \qquad \frac{\textbf{wlp}}{\textbf{wp}} = \text{bogus}$$



A Somewhat Intricate Example

$P_{div} =$

$x \coloneqq 1;$
$\texttt{while}\ (x = 1)\ \{$
$\quad x \coloneqq 1$
$\}$

$P_{andiv} =$

$x \coloneqq 1;$
$\texttt{while}\ (x = 1)\ \{$
$\quad \{x \coloneqq 1\}\ [1/2]\ \{x \coloneqq 0\};$
$\quad \texttt{observe}\ x = 1$
$\}$

A Somewhat Intricate Example

$$
\begin{aligned}
&P_{div} = \\
&\qquad x := 1; \\
&\qquad \texttt{while}\ (x = 1)\ \{ \\
&\qquad\qquad x := 1 \\
&\qquad \}
\end{aligned}
$$

$$
\begin{aligned}
&P_{andiv} = \\
&\qquad x := 1; \\
&\qquad \texttt{while}\ (x = 1)\ \{ \\
&\qquad\qquad \{x := 1\}\ [1/2]\ \{x := 0\}; \\
&\qquad\qquad \texttt{observe}\ x = 1 \\
&\qquad \}
\end{aligned}
$$

cwp of P_{div} and P_{andiv} yields:

A Somewhat Intricate Example

$$
\begin{aligned}
&P_{div} = \\
&\qquad x := 1; \\
&\qquad \texttt{while } (x = 1)\ \{ \\
&\qquad\qquad x := 1 \\
&\qquad \}
\end{aligned}
$$

$$
\begin{aligned}
&P_{andiv} = \\
&\qquad x := 1; \\
&\qquad \texttt{while } (x = 1)\ \{ \\
&\qquad\qquad \{x := 1\}\ [1/2]\ \{x := 0\}; \\
&\qquad\qquad \texttt{observe } x = 1 \\
&\qquad \}
\end{aligned}
$$

cwp of P_{div} and P_{andiv} yields:

- $\text{cwp}[P_{div}](f, \mathbf{1}) = (\mathbf{0}, \mathbf{1})$ $\qquad$ $\text{cwp}[P_{andiv}](f, \mathbf{1}) = (\mathbf{0}, \mathbf{0})$

A Somewhat Intricate Example

$P_{div} =$

$x := 1;$

$\texttt{while } (x = 1) \ \{$

$\quad x := 1$

$\}$

$P_{andiv} =$

$x := 1;$

$\texttt{while } (x = 1) \ \{$

$\quad \{x := 1\} \ [1/2] \ \{x := 0\};$

$\quad \texttt{observe } x = 1$

$\}$

cwp of P_{div} and P_{andiv} yields:

- $\mathsf{cwp}[P_{div}](f, \mathbf{1}) = (\mathbf{0}, \mathbf{1})$ $\qquad \mathsf{cwp}[P_{andiv}](f, \mathbf{1}) = (\mathbf{0}, \mathbf{0})$

P_{div} and P_{andiv} <u>are not</u> semantically equivalent!

A Somewhat Intricate Example

$P_{div} =$
  $x := 1;$
  $\texttt{while}\ (x = 1)\ \{$
    $x := 1$
  $\}$

$P_{andiv} =$
  $x := 1;$
  $\texttt{while}\ (x = 1)\ \{$
    $\{x := 1\}\ [1/2]\ \{x := 0\};$
    $\texttt{observe}\ x = 1$
  $\}$

cwp of P_{div} and P_{andiv} yields:

- $\mathsf{cwp}[P_{div}](f,\, \mathbf{1}) = (\mathbf{0},\, \mathbf{1})$ $\mathsf{cwp}[P_{andiv}](f,\, \mathbf{1}) = (\mathbf{0},\, \mathbf{0})$
- "$\frac{\mathsf{wp}}{\mathsf{wp}}$"$[P_{div}](f) = \frac{0}{0} =$ undef.   "$\frac{\mathsf{wp}}{\mathsf{wp}}$"$[P_{andiv}](f) = \frac{0}{0} =$ undef.

P_{div} and P_{andiv} <u>are not</u> semantically equivalent!

A Somewhat Intricate Example

$$
\begin{array}{l}
P_{div} = \\
\quad x := 1; \\
\quad \texttt{while } (x = 1) \ \{ \\
\qquad x := 1 \\
\quad \}
\end{array}
\qquad
\begin{array}{l}
P_{andiv} = \\
\quad x := 1; \\
\quad \texttt{while } (x = 1) \ \{ \\
\qquad \{x := 1\} \ [1/2] \ \{x := 0\}; \\
\qquad \texttt{observe } x = 1 \\
\quad \}
\end{array}
$$

cwp of P_{div} and P_{andiv} yields:

- $\mathsf{cwp}[P_{div}](f, \mathbf{1}) = (\mathbf{0}, \mathbf{1})$ $\qquad$ $\mathsf{cwp}[P_{andiv}](f, \mathbf{1}) = (\mathbf{0}, \mathbf{0})$
- "$\frac{\mathsf{wp}}{\mathsf{wp}}$"$[P_{div}](f) = \frac{0}{0} =$ undef. $\qquad$ "$\frac{\mathsf{wp}}{\mathsf{wp}}$"$[P_{andiv}](f) = \frac{0}{0} =$ undef.

P_{div} and P_{andiv} are not semantically equivalent!

Not distinguished by Nori et al.'s semantics!

Theorem: cwp is “Backward Compatible”

Theorem: cwp is "Backward Compatible"

Let $Q \in$ pGCL (i.e. Q contains no observations),

Theorem: cwp is "Backward Compatible"

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$

Theorem: cwp is "Backward Compatible"

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f', g') = \mathsf{cwp}[Q](f, \mathbf{1})$.

Theorem: cwp is “Backward Compatible”

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f',\, g') = \mathsf{cwp}[Q](f,\, \mathbf{1})$. Then

$$\mathsf{wp}[Q](f)(\sigma) \;=\; \frac{f'(\sigma)}{g'(\sigma)}$$

Theorem: cwp is "Backward Compatible"

Let $Q \in \mathsf{pGCL}$ (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f',\, g') = \mathsf{cwp}[Q](f,\, \mathbf{1})$. Then

$$\mathsf{wp}[Q](f)(\sigma) \;=\; \frac{f'(\sigma)}{g'(\sigma)} \;=\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wlp}[Q](\mathbf{1})}$$

Theorem: cwp is "Backward Compatible"

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f',\, g') = \mathsf{cwp}[Q](f,\, \mathbf{1})$. Then

$$\mathsf{wp}[Q](f)(\sigma) \;=\; \frac{f'(\sigma)}{g'(\sigma)} \;=\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wlp}[Q](\mathbf{1})} \;\neq\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wp}[Q](\mathbf{1})}$$

Theorem: cwp is "Backward Compatible"

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f',\, g') = \mathsf{cwp}[Q](f,\, \mathbf{1})$. Then

$$\mathsf{wp}[Q](f)(\sigma) \;=\; \frac{f'(\sigma)}{g'(\sigma)} \;=\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wlp}[Q](\mathbf{1})} \;\neq\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wp}[Q](\mathbf{1})}$$

Theorem: Correspondence Theorem

Theorem: cwp is "Backward Compatible"

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f', g') = \mathsf{cwp}[Q](f, \mathbf{1})$. Then

$$\mathsf{wp}[Q](f)(\sigma) \;=\; \frac{f'(\sigma)}{g'(\sigma)} \;=\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wlp}[Q](\mathbf{1})} \;\neq\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wp}[Q](\mathbf{1})}$$

Theorem: Correspondence Theorem

Let $P \in \mathsf{cpGCL}^{\boxtimes}$,

Theorem: cwp is "Backward Compatible"

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f', g') = \mathsf{cwp}[Q](f, \mathbf{1})$. Then

$$\mathsf{wp}[Q](f)(\sigma) \;=\; \frac{f'(\sigma)}{g'(\sigma)} \;=\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wlp}[Q](\mathbf{1})} \neq \frac{\mathsf{wp}[Q](f)}{\mathsf{wp}[Q](\mathbf{1})}$$

Theorem: Correspondence Theorem

Let $P \in \mathsf{cpGCL}^{\boxtimes}$, $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$

Theorem: cwp is “Backward Compatible”

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f',\, g') = \mathsf{cwp}[Q](f,\, \mathbf{1})$. Then

$$\mathsf{wp}[Q](f)(\sigma) \;=\; \frac{f'(\sigma)}{g'(\sigma)} \;=\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wlp}[Q](\mathbf{1})} \;\neq\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wp}[Q](\mathbf{1})}$$

Theorem: Correspondence Theorem

Let $P \in \mathsf{cpGCL}^{\boxtimes}$, $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f',\, g') = \mathsf{cwp}[P](f,\, \mathbf{1})$.

Theorem: cwp is "Backward Compatible"

Let $Q \in$ pGCL (i.e. Q contains no observations), $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f', g') = \mathsf{cwp}[Q](f, \mathbf{1})$. Then

$$\mathsf{wp}[Q](f)(\sigma) \;=\; \frac{f'(\sigma)}{g'(\sigma)} \;=\; \frac{\mathsf{wp}[Q](f)}{\mathsf{wlp}[Q](\mathbf{1})} \neq \frac{\mathsf{wp}[Q](f)}{\mathsf{wp}[Q](\mathbf{1})}$$

Theorem: Correspondence Theorem

Let $P \in \mathsf{cpGCL}^{\boxtimes}$, $f \in \mathbb{E}$, $\sigma \in \mathbb{S}$ and $(f', g') = \mathsf{cwp}[P](f, \mathbf{1})$. Then ${f'(\sigma)}/{g'(\sigma)}$ corresponds to the conditional expected reward of the operational RMDP.

A Mild Assumption

The non–deterministic choice $\{P_1\} \square \{P_2\}$ is an implementation choice.

A Mild Assumption

The non–deterministic choice $\{P_1\} \,\Box\, \{P_2\}$ is an implementation choice. More formally: If it holds that

$$\mathsf{cwp}\big[\{P_1\} \,\Box\, \{P_2\}\big] = \mathsf{cwp}[P_1]$$

A Mild Assumption

The non–deterministic choice $\{P_1\} \Box \{P_2\}$ is an implementation choice. More formally: If it holds that

$$\mathsf{cwp}\big[\{P_1\} \Box \{P_2\}\big] = \mathsf{cwp}[P_1]$$

then it should also hold that

$$\mathsf{cwp}\big[\{\{P_1\} \Box \{P_2\}\}\ [p]\ \{P_3\}\big] = \mathsf{cwp}\big[\{P_1\}\ [p]\ \{P_3\}\big].$$

A Mild Assumption

The non–deterministic choice $\{P_1\} \Box \{P_2\}$ is an implementation choice. More formally: If it holds that

$$\mathsf{cwp}\big[\{P_1\} \Box \{P_2\}\big] = \mathsf{cwp}[P_1]$$

then it should also hold that

$$\mathsf{cwp}\big[\{\{P_1\} \Box \{P_2\}\} \; [p] \; \{P_3\}\big] = \mathsf{cwp}\big[\{P_1\} \; [p] \; \{P_3\}\big].$$

Theorem: Adding Non–Determinism to cwp

Under this mild assumption, it is not possible to extend the rules for cwp by a rule for non–deterministic choice.

Conclusion

Conclusion

- We introduce operational semantics à la Gretz et al. for probabilistic programs with conditioning.

Conclusion

- We introduce operational semantics à la Gretz et al. for probabilistic programs with conditioning.
- We introduce a reasonable, backward compatible denotational semantics à la McIver & Morgan.

Conclusion

- We introduce operational semantics à la Gretz et al. for probabilistic programs with conditioning.
- We introduce a reasonable, backward compatible denotational semantics à la McIver & Morgan.
- We prove that operational and denotational semantics coincide for the fully probabilistic fragment of cpGCL.

Conclusion

- We introduce operational semantics à la Gretz et al. for probabilistic programs with conditioning.
- We introduce a reasonable, backward compatible denotational semantics à la McIver & Morgan.
- We prove that operational and denotational semantics coincide for the fully probabilistic fragment of cpGCL.
- We prove that cwp cannot be extended to non–deterministic choice.

Conclusion

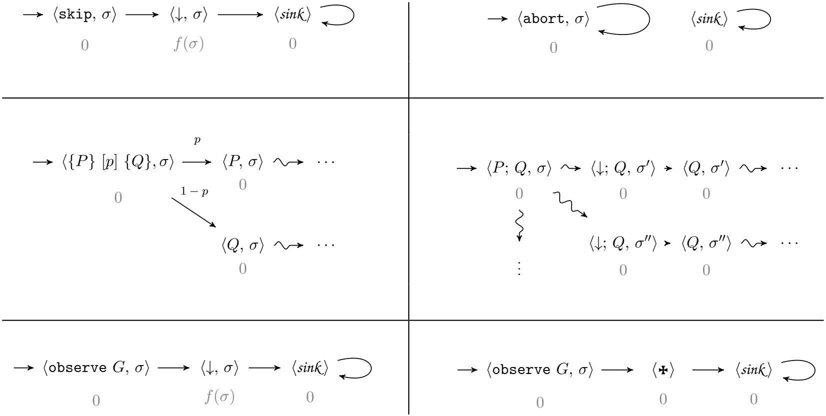
- We introduce operational semantics à la Gretz et al. for probabilistic programs with conditioning.
- We introduce a reasonable, backward compatible denotational semantics à la McIver & Morgan.
- We prove that operational and denotational semantics coincide for the fully probabilistic fragment of cpGCL.
- We prove that cwp cannot be extended to non–deterministic choice.
- We can use cwp to prove the correctness of several program transformations.

Conclusion

- We introduce operational semantics à la Gretz et al. for probabilistic programs with conditioning.
- We introduce a reasonable, backward compatible denotational semantics à la McIver & Morgan.
- We prove that operational and denotational semantics coincide for the fully probabilistic fragment of cpGCL.
- We prove that cwp cannot be extended to non–deterministic choice.
- We can use cwp to prove the correctness of several program transformations.

Thank you for your attention! :-)

Operational Semantics for cpGCL



Rules for cwp

$\boldsymbol{P}$	$\mathsf{\mathbf{cwp}}[\boldsymbol{P}](\boldsymbol{f},\, \boldsymbol{g})$
$x := E$	$(f[x/E],\, g[x/E])$
$\texttt{observe}\, G$	$\chi_G \cdot (f,\, g)$
$P_1;\;\, P_2$	$(\mathsf{cwp}[P_1] \circ \mathsf{cwp}[P_2])(f, g)$
$\texttt{if}\,(G)\,\{P_1\}\,\texttt{else}\,\{P_2\}$	$\chi_G \cdot \mathsf{cwp}[P_1](f,\, g) + \chi_{\neg G} \cdot \mathsf{cwp}[P_2](f,\, g)$
$\{P_1\}\;[p]\;\{P_2\}$	$p \cdot \mathsf{cwp}[P_1](f,\, g) + (1-p) \cdot \mathsf{cwp}[P_2](f,\, g)$
$\{P_1\}\,\square\,\{P_2\}$	— not defined —
$\texttt{while}\,(G)\,\{P'\}$	$\boldsymbol{\mu}_{\sqsubseteq,\sqsupseteq}(\hat{f},\, \hat{g})\bullet\; \Big(\chi_G \cdot \mathsf{cwp}[P'](\hat{f},\, \hat{g}) + \chi_{\neg G} \cdot (f,\, g)\Big)$

Rules for cwp

$\boldsymbol{P}$	$\textbf{cwp}[\boldsymbol{P}](\boldsymbol{f},\, \boldsymbol{g})$
$x := E$	$(f[x/E],\, g[x/E])$
`observe` G	$\chi_G \cdot (f,\, g)$
$P_1;\ P_2$	$(\mathsf{cwp}[P_1] \circ \mathsf{cwp}[P_2])(f, g)$
`if` $(G)\,\{P_1\}$ `else` $\{P_2\}$	$\chi_G \cdot \mathsf{cwp}[P_1](f,\, g) + \chi_{\neg G} \cdot \mathsf{cwp}[P_2](f,\, g)$
$\{P_1\}\ [p]\ \{P_2\}$	$p \cdot \mathsf{cwp}[P_1](f,\, g) + (1-p) \cdot \mathsf{cwp}[P_2](f,\, g)$
$\{P_1\} \,\square\, \{P_2\}$	— not defined —
`while` $(G)\,\{P'\}$	$\boldsymbol{\mu}_{\sqsubseteq,\sqsupseteq}(\hat{f},\, \hat{g})\bullet\ \left(\chi_G \cdot \mathsf{cwp}[P'](\hat{f},\, \hat{g}) + \chi_{\neg G} \cdot (f,\, g)\right)$