

Conditioning in Probabilistic Programming

MFPS XXXI 2015

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



23.6.2015

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \text{if } (G) \{P\} \text{ else } \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \text{while } (G) \{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \text{if } (G) \{P\} \text{ else } \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \text{while } (G) \{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \mathbf{if} (G) \{P\} \mathbf{else} \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \mathbf{while} (G) \{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \text{if } (G) \{P\} \text{ else } \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \text{while } (G) \{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \text{if } (G) \{P\} \text{ else } \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \text{while } (G) \{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \text{if } (G) \{P\} \text{ else } \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \text{while } (G) \{P\}$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \text{if } (G) \{P\} \text{ else } \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \text{while } (G) \{P\}$$

Syntax of cpGCL Programs

$$P \longrightarrow \text{"see pGCL"} \mid \text{observe } G$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \text{if } (G) \{P\} \text{ else } \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \text{while } (G) \{P\}$$

Syntax of cpGCL Programs

$$P \longrightarrow \text{"see pGCL"} \mid \text{observe } G$$

Motivation

Syntax of pGCL Programs [McIver & Morgan '06]

$$P \longrightarrow x := E \mid \text{if } (G) \{P\} \text{ else } \{P\} \mid \{P\} [p] \{P\} \\ \mid \{P\} \square \{P\} \mid \text{while } (G) \{P\}$$

Syntax of cpGCL Programs

$$P \longrightarrow \text{"see pGCL"} \mid \text{observe } G$$

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

$\mathbb{S} = \{\sigma \mid \sigma: \mathcal{Vars} \rightarrow \mathbb{R}\}$ denotes the set of program states.

Expectations

Unbounded and Bounded Expectations

$\mathbb{S} = \{\sigma \mid \sigma: \mathcal{Vars} \rightarrow \mathbb{R}\}$ denotes the set of program states. The set of **expectations** \mathbb{E} and the set of **bounded expectations** $\mathbb{E}_{\leq 1}$ are

$$\mathbb{E} = \{f \mid f: \mathbb{S} \rightarrow \mathbb{R}_{\geq 0}^{\infty}\} \quad \mathbb{E}_{\leq 1} = \{g \mid g: \mathbb{S} \rightarrow [0, 1]\} .$$

Expectations

Unbounded and Bounded Expectations

$\mathbb{S} = \{\sigma \mid \sigma: \mathcal{Vars} \rightarrow \mathbb{R}\}$ denotes the set of program states. The set of **expectations** \mathbb{E} and the set of **bounded expectations** $\mathbb{E}_{\leq 1}$ are

$$\mathbb{E} = \{f \mid f: \mathbb{S} \rightarrow \mathbb{R}_{\geq 0}^{\infty}\} \quad \mathbb{E}_{\leq 1} = \{g \mid g: \mathbb{S} \rightarrow [0, 1]\} .$$

random variable = expectation

Expectations

Unbounded and Bounded Expectations

$\mathbb{S} = \{\sigma \mid \sigma: \mathcal{Vars} \rightarrow \mathbb{R}\}$ denotes the set of program states. The set of **expectations** \mathbb{E} and the set of **bounded expectations** $\mathbb{E}_{\leq 1}$ are

$$\mathbb{E} = \{f \mid f: \mathbb{S} \rightarrow \mathbb{R}_{\geq 0}^{\infty}\} \quad \mathbb{E}_{\leq 1} = \{g \mid g: \mathbb{S} \rightarrow [0, 1]\} .$$

random variable = expectation \neq 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: $P \in \text{cpGCL}$,

Operational Reward Markov Decision Process (RMDP)

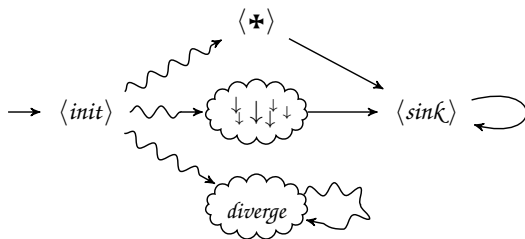
- Given: $P \in \text{cpGCL}$, $f \in \mathbb{E}$.

Operational Reward Markov Decision Process (RMDP)

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

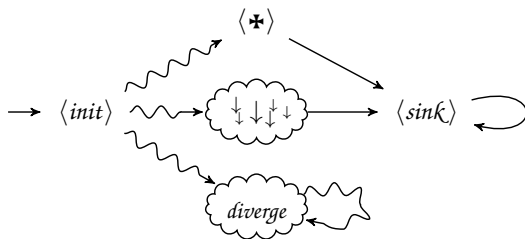
Operational Reward Markov Decision Process (RMDP)

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

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

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 $\text{wp}[P](f)$ 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

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

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

$$\mathbf{E}_{\llbracket P \rrbracket(\sigma)}(f) = \mathbf{E}_{\delta_\sigma}(\text{wp}[P](f))$$

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

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

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

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

For $g \in \mathbb{E}_{\leq 1}$, an expectation $\text{wlp}[P](g)$ 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

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

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

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

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

$$\text{wlp}[P](g) = \text{wp}[P](g) + \Pr(\text{"}P \text{ diverges"})$$

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 $P \in \text{cpGCL}^{\boxtimes}$ (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

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 $P \in \text{cpGCL}^{\boxtimes}$ (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 $\text{cwp}[P](f, \mathbf{1}) = (f', g')$.
Furthermore:

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

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

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

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

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

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

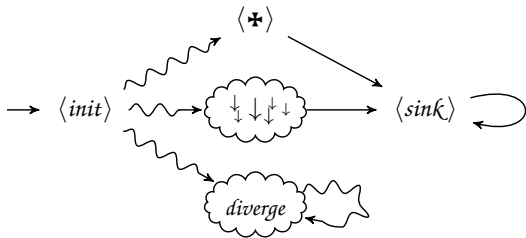
$$\frac{\mathbf{wp}}{\mathbf{wlp}}$$

The Four Possibilities

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

The Four Possibilities

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



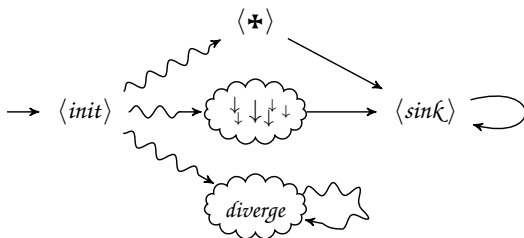
The Four Possibilities

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

$$\frac{\text{wp}}{\text{wp}}$$

$$\frac{\text{wlp}}{\text{wlp}}$$

$$\frac{\text{wlp}}{\text{wp}}$$



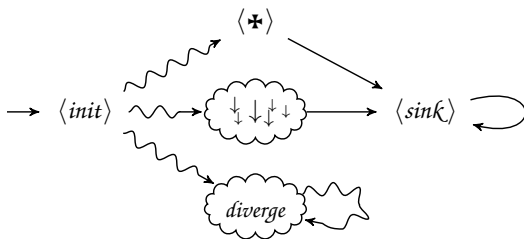
The Four Possibilities

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

$$\frac{\text{wp}}{\text{wp}} = \text{Nori et al.}$$

$$\frac{\text{wlp}}{\text{wlp}}$$

$$\frac{\text{wlp}}{\text{wp}}$$



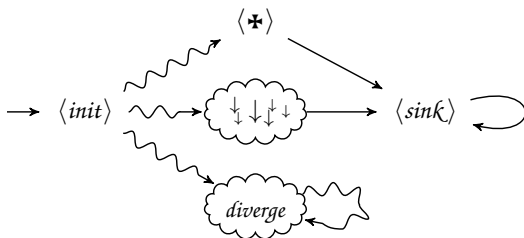
The Four Possibilities

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

$$\frac{\text{wp}}{\text{wp}} = \text{Nori et al.}$$

$$\frac{\text{wlp}}{\text{wlp}} \sim \text{cwlp}$$

$$\frac{\text{wlp}}{\text{wp}}$$



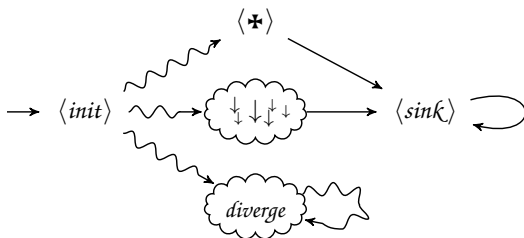
The Four Possibilities

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

$$\frac{\text{wp}}{\text{wp}} = \text{Nori et al.}$$

$$\frac{\text{wlp}}{\text{wlp}} \sim \text{cwlp}$$

$$\frac{\text{wlp}}{\text{wp}} = \text{bogus}$$



A Somewhat Intricate Example

 $P_{div} =$

```
 $x := 1;$   
while ( $x = 1$ ) {  
     $x := 1$   
}
```

 $P_{andiv} =$

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


A Somewhat Intricate Example

 $P_{div} =$

```

x := 1;
while (x = 1) {
  x := 1
}

```

 $P_{andiv} =$

```

x := 1;
while (x = 1) {
  {x := 1} [1/2] {x := 0};
  observe x = 1
}

```

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

A Somewhat Intricate Example

$$P_{div} =$$

```

x := 1;
while (x = 1) {
  x := 1
}

```

$$P_{andiv} =$$

```

x := 1;
while (x = 1) {
  {x := 1} [1/2] {x := 0};
  observe x = 1
}

```

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

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

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

A Somewhat Intricate Example

$$P_{div} =$$

```

x := 1;
while (x = 1) {
  x := 1
}

```

$$P_{andiv} =$$

```

x := 1;
while (x = 1) {
  {x := 1} [1/2] {x := 0};
  observe x = 1
}

```

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

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

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

A Somewhat Intricate Example

$$P_{div} =$$

```

x := 1;
while (x = 1) {
  x := 1
}

```

$$P_{andiv} =$$

```

x := 1;
while (x = 1) {
  {x := 1} [1/2] {x := 0};
  observe x = 1
}

```

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

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

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

A Somewhat Intricate Example

$$P_{div} =$$

```

x := 1;
while (x = 1) {
  x := 1
}

```

$$P_{andiv} =$$

```

x := 1;
while (x = 1) {
  {x := 1} [1/2] {x := 0};
  observe x = 1
}

```

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

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

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 $Q \in \text{pGCL}$ (i.e. Q contains no observations),

Theorem: cwp is “Backward Compatible”

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

Theorem: cwp is “Backward Compatible”

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

Theorem: cwp is “Backward Compatible”

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

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

Theorem: cwp is “Backward Compatible”

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

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

Theorem: cwp is “Backward Compatible”

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

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

Theorem: cwp is “Backward Compatible”

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

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

Theorem: Correspondence Theorem

Theorem: cwp is “Backward Compatible”

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

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

Theorem: Correspondence Theorem

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

Theorem: cwp is “Backward Compatible”

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

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

Theorem: Correspondence Theorem

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

Theorem: cwp is “Backward Compatible”

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

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

Theorem: Correspondence Theorem

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

Theorem: cwp is “Backward Compatible”

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

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

Theorem: Correspondence Theorem

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

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 $\{P_1\} \square \{P_2\}$ is an implementation choice. More formally: If it holds that

$$\text{cwp}[\{P_1\} \square \{P_2\}] = \text{cwp}[P_1]$$

A Mild Assumption

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

$$\text{cwp}[\{\{P_1\} \square \{P_2\}\}] = \text{cwp}[P_1]$$

then it should also hold that

$$\text{cwp}[\{\{\{P_1\} \square \{P_2\}\} [p] \{P_3\}\}] = \text{cwp}[\{P_1\} [p] \{P_3\}].$$

A Mild Assumption

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

$$\text{cwp}[\{\{P_1\} \square \{P_2\}\}] = \text{cwp}[P_1]$$

then it should also hold that

$$\text{cwp}[\{\{\{P_1\} \square \{P_2\}\} [p] \{P_3\}\}] = \text{cwp}[\{P_1\} [p] \{P_3\}].$$

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

$$\begin{array}{c} \rightarrow \langle \text{skip}, \sigma \rangle \longrightarrow \langle \downarrow, \sigma \rangle \longrightarrow \langle \text{sink} \rangle \curvearrowright \\ 0 \qquad f(\sigma) \qquad 0 \end{array}$$

$$\begin{array}{c} \rightarrow \langle \text{abort}, \sigma \rangle \curvearrowright \qquad \langle \text{sink} \rangle \curvearrowright \\ 0 \qquad 0 \end{array}$$

$$\begin{array}{c} \rightarrow \langle \{P\} [p] \{Q\}, \sigma \rangle \longrightarrow \langle P, \sigma \rangle \rightsquigarrow \dots \\ 0 \qquad \begin{array}{l} p \\ \searrow \\ \langle Q, \sigma \rangle \rightsquigarrow \dots \\ 0 \end{array} \\ \qquad \begin{array}{l} 1-p \\ \searrow \\ \langle Q, \sigma \rangle \rightsquigarrow \dots \\ 0 \end{array} \end{array}$$

$$\begin{array}{c} \rightarrow \langle P; Q, \sigma \rangle \rightsquigarrow \langle \downarrow; Q, \sigma' \rangle \rightsquigarrow \langle Q, \sigma' \rangle \rightsquigarrow \dots \\ 0 \qquad \begin{array}{l} \rightsquigarrow \\ \rightsquigarrow \\ \rightsquigarrow \\ \vdots \end{array} \qquad \begin{array}{l} \langle \downarrow; Q, \sigma'' \rangle \rightsquigarrow \langle Q, \sigma'' \rangle \rightsquigarrow \dots \\ 0 \qquad 0 \end{array} \end{array}$$

$$\begin{array}{c} \rightarrow \langle \text{observe } G, \sigma \rangle \longrightarrow \langle \downarrow, \sigma \rangle \longrightarrow \langle \text{sink} \rangle \curvearrowright \\ 0 \qquad f(\sigma) \qquad 0 \end{array}$$

$$\begin{array}{c} \rightarrow \langle \text{observe } G, \sigma \rangle \longrightarrow \langle * \rangle \longrightarrow \langle \text{sink} \rangle \curvearrowright \\ 0 \qquad 0 \qquad 0 \end{array}$$

Rules for cwp

P	$\mathbf{cwp}[P](f, g)$
$x := E$	$(f[x/E], g[x/E])$
$\mathbf{observe} G$	$\chi_G \cdot (f, g)$
$P_1; P_2$	$(\mathbf{cwp}[P_1] \circ \mathbf{cwp}[P_2])(f, g)$
$\mathbf{if} (G) \{P_1\} \mathbf{else} \{P_2\}$	$\chi_G \cdot \mathbf{cwp}[P_1](f, g) + \chi_{\neg G} \cdot \mathbf{cwp}[P_2](f, g)$
$\{P_1\} [p] \{P_2\}$	$p \cdot \mathbf{cwp}[P_1](f, g) + (1 - p) \cdot \mathbf{cwp}[P_2](f, g)$
$\{P_1\} \square \{P_2\}$	— not defined —
$\mathbf{while} (G) \{P'\}$	$\mu_{\square, \supseteq}(\hat{f}, \hat{g}) \cdot \left(\chi_G \cdot \mathbf{cwp}[P'](f, g) + \chi_{\neg G} \cdot (f, g) \right)$

Rules for cwp

P	$\mathbf{cwp}[P](f, g)$
$x := E$	$(f[x/E], g[x/E])$
$\mathbf{observe} G$	$\chi_G \cdot (f, g)$
$P_1; P_2$	$(\mathbf{cwp}[P_1] \circ \mathbf{cwp}[P_2])(f, g)$
$\mathbf{if} (G) \{P_1\} \mathbf{else} \{P_2\}$	$\chi_G \cdot \mathbf{cwp}[P_1](f, g) + \chi_{\neg G} \cdot \mathbf{cwp}[P_2](f, g)$
$\{P_1\} [p] \{P_2\}$	$p \cdot \mathbf{cwp}[P_1](f, g) + (1 - p) \cdot \mathbf{cwp}[P_2](f, g)$
$\{P_1\} \square \{P_2\}$	— not defined —
$\mathbf{while} (G) \{P'\}$	$\mu_{\square, \supseteq}(\hat{f}, \hat{g}) \cdot \left(\chi_G \cdot \mathbf{cwp}[P'](f, g) + \chi_{\neg G} \cdot (f, g) \right)$