Probabilistic Programming

Lecture #7: Probabilistic Weakest Preconditions

Joost-Pieter Katoen

RWTH Lecture Series on Probabilistic Programming 2018

Overview

Code-level reasoning

Proving properties of probabilistic programs: not by executing them, but by reasoning at the syntax level of programs.

Compositionality: determine the correctness of composed program P by reasoning about its parts in isolation and then obtain P's correctness result by combining those parts' analyses.

Overview

Elementary pGCL ingredients

- Program variables $x \in \mathit{Vars}$ whose values are fractional numbers
- Arithmetic expressions E over the program variables
- Boolean expressions G (guarding a choice or loop) over the program variables

unif [0..x]

x=2

$\frac{x}{x+1}$

- A distribution expression $\mu : \Sigma \rightarrow \mathit{Dist}(\mathbb{Q})$
- A probability expression $p : \Sigma \rightarrow [0,1] \cap \mathbb{Q}$

Probabilistic GCL: Syntax

Kozen

McIver

Morgan

- `skip` empty statement
- `diverge` divergence
- `x := E` assignment
- `x :r= mu` **random assignment** $(x :\approx \mu)$
- `prog1 ; prog2` sequential composition
- `if (G) prog1 else prog2` choice
- `prog1 [p] prog2` **probabilistic choice**
- `while (G) prog` iteration

Conditioning will be treated later. For the moment: no conditioning.

Examples: Intuition

1. Let program P be:

```
x := 5 [4/5] x := 10
```

The expected value of x on P's termination is: $\frac{4}{5} \cdot 5 + \frac{1}{5} \cdot 10 = 6$

2. Let program Q be:

```
x:=2 ; ( x := x+5 [4/5] x := 10 )
```

value of x when starting Q

The expected value of x on Q's termination is: $\frac{4}{5} \cdot (x+5) + \frac{1}{5} \cdot 10 = \frac{4x}{5} + 6$

$\frac{4 \cdot 2}{5} + 6$

Examples: Intuition

1. Let program P be:

```
x := 5 [4/5] x := 10
```

The expected value of `x` on P's termination is: $\frac{4}{5}\cdot 5 + \frac{1}{5}\cdot 10 = 6$

2. Let program Q be:

```
x := x+5 [4/5] x := 10
```

The expected value of `x` on Q's termination is: $\frac{4}{5}\cdot(x+5) + \frac{1}{5}\cdot 10 = \frac{4x}{5} + 6$

$[x=10]$ $\quad [x=10](x:=10)$

3. The probability that `x` = 10 on Q's termination is:

value of x when starting Q

$$\frac{4}{5}\cdot[x+5=10] + \frac{1}{5}\cdot 1 \; = \frac{4\cdot[x=5]+1}{5}$$

Expected values

A probability distribution μ on a countable set X is a function $\mu : X \to [0,1]$ such that $\sum_{x \in X} \mu(x) = 1$.

The expected value of random variable $f : X \to \mathbb{R}$ under distribution μ is defined by:

$$E_\mu(f) = \sum_{x \in X} f(x) \cdot \mu(x) = \int_X f \, d\mu$$

3 x fair coin $\mu(hd) = \mu(tl) = \frac{1}{2}$

$$f = \begin{cases} 15 & \text{, if } 3 \times \text{hds} \\ 5 & \text{, if } 3 \times \text{tls} \\ 0 & \text{, otherwise} \end{cases}$$

$$E_\mu(f) = 15 \cdot \frac{1}{8} + 5 \cdot \frac{1}{8} + 0 \cdot \frac{6}{8}$$

Expectations

Predicates

A predicate F maps program states onto Booleans, i.e., $F : \mathbb{S} \to \mathbb{B}$.

Let $\mathbb{P}$ denote the set of all predicates and $F \sqsubseteq G$ if and only if $F \Rightarrow G$.

Expectations are the quantitative analogue of predicates.

Expectations

A expectation[1] (read: random variable) f maps program states onto non-negative reals extended with infinity, i.e., $f : \mathbb{S} \to \mathbb{R}_{\geq 0} \cup \{ \infty \}$.

Let $\mathbb{E}$ denote the set of all expectations and let $\sqsubseteq$ be defined for $f, g \in \mathbb{E}$ by:

$$f \sqsubseteq g \quad \text{if and only if} \quad f(s) \leq g(s) \quad \text{for all } s \in \mathbb{S}.$$

[1] $\neq$ expectations in probability theory.

Expectations

$(\mathbb{E}, \sqsubseteq)$ is a complete lattice.

Proof.

Left as exercise. The least element of $(\mathbb{E}, \sqsubseteq)$ is the constant function $\lambda s.0$, also denoted as $\mathbf{0}$ defined by $\mathbf{0}(s) = 0$. The supremum of a subset $S \subseteq \mathbb{E}$ is constructed point-wise by $\sup S = \sup_{f \in S} f$. $\square$

Operations on expectations

- For $k \in \mathbb{R}_{\geq 0} \cup \{\infty\}$, let $\lambda s.k$ denote the expectation that is constantly k for all s
- For expression E, $x \in Vars$ and $f \in \mathbb{E}$,

$$f[x := E](s) = \begin{cases} f(y) & \text{if } x \neq y \\ [\![E]\!]_s & \text{otherwise} \end{cases}$$

- For $f \in \mathbb{E}$ and $c \in \mathbb{R}_{\geq 0}$, $(c \cdot f)(s) = c \cdot f(s)$
- For $f, g \in \mathbb{E}$, let $(f + g)(s) = f(s) + g(s)$. Multiplication and subtraction are defined analogously.

Overview

Expectation transformers

Predicate transformer

A predicate transformer Φ is a total function between predicates, i.e., $\Phi : \mathbb{P} \rightarrow \mathbb{P}$.

Expectation transformer

An expectation transformer Φ is a total function between expectations, i.e., $\Phi : \mathbb{E} \rightarrow \mathbb{E}$.

Weakest pre-expectations

Weakest precondition

For probabilistic program P and $e, f \in \mathbb{E}$, the expectation transformer $wp(P, \cdot) : \mathbb{E} \to \mathbb{E}$ is defined by $wp(P, f) = e$ iff e maps each (initial) state s to the expected value of f after executing P on s.

The characterising equation of a weakest pre-expectation is given by:

$$wp(P, f) \;=\; \lambda s. \int_{\mathbb{S}} f \; dP_s$$

where P_s is the distribution over the final states (reached on termination of P) when executing P on the initial state s.

$$s = \begin{cases} x=10 \\ y=2 \end{cases} \quad \overset{P_s}{\rightsquigarrow} \quad Dist(\mathbb{S})$$

Weakest pre-expectations

Weakest precondition

For probabilistic program P and $e, f \in \mathbb{E}$, the expectation transformer $wp(P, \cdot) : \mathbb{E} \rightarrow \mathbb{E}$ is defined by $wp(P, f) = e$ iff e maps each (initial) state s to the expected value of f after executing P on s.

The characterising equation of a weakest pre-expectation is given by:

$$wp(P, f) \;=\; \lambda s. \int_{\mathbb{S}} f \, dP_s$$

where P_s is the distribution over the final states (reached on termination of P) when executing P on the initial state s.

Examples.

$$wp\left(x:=0 \; [1/2] \; x:=1 ,\; x \right) = \frac{1}{2}$$

post "r.v"

$$wp\Big(c:=0 \; [\tfrac{1}{2}] \; c:=1;$$
$$\text{if}(c)\ \{\text{skip}\}\ \text{else}\ \{\ x:=x+1;$$
$$c:=0 \; [\tfrac{1}{2}] \; c:=1;$$
$$\text{if}(c)\ \{\text{skip}\}\ \text{else}\ \{x:=x+1\}$$
$$\};$$
$$x \Big)$$

$$\begin{cases} x & pr = \frac{1}{2} \\ x+1 & pr = \frac{1}{4} \\ x+2 & pr = \frac{1}{4} \end{cases}$$

$$\frac{1}{2}x + \frac{1}{4}(x+1) + \frac{1}{4}(x+2) = \ldots$$

3) $wp\Big(x:=0;\ c:=0;$

$\quad \text{while}\ (c=0)\ \{\ c:=1\ [p]\ x:=x+1\},$

$\quad x \Big) = \dfrac{1-p}{p}$

Reasoning about probabilities

$\underbrace{[x=10]}_{F}$

f

An important special case is when the $\overbrace{\text{post-expectation}}^{f}$ is given as $[F]$ with $F \in \mathbb{P}$. We then can consider F as an event and $wp(P, [F])(s)$ as the probability that executing P on input s will terminate in a final state $\tau \models F$.

$$\left.\begin{array}{l} P :: \quad x := x+5 \; [4/5] \; x := 10 \\ f = [x=10] \end{array}\right\} \qquad wp(P, f) = \underbrace{\frac{4 \cdot [x=5] + 1}{5}}$$

Reasoning about probabilities

An important special case is when the post-expectation is given as $[F]$ with $F \in \mathbb{P}$. We then can consider F as an event and $wp(P, [F])(s)$ as the probability that executing P on input s will terminate in a final state $\tau \models F$.

Example

See the third example a few slides ago. More examples later.

Expectation transformer semantics of pGCL

pGCL

Syntax

- skip
- diverge
- x := E
- x :≈ μ
- P1 ; P2
- if (G) P1 else P2
- P1 [p] P2
- while (G)P

$wp(P, f)$

$\hookrightarrow f \in \mathbb{E}$

f

$\underline{0}$

$f(x := E)$

$wp(P_1, wp(P_2, f))$

$P_1 \quad ; \quad P_2 \quad f$

$[G] \cdot wp(P_1, f)$

$+ [\neg G] \cdot wp(P_2, f)$

$p \cdot wp(P_1, f)$

$+ (1-p) \cdot wp(P_2, f)$

Expectation transformer semantics of pGCL

Syntax

- **skip**
- **diverge**
- `x := E`
- `x :≈` μ
- `P1 ; P2`
- **if** `(G) P1` **else** `P2`
- `P1 [p] P2`
- **while** `(G)P`

Semantics $wp(P, f)$

- f
- 0
- $f[x := E]$
- $\lambda s. \int_{\mathbb{Q}} (\lambda v. f(s[x := v]))\, d\mu_s$

$wp(P, f) = e \in \mathbb{E}$

$\lambda x. (x^2 + 2)$

$f(x) =$

$x^2 + 2$

unif $[1 \ldots x]$

$s(x) = 10$

$\mu_s =$ unif $[1 .. 10]$

Expectation transformer semantics of pGCL

Syntax

- **skip**
- **diverge**
- `x := E`
- `x :≈` μ
- `P1 ; P2`
- **if** `(G) P1` **else** `P2`
- `P1 [p] P2`
- **while** `(G)P`

Semantics $wp(P, f)$

- f
- 0
- $f[x := E]$
- $\lambda s. \int_{\mathbb{Q}} (\lambda v. f(s[x := v]))\, d\mu_s$
- $wp(P_1, wp(P_2, f))$
- $[G] \cdot wp(P_1, f) + [\neg G] \cdot wp(P_2, f)$
- $p \cdot wp(P_1, f) + (1-p) \cdot wp(P_2, f)$
- $\mathrm{lfp}\ X.\ ([G] \cdot wp(P, X) + [\neg G] \cdot f)$

lfp is the least fixed point operator wrt. the ordering $\sqsubseteq$ on expectations $\mathbb{E}$.

Examples

1. Let program P be:

   ```
   x := 5 [4/5] x := 10
   ```

   For $f = x$, we have

   $$wp(P, x) = \frac{4}{5} \cdot wp(x := 5, x) + \frac{1}{5} \cdot wp(x := 10, x) = \frac{4}{5} \cdot 5 + \frac{1}{5} \cdot 10 = 6$$

   def. of wp for P_1 [p] P_2

   def. of wp. for assignment

Examples

1. Let program P be:

```
x := 5 [4/5] x := 10
```

   For $f = x$, we have

$$wp(P, x) = \frac{4}{5} \cdot wp(x := 5, x) + \frac{1}{5} \cdot wp(x := 10, x) = \frac{4}{5} \cdot 5 + \frac{1}{5} \cdot 10 = 6$$

2. Let program P' be:

```
x := x+5 [4/5] x := 10
```

   For $f = x$, we have:

$$\underline{wp(P', x)} = \frac{4}{5} \cdot wp(x \mathrel{+}:= 5, x) + \frac{1}{5} \cdot wp(x := 10, x) = \frac{4}{5} \cdot (x+5) + \frac{1}{5} \cdot 10 = \frac{4x}{5} + 6$$

wp for

[p]

Examples

1. Let program P be:

```
x := 5 [4/5] x := 10
```

   For $f = x$, we have

$$wp(P, x) = \tfrac{4}{5}{\cdot}wp(x := 5, x) + \tfrac{1}{5}{\cdot}wp(x := 10, x) = \tfrac{4}{5}{\cdot}5 + \tfrac{1}{5}{\cdot}10 = 6$$

2. Let program P' be:

```
x := x+5 [4/5] x := 10
```

   For $f = x$, we have:

$$wp(P', x) = \tfrac{4}{5}{\cdot}wp(x \mathrel{+}:= 5, x) + \tfrac{1}{5}{\cdot}wp(x := 10, x) = \tfrac{4}{5}{\cdot}(x{+}5) + \tfrac{1}{5}{\cdot}10 = \frac{4x}{5} + 6$$

3. For program P' (again) and $f = [x = 10]$, we have:

$$\begin{aligned} wp(P', [x{=}10]) &= \tfrac{4}{5} \cdot wp(x := x{+}5, [x{=}10]) + \tfrac{1}{5} \cdot wp(x := 10, [x{=}10]) \\ &= \tfrac{4}{5} \cdot \underbrace{[x{+}5 = 10]}_{[x=5]} + \tfrac{1}{5} \cdot \underbrace{[10 = 10]}_{=1} \\ &= \frac{4{\cdot}[x{=}5]{+}1}{5} \end{aligned}$$

```
x := 0 [1/2] x := 1; // command c1
y := 0 [1/3] y := 1; // command c2
```

$$wp(c_1; c_2, [x = y])$$
$$=$$
$$wp(c_1, wp(c_2, [x = y]))$$
$$=$$
$$wp(c_1, {}^1\!/\!_3 \cdot wp(y := 0, [x = y]) + {}^2\!/\!_3 \cdot wp(y := 1, [x = y]))$$

```
x := 0 [1/2] x := 1; // command c1
y := 0 [1/3] y := 1; // command c2
```

$$wp(c_1;\, c_2,\, [x = y])$$

$$=$$

$$wp(c_1,\, wp(c_2,\, [x = y]))$$

$$=$$

$$wp(c_1,\, {}^1\!/\!_3 \cdot wp(y := 0,\, [x = y]) + {}^2\!/\!_3 \cdot wp(y := 1,\, [x = y]))$$

$$=$$

$$wp(c_1,\, {}^1\!/\!_3 \cdot [x = 0] + {}^2\!/\!_3 \cdot [x = 1])$$

```
x := 0 [1/2] x := 1; // command c1
y := 0 [1/3] y := 1; // command c2
```

$$wp(c_1; c_2, [x = y])$$
$$=$$
$$wp(c_1, wp(c_2, [x = y]))$$
$$=$$
$$wp(c_1, {}^1\!/_3 \cdot wp(y := 0, [x = y]) + {}^2\!/_3 \cdot wp(y := 1, [x = y]))$$
$$=$$
$$wp(c_1, {}^1\!/_3 \cdot [x = 0] + {}^2\!/_3 \cdot [x = 1])$$
$$=$$
$${}^1\!/_2 \cdot wp(x := 0, {}^1\!/_3 \cdot [x = 0] + {}^2\!/_3 \cdot [x = 1]) + {}^1\!/_2 \cdot wp(x := 1, {}^1\!/_3 \cdot [x = 0] + {}^2\!/_3 \cdot [x = 1])$$

```
x := 0 [1/2] x := 1; // command c1
y := 0 [1/3] y := 1; // command c2
```

$$wp(c_1; c_2, [x = y])$$
$$=$$
$$wp(c_1, wp(c_2, [x = y]))$$
$$=$$
$$wp(c_1, {}^1/_3 \cdot wp(y := 0, [x = y]) + {}^2/_3 \cdot wp(y := 1, [x = y]))$$
$$=$$
$$wp(c_1, {}^1/_3 \cdot [x = 0] + {}^2/_3 \cdot [x = 1])$$
$$=$$
$${}^1/_2 \cdot wp(x := 0, {}^1/_3 \cdot [x = 0] + {}^2/_3 \cdot [x = 1]) + {}^1/_2 \cdot wp(x := 1, {}^1/_3 \cdot [x = 0] + {}^2/_3 \cdot [x = 1])$$
$$=$$
$${}^1/_2 \cdot ({}^1/_3 \cdot \underbrace{[0 = 0]}_{=1} + {}^2/_3 \cdot \underbrace{[0 = 1]}_{=0}) + {}^1/_2 \cdot ({}^1/_3 \cdot \underbrace{[1 = 0]}_{=0} + {}^2/_3 \cdot \underbrace{[1 = 1]}_{=1})$$

```
x := 0 [1/2] x := 1; // command c1
y := 0 [1/3] y := 1; // command c2
```

$$
\begin{aligned}
& wp(c_1; c_2, [x = y]) \\
&= \\
& wp(c_1, wp(c_2, [x = y])) \\
&= \\
& wp(c_1, {}^1/_3 \cdot wp(y := 0, [x = y]) + {}^2/_3 \cdot wp(y := 1, [x = y])) \\
&= \\
& wp(c_1, {}^1/_3 \cdot [x = 0] + {}^2/_3 \cdot [x = 1]) \\
&= \\
& {}^1/_2 \cdot wp(x := 0, {}^1/_3 \cdot [x = 0] + {}^2/_3 \cdot [x = 1]) + {}^1/_2 \cdot wp(x := 1, {}^1/_3 \cdot [x = 0] + {}^2/_3 \cdot [x = 1]) \\
&= \\
& {}^1/_2 \cdot ({}^1/_3 \cdot [0 = 0] + {}^2/_3 \cdot [0 = 1]) + {}^1/_2 \cdot ({}^1/_3 \cdot [1 = 0] + {}^2/_3 \cdot [1 = 1]) \\
&= \\
& {}^1/_2 \cdot ({}^1/_3 \cdot \mathbf{1} + {}^2/_3 \cdot \mathbf{0}) + {}^1/_2 \cdot ({}^1/_3 \cdot \mathbf{0} + {}^2/_3 \cdot \mathbf{1}) \\
&= \\
& {}^1/_2 \cdot ({}^1/_3 + {}^2/_3) \\
&= \\
& {}^1/_2
\end{aligned}
$$

A simple slot machine

```
void flip {
  d1 := ♡ [1/2] ◇;
  d2 := ♡ [1/2] ◇;
  d3 := ♡ [1/2] ◇;
}
```

Example weakest pre-expectations

Let $\mathit{all}(x) \;\equiv\; (x = d_1 = d_2 = d_3)$.

- If $f = [\mathit{all}(\heartsuit)]$, then $wp(\mathit{flip}, f) = \frac{1}{8}$.
- If $g = 10 \cdot [\mathit{all}(\heartsuit)] + 5 \cdot [\mathit{all}(\diamondsuit)]$, then:

$$wp(\mathit{flip}, g) \;=\; \frac{15}{8} \;=\; 6 \cdot \frac{1}{8} \cdot 0 + 1 \cdot \frac{1}{8} \cdot 10 + 1 \cdot \frac{1}{8} \cdot 5$$

So the least fraction of the jackpot the gamer can expect to win is $\frac{15}{8}$.

Loops

$$wp(\text{while}\,(G)\{\,P\,\}, f) \;=\; \text{lfp}\, X.\ \underbrace{([G]\cdot wp(P,X) + [\neg G]\cdot f)}_{\Psi(X)}$$

Scott continuity of Ψ

The function $\Psi : \mathbb{E} \to \mathbb{E}$ (defined as above) is continuous on $(\mathbb{E}, \sqsubseteq)$.

Proof.

Left as an exercise. By structural induction on pGCL programs. $\square$

Corollary

By Kleene's fixpoint theorem, it follows $\text{lfp}\ \Psi = \sup_{n\in\mathbb{N}} \Psi^n(\mathbf{0})$.

$\Psi^n(\mathbf{0})$ is the expected value over the final states of running while $(G)\{\,P\,\}$ exactly n times when starting with the constant expectation $\mathbf{0}$.

A simple loopy program

```
x := 0;
while (c) {
    { c := 0 } [0.5] { x++ }
}
```

What is the expected value of x on termination?

$$wp(\text{while}(c)\ \{c:=0\ [1/2]\ x{+}{+}\},\ X)$$

$$\Psi(X) = [c=1]\ wp(c:=0\ [\tfrac{1}{2}]\ x{+}{+},\ X) + [c\neq 1]\cdot X$$

$$= \ldots \text{calculate} \ldots$$

$$= [c=1]\left(\tfrac{1}{2}\cdot X(c:=0) + \tfrac{1}{2}X(x:=x+1)\right)$$

$$+ [c\neq 1]\cdot x$$

Iterating: $\Psi^0(\underline{0}) = \underline{0}$

$$\Psi^1(\underline{0}) = [c\neq 1]\cdot x$$

$$\Psi^2(\underline{0}) = \Psi([c\neq 1]\cdot x)$$

$$= [c=1]\left(\tfrac{1}{2}\cdot [c\neq 1]\cdot x\,(c:=0) + \tfrac{1}{2}[c\neq 1]\cdot x\,(x{+}{+})\right)$$

$$+ [c\neq 1]\cdot x$$

$$= [c=1]\left(\tfrac{1}{2}\cdot x + \tfrac{1}{2}[c\neq 1](x+1)\right) + [c\neq 1]\cdot x$$

$$= [c=1]\cdot\tfrac{1}{2}x + [c\neq 1]\cdot x$$

$$\Psi^3(\underline{0}) = \Psi\left([c=1]\tfrac{1}{2}x + [c\neq 1]\cdot x\right)$$

$$= [c=1]\cdot\left(\tfrac{1}{2}x + \tfrac{1}{4}(x+1)\right) + [c\neq 1]\cdot x$$

claim

$$\Psi^n(\underline{0}) = [c=1]\cdot\sum_{0<i<n}\left(\tfrac{1}{2}\right)^i(x+i-1) + [c\neq 1]\cdot x$$

$$wp\left(\mathrm{while}(c)\ \{c:=0\ [1/2]\ x:=x+1\},\ x\right)$$

$$= \boxed{[c=1] \sum_{i=1}^{\infty} \left(\tfrac{1}{2}\right)^{i} (\underset{\overset{\uparrow}{0}}{x}+i-1) + \underbrace{[c \neq 1] \cdot x}_{x=0}}$$

$$wp\left(x:=0\ ;\ \mathrm{loop}\right)$$

$$= [c=1] \sum_{i=1}^{\infty} \left(\tfrac{1}{2}\right)^{i} (i-1)$$

know: $\displaystyle\sum_{i=1}^{\infty} p^{i}(i-1) = \frac{p^2}{(1-p)^2}$ for $|p|<1$

$$= [c=1] \cdot 1$$

the exp. value of x equals 1 if $c=1$ at the start

0, otherwise

Approximating while-loops

$\Psi^0(\underline{0})$ $\Psi(\Psi(\underline{0}))$

Let:

$$
\begin{aligned}
\text{while}^0(G)\{\,P\,\}) &= \text{diverge} \\
\text{while}^{n+1}(G)\{\,P\,\}) &= \text{if } (G) \text{ then } P; \text{while}^n(G)\{\,P\,\}) \text{ else skip}
\end{aligned}
$$

Approximating while-loops

Let:

$$\begin{aligned} \text{while}^{0}(G)\{\,P\,\}) &= \text{diverge} \\ \text{while}^{n+1}(G)\{\,P\,\}) &= \text{if } (G) \text{ then } P; \text{while}^{n}(G)\{\,P\,\}) \text{ else skip} \end{aligned}$$

Let $\Psi(X) = ([G] \cdot wp(P, X) + [\neg G] \cdot f)$. Then for all $n \in \mathbb{N}$ it holds:

$$\Psi^{n}(\mathbf{0}) = wp(\text{while}^{n}(G)\{\,P\,\}, f)$$

Approximating while-loops

Let:

$$\begin{aligned} \text{while}^{0}(G)\{\,P\,\}) &= \text{diverge} \\ \text{while}^{n+1}(G)\{\,P\,\}) &= \text{if } (G) \text{ then } P; \text{while}^{n}(G)\{\,P\,\}) \text{ else skip} \end{aligned}$$

Let $\Psi(X) = ([G] \cdot wp(P, X) + [\neg G] \cdot f)$. Then for all $n \in \mathbb{N}$ it holds:

$$\Psi^{n}(\mathbf{0}) \;=\; wp(\text{while}^{n}(G)\{\,P\,\}, f)$$

Proof.

By induction on n using the inductive definition of wp. □

Overview

Properties of weakest pre-expectations

For all `pGCL` programs P and expectations f, g it holds:

- Continuity: $wp(P, \cdot)$ is continuous on $(\mathbb{E}, \sqsubseteq)$.

Properties of weakest pre-expectations

For all pGCL programs P and expectations f, g it holds:

- Continuity: $wp(P, \cdot)$ is continuous on $(\mathbb{E}, \sqsubseteq)$.

- Monotonicity: $f \leq g$ implies $wp(P, f) \leq wp(P, g)$

$f(s) \leq g(s)$

Properties of weakest pre-expectations

For all pGCL programs P and expectations f, g it holds:

- Continuity: $wp(P, \cdot)$ is continuous on $(\mathbb{E}, \sqsubseteq)$.
- Monotonicity: $f \leq g$ implies $wp(P, f) \leq wp(P, g)$
- Feasibility: $f \leq \mathbf{k}$ implies $wp(P, f) \leq \mathbf{k}$

$f : S \longrightarrow \mathbb{R}_{\geq 0} + \infty$

$\forall s.\ f(s) \leq k$

Properties of weakest pre-expectations

For all pGCL programs P and expectations f, g it holds:

- Continuity: $wp(P, \cdot)$ is continuous on $(\mathbb{E}, \sqsubseteq)$.
- Monotonicity: $f \leq g$ implies $wp(P, f) \leq wp(P, g)$
- Feasibility: $f \leq \mathbf{k}$ implies $wp(P, f) \leq \mathbf{k}$
- Linearity: $wp(P, r \cdot f + g) \;=\; r \cdot wp(P, f) + wp(P, g)$ for every $r \in \mathbb{R}_{\geq 0}$

$$wp(P, r.f) = r.\, wp(P, f)$$

Properties of weakest pre-expectations

For all pGCL programs P and expectations f, g it holds:

- Continuity: $wp(P, \cdot)$ is continuous on $(\mathbb{E}, \sqsubseteq)$.
- Monotonicity: $f \leq g$ implies $wp(P, f) \leq wp(P, g)$
- Feasibility: $f \leq \mathbf{k}$ implies $wp(P, f) \leq \mathbf{k}$
- Linearity: $wp(P, r \cdot f + g) \;=\; r \cdot wp(P, f) + wp(P, g)$ for every $r \in \mathbb{R}_{\geq 0}$
- Strictness: $wp(P, \mathbf{0}) = \mathbf{0}$

It is good to know: $wp(P, \mathbf{1})$ = termination probability of program P

Backward compatibility

The wp-semantics of pGCL is a conservative extension of Dijkstra's wp-semantics.

For any ordinary GCL program P and predicate $F \in \mathbb{P}$:

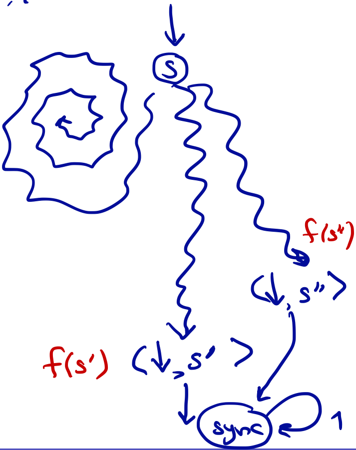
$$\underbrace{\left[wp(P, [F]) \right]}_{\text{pGCL}} = \underbrace{wp(P, F)}_{\text{Dijkstra}}$$

Recall: operational semantics of pGCL

pGCL program P ⟶ reward Markov chain $[\![P]\!]$

?

$wp(P, f)$



Weakest pre-expectations = expected rewards

Compatibility theorem

For every pGCL program P, input s and expectation f:

$$wp(P, f)(s) \quad = \quad ER^{[\![P]\!]}(s, \Diamond sink)$$

In words: the $wp(P, f)$ for input s equals the expected reward to reach final state $sink$ in MC $[\![P]\!]$ where reward function r in $[\![P]\!]$ is defined by: $r(\langle \downarrow, s' \rangle) = f(s')$ and $r(\cdot) = 0$ otherwise.

For finite-state programs, wp-reasoning can be done
with model checkers such as PRISM and Storm (www.stormchecker.org).

Example

Overview

A more tricky loopy program

```
    c := 1;
    while (c = 1) {
diverge   { abort } [0.5] { x++ };
          { skip } [0.5] { c := 0 }
    }
```

What is the probability that
either x is even on termination, or the program diverges?

Bounded expectations

$f : \mathbb{S} \to \underline{\mathbb{R}_{\geq 0} + \infty}$

Bounded expectations

The set of (one-)bounded expectations, denoted $\mathbb{E}_{\leq 1}$ is defined as:

$$\mathbb{E}_{\leq 1} \;=\; \{\, f \in \mathbb{E} \mid f \sqsubseteq \mathbf{1} \,\}$$

$\leq$

$\mathbf{1}(s) = 1 \quad \forall s$

Bounded expectations

Bounded expectations

The set of (one-)bounded expectations, denoted $\mathbb{E}_{\leq 1}$ is defined as:

$$\mathbb{E}_{\leq 1} \ = \ \{ \, f \in \mathbb{E} \mid f \sqsubseteq \mathbf{1} \, \}$$

$(\mathbb{E}_{\leq 1}, \sqsubseteq)$ is a complete lattice.

Proof.

Left as an exercise. The least element is $\lambda s.0$; the greatest element is $\lambda s.1$ and suprema are defined as for $\mathbb{E}$. □

Weakest liberal pre-expectations

Weakest liberal pre-expectation

For probabilistic program P and $e, f \in \mathbb{E}_{\leq 1}$, the expectation transformer $wlp(P, \cdot) : \mathbb{E}_{\leq 1} \to \mathbb{E}_{\leq 1}$ is defined by $wlp(P, f) = e$ such that e equals the expected value of f after executing P on s plus the probability that P diverges on s.

The characterising equation of a weakest liberal pre-expectation is given by:

$$wlp(P, f) \;=\; \overbrace{\lambda s. \int_{\mathbb{S}} f \, dP_s}^{= wp(P,f)} + \left(1 - \overbrace{\int_{\mathbb{S}} 1 \, dP_s}^{= wp(P,1)}\right)$$

where P_s is the distribution over the final states when executing P (reached on termination) on the initial state s.

$wp(P,f) = e$

$wlp(P,f) = wp(P,f)$

$wp(P,1) =$ prob. of P terminating

Weakest liberal pre-expectations

Weakest liberal pre-expectation

For probabilistic program P and $e, f \in \mathbb{E}_{\leq 1}$, the expectation transformer $wlp(P, \cdot) : \mathbb{E}_{\leq 1} \to \mathbb{E}_{\leq 1}$ is defined by $wlp(P, f) = e$ such that e equals the expected value of f after executing P on s plus the probability that P diverges on s.

The characterising equation of a weakest liberal pre-expectation is given by:

$$wlp(P, f) \;=\; \lambda s. \int_{\mathbb{S}} f \, dP_s + \left(1 - \int_{\mathbb{S}} 1 \, dP_s\right)$$

where P_s is the distribution over the final states when executing P (reached on termination) on the initial state s.

Examples.

Weakest liberal pre-expectation $wlp(P, f) \;=\;$ "$wp(P, f) + Pr[P \text{ diverges}]$".

$P :: \text{diverge} \; [\tfrac{1}{3}] \; x := 10$

$f = [x = 10]$

$$wlp(P, f) = \frac{1}{3} \cdot wlp(\text{diverge}, [x=10])$$

$$+ \frac{2}{3} \underbrace{wlp(x := 10, [x=10])}$$

$$= wp(x := 10, \ldots)$$

$$= \frac{1}{3} \cdot \underbrace{wlp(\text{diverge}, [x=10])}_{=1} + \frac{2}{3} \cdot \underbrace{[10=10]}_{=1}$$

$$= 1$$

$P ::$

$c := 1;$ while (c) $\{$ diverge $[\frac{1}{2}]$ $x{+}{+}$;

skip $[\frac{1}{2}]$ $c := 0$ $\}$

$f = [x \text{ is even}]$

$wlp(P, [x \text{ is even}]) =$

$$\frac{2}{3} + \frac{4 \cdot [x \text{ odd}]}{15} + \frac{[x \text{ even}]}{15}$$

Bounded expectation transformer semantics of pGCL

Syntax

- **skip**
- **diverge**
- x := E
- x :≈ μ
- P1 ; P2
- **if** (G) P1 **else** P2
- P1 [p] P2
- **while** (G)P

$wlp(P, f)$

f

$1 = \text{greatest elt } (\mathbb{E}_{\leq 1}, \sqsubseteq)$

$f(x := E)$

$wlp(P_1, wlp(P_2, f))$

gfp

$wlp(P, \cdot) : \mathbb{E}_{\leq 1} \longrightarrow \mathbb{E}_{\leq 1}$

$wp(P, \cdot) : \mathbb{E} \longrightarrow \mathbb{E}$

Bounded expectation transformer semantics of pGCL

Syntax

- `skip`
- `diverge`
- `x := E`
- `x :≈` μ
- `P1 ; P2`
- `if (G) P1 else P2`
- `P1 [p] P2`
- `while (G)P`

Semantics $wlp(P, f)$

- f
- 1
- $f[x := E]$
- $\lambda s. \int_{\mathbb{Q}} (\lambda v. f(s[x := v]))\, d\mu_s$
- $wlp(P_1, wlp(P_2, f))$
- $[G] \cdot wlp(P_1, f) + [\neg G] \cdot wlp(P_2, f)$
- $p \cdot wlp(P_1, f) + (1-p) \cdot wlp(P_2, f)$
- $\text{gfp } X.\ ([G] \cdot wlp(P, X) + [\neg G] \cdot f)$

gfp is the greatest fixed point operator wrt. the ordering $\sqsubseteq$ on bounded expectations $\mathbb{E}_{\leq 1}$.

Loops

$$wlp(\text{while } (G)\{ P \}, f) \;=\; \text{gfp}\, X.\; \underbrace{([G] \cdot wlp(P, X) + [\neg G] \cdot f)}_{\Psi(X)}$$

Loops

$$wlp(\text{while }(G)\{\,P\,\}, f) \;=\; \text{gfp}\,X.\,\underbrace{([\,G\,]\cdot wlp(P,X) + [\neg G\,]\cdot f)}_{\Psi(X)}$$

Scott continuity of Ψ

The function $\Psi : \mathbb{E}_{\leq 1} \to \mathbb{E}_{\leq 1}$ (defined as above) is continuous on $(\mathbb{E}_{\leq 1}, \sqsubseteq)$.

Proof.

Left as an exercise. $\square$

Loops

$$wlp(\text{while}\ (G)\{\ P\ \}, f) \ = \ \text{gfp}\, X.\ \underbrace{([\,G\,]\cdot wlp(P, X) + [\neg G\,]\cdot f)}_{\Psi(X)}$$

Scott continuity of Ψ

The function $\Psi : \mathbb{E}_{\leq 1} \rightarrow \mathbb{E}_{\leq 1}$ (defined as above) is continuous on $(\mathbb{E}_{\leq 1}, \sqsubseteq)$.

Proof.

Left as an exercise. □

Corollary

By Kleene's fixpoint theorem, it follows gfp $\Psi = \sup_{n\in\mathbb{N}} \Psi^n(\mathbf{1})$.

$\Phi^n(\mathbf{1})$ denotes the expected value over the final states of running while $(G)\{\ P\ \}$ exactly n times for the constant expectation $\mathbf{1}$.

A more tricky loopy program

```
c := 1;
while (c = 1) {
    { abort } [0.5] { x++ };
    { skip } [0.5] { c := 0 }
}
```

What is the probability that
either x is even on termination, or the program diverges?

$c := 1 ;$ while (c) $\{$ div $[\frac{1}{2}]$ $x{+}{+}$; skip $[\frac{1}{2}]$ $c := 0 \}$

$f = [x \text{ is even}]$

$$\Psi(X) = [c \neq 1] \cdot [x \text{ even}] + [c=1] \left(\frac{1}{2} + \frac{X(x := x+1) + X(c := 0, x := x+1)}{4} \right.$$

$$\Psi(1) = [c \neq 1] \cdot [x \text{ even}] + [c=1]$$

$$\Psi^2(1) = [c \neq 1] \cdot [x \text{ even}] + [c=1] \left(\frac{3}{4} + \frac{[x \text{ odd}]}{4} \right)$$

$$\Psi^3(1) = \quad 〃 \quad 〃 \quad + [c=1] \left(\frac{11}{16} + \frac{[x \text{ even}]}{16} + \frac{[x \text{ odd}]}{4} \right)$$

this yields the pattern:

$$\Psi^n(1) = [c \neq 1] \cdot [x \text{ even}] + [c=1] \left(\frac{2^{n-1}+1}{4^{n-1}} + \sum_{i=0}^{\lfloor \frac{n-3}{2} \rfloor} \frac{[x \text{ even}]}{4^{2(i+1)}} + \sum_{i=0}^{\lfloor \frac{n-2}{2} \rfloor} \frac{[x \text{ odd}]}{4^{2i+1}} \right)$$

$$wp(\text{while} \ldots, [x \text{ even}]) = \sup_{n \in \mathbb{N}}$$

$$[c \neq 1] \cdot [x \text{ even}] + [c=1] \left(\frac{2}{3} + \frac{4 [x \text{ odd}]}{15} + \frac{[x \text{ even}]}{15} \right)$$

e.g. $$\sum_{i=0}^{\infty} \frac{1}{4^{2(i+1)}} = \frac{1}{4} \sum_{i=0}^{\infty} \left(\frac{1}{4^i} \right)^2 = \frac{1}{4} \cdot \frac{1}{1 - \frac{1}{16}} = \frac{4}{15}$$

$$wlp(\text{program}, [x \text{ even}]) = \frac{2}{3} + \frac{4 [x \text{ odd}]}{15} + \frac{[x \text{ even}]}{15}$$

Properties of weakest liberal pre-expectations

For all pGCL programs P and bounded expectations f, g it holds:

- Continuity: $wlp(P, \cdot)$ is continuous on $(\mathbb{E}_{\leq 1}, \sqsubseteq)$
- Monotonicity: $f \leq g$ implies $wlp(P, f) \leq wlp(P, g)$
- Superlinearity: $r \cdot wlp(P, f) + wlp(P, g) \leq wlp(P, r \cdot f + g)$ for every $r \in \mathbb{R}_{\geq 0}$
- Duality: $wlp(P, f) = wp(P, f) + (1 - wp(P, \mathbf{1}))$

  $wp(P, \mathbf{1})$ = termination probability of program P
- Coincidence: $\boxed{wlp(P, f) = wp(P, f)}$ for a.s.-terminating P

  $wp(P,1) = 1$
- Co-strictness: $wlp(P, \mathbf{1}) = \mathbf{1}$