Probabilistic Programming

Lecture #9: Probabilistic Weakest Preconditions

Joost-Pieter Katoen

RWTH Lecture Series on Probabilistic Programming 2022-23

Overview

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.

Probabilistic GCL: Syntax

Kozen | McIver | Morgan



- `skip` — empty statement
- `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.

Overview

Some preliminaries

- Variable valuation $s : \mathit{Vars} \to \mathbb{Q}$ maps each program variable onto a value (here: rational numbers)
- Let $\mathbb{S}$ denote the set of variable valuations.
- Let $[\![E]\!]_s$ denote the valuation of expression E in s
- The indicator function of guard φ is denoted by $[\varphi]$:

$$[\varphi](s) \;=\; \begin{cases} 1 & \text{if } s \models \varphi \\ 0 & \text{if } s \not\models \varphi \end{cases}$$

These are also known as Iverson brackets.

From predicates to quantities

- Let program P be:

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

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

- Let program P' be:

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

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

- The probability that `x` $= 10$ on P''s termination is:

$$\frac{4}{5}\underbrace{\cdot[x+5=10]}_{\text{Iverson brackets}} + \frac{1}{5}\cdot 1 \;=\; \frac{4\cdot[x=5]+1}{5}$$

Expectation transformers

Let the set of expectations[1] (read: random variables):

$$\mathbb{E} \;=\; \left\{ f \mid f : \underbrace{\mathbb{S}}_{\text{states}} \to \mathbb{R}_{\geqslant 0} \cup \{\infty\} \right\}$$

$(\mathbb{E}, \sqsubseteq)$ is a complete lattice where $f \sqsubseteq g$ if and only if $\forall s \in \mathbb{S}.\, f(s) \leqslant g(s)$

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

Function $\Phi : \mathbb{E} \to \mathbb{E}$ is an expectation transformer

expectations are the quantitative analogue of predicates

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

Weakest pre-expectations

For pGCL program P, let $wp[\![P]\!] : \mathbb{E} \to \mathbb{E}$ an expectation transformer.

$g = wp[\![P]\!](f)$ is P's weakest pre-expectation w.r.t. post-expectation f iff

g maps initial state s to the expected value of f after executing P on s.

Examples:

- For P_1:: `x := 5 [4/5] x := 10`, we have $wp[\![P_1]\!](x) = 6$
- For P_2:: `x := x+5 [4/5] x := 10`, we have:

$$wp[\![P_2]\!](x) = \frac{4x}{5} + 6 \quad \text{and} \quad wp[\![P_2]\!]([x = 10]) = \frac{4 \cdot [x = 5] + 1}{5}$$

Expected values

A probability distribution μ on a countable set X is a function

$$\mu : X \to [0,1] \quad \text{such that} \quad \sum_{x \in X} \mu(x) = 1.$$

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

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

Kozen's duality theorem

If μ_P^s is the distribution over the final states obtained by running P on the initial state s, then for post-expectation f:

$$\underbrace{\sum_{t \in \mathbb{S}} \mu_P^s(t) \cdot f(t)}_{\text{forward}} \;=\; \underbrace{wp[\![P]\!](f)(s)}_{\text{backward}}$$

Pictorially:

Reasoning about probabilities

A special case is when the post-expectation equals $[F]$ with $F \in \mathbb{P}$.

Then one can consider F as an "event" and, most importantly,

$wp[\![P]\!]([F])(s)$ is the probability that running program P on input s terminates in a final state $\tau \models F$.

Note: the special case $wp[\![P]\!](\mathbf{1}) =$ termination probability of program P

Operations on expectations

- For $k \in \mathbb{R}_{\geqslant 0} \cup \{\infty\}$, let $\lambda s.k$, also denoted $\mathbf{k}$, denote the expectation that is constantly k for all s
- For expression E, $x \in \mathit{Vars}$ 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 $c \in \mathbb{R}_{\geqslant 0}$, $(c \cdot f)(s) = c \cdot f(s)$
- For $f, g \in \mathbb{E}$, let $(f+g)(s) = f(s) + g(s)$.
  Multiplication and subtraction are defined analogously.

Overview

Expectation transformer semantics

Syntax probabilistic program P	Expectation $wp[\![P]\!](f)$
`skip`	f
$x := E$	$f[x := E]$
$x :\approx \mu$	$\lambda s. \int_{\mathbb{Q}} (\lambda v. f(s[x := v]))\, d\mu_s$
$P; Q$	$wp[\![P]\!]\,(wp[\![Q]\!](f))$
`if` (φ) P `else` Q	$[\varphi] \cdot wp[\![P]\!](f) + [\neg\varphi] \cdot wp[\![Q]\!](f)$
$P\,[p]\,Q$	$p \cdot wp[\![P]\!](f) + (1-p) \cdot wp[\![Q]\!](f)$
`while` (φ) $\{P\}$	$\mathit{lfp}\, X.\, \underbrace{(([\varphi] \cdot wp[\![P]\!](X)) + [\neg\varphi] \cdot f)}_{\text{loop characteristic function } \Phi_f(X)}$

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

Examples

1. Let program P be:

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

   For $f = x$, we have
   $wp[\![P]\!](x) = \frac{4}{5} \cdot wp[\![x := 5]\!](x) + \frac{1}{5} \cdot wp[\![x := 10]\!](x) = \frac{4}{5} \cdot 5 + \frac{1}{5} \cdot 10 = 6$
2. Let program P' be:

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

   For $f = x$, we have:
   $$\begin{aligned} wp[\![P']\!](x) &= \tfrac{4}{5} \cdot wp[\![x := 5]\!](x) + \tfrac{1}{5} \cdot wp[\![x := 10]\!](x) \\ &= \tfrac{4}{5} \cdot (x+5) + \tfrac{1}{5} \cdot 10 \;=\; \frac{4x}{5} + 6 \end{aligned}$$
3. For program P' (again) and $f = [x = 10]$, we have:
   $$\begin{aligned} wp[\![P']\!]([x{=}10]) &= \tfrac{4}{5} \cdot wp[\![x := x+5]\!]([x{=}10]) + \tfrac{1}{5} \cdot wp[\![x := 10]\!]([x{=}10]) \\ &= \tfrac{4}{5} \cdot [x+5 = 10] + \tfrac{1}{5} \cdot [10 = 10] \\ &= \frac{4 \cdot [x = 5] + 1}{5} \end{aligned}$$

Loops

$$wp[\![\text{while } (\varphi)\,\{\,P\,\}]\!](f) \;=\; \text{lfp}\, X.\; \underbrace{([\varphi] \cdot wp[\![P]\!](X) + [\neg\varphi] \cdot f)}_{\text{loop characteristic function } \Phi_f(X)}$$

- Function $\Phi_f : \mathbb{E} \to \mathbb{E}$ (defined above) is Scott continuous on $(\mathbb{E}, \sqsubseteq)$.
- By Kleene's fixed point theorem, it follows: $\text{lfp}\, \Phi_f = \sup_{n \in \mathbb{N}} \Phi_f^n(\mathbf{0})$.
- $\Phi^n(\mathbf{0})$ is the expected value over the final states of running the loop n times when starting with the constant expectation $\mathbf{0}$.

A simple loopy program

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

Q: What is the expected value of x on termination?

Practical relevance

- Formal verification of randomised algorithms
- Exact inference for probabilistic programs
- Deciding program equivalence
- Proving program transformations
- Expected resource consumption
- Proving almost-sure termination

Overview

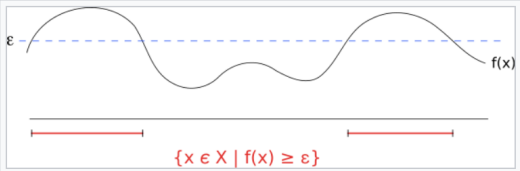
Properties of weakest pre-expectations

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

- Continuity: $wp[\![P]\!](\cdot)$ is Scott continuous on $(\mathbb{E}, \sqsubseteq)$.
- Monotonicity: $f \sqsubseteq g$ implies $wp[\![P]\!](f) \sqsubseteq wp[\![P]\!](g)$
- Feasibility: $f \sqsubseteq \mathbf{k}$ implies $wp[\![P]\!](f) \sqsubseteq \mathbf{k}$ for any constant function $\mathbf{k}$
- Linearity: $wp[\![P]\!](r \cdot f + g) = r \cdot wp[\![P]\!](f) + wp[\![P]\!](g)$ $\quad \forall r \in \mathbb{R}_{\geqslant 0}$
- Strictness: $wp[\![P]\!](\mathbf{0}) = \mathbf{0}$

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

Markov's inequality



Markov's inequality gives an upper bound for the measure of the set (indicated in red) where $f(x)$ exceeds a given level ε. The bound combines the level ε with the average value of f.

For all pGCL programs P and expectation f and $\varepsilon \in \mathbb{R}_{>0}$ it holds:

$$wp[\![P]\!]([f \geqslant \varepsilon]) \sqsubseteq \frac{wp[\![P]\!](f)}{\varepsilon}$$

Proof: by using monotonicity and linearity (for $g = \mathbf{0}$)

Backward compatibility

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

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

$$\underbrace{wp[\![P]\!]([F])}_{\text{pGCL, Lec \#9}} = \underbrace{[wp[\![P]\!](F)]}_{\text{Dijkstra, Lec \#8}}$$

For ordinary programs, probabilistic wp equals Dijkstra's wp

Overview

Recall: operational semantics of pGCL

Rewards

To reason about resource usage in MCs: use rewards.

A reward MC is a pair (D, r) with D an MC with state space Σ and $r : \Sigma \to \mathbb{R}$ a function assigning a real reward to each state.

The reward $r(\sigma)$ stands for the reward earned on leaving state σ.

Let $\pi = \sigma_0 \ldots \sigma_n$ be a finite path in (D, r) and $G \subseteq \Sigma$ a set of target states with $\pi \in \Diamond G$. The cumulative reward along π until reaching G is:

$$r_G(\pi) = r(\sigma_0) + \ldots + r(\sigma_{k-1}) \text{ where } \sigma_i \notin G \text{ for all } i < k \text{ and } \sigma_k \in G.$$

If $\pi \notin \Diamond G$, then $r_G(\pi) = \infty$.

Expected reward for reachability

Let σ be such that $Pr(\sigma \models \Diamond G) = 1$.

Then: the expected reward until reaching $G \subseteq \Sigma$ from $\sigma \in \Sigma$ is:

$$\mathrm{ER}(\sigma, \Diamond G) \;=\; \sum_{\widehat{\pi}} Pr(\widehat{\pi}) \cdot r_G(\widehat{\pi})$$

where $\widehat{\pi} = \sigma_0 \ldots \sigma_k$ is such that $\sigma_k \in G$, $\sigma_0 = \sigma$ and $\sigma_i \notin G$ for all $i < k$.

If $Pr(\sigma \models \Diamond G) < 1$, then let $\mathrm{ER}(\sigma, \Diamond G) = \infty$.

On computing expected rewards

Expected rewards in finite Markov chains can be computed in polynomial time by solving a system of linear equations.
(details on the black board.)

Equation system for expected rewards

Weakest pre-expectations = expected rewards

Compatibility theorem

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

$$\underbrace{wp[\![P]\!](f)(s)}_{\text{wp-semantics}} \quad = \quad \underbrace{ER^{[\![P]\!]}(s, \Diamond sink)}_{\text{operational semantics}}$$

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:

$$r(\langle \downarrow, s' \rangle) = f(s') \quad \text{and} \quad r(\cdot) = 0 \text{ otherwise.}$$

For finite-state programs, weakest pre-expectations can be computed by solving a system of linear equations, cf. a previous lecture.

Take-home messages

- Expectations are the quantitative analogue of predicates
- Expectations are mappings from states to extended reals
- Probabilistic weakest preconditions relate to expected values at termination
- Loops are defined by least fixed points
- For ordinary programs, probabilistic wp equals Dijkstra's wp

Next lecture:

weakest liberal expectations and syntax for expectations

Next lecture

Thursday Nov 17, 16:30