CS512 - Formal Methods

Thursday, March 29th, 2018

Note-taker: Glib Dolotov

Announcements: April 3, 5, 10 - Lectures by Alley Stoughton on *EasyCrypt*.
Assignment #7: due Tomorrow, March 30th.

Consider

$$\mathcal{P}: \quad ((x := 0) \oplus_{1/2} (x := 1)) \oplus_{1/2} (x := 2);$$
$$y := 3;$$

SET A: countable
$\vartheta : A \to [0,1]$
ϑ is a probability distribution such that

$$\sum_{a \in A} \vartheta(a) = 1.$$

If, however, we write

$$\sum_{a \in A} \vartheta(a) \le 1,$$

then ϑ is a sub-probability distribution. We allow for an event that we don't know the probability of.

$$\mathcal{D}(A) \triangleq \left\{ \vartheta : A \to [0,1] \mid \sum_{a \in A} \le 1 \right\}$$

For a *deterministic* program P

$$[\![P]\!] : \mathcal{S} \to \mathcal{S}$$

where $[\![P]\!]$ is interpreted as a "state transformer".

Note: Previous versions of the lecture notes used "Σ" to represent the set of possible states, we are now using "$\mathcal{S}$" instead to avoid confusing it with "summation".
,

For a *probabilistic* program $\mathcal{P}$

$$\langle\!\langle \mathcal{P} \rangle\!\rangle : \Theta \to \Theta$$

$\mathcal{S}$ (formerly called Σ) is the set of states σ.

$$\mathcal{S} = \{ \sigma : \mathcal{V} \to \mathbb{Z} \}$$

$\mathcal{V}$ is the set of all variables.
Θ is the set of all *probabilistic* states ϑ.

$$\vartheta \in \mathcal{D}(\mathcal{S}) = \{ \mathcal{S} \to [0,1] \mid \dots \}$$

Example:

$$\mathcal{V} = \{ x, y \}$$
$$\sigma : \{ x, y \} \to \mathbb{Z}$$
$$\sigma = \langle \sigma(x), \sigma(y) \rangle$$
$$\mathcal{S} = \mathbb{Z} \times \mathbb{Z}$$

$\sigma = \langle m, n \rangle$ where $m, n \in \mathbb{Z}$ has infinitely many possible states.

Suppose we start $\mathcal{P}$ from $\langle 5, 5 \rangle$. What is the probabilistic state when $\mathcal{P}$ terminates?

1. Note: we are inputting a *state* into a *probabilistic* program. However, *probabilistic* programs accept only *probabilistic states*. So we must first convert the state into a probabilistic state as follows:

$$\vartheta_\sigma(\sigma') = \begin{cases} 1 & \text{if } \sigma' = \sigma \\ 0 & \text{if } \sigma' \ne \sigma \end{cases}$$

2. Inputting $\langle 5, 5 \rangle$ into $\mathcal{P}$ yields a distribution function ϑ_1 which produces probabilities when given a possible output state.

$$\vartheta_1(\sigma) = \begin{cases} 1/4 & \text{if } \sigma = \langle 0, 3 \rangle \\ 1/4 & \text{if } \sigma = \langle 1, 3 \rangle \\ 1/2 & \text{if } \sigma = \langle 2, 3 \rangle \\ 0 & \text{otherwise} \end{cases}$$

Let us now consider a slightly different probabilistic program, $\mathcal{P}_2$. Differences between $\mathcal{P}$ and $\mathcal{P}_2$ are printed in blue.

$$\mathcal{P}_2: \quad ((x := 0) \oplus_{1/2} (x := 1)) \oplus_{1/2} (x := 2);$$
$$((y := 3) \oplus_{1/2} (y := 4));$$
$$\text{while } y = 4 \text{ do}$$
$$\quad \text{skip}$$
$$\text{od;}$$

Working with $\mathcal{P}_2$, we must note that the program will never terminate with a probability of $1/2$. The yield of $\mathcal{P}_2$ therefore becomes:

$$\vartheta_2(\sigma) = \begin{cases} 1/8 & \text{if } \sigma = \langle 0, 3 \rangle \\ 1/8 & \text{if } \sigma = \langle 1, 3 \rangle \\ 1/4 & \text{if } \sigma = \langle 2, 3 \rangle \\ 0 & \text{otherwise} \end{cases}$$

Consider $\mathcal{P}_3$ (changes between $\mathcal{P}_3$ and $\mathcal{P}_2$ are in purple):

$$
\begin{aligned}
\mathcal{P}_3: \quad & ((x := 0) \oplus_{1/2} (x := 1)) \oplus_{1/2} (x := 2); \\
& ((y := 3) \oplus_{2/3} (y := 4)); \\
& \text{while } y = 4 \text{ do} \\
& \quad \text{skip} \\
& \text{od};
\end{aligned}
$$

The yield of $\mathcal{P}_3$ becomes:

$$
\vartheta_2(\sigma) = \begin{cases} 1/6 & \text{if } \sigma = \langle 0, 3 \rangle \\ 1/6 & \text{if } \sigma = \langle 1, 3 \rangle \\ 1/3 & \text{if } \sigma = \langle 2, 3 \rangle \\ 0 & \text{otherwise} \end{cases}
$$

Consider $\mathcal{P}_4$ (changes between $\mathcal{P}_4$ and $\mathcal{P}_2$ are in red):

$$
\begin{aligned}
\mathcal{P}_2: \quad & ((x := 0) \oplus_{1/2} (x := 1)) \oplus_{1/2} (x := 2); \\
& ((y := 3) \oplus_{1/2} (y := 4)); \\
& \text{while } y = 4 \text{ do} \\
& \quad (y := 3) \oplus_{1/2} (y := 4); \\
& \text{od};
\end{aligned}
$$

$\mathcal{P}_4$ does, in fact, terminate because

$$
\lim_{n \to \infty} p^n = 0.
$$

In other words, each iteration has a possibility of continuing the program with a probability (p) of $1/2$. As the number of iterations (n) approaches infinity, the probability that the program hasn't yet terminated collapses to 0.

Consider integer expressions in classical Hoare logic:

$$
E = x + y - 4
$$

$$
\sigma = \langle 2, 13 \rangle
$$

Recall, E is used to denote integer expressions.
$[\![E]\!]\sigma = 2 + 13 - 4 = 11$ $\qquad$ $[\![E]\!] : \mathcal{S} \to \mathbb{Z}$

For probabilistic Hoare logic, integer expressions work as follows:

$$
\langle\!\langle E \rangle\!\rangle : \Theta \to \mathcal{D}(\mathbb{Z}) \qquad \text{note: } \Theta = \mathcal{D}(\mathcal{S}).
$$

Therefore:

$$
\langle\!\langle E \rangle\!\rangle : \mathcal{D}(\mathcal{S}) \to (\mathbb{Z} \to [0, 1]).
$$

Example: let's work with $\mathcal{P}$ and it's output

$$
\vartheta_1(\sigma) = \begin{cases} 1/4 & \text{if } \sigma = \langle 0, 3 \rangle \\ 1/4 & \text{if } \sigma = \langle 1, 3 \rangle \\ 1/2 & \text{if } \sigma = \langle 2, 3 \rangle \\ 0 & \text{otherwise} \end{cases}
$$

$$
E \triangleq x + y = 4
$$

$$
\langle\!\langle E \rangle\!\rangle \; \vartheta \; n = p
$$

$$
\langle\!\langle x + y - 4 \rangle\!\rangle \vartheta_1(-1) = 1/4
$$

$$
\langle\!\langle x + y - 4 \rangle\!\rangle \vartheta_1(n) = \begin{cases} 1/4 & \text{if } n = -1 \\ 1/4 & \text{if } n = 0 \\ 1/2 & \text{if } n = 1 \\ 0 & \text{otherwise} \end{cases}
$$

Now for Boolean expressions:
In **classical Hoare logic**: $[\![B]\!] : \mathcal{S} \to \mathbb{B}$
In **probabilistic Hoare logic**: $\langle\!\langle B \rangle\!\rangle : \vartheta = \mathcal{D}(\mathcal{S}) \to \mathcal{D}(\mathbb{B})$.

A probabilistic program is a transformer of probabilistic states.

Binary relation vs. a function: a function is a specific case of a binary relation.

In classical Hoare logic:

$$
[\![x := E]\!] = \{ (\sigma, \sigma[x \mapsto n]) \mid \sigma \in \mathcal{S}, \; n = [\![E]\!]\sigma \}
$$

At this point in the lecture, we began looking at the denotational semantics of commands of probabilistic programs in the lecture notes.

Pre- & Post-Conditions of Probabilistic H.L.

$\{\Phi\}\mathcal{P}\{\Psi\}$. We must somehow include probabilistic data in both pre- and post-conditions. We have yet to formally define these.