A Formalization of Probabilistic Programs in First-Order Logic

Pritom Rajkhowa

May 2019

1 Motivating Examples

1.1 Example 1

We first introduce probabilistic programs with an example. Then demonstrate how the proposed approach works.

Consider the following program probabilistic program. This program represents a game where a player flips a coin for y times. The player wins k dollars if the head turns up at the k^{th} flip. The following program models this game when the head probability of the coin is 0.5:

```
x=0; while (0<y) { x = x + y [1/2] x; y = y - 1;}
```

Our translator would be translated to a set of axioms $\Pi_P^{\vec{X}}$ like the following where variables $\vec{X} = \{x, y\}$

$$
\begin{aligned}
&x_1 = x_3(N_1), y_1 = y_3(N_1), \\
&\forall n_1.y_3(n_1+1) = y_3(n_1) - 1, \\
&\forall n_1.x_3(n_1+1) = 0.5 \times (x_3(n_1) + y_3(n_1)) + (1 - 0.5) \times x_3(n_1), \\
&x_3(0) = 0, y_3(0) = y \\
&\neg(y_3(N_1) > 0), \\
&\forall n_1.n_1 < N_1 \rightarrow (y_3(n_1) > 0).
\end{aligned}
$$

where x_1 and y_1 denote the output values of x and y, respectively, $x_3(n_1)$ and $y_3(n_1)$ the values of x and y during the n_1th iteration of the loop, respectively. Also, N_1 is a system-generated natural number constant denoting the number of iterations that the outer loop runs before exiting. The expected amount of money the player can win from this game is $E[x_1] = E[x_3(N_1)]$.

By using RS, our translator computes closed-form solution for $x_3(n_1+1)$ and moment based closed-form solution $E[y_3(n_1+1)]$, eliminates them, and produces the following axioms:

$$
\begin{aligned}
&E[x_1] = \frac{2 \times y \times N_1 - N_1^2 - N_1}{4}, y_1 = y - N_1,\\
&\neg((y - N_1) > 0),\\
&\forall n_1 . n_1 < N_1 \to (y - n_1) > 0.
\end{aligned}
$$

From these axioms, our system can show the post expectation of the above algorithm that expected the amount of the money a player can win is $E[x_1] = \frac{y^2-y}{4}$ with respect to pre-expectation $y > 0$. Our system can also show the post-expectation the program for the higher moment such as $E[x_1^2] = \frac{y^2-y}{4}$ and $E[x_1^3] = \frac{y^2-y}{4}$.

1.2 Example 2

We illustrate how our approach effectively handle nested loop where other approaches failed using the above simple example. The C code snippet P is from the benchmark used in [1] with variables $\vec{X} = \{x, y, k, L, M\}$. Our translator would be translated to a set of axioms $\Pi_P^{\vec{X}}$ like the following:

```
real x, y;
int k = 0 ;
while(x<=L)
{
    y = 0;
    while(y<=M){
        y = y + UniReal(-0.1,0.2);
    }
    x = x + UniReal(-0.1,0.2);
    k = k+1;
}
```

Our translator would be translated to a set of axioms $\Pi_P^{\vec{X}}$ like the following:

$$
\begin{aligned}
& x_1 = x_6(N_2), y_1 = y_6(N_2), k_1 = k_6(N_2), M_1 = M, L_1 = L, \\
& \forall n_1, n_2.y_2(n_1+1, n_2) = y_2(n_1, n_2) + uniReal(-0.1, 0.2, n_1, n_2) \\
& \forall n_2.y_2(0, n_2) = 0, \\
& \forall n_1, n_2.\neg(y_3(N_1(n_2), n_2) < M), \\
& \forall n_1, n_2.n_1 < N_1(n_2) \rightarrow (y_3(n_1, n_2) < M), \\
& \forall n_2.y_6(n_2+1) = y_2(N_1(n_2), n_2), \\
& \forall n_2.x_6(n_2+1) = x_6(n_2) + uniReal(-0.1, 0.2, n_2) \\
& \forall n_2.k_6(n_2+1) = k_6(n_2) + 1 \\
& x_6(0) = y, y_6(0) = 0, k_6(0) = 0 \\
& \neg(x_6(N_2) < L), \\
& \forall n_2.n_2 < N_2 \rightarrow (x_6(N_2) < L)
\end{aligned}
$$

where x_1, y_1, k_1, M_1 and N_1 denote the output values of x, y, k, M and N, respectively, $y_2(n_1, n_2)$ the value of y at n_1th iteration of the inter loop during n_2th iteration of the outer loop. $x_6(n_2)$, $y_6(n_2)$ and $k_6(n_2)$ the values of x, y and k during the n_2th iteration of the loop, respectively. Also N_2 is a system generated natural number constant denoting the number of iterations that the outer loop runs before exiting; N_1 is a system generated natural number function denoting that for each n_2, $N_1(n_2)$ is the number of iterations that the inner loop runs during the n_2th iteration of the outer loop.

It tries to find the closed form solutions of moment base recurrences relations generated by translator using our desgined system recurrence solver module(RS). After substituting moment base recurrences relations closed-form solution with resulted set of axoims $\Pi'^{\vec{X}}_P$ are as follows:

$$
\begin{aligned}
& E[x_1] = \frac{10 \times N_2}{3}, E[y_1] = \frac{10 \times N_1(N_2)}{3}, E[k_1] = N_2, , M_1 = M, L_1 = L \\
& \forall n_2.\neg(\frac{10 \times N_1(n_2)}{3} < M), \forall n_1, n_2.n_1 < N_1(n_2) \rightarrow (\frac{10 \times N_1(n_2)}{3} < M), \\
& \neg(\frac{10 \times N_2}{3} < L), \forall n_2.n_2 < N_2 \rightarrow (\frac{10 \times N_2}{3}) < L)
\end{aligned}
$$

For the above program, our system can show the post expectation of variable x and y as $\frac{10 \times N_2}{3}$ and $\frac{10 \times N_1(N_2)}{3}$ respectively.

When it tries to find the closed form solutions of higher moment or mixed moment using our desgined system recurrence solver module(RS). After substituting higher moment or mixed moment closed-form solution with resulted set of axoims $\Pi'^{\vec{X}}_{P}$ are as follows:

$$
\begin{aligned}
&Var[x_1] = \frac{9 \times N_1^4}{16} - \frac{9 \times N_1^3}{8} + \frac{15 \times N_1^2}{16} - \frac{3 \times N_1}{2} + 1,\\
&Var[y_1] = \frac{9 \times N_1^4}{16} - \frac{9 \times N_1^3}{8} + \frac{33 \times N_1^2}{16} - \frac{3 \times N_1}{2} + 1,\\
&Var[f_1] = \frac{9}{16},\\
&\neg((\frac{9 \times N_1^4}{16} - \frac{9 \times N_1^3}{8} + \frac{15 \times N_1^2}{16} - \frac{3 \times N_1}{2} + 1) < M),\\
&\forall n_1 . n_1 < N_1 \rightarrow ((\frac{9 \times n_1^4}{16} - \frac{9 \times n_1^3}{8} + \frac{15 \times n_1^2}{16} - \frac{3 \times n_1}{2} + 1) < M).
\end{aligned}
$$

References

[1] K. Chatterjee, H. Fu, P. Novotný, and R. Hasheminezhad, "Algorithmic analysis of qualitative and quantitative termination problems for affine probabilistic programs," *CoRR*, vol. abs/1510.08517, 2015. [Online]. Available: http://arxiv.org/abs/1510.08517