On lightweight Hoare logic of probabilistic programs: a bound tighter than the union bound

Xingyu Xie

Tsinghua University

Abstarct

In the formal verification of probabilistic programming, lightweight Hoare logics are proposed to reason about a bound of the failure probability of non-probabilistic assertions. The existing lightweight Hoare logic, aHL, relies on the union bound, a simple tool from probabilistic theory. However, we found that the union bound is loose in general.

In this work, we tighten the bound in aHL and prove its soundness and tightness. Downstream tools that rely on aHL can directly benefit from our out-of-the-box improvement. Practical applications to demonstrate the superiority of our theoretical improvements are in the plan.

Starting Point: aHL

aHL[1] is based on a standard probabilistic imperative language, whose core grammar is

$$c ::= x \leftarrow e \mid x \xleftarrow{\$} d \mid c; c \mid \textbf{if } b \textbf{ then } c \textbf{ else } c \mid \textbf{while } b \textbf{ do } c.$$

An aHL judgment is of the form

$$\vdash_\beta c : \Phi \Longrightarrow \Psi,$$

which means that from any initial program state satisfying Φ, after executing program c, Ψ holds except with a probability at most β. In other words, β is a bound of the failure probability for c with respect to the specification of precondition Φ and postcondition Ψ.

A key of aHL is how to give a bound for the rule of sequential composition. The solution is provided by the *union bound*: for events A and B, $\Pr[A \cup B] \leq \Pr[A] + \Pr[B]$. Internalizing the union bound in the logic, the sequential composition rule [Seq] of aHL is as follows.

$$[\textsc{Seq}]\frac{\vdash_{\beta_1} c_1 : \Phi \Longrightarrow \Xi \qquad \vdash_{\beta_2} c_2 : \Xi \Longrightarrow \Psi}{\vdash_{\beta_1+\beta_2} c_1; c_2 : \Phi \Longrightarrow \Psi}$$

The above rule expresses that if the failure probability for c_1 is no more than β_1 and the failure probability for c_2 is no more than β_2, the failure probability for $c_1; c_2$ is no more than $\beta_1 + \beta_2$. More detailedly, this rule expresses that if (1) from any state satisfying Φ, after executing program c_1, Ξ holds except with a probability at most β_1, and (2) from any state satisfying Ξ, after executing program c_2, Ψ holds except with a probability at most β_2, then from any state satisfying Φ, after executing program $c_1; c_2$, Ψ holds except with a probability at most $\beta_1 + \beta_2$.

Figure 1 shows the intuition of this bound. The blue ellipse represents the event that c_1 fails, and the yellow one represents the event that c_2 fails. The blue area is no more than β_1, and the yellow area is no more than β_2. Thus, the colored area is no more than $\beta_1 + \beta_2$, which indicates a bound for the event that $c_1; c_2$ fails.

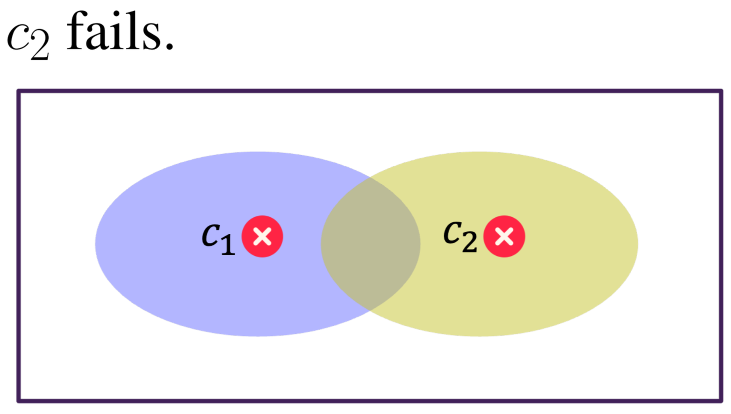

Figure 1: union bound: $\beta_1 + \beta_2$

Critique to aHL

Question We observe that the union bound, $\beta_1 + \beta_2$, may exceed 1, which is a useless case since the probability is always no more than 1 by definition.

However, finding a tight (accurate) failure bound is a fundamental quantitative analysis task for probabilistic programs. Now, a question arises naturally: What bound is tight enough for aHL?

Analysis We point out that the union bound ignores the dependence between the two composed programs: only when c_1 does not fail, it is meaningful for us to consider whether c_2 fails, as shown in Figure 2. Roughly speaking, $c_1; c_2$ fails only when c_1 and c_2 do not both succeed, which indicates that $1 - (1 - \beta_1)(1 - \beta_2)$ is a bound.

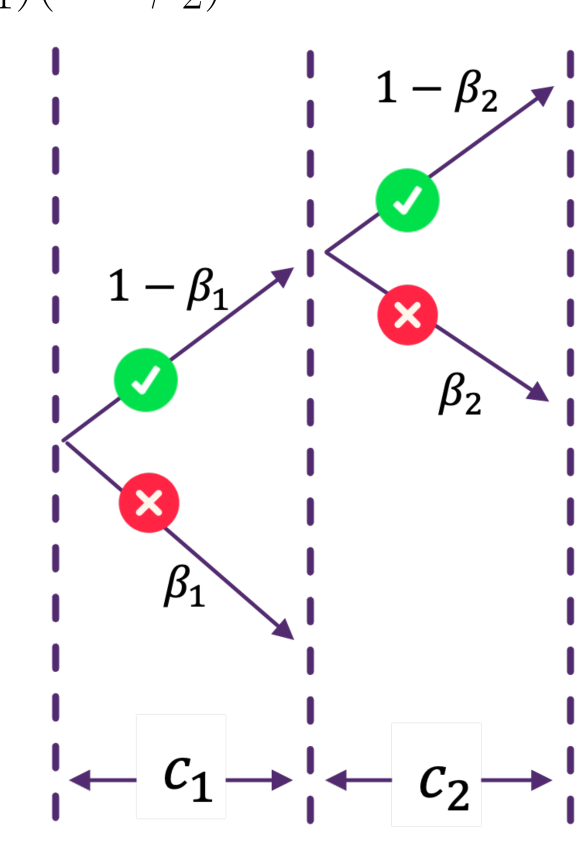

Figure 2: our bound:
$1 - (1 - \beta_1)(1 - \beta_2) = \beta_1 + \beta_2 - \beta_1\beta_2$

Results

We improve the crucial sequential composition rule [Seq] as follows.

$$[\textsc{Seq-X}]\frac{\vdash_{\beta_1} c_1 : \Phi \Longrightarrow \Xi \qquad \vdash_{\beta_2} c_2 : \Xi \Longrightarrow \Psi}{\vdash_{\beta_1+\beta_2-\beta_1\beta_2} c_1; c_2 : \Phi \Longrightarrow \Psi}$$

We prove (with pen and paper) that the rule [Seq-X] is sound and tight.

Theorem 1 (soundness) *For the rule* [Seq-X], *if the premise judgments are valid, then the conclusion judgment is valid.*

Theorem 2 (tightness) *There exist programs and specifications so that applying the rule* [Seq-X] *produces the* exact *failure probability of the conclusion.*

In other words, this tightness means that there *is* a practical application of the rule achieving the bound, which cannot be satisfied in aHL.

References

[1] G. Barthe et al. "A Program Logic for Union Bounds". In: *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).*