Lower Bounds for Possibly Divergent Probabilistic Programs

Shenghua Feng, Mingshuai Chen, Han Su, Benjamin L. Kaminski,
Joost-Pieter Katoen, Naijun Zhan

A Fun Fact

"A drunk man will find his way home, but a drunk bird may get lost forever."
— Kakutani' interpretation of [Pólya, Math. Ann. '21]

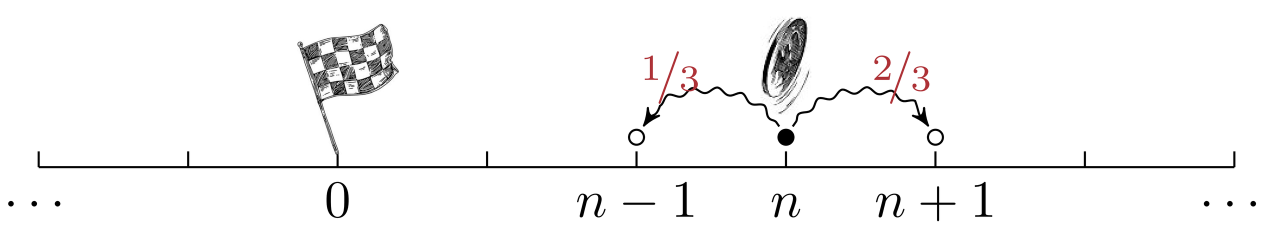
Yet, can we show *how likely at least* can a drunk bird return home?

In a Nutshell

Problem: How can we establish lower bounds on expected values of probabilistic programs that may exhibit divergence?

Example: Determine a non-trivial *lower bound on the termination probability* of the 1-D biased random walk which *diverges with non-zero probability*:

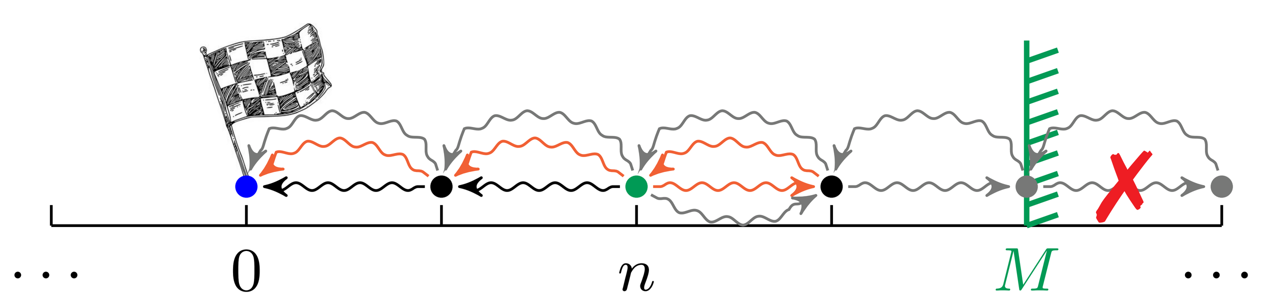
$$C_{\text{brw}}:\quad \texttt{while}\,(\,0 < n\,)\,\{\,n \coloneqq n-1\ [1/3]\ n \coloneqq n+1\,\}$$



Challenge: Existing lower bound induction rules are confined to either (i) *bounded* random variables with a priori knowledge on the termination probability [McIver & Morgan 2005]; or (ii) (universally) *almost-surely terminating* (AST) programs [Hark et al., POPL '20].

Our solution: The guard-strengthening technique. Strengthening the loop guard (and thereby the reachable state space) to forge an AST program, which can then be tackled by existing proof rules.

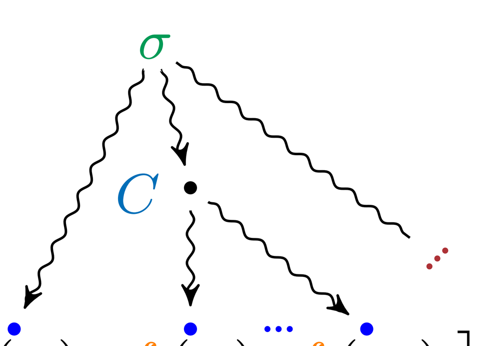
$$C^{M}_{\text{brw}}:\quad \texttt{while}\,(\,0 < n < M\,)\,\{\,n \coloneqq n-1\ [1/3]\ n \coloneqq n+1\,\}$$



Our guard-strengthening technique makes a connection between the lower bound for C_{brw} and the lower bound for C^{M}_{brw}. Moreover, the strengthened program C^{M}_{brw} features a *stronger* termination property (i.e., becoming AST), and thus is amenable to existing induction rules.

Formalization in the wp-Calculus

The expected value of function f after program C terminates is precisely captured by *weakest preexpectations* [Kozen; McIver, Morgan; Kaminski]:



$$\mathsf{wp}[\![C]\!](f)(\sigma) \triangleq \operatorname{Exp}\big[\, f(\tau_1)\quad f(\tau_2) \cdots f(\tau_m)\,\big]$$

Intuitively, $\mathsf{wp}[\![C]\!](f)(\sigma)$ represents the expected value of f evaluated in the final states reached after termination of C on input σ.

The crux of (probabilistic) program verification is to reason about `while`-loops: It amounts to determining the *quantitative least fixed point* (of some monotonic operator Φ_f capturing the loop semantics w.r.t. f) which is often difficult or even impossible to compute [Kaminski et al., Acta Inform. '19]:

$$\mathsf{wp}[\![\texttt{while}\,(\,\varphi\,)\,\{\,C\,\}]\!](f) \;=\; \mathsf{lfp}\ \Phi_f \;=\; ?$$

Limitations of Existing Lower Bound Rule

As computing the exact least fixed point $\mathsf{lfp}\ \Phi_f$ is often intractable, researchers seek to bound it from above and/or from below:

- Upper bounds (Park induction [Park, Mach. Intel. '69]) :

$$\Phi_f(u) \preceq u \quad\text{implies}\quad \mathsf{lfp}\ \Phi_f \preceq u$$

- Lower bounds [Hark et al., POPL '20]:

$$l \preceq \Phi_f(l) \;\wedge\; l \text{ is } \boxed{\text{uniformly integrable}} \quad\text{implies}\quad l \preceq \mathsf{lfp}\ \Phi_f$$

almost-sure termination
bounded expectations
...

The above lower bound rule does not apply to *divergent* programs, and even for AST ones, it requires extra proof efforts in, e.g., looking for supermartingales witnessing AST and reasoning about the looping time or establishing bounds on expectations to achieve uni. int. of l.

Our Approach: Guard-Strengthening

Guard-Strengthening Rule

$$C_{\text{loop}}:\quad \texttt{while}\,(\,\varphi\,)\,\{\,C\,\} \quad\rightsquigarrow\quad C'_{\text{loop}}:\quad \texttt{while}\,(\,\varphi'\,)\,\{\,C\,\}$$

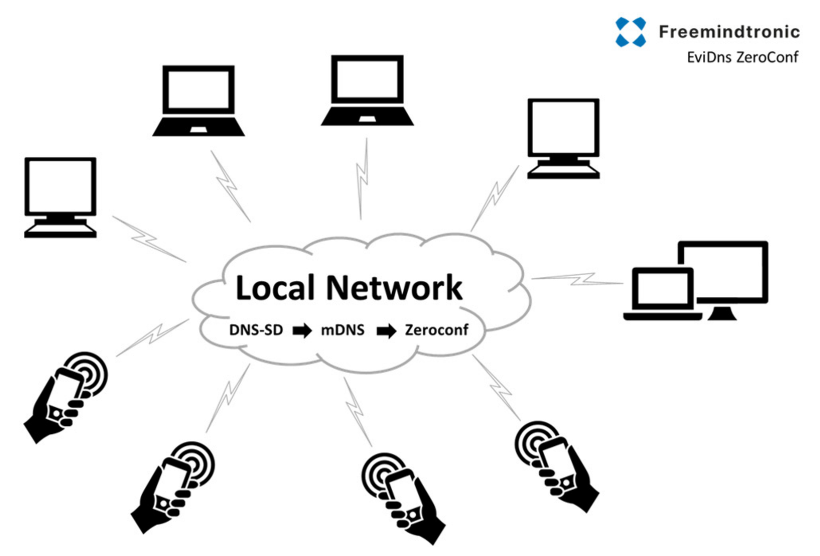
$$\frac{\varphi' \Longrightarrow \varphi \qquad l \preceq \mathsf{wp}[\![C'_{\text{loop}}]\!]([\neg\varphi]\cdot f)}{l \preceq \mathsf{wp}[\![C_{\text{loop}}]\!](f)}$$

Example: $l_M \;=\; [n<0] + [0 \le n \le M]\cdot\left((1/2)^n - (1/2)^M\right)$

$$\forall M \in \mathbb{N}:\quad l_M \overset{\text{Hark}}{\preceq} \mathsf{wp}[\![C^{M}_{\text{brw}}]\!]([n \le 0]\cdot 1) \overset{\text{Stren.}}{\preceq} \mathsf{wp}[\![C_{\text{brw}}]\!](1)$$

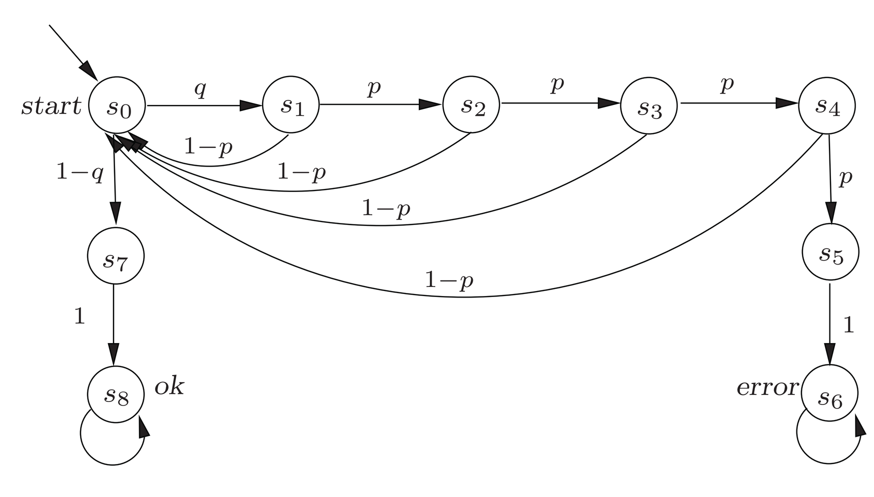
- **Complete:** where there's l, there's φ' (albeit not "good" enough).
- **General:** applicable to possibly *divergent* C_{loop} and *unbounded* f, l.
- **Tight:** the underapproximation error *approaches* 0 as $\varphi' \to \varphi$.
- **Automatable:** reducible to *probabilistic model checking* for finite-state C'_{loop} (which presents a solution to the drunk bird problem).

A "Real" Application: Zeroconf Protocol



©Freemindtronic

Self-configuring IP network



©[Baier & Katoen 2008]

Markov-chain snippet ($N = 4$)

C_{zc}: $\quad start = 1\,\mathring{,}\ established = 0\,\mathring{,}\ probe = 0\,\mathring{,}$
$\quad\texttt{while}\,(\,start \le 1 \wedge established \le 0 \wedge probe < N \wedge N \ge 4\,)\,\{$
$\quad\quad\texttt{if}\,(\,start = 1\,)\,\{$
$\quad\quad\quad\{\,start \coloneqq 0\,\}\ [0.5]\ \{\,start \coloneqq 0\,\mathring{,}\ established \coloneqq 1\,\}\,\}$
$\quad\quad\texttt{else}\,\{\,\{\,probe \coloneqq probe + 1\,\}\ [0.001]\ \{\,start \coloneqq 1\,\mathring{,}\ probe \coloneqq 0\,\}\,\}\,\}$

By strengthening the guard of C_{zc} with $N \le 10$, we are able to establish

$$\Pr(\text{"starting within the loop guard, } C_{\text{zc}} \text{ terminates with } established = 1\text{"}) \;\ge\; 0.99999999999$$