Lower Bounds for Possibly Divergent Probabilistic Programs

Mingshuai Chen

—Joint work with S. Feng, B. L. Kaminski, J.-P. Katoen, H. Su, and N. Zhan—

ROCKS · Nijmegen · May 2022
Work in Progress ...
\[C_{\text{brw}}: \quad \text{while}(n > 0)\{ n := n - 1 \quad [0.3] \quad n := n + 1 \}\]
The crux of probabilistic programming is to treat normal-looking programs as if they were probability distributions. — Michael Hicks, The PL Enthusiast
\( C_{\text{brw}}: \) while \((n > 0)\) \{ \( n := n - 1 \) \[0.3\] \( n := n + 1 \) \}
$C_{\text{brw}}$: while ($n > 0$) \{ $n := n - 1$ [0.3] $n := n + 1$ \}
The crux of probabilistic programming is to treat normal-looking programs as if they were probability distributions.

— Michael Hicks, The PL Enthusiast

\[ C_{\text{brw}}: \quad \text{while}(n > 0) \{ n := n - 1 \ [0.3] \ n := n + 1 \} \]
The crux of probabilistic programming is to treat normal-looking programs as if they were probability distributions.

— Michael Hicks, The PL Enthusiast

\[ C_{\text{brw}} : \quad \text{while}(n > 0) \{ n := n - 1 \ [0.3] \ n := n + 1 \} \]
The crux of probabilistic programming is to treat normal-looking programs as if they were probability distributions.

— Michael Hicks, The PL Enthusiast
Quantitative Reasoning about Probabilistic Loops

\[
\sigma \\
C \\
\tau_1 \quad \tau_2 \quad \tau_m \\

wp \ J C K (f) (\sigma) \equiv \text{Exp} [ \ ] \ \\
\text{while} (\phi) \{ C \} K (f) = \text{lfp} \Phi f = ?
\]
Quantitative Reasoning about Probabilistic Loops

\[ \sigma \]

\[ C \]

\[ f(\tau_1) \quad f(\tau_2) \quad f(\tau_m) \]

\[ \text{wp} \]

\[ J \]

\[ K \]

\[ \text{Exp} \]

\[ \text{while} (\phi) \{ C \} K (f) = \text{lfp} \Phi f = \text{?} \]
Quantitative Reasoning about Probabilistic Loops

\[ \sigma \\
C \\
\vdots \\
f(\tau_1) \\
f(\tau_2) \\
f(\tau_m) \\
\]

\[ \text{wp}([C](f)(\sigma)) \triangleq \text{Exp}[f(\tau_1), f(\tau_2), f(\tau_m)] \]
Quantitative Reasoning about Probabilistic Loops

\[
\begin{align*}
\wp [C] (f)(\sigma) & \triangleq \text{Exp} \left[ f(\tau_1), f(\tau_2), \ldots, f(\tau_m) \right] \\
\wp [\text{while}(\varphi) \{ C \}] (f) & = \text{lfp} \ \Phi_f = ?
\end{align*}
\]
Bounding the Least Fixed Point

\[ l \preceq \lfp \Phi_f \preceq u \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- **Upper bounds (Park induction):**

  \[ \Phi_f(u) \preceq u \quad \text{implies} \quad \text{lfp } \Phi_f \preceq u. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- Upper bounds (Park induction):
  \[ \Phi_f(u) \preceq u \implies \text{lfp } \Phi_f \preceq u. \]
Bounding the Least Fixed Point

\[ l \preceq \lfp \Phi_f \preceq u \]

- Upper bounds (Park induction):
  \[ \Phi_f(u) \preceq u \] implies \[ \lfp \Phi_f \preceq u \].

\[ u \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- Upper bounds (Park induction):
  \[ \Phi_f(u) \preceq u \implies \text{lfp } \Phi_f \preceq u . \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- **Upper bounds (Park induction):**
  \[ \Phi_f(u) \preceq u \implies \text{lfp } \Phi_f \preceq u. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- **Upper bounds (Park induction):**
  \[ \Phi_f(u) \preceq u \text{ implies } \text{lfp } \Phi_f \preceq u. \]

- **Lower bounds:**
  \[ l \preceq \Phi_f(l) \text{ implies } l \preceq \text{lfp } \Phi_f. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- **Upper bounds (Park induction):**
  \[ \Phi_f(u) \preceq u \text{ implies } \text{lfp } \Phi_f \preceq u. \]

- **Lower bounds:**
  \[ l \preceq \Phi_f(l) \text{ implies } l \preceq \text{lfp } \Phi_f. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- **Upper bounds (Park induction):**
  \[ \Phi_f(u) \preceq u \implies \text{lfp } \Phi_f \preceq u. \]

- **Lower bounds:**
  \[ l \preceq \Phi_f(l) \implies l \preceq \text{lfp } \Phi_f. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- Upper bounds (Park induction):
  \[ \Phi_f(u) \preceq u \text{ implies } \text{lfp } \Phi_f \preceq u. \]

- Lower bounds:
  \[ l \preceq \Phi_f(l) \text{ implies } l \preceq \text{lfp } \Phi_f. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- **Upper bounds (Park induction):**
  \[ \Phi_f(u) \preceq u \implies \text{lfp } \Phi_f \preceq u. \]

- **Lower bounds:**
  \[ l \preceq \Phi_f(l) \implies l \preceq \text{lfp } \Phi_f. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- Upper bounds (Park induction):
  \[ \Phi_f(u) \preceq u \text{ implies } \text{lfp } \Phi_f \preceq u. \]

- Lower bounds:
  \[ l \preceq \Phi_f(l) \text{ implies } l \preceq \text{lfp } \Phi_f. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- **Upper bounds (Park induction):**
  \[ \Phi_f(u) \preceq u \implies \text{lfp } \Phi_f \preceq u. \]

- **Lower bounds (Hark et al.'s rule):**
  \[ l \preceq \Phi_f(l) \land l \text{ is uni. int.} \implies l \preceq \text{lfp } \Phi_f. \]
Bounding the Least Fixed Point

\[ l \preceq \text{lfp } \Phi_f \preceq u \]

- **Upper bounds (Park induction):**
  \[ \Phi_f(u) \preceq u \implies \text{lfp } \Phi_f \preceq u \]

- **Lower bounds (Hark et al.’s rule):**
  \[ l \preceq \Phi_f(l) \land l \text{ is uni. int.} \implies l \preceq \text{lfp } \Phi_f \]

almost-sure termination
bounded expectations

...
A New Proof Rule for Lower Bounds

$\textbf{Loop}: \quad \text{while}(\varphi)\{C\} \quad \sim \quad \text{Loop'}: \quad \text{while}(\varphi')\{C\}$

Applicable to possibly divergent Loop.

$l$ can be arbitrarily tight.

Reducible to probabilistic BMC.

Easier to ensure uni. int. for Loop'.

Mingshuai Chen · i2, RWTH Aachen
A New Proof Rule for Lower Bounds

\[ \text{Loop: } \text{while}(\varphi)\{C\} \quad \overset{\sim}{\rightarrow} \quad \text{Loop': } \text{while}(\varphi')\{C\} \]

\[
\begin{align*}
\varphi' & \implies \varphi \\
\frac{\varphi' \implies \varphi}{l \leq \text{wp}[\text{Loop'}](\neg \varphi \cdot f)} \\
\frac{\varphi' \implies \varphi}{l \leq \text{wp}[\text{Loop}](f)}
\end{align*}
\]
A New Proof Rule for Lower Bounds

\[ \text{Loop: while}(\varphi)\{ C \} \quad \rightsquigarrow \quad \text{Loop': while}(\varphi')\{ C \} \]

\[ \varphi' \Rightarrow \varphi \quad l \leq \wp[\text{Loop'}](\lnot \varphi \cdot f) \]
\[ l \leq \wp[\text{Loop}](f) \]

- Applicable to possibly divergent Loop.
A New Proof Rule for Lower Bounds

\[
\text{Loop}: \quad \text{while}(\varphi)\{C\} \quad \sim \sim \quad \text{Loop'}: \quad \text{while}(\varphi')\{C\}
\]

\[
\varphi' \implies \varphi\quad l \leq \text{wp}[\text{Loop'}] (\lnot\varphi \cdot f) \\
l \leq \text{wp}[\text{Loop}] (f)
\]

- Applicable to \textit{possibly divergent} \text{Loop}.
- \textit{l} can be \textit{arbitrarily tight}.
A New Proof Rule for Lower Bounds

\[ \text{Loop: } \text{while}(\varphi)\{C\} \quad \mapsto \quad \text{Loop': } \text{while}(\varphi')\{C\} \]

\[ \varphi' \implies \varphi \quad l \leq \text{wp}[\text{Loop'}](\neg \varphi \cdot f) \]
\[ l \leq \text{wp}[\text{Loop}](f) \]

- Applicable to \textit{possibly divergent Loop}.
- \( l \) can be \textit{arbitrarily tight}.
- Reducible to \textit{probabilistic BMC}.
A New Proof Rule for Lower Bounds

**Loop**: \( \text{while}(\varphi)\{ C \} \quad \leadsto \quad \text{Loop}'\): \( \text{while}(\varphi')\{ C \} \)

\[
\varphi' \implies \varphi \quad l \leq \wp[\text{Loop}'] ([\neg \varphi] \cdot f) \\
\quad l \leq \wp[\text{Loop}'] (f)
\]

- Applicable to *possibly divergent Loop*.
- \( l \) can be *arbitrarily tight*.
- Reducible to *probabilistic BMC*.
- Easier to ensure *uni. int.* for \( \text{Loop}' \).
- ...

Mingshuai Chen · i2, RWTH Aachen
Example: Biased Random Walk

\( C_{brw} : \)  \( \text{while} \ (0 < n) \) \{ 
\[ n := n - 1 \] 0.3 \[ n := n + 1 \] 
\} 

\( C_{brw}^M : \)  \( \text{while} \ (0 < n < M) \) \{ 
\[ n := n - 1 \] 0.3 \[ n := n + 1 \] 
\}
Example: Biased Random Walk

\[ C_{brw} : \quad \text{while} (0 < n) \{
\quad n := n - 1 \quad [0.3] \quad n := n + 1
\} \]

\[ C_{brw}^M : \quad \text{while} (0 < n < M) \{
\quad n := n - 1 \quad [0.3] \quad n := n + 1
\} \]
Example: Biased Random Walk

\[ C_{\text{brw}}: \quad \text{while } (0 < n) \{ \]
\[ n := n - 1 \quad [0.3] \quad n := n + 1 \]
\[ \} \]

\[ C_{\text{brw}}^M: \quad \text{while } (0 < n < M) \{ \]
\[ n := n - 1 \quad [0.3] \quad n := n + 1 \]
\[ \} \]

\[ \ell_M(n) = (\frac{3}{7})^n - (\frac{3}{7})^M. \]
Example: Biased Random Walk

\[ C_{brw}: \quad \text{while}(0 < n) \{ \]
\[ n := n - 1 \quad [0.3] \quad n := n + 1 \]
\[ \} \]

\[ C^M_{brw}: \quad \text{while}(0 < n < M) \{ \]
\[ n := n - 1 \quad [0.3] \quad n := n + 1 \]
\[ \} \]

\[ l_M(n) = (\frac{3}{7})^n - (\frac{3}{7})^M. \]

\[ \forall M \in \mathbb{N}: \quad l_M \leq \text{wp}[C^M_{brw}](\{n \leq 0\} \cdot 1) \leq \text{wp}[C_{brw}](1). \]
Example: Biased Random Walk

\[ C_{\text{brw}} : \text{while} (0 < n) \{ \]
\[ n := n - 1 \]
\[ n := n + 1 \]
\[ \} \]

\[ \]

\[ C_{\text{brw}} : \text{while} (0 < n < M) \{ \]
\[ n := n - 1 \]
\[ n := n + 1 \]
\[ \} \]

\[ l_M(n) = (\frac{3}{7})^n - (\frac{3}{7})^M. \]

\[ \forall M \in \mathbb{N} : \quad l_M \leq \wp[C_{\text{brw}}^M]([n \leq 0] \cdot 1) \leq \wp[C_{\text{brw}}](1). \]

\[ (\frac{3}{7})^n = \lim_{M \to \infty} l_M \leq \wp[C_{\text{brw}}](1). \]
Example: Biased Random Walk

\[ C_{\text{brw}} : \quad \text{while} \ (0 < n) \{ \]
\[ n := n - 1 \quad \text{[0.3]} \quad n := n + 1 \]
\[ \} \]

\[ C^M_{\text{brw}} : \quad \text{while} \ (0 < n < M) \{ \]
\[ n := n - 1 \quad \text{[0.3]} \quad n := n + 1 \]
\[ \} \]

\[ l_M(n) = (3/7)^n - (3/7)^M . \]

\[ \forall M \in \mathbb{N} : \quad l_M \leq \text{wp}[C^M_{\text{brw}}]([n \leq 0] \cdot 1) \leq \text{wp}[C_{\text{brw}}] (1) . \]

\[ (3/7)^n = \lim_{M \to \infty} l_M \overset{\text{Park}}{=} \text{wp}[C_{\text{brw}}] (1) . \]