

# Lower Bounds for Possibly Divergent Probabilistic Programs

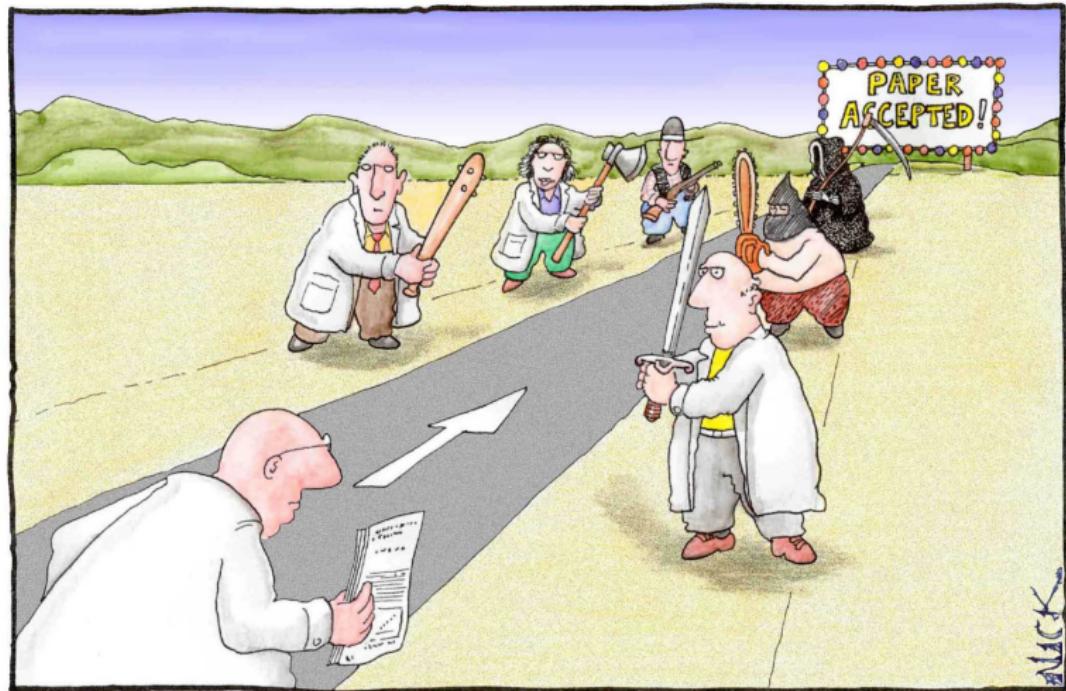
Mingshuai Chen

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



ROCKS · Nijmegen · May 2022

# Work in Progress ...

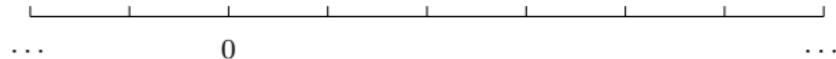


# Probabilistic Programs

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

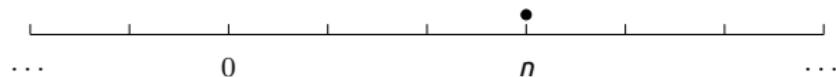
# Probabilistic Programs

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



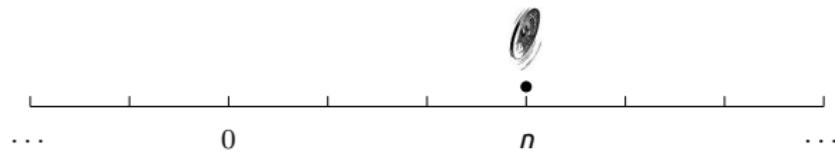
# Probabilistic Programs

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



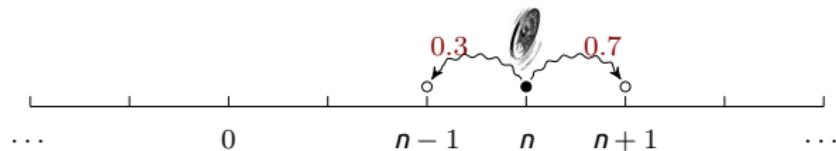
# Probabilistic Programs

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



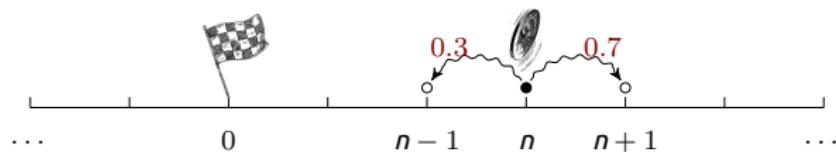
# Probabilistic Programs

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



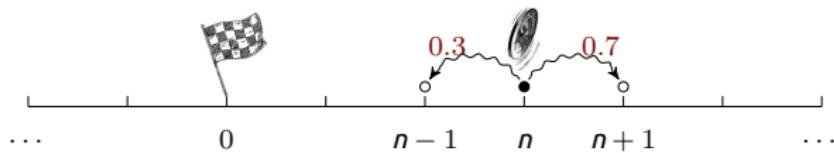
# Probabilistic Programs

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



# Probabilistic Programs

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

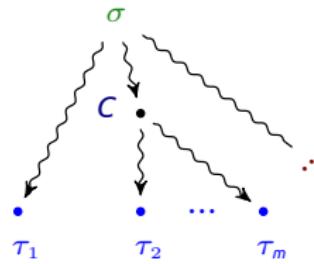


*"The crux of probabilistic programming is to treat normal-looking programs as if they were probability distributions."*

— Michael Hicks, The PL Enthusiast

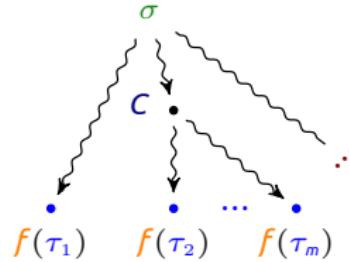
# Quantitative Reasoning about Probabilistic Loops

[Kozen; McIver, Morgan; Kaminski]



# Quantitative Reasoning about Probabilistic Loops

[Kozen; McIver, Morgan; Kaminski]



The diagram illustrates a probabilistic loop structure. It features a central node labeled  $C$  in blue. From this node, several arrows point downwards to other nodes, which are represented by small black dots. One of these arrows is explicitly labeled with the Greek letter  $\sigma$  above it in green. Below the diagram, there is a mathematical expression:

$$\text{wp}[\![C]\!](f)(\sigma) \triangleq \text{Exp}[f(\tau_1) \quad f(\tau_2) \quad \dots \quad f(\tau_m)]$$

# Quantitative Reasoning about Probabilistic Loops

[Kozen; McIver, Morgan; Kaminski]

$$\text{wp}[\![C]\!](f)(\sigma) \triangleq \text{Exp}[f(\tau_1) \quad f(\tau_2) \quad \dots \quad f(\tau_m)]$$

$$\text{wp}[\!\![\text{while } (\varphi) \{ C \}]\!](f) = \text{lfp } \Phi_f = ?$$

# Bounding the Least Fixed Point

$$l \preceq \text{lfp } \Phi_f \preceq u$$

# 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) :

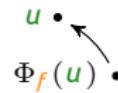
$$\Phi_f(u) \preceq u \text{ implies } \text{lfp } \Phi_f \preceq u.$$

# Bounding the Least Fixed Point

$$l \preceq \text{lfp } \Phi_f \preceq u$$

- Upper bounds (Park induction) :

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

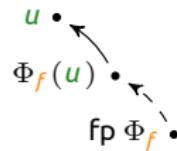


# Bounding the Least Fixed Point

$$l \preceq \text{lfp } \Phi_f \preceq u$$

- Upper bounds (Park induction) :

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

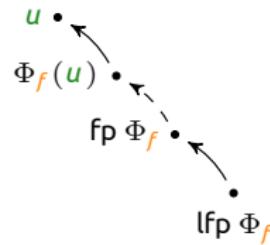


# Bounding the Least Fixed Point

$$l \preceq \text{lfp } \Phi_f \preceq u$$

- Upper bounds (Park induction) :

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



# Bounding the Least Fixed Point

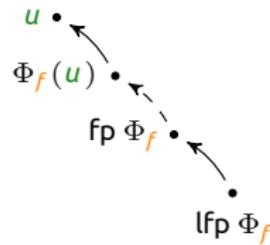
$$l \preceq \text{lfp } \Phi_f \preceq u$$

- 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

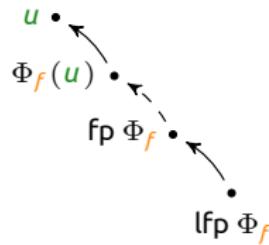
$$l \preceq \text{lfp } \Phi_f \preceq u$$

- 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

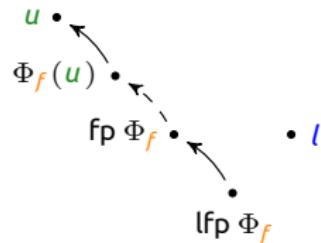
$$l \preceq \text{lfp } \Phi_f \preceq u$$

- Upper bounds (Park induction) :

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

- Lower bounds :

$$l \preceq \Phi_f(l) \text{ implies } l \preceq \text{lfp } \Phi_f.$$



# Bounding the Least Fixed Point

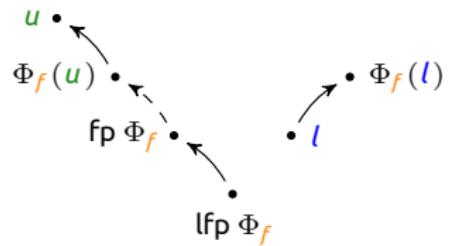
$$l \preceq \text{lfp } \Phi_f \preceq u$$

- 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

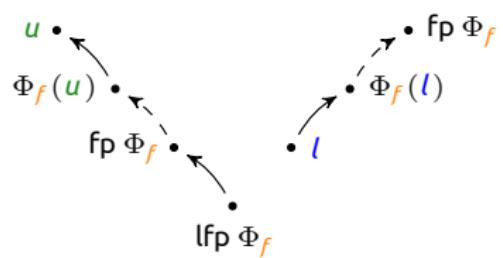
$$l \preceq \text{lfp } \Phi_f \preceq u$$

- Upper bounds (Park induction) :

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

- Lower bounds :

$$l \preceq \Phi_f(l) \text{ implies } l \preceq \text{lfp } \Phi_f.$$



# Bounding the Least Fixed Point

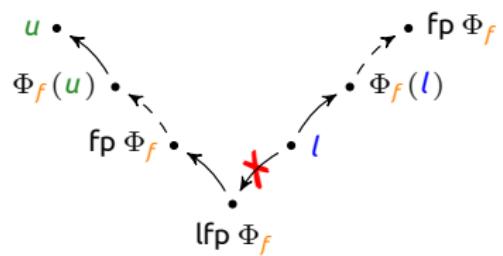
$$l \preceq \text{lfp } \Phi_f \preceq u$$

- 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

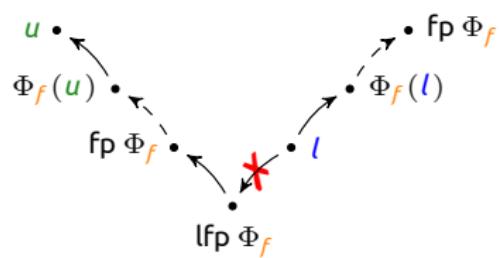
$$l \preceq \text{lfp } \Phi_f \preceq u$$

- Upper bounds (Park induction) :

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

- Lower bounds (Hark et al.'s rule) :

$$l \preceq \Phi_f(l) \wedge l \text{ is uni. int. implies } l \preceq \text{lfp } \Phi_f.$$



# Bounding the Least Fixed Point

$$l \preceq \text{lfp } \Phi_f \preceq u$$

- Upper bounds (Park induction) :

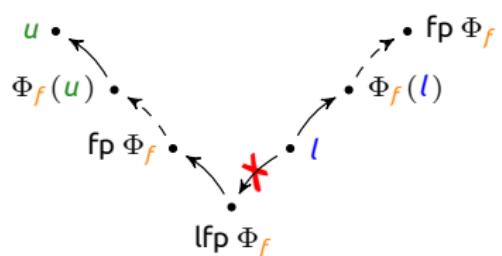
$$\Phi_f(u) \preceq u \text{ implies } \text{lfp } \Phi_f \preceq u.$$

- Lower bounds (Hark et al.'s rule) :

$$l \preceq \Phi_f(l) \wedge l \text{ is uni. int.} \text{ implies } l \preceq \text{lfp } \Phi_f.$$

↑  
almost-sure termination  
bounded expectations

...



# A New Proof Rule for Lower Bounds

*Loop*: `while`( $\varphi$ ) {  $C$  }  $\rightsquigarrow$  *Loop'*: `while`( $\varphi'$ ) {  $C$  }

# A New Proof Rule for Lower Bounds

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

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

# A New Proof Rule for Lower Bounds

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

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

- Applicable to *possibly divergent Loop*.

# A New Proof Rule for Lower Bounds

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

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

- Applicable to *possibly divergent Loop*.
- $l$  can be *arbitrarily tight*.

# A New Proof Rule for Lower Bounds

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

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

- Applicable to *possibly divergent Loop*.
- $l$  can be *arbitrarily tight*.
- Reducible to *probabilistic BMC*.

# A New Proof Rule for Lower Bounds

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

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

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

## Example : Biased Random Walk

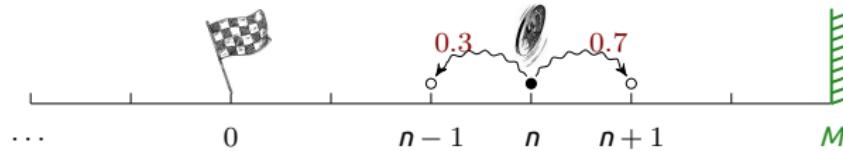
$C_{\text{brw}}$ :    `while ( $0 < n$ ) {  
     $n := n - 1$  [0.3]  $n := n + 1$   
}`

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

## Example : Biased Random Walk

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

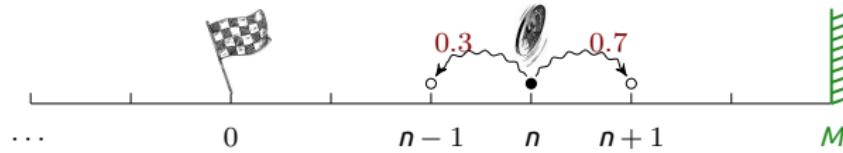
$C_{\text{brw}}^M$ :    `while ( $0 < n < M$ ) {  
     $n := n - 1 [0.3]$   $n := n + 1$   
}`



## Example : Biased Random Walk

```
Cbrw: while (0 < n) {
    n := n - 1 [0.3] n := n + 1
}
```

```
CbrwM: while (0 < n < M) {
    n := n - 1 [0.3] n := n + 1
}
```

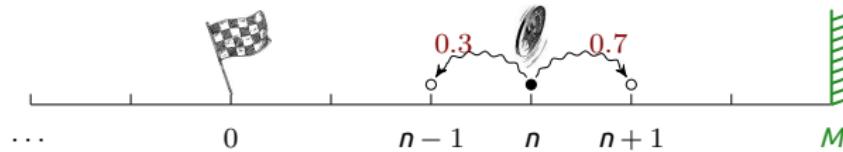


$$l_M(n) = (3/7)^n - (3/7)^M .$$

## Example : Biased Random Walk

```
Cbrw: while (0 < n) {
    n := n - 1 [0.3] n := n + 1
}
```

```
CbrwM: while (0 < n < M) {
    n := n - 1 [0.3] n := n + 1
}
```

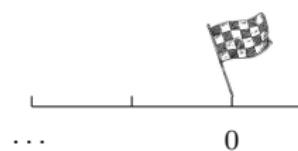


$$l_M(n) = (3/7)^n - (3/7)^M .$$

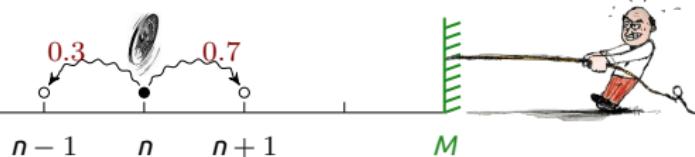
$$\forall M \in \mathbb{N}: l_M \preceq \text{wp}[\![C_{\text{brw}}^M]\!] ([n \leq 0] \cdot \textcolor{orange}{1}) \preceq \text{wp}[\!C_{\text{brw}}]\!(\textcolor{orange}{1}) .$$

## Example : Biased Random Walk

```
Cbrw: while (0 < n) {
    n := n - 1 [0.3] n := n + 1
}
```



```
CbrwM: while (0 < n < M) {
    n := n - 1 [0.3] n := n + 1
}
```



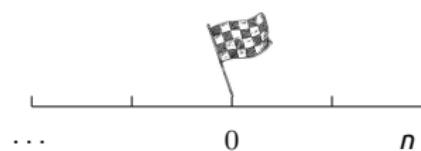
$$l_M(n) = (3/7)^n - (3/7)^M .$$

$$\forall M \in \mathbb{N}: l_M \preceq \text{wp}[\mathcal{C}_{\text{brw}}^M]([n \leq 0] \cdot \textcolor{orange}{1}) \preceq \text{wp}[\mathcal{C}_{\text{brw}}](\textcolor{orange}{1}) .$$

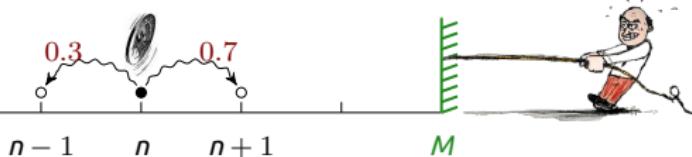
$$(3/7)^n = \lim_{M \rightarrow \infty} l_M \preceq \text{wp}[\mathcal{C}_{\text{brw}}](\textcolor{orange}{1}) .$$

## Example : Biased Random Walk

```
Cbrw: while (0 < n) {
    n := n - 1 [0.3] n := n + 1
}
```



```
CbrwM: while (0 < n < M) {
    n := n - 1 [0.3] n := n + 1
}
```



$$l_M(n) = (3/7)^n - (3/7)^M .$$

$$\forall M \in \mathbb{N}: l_M \preceq \text{wp}[\mathcal{C}_{\text{brw}}^M]([n \leq 0] \cdot \textcolor{orange}{1}) \preceq \text{wp}[\mathcal{C}_{\text{brw}}](\textcolor{orange}{1}) .$$

$$(3/7)^n = \lim_{M \rightarrow \infty} l_M \stackrel{\text{Park}}{=} \text{wp}[\mathcal{C}_{\text{brw}}](\textcolor{orange}{1}) .$$