Automated Run–Time Analysis of Probabilistic Programs

Marcel Hark

Outline

Introduction

Basics of Probabilistic Programs

Semantics of Probabilistic Programs

My Work

Outline

Non–Probabilistic Programs

Example

$x := n$
while $x > 0$ **do**
| $x := x - 1$

Non–Probabilistic Programs

Example

$x := n$
while $x > 0$ **do**
| $x := x - 1$

$\rightarrow$ How to estimate number of loop-iterations?

Non–Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

| $x := x - 1$

$\rightarrow$ How to estimate number of loop-iterations? $\Rightarrow$ ranking function.

Non–Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

| $x := x - 1$

$\rightarrow$ How to estimate number of loop-iterations? $\Rightarrow$ ranking function.

$\rightarrow$ $R : \mathbb{Z}^2 \rightarrow \mathbb{Z}, (x, n) \mapsto x$ is a ranking function for the loop.

Non–Probabilistic Programs

Example

$x := n$
while $x > 0$ **do**
| $x := x - 1$

$\rightarrow$ R strictly positive

$\rightarrow$ How to estimate number of loop-iterations? $\Rightarrow$ ranking function.

$\rightarrow$ $R : \mathbb{Z}^2 \rightarrow \mathbb{Z}, (x, n) \mapsto x$ is a ranking function for the loop.

Non–Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\mid\ x := x - 1$

$\rightarrow$ R strictly positive

$\rightarrow$ R strictly decreasing

$\rightarrow$ How to estimate number of loop-iterations? $\Rightarrow$ ranking function.

$\rightarrow$ $R : \mathbb{Z}^2 \rightarrow \mathbb{Z}, (x, n) \mapsto x$ is a ranking function for the loop.

Non–Probabilistic Programs

Example

$x := n$
while $x > 0$ **do**
| $x := x - 1$

$\rightarrow$ R strictly positive
$\rightarrow$ R strictly decreasing

$\rightarrow$ How to estimate number of loop-iterations? $\Rightarrow$ ranking function.

$\rightarrow$ $R : \mathbb{Z}^2 \rightarrow \mathbb{Z}, (x, n) \mapsto x$ is a ranking function for the loop.

$\rightarrow$ Therefore the loop is executed at most $\max(x, 0)$ times.

Non–Probabilistic Programs

Example

$x := n$
while $x > 0$ **do**
| $x := x - 1$

→ R strictly positive
→ R strictly decreasing

→ How to estimate number of loop-iterations? ⇒ ranking function.

→ $R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto x$ is a ranking function for the loop.

→ Therefore the loop is executed at most $\max(x, 0)$ times.

→ x has size n.

Introduction

Non–Probabilistic Programs

Example

$x := n$
while $x > 0$ **do**
$\mid\ x := x - 1$

$\rightarrow R$ strictly positive
$\rightarrow R$ strictly decreasing

$\rightarrow$ How to estimate number of loop-iterations? $\Rightarrow$ ranking function.

$\rightarrow R : \mathbb{Z}^2 \rightarrow \mathbb{Z}, (x, n) \mapsto x$ is a ranking function for the loop.

$\rightarrow$ Therefore the loop is executed at most $\max(x, 0)$ times.

$\rightarrow x$ has size n.

$\rightarrow max(n, 0)$ is an upper bound for number of loop iterations.

Non–Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

| $x := x - 1$

→ R strictly positive

→ R strictly decreasing

→ How to estimate number of loop-iterations? ⇒ ranking function.

→ $R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto x$ is a ranking function for the loop.

→ Therefore the loop is executed at most $\max(x, 0)$ times.

→ x has size n.

→ $max(n, 0)$ is an upper bound for number of loop iterations.

→ $max(n, 0) + 1$ is an upper bound on the worst–case runtime of the program.

Non–Probabilistic Programs

Example

$x := n$
while $x > 0$ **do**
| $x := x - 1$

→ R strictly positive
→ R strictly decreasing

→ How to estimate number of loop-iterations? ⇒ ranking function.

→ $R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto x$ is a ranking function for the loop.

→ Therefore the loop is executed at most $\max(x, 0)$ times.

→ x has size n.

→ $max(n, 0)$ is an upper bound for number of loop iterations.

→ $max(n, 0) + 1$ is an upper bound on the worst–case runtime of the program.

→ Automated in our tool „KoAT“ [ACM–TOPLAS 2016].

Outline

Probabilistic Programs

Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

  $x := x - 1$

Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

$n = 1$

Probabilistic Programs

Example

$x := n$

while $x > 0$ do

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

$n = 1$

$x = 1$

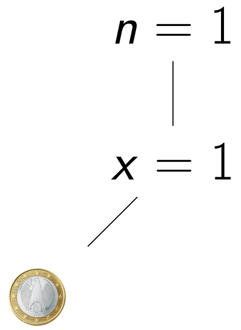
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1 \rangle + \frac{1}{2}\langle x \rangle$



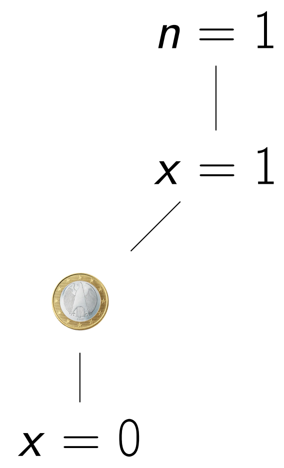
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



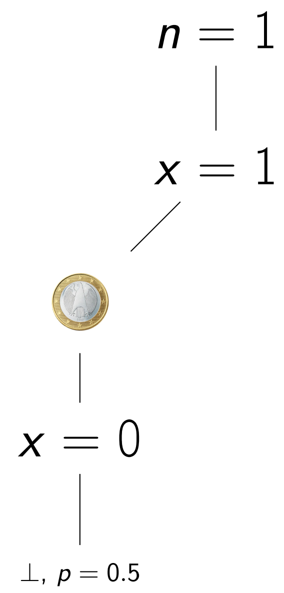
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



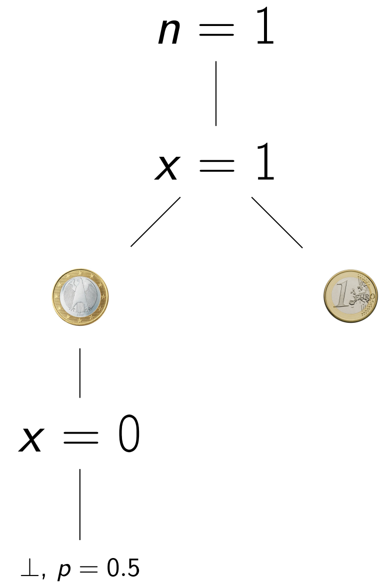
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



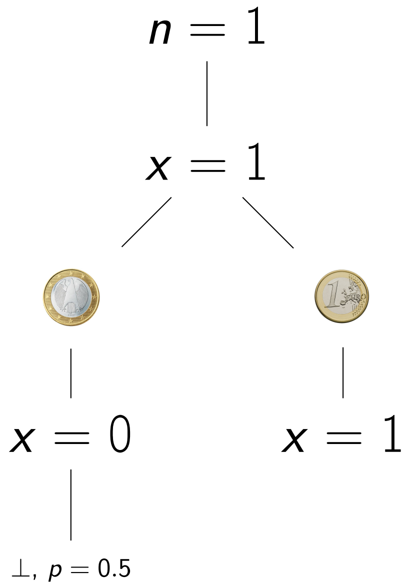
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



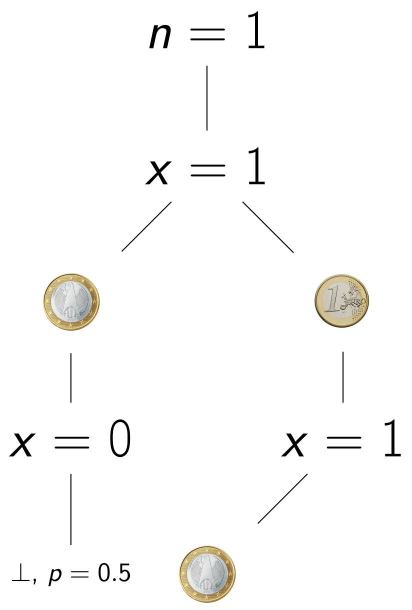
Probabilistic Programs

Example

$x := n$

while $x > 0$ do

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



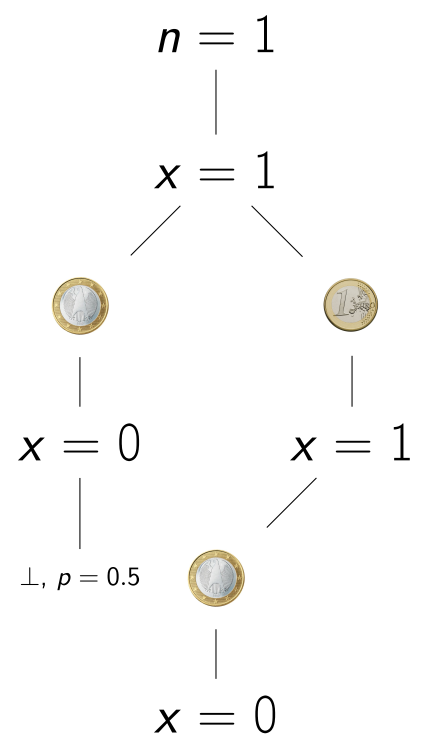
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1 \rangle + \frac{1}{2}\langle x \rangle$



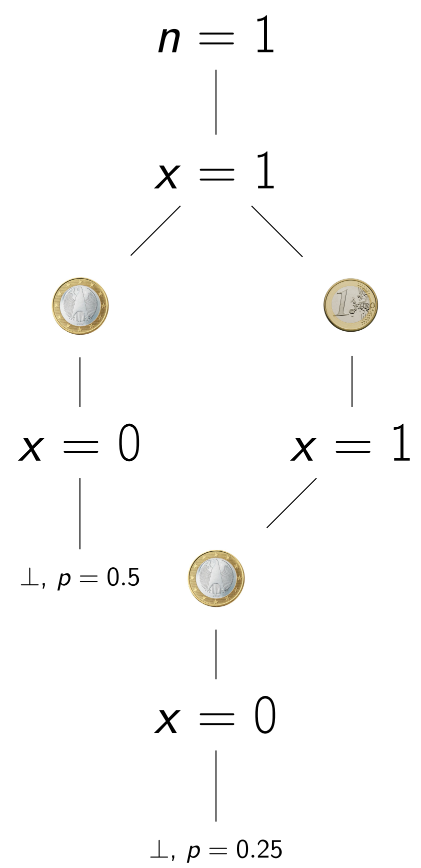
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



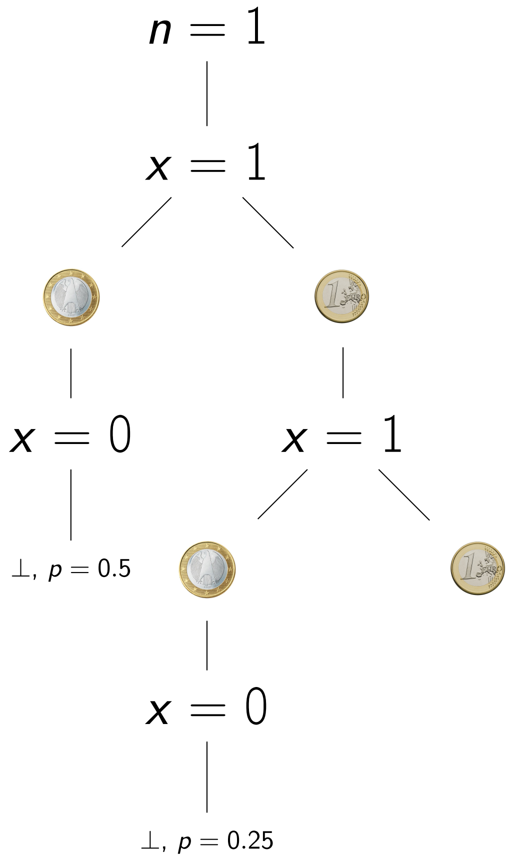
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



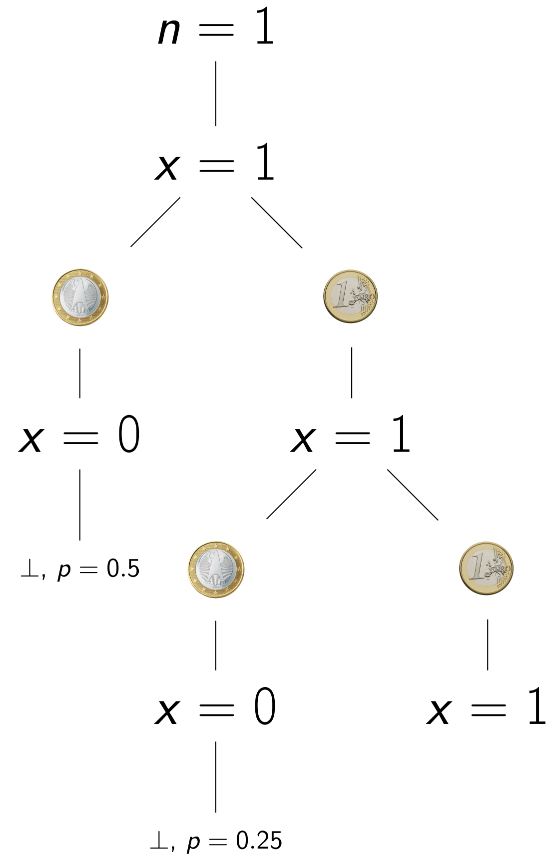
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



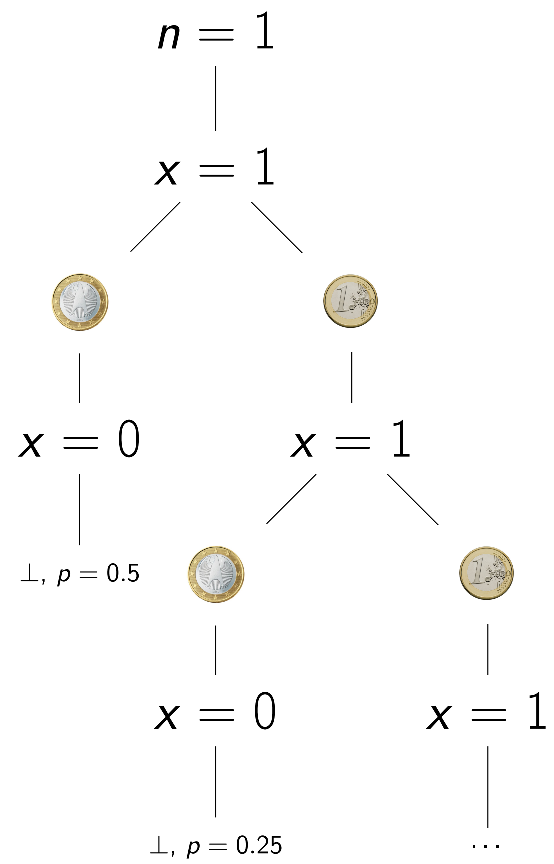
Probabilistic Programs

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



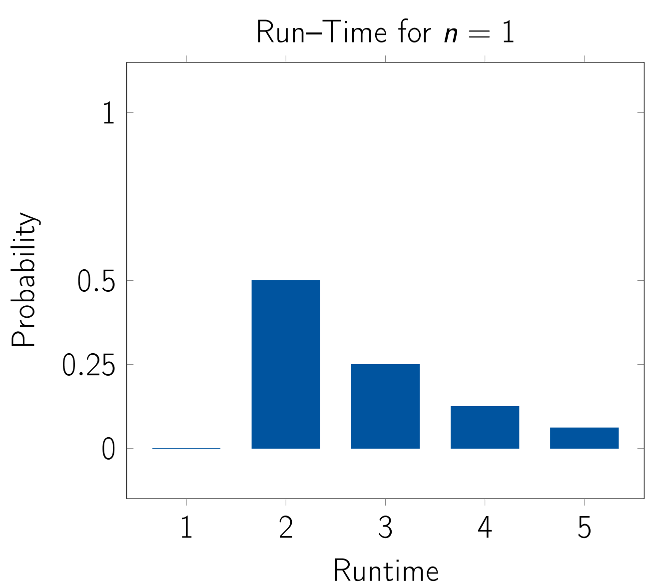
Probabilistic Programs

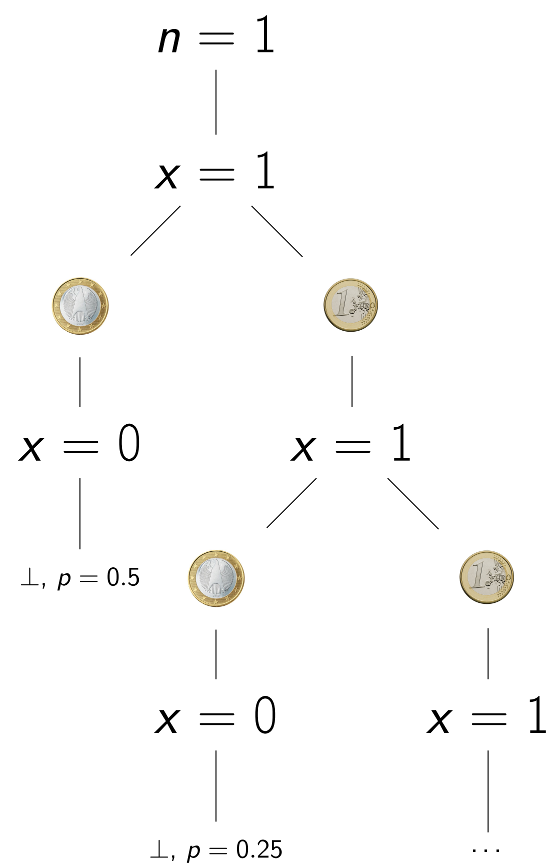
Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$





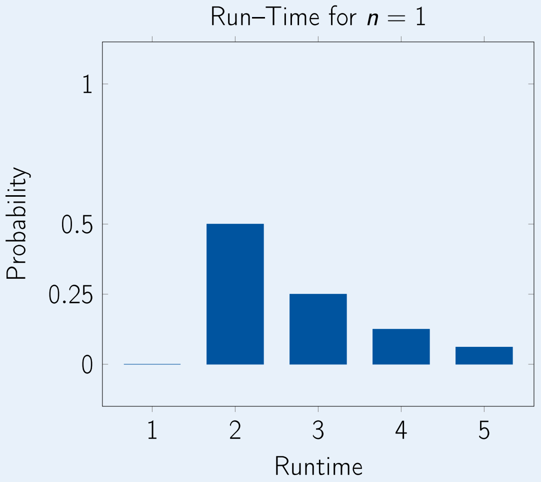
Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



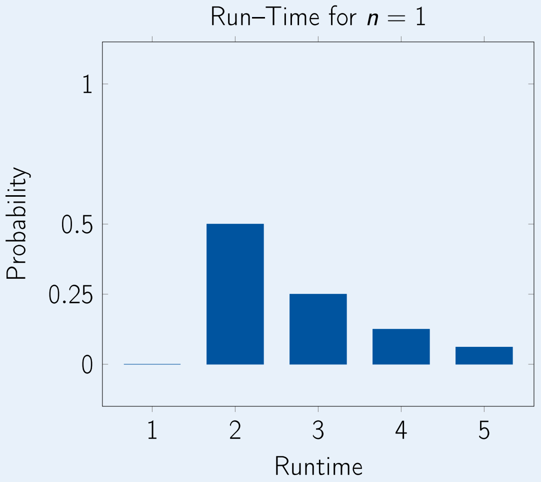
Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



- Runtime is a positive random-variable *Term*

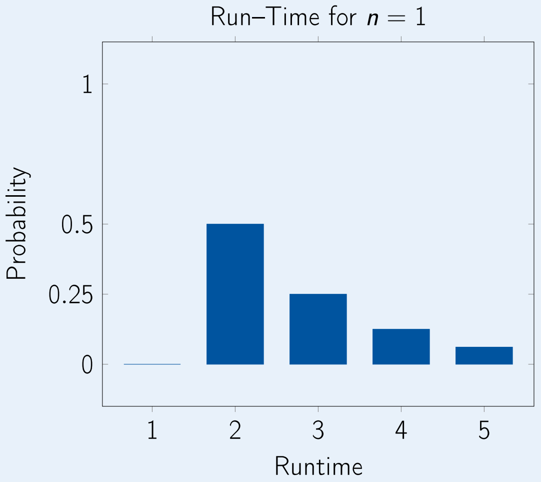
Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$



- Runtime is a positive random-variable *Term* $\Rightarrow \mathbb{P}(Term = m + 1) = \left(\frac{1}{2}\right)^m, \; m \geq 1$

Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \left(\frac{1}{2}\right)^m, \ m \geq n \geq 1$

Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m + 1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \; m \geq n \geq 1$

Types of Termination

Example

$x := n$
while $x > 0$ **do**
$\quad x := \frac{1}{2}\langle x-1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \ m \geq n \geq 1$

Definition

A probabilistic program terminates

Types of Termination

Example

$x := n$
while $x > 0$ **do**
$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m + 1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \; m \geq n \geq 1$

Definition

A probabilistic program terminates

- **almost surely** if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)

Types of Termination

Example

$x := n$
while $x > 0$ **do**
$\quad x := \frac{1}{2}\langle x-1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \; m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) =$

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)

Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m + 1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \ m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m + 1)$

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)

Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x-1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \; m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m+1) = 1$

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)

Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \ m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m+1) = 1 \Rightarrow$ A.S.T

Definition

A probabilistic program terminates

- **almost surely** if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)

Types of Termination

Example

$x := n$
while $x > 0$ **do**
$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m + 1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \; m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m + 1) = 1 \Rightarrow$ A.S.T

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- **positive almost surely** if $\mathbb{E}(\mathit{Term}) < \infty$ (**P.A.S.T.**)

Types of Termination

Example

$x := n$
while $x > 0$ **do**
$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m + 1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \ m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m + 1) = 1 \Rightarrow$ A.S.T
- **Expected run–time:**

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Types of Termination

Example

$x := n$
while $x > 0$ **do**
$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \; m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m+1) = 1 \Rightarrow$ A.S.T
- **Expected run–time:** $\mathbb{E}(\mathit{Term})$

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Types of Termination

Example

$x := n$
while $x > 0$ **do**
  $x := \frac{1}{2}\langle x-1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \ m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m+1) = 1 \Rightarrow$ A.S.T
- **Expected run–time:** $\mathbb{E}(\mathit{Term}) = \sum_{m \geq n} (m+1) \cdot \mathbb{P}(\mathit{Term} = m+1)$

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Types of Termination

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \; m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m+1) = 1 \Rightarrow$ A.S.T
- **Expected run–time:** $\mathbb{E}(\mathit{Term}) = \sum_{m \geq n} (m+1) \cdot \mathbb{P}(\mathit{Term} = m+1) = 2n+1 < \infty$

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Types of Termination

Example

$x := n$
while $x > 0$ **do**
$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

- Runtime is a positive random-variable $\mathit{Term} \Rightarrow \mathbb{P}(\mathit{Term} = m+1) = \binom{m-1}{n-1} \cdot \left(\frac{1}{2}\right)^m, \; m \geq n \geq 1$
- $\mathbb{P}(\mathit{Term} < \infty) = \sum_{m \geq n} \mathbb{P}(\mathit{Term} = m+1) = 1 \Rightarrow$ A.S.T
- Expected run–time: $\mathbb{E}(\mathit{Term}) = \sum_{m \geq n} (m+1) \cdot \mathbb{P}(\mathit{Term} = m+1) = 2n+1 < \infty \Rightarrow$ P.A.S.T.

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- **positive almost surely** if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Theorem

P.A.S.T. implies A.S.T.

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Example

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Example

A.S.T. is weaker than P.A.S.T

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Example

A.S.T. is weaker than P.A.S.T

$x := n$
while $x > 0$ **do**
  $x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x + 1\rangle$

Types of Termination

Definition

A probabilistic program terminates

- **almost surely** if $\mathbb{P}(Term < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(Term) < \infty$ (P.A.S.T.)

Example

A.S.T. is weaker than P.A.S.T

$x := n$
while $x > 0$ **do**
  $x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x + 1\rangle$

$$\mathbb{P}(Term < \infty) = 1$$

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Example

A.S.T. is weaker than P.A.S.T

$x := n$

while $x > 0$ **do**

| $x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x + 1\rangle$

$$\mathbb{P}(\mathit{Term} < \infty) = 1$$

$$\mathbb{E}(\mathit{Term}) = \infty$$

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Objective

Reason about termination types fully automatically.

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Objective

Reason about termination types fully automatically.
Compute bounds on $\mathbb{E}(\mathit{Term})$.

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Objective

Reason about termination types fully automatically.
Compute bounds on $\mathbb{E}(\mathit{Term})$.

- Kaminski, Katoen, et al., POPL 2018.

Types of Termination

Definition

A probabilistic program terminates

- **almost surely** if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Objective

Reason about termination types **fully automatically**.
Compute bounds on $\mathbb{E}(\mathit{Term})$.

- Kaminski, Katoen, et al., POPL 2018.
- **Chatterjee et al., POPL 2018**

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- **positive almost surely** if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Objective

Reason about termination types fully automatically.
Compute bounds on $\mathbb{E}(\mathit{Term})$.

- Kaminski, Katoen, et al., POPL 2018.
- Chatterjee et al., POPL 2018
- **Kaminski, Katoen, Matheja, Olmedo, ESOP 2016.**

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- **positive almost surely** if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Objective

Reason about termination types **fully automatically**.
Compute bounds on $\mathbb{E}(\mathit{Term})$.

- Kaminski, Katoen, et al., POPL 2018.
- Chatterjee et al., POPL 2018
- Kaminski, Katoen, Matheja, Olmedo, ESOP 2016.
- **Hoffmann et al., to appear.**

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- **positive almost surely** if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Objective

Reason about termination types **fully automatically**.
Compute bounds on $\mathbb{E}(\mathit{Term})$.

- Kaminski, Katoen, et al., POPL 2018.
- **Chatterjee et al., POPL 2018 based on Bournez, Garnier, RTA 05.**
- Kaminski, Katoen, Matheja, Olmedo, ESOP 2016.
- Hoffmann et al., to appear.

Types of Termination

Definition

A probabilistic program terminates

- almost surely if $\mathbb{P}(\mathit{Term} < \infty) = 1$ (A.S.T.)
- positive almost surely if $\mathbb{E}(\mathit{Term}) < \infty$ (P.A.S.T.)

Remark

A.S.T. and P.A.S.T are **undecidable**.

Outline

Introduction

Basics of Probabilistic Programs

Semantics of Probabilistic Programs

My Work

Probabilistic Control Flow Graph

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

Probabilistic Control Flow Graph

Example

$x := n$
while $x > 0$ do
  if $prob(0.5)$ then
    $x := x - 1$
  else
    $x := x$

Semantics of Probabilistic Programs

Probabilistic Control Flow Graph

Example

$x := n$

while $x > 0$ **do**

  if $prob(0.5)$ **then**

    $x := x - 1$

  else

    $x := x$

Probabilistic Control Flow Graph

Example

ℓ_0

$x := n$

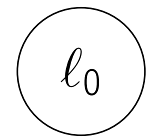
while $x > 0$ **do**

  if $prob(0.5)$ **then**

    $x := x - 1$

  else

    $x := x$



Probabilistic Control Flow Graph

Example

ℓ_0

$x := n$

ℓ_1

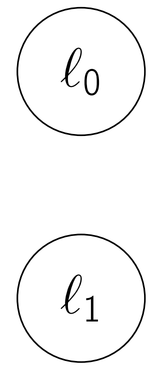
while $x > 0$ do

  if $prob(0.5)$ then

    $x := x - 1$

  else

    $x := x$



Probabilistic Control Flow Graph

Example

ℓ_0

$x := n$

ℓ_1

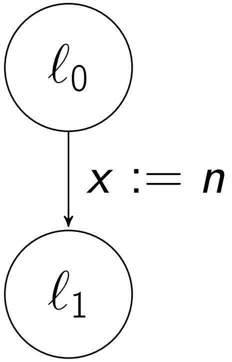
while $x > 0$ **do**

  if $prob(0.5)$ **then**

    $x := x - 1$

  else

    $x := x$



Probabilistic Control Flow Graph

Example

(ℓ_0)

$x := n$

(ℓ_1)

while $x > 0$ **do**

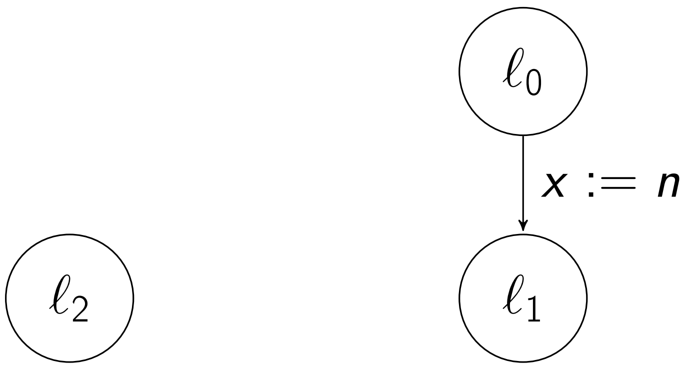
  (ℓ_2)

  if $\mathit{prob}(0.5)$ **then**

    $x := x - 1$

  else

    $x := x$



Probabilistic Control Flow Graph

Example

ℓ_0

$x := n$

ℓ_1

while $x > 0$ do

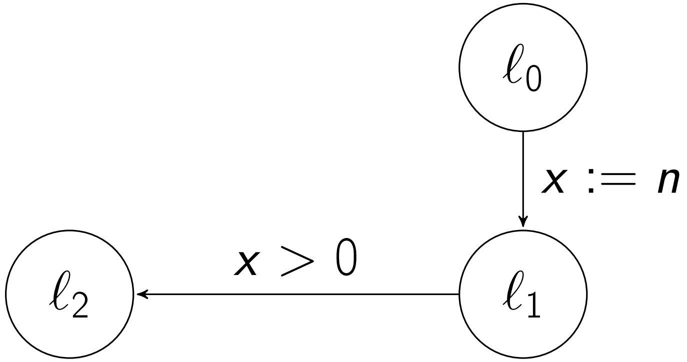
ℓ_2

if $prob(0.5)$ then

$x := x - 1$

else

$x := x$



Probabilistic Control Flow Graph

Example

ℓ_0

$x := n$

ℓ_1

while $x > 0$ **do**

  ℓ_2

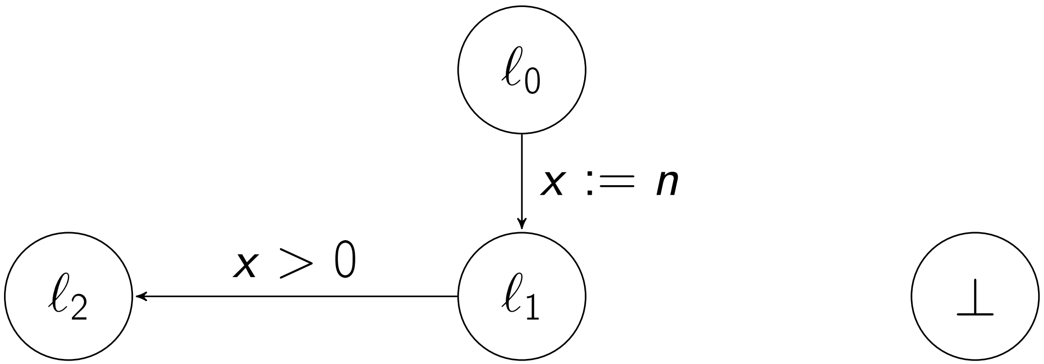
  if $prob(0.5)$ **then**

    $x := x - 1$

  else

    $x := x$

$\bot$



Probabilistic Control Flow Graph

Example

(ℓ_0)

$x := n$

(ℓ_1)

while $x > 0$ **do**

  (ℓ_2)

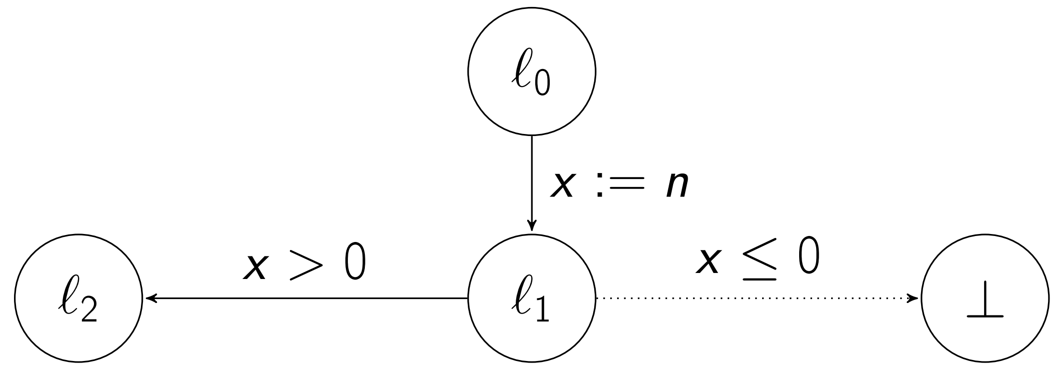
  if $prob(0.5)$ **then**

    $x := x - 1$

  else

    $x := x$

($\bot$)



Probabilistic Control Flow Graph

Example

ℓ_0

$x := n$

ℓ_1

while $x > 0$ do

ℓ_2

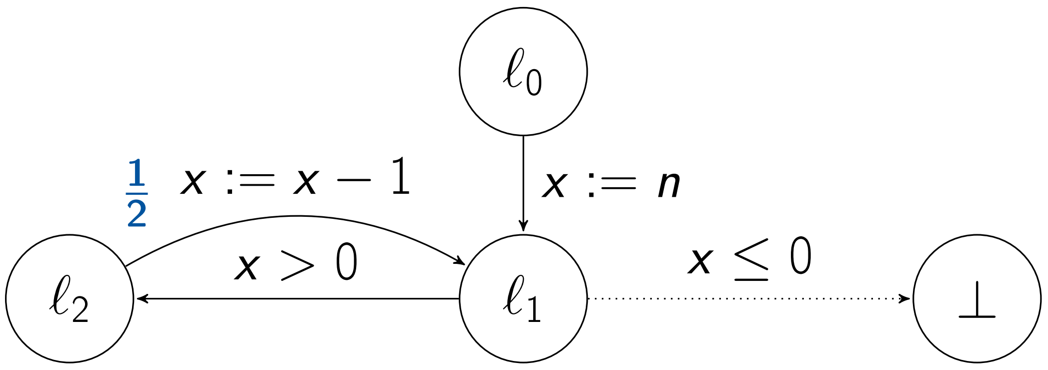
if $prob(0.5)$ then

$x := x - 1$

else

$x := x$

$\bot$



Probabilistic Control Flow Graph

Example

ℓ_0

$x := n$

ℓ_1

while $x > 0$ **do**

  ℓ_2

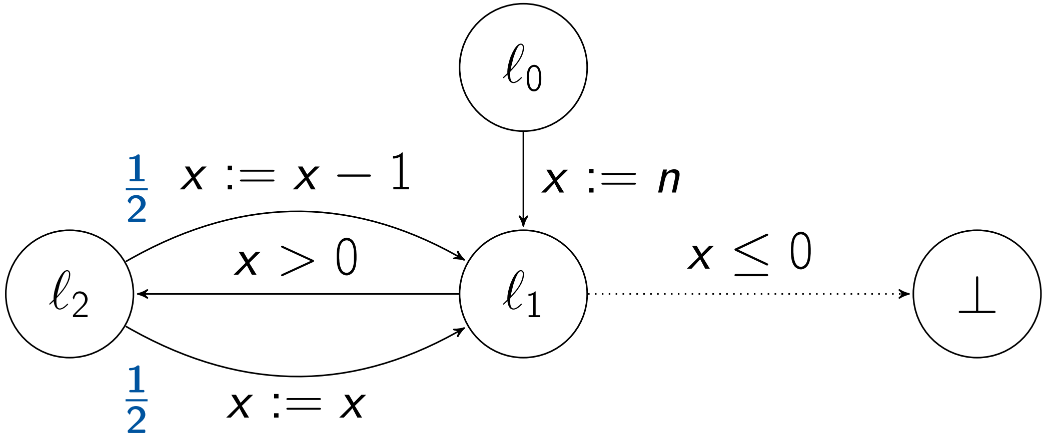
  if $\color{red}{prob(0.5)}$ **then**

  $x := x - 1$

  else

  $\color{red}{x := x}$

$\bot$



Semantics of Probabilistic Programs

Probabilistic Control Flow Graph

Example

ℓ_0

$x := n$

ℓ_1

while $x > 0$ **do**

  ℓ_2

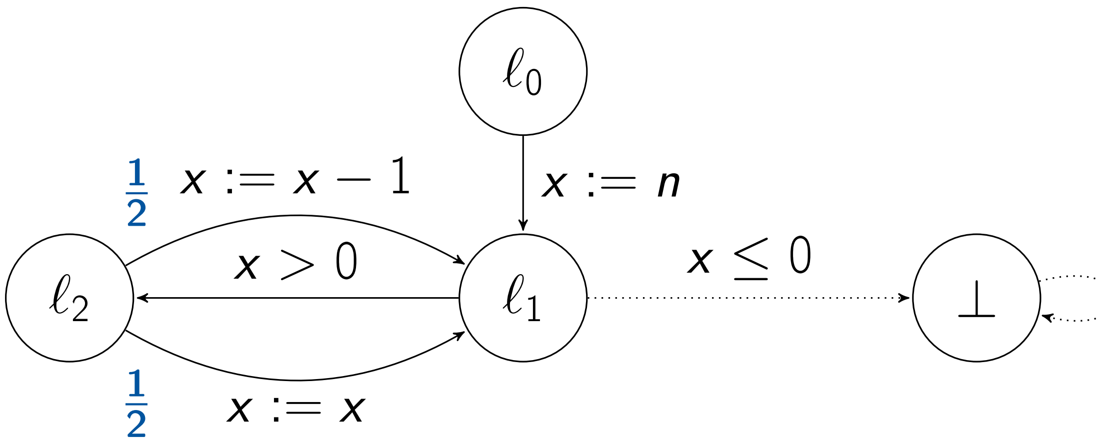
  if $prob(0.5)$ **then**

    $x := x - 1$

  else

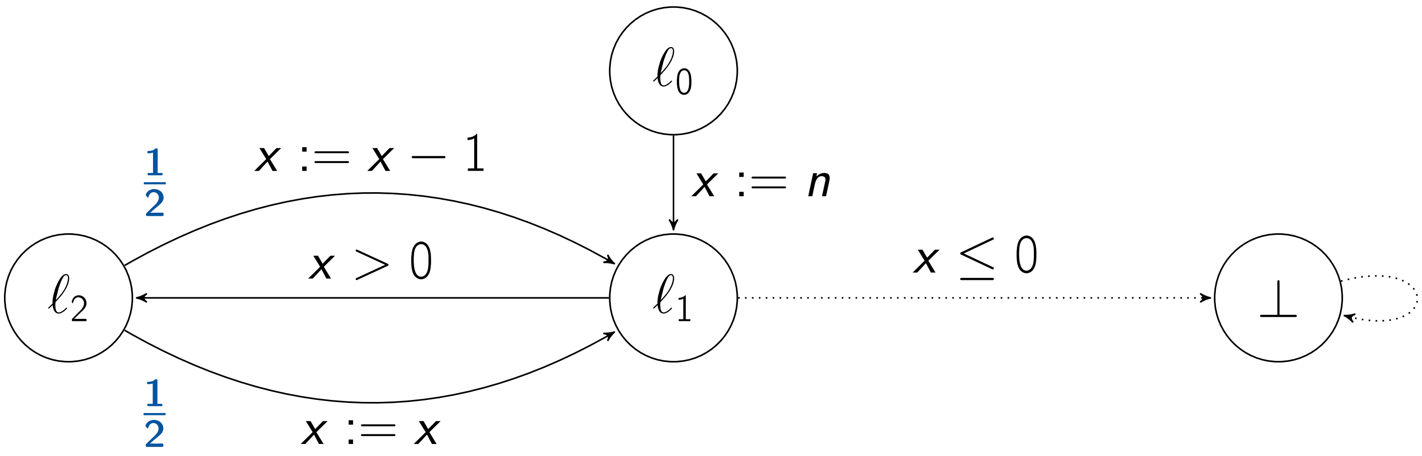
    $x := x$

$\bot$

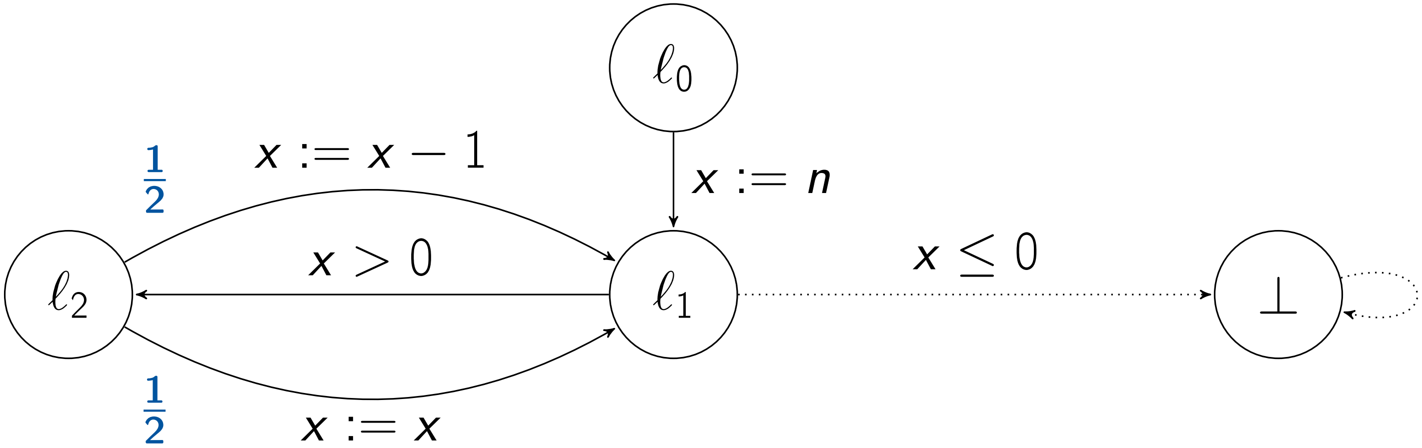


Markov Decision Processes

Markov Decision Processes



Markov Decision Processes

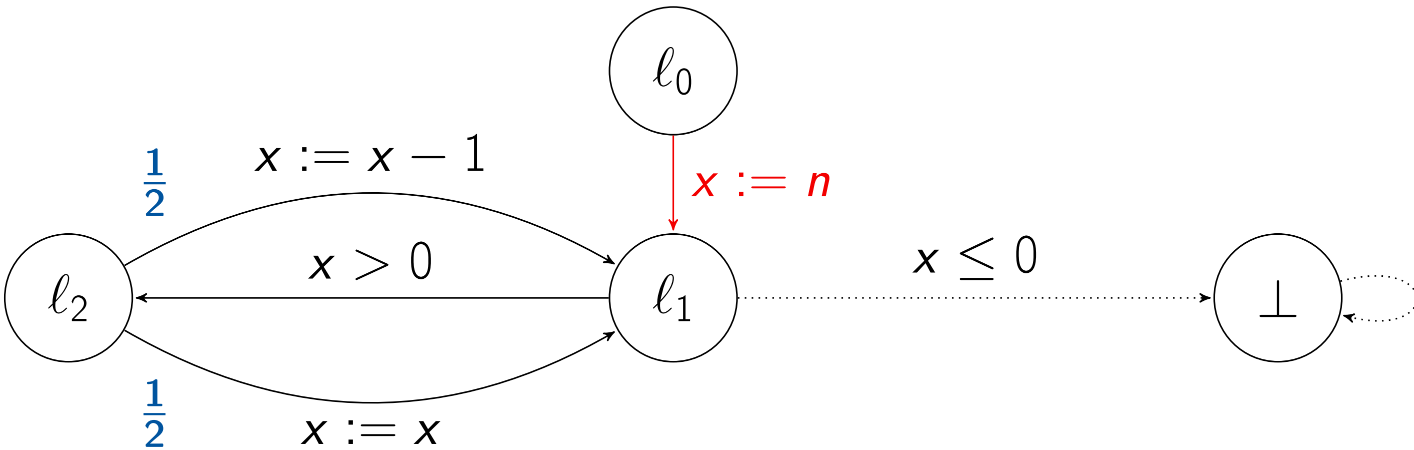


Example

$$(\ell_0, (0, 1))$$

Semantics of Probabilistic Programs

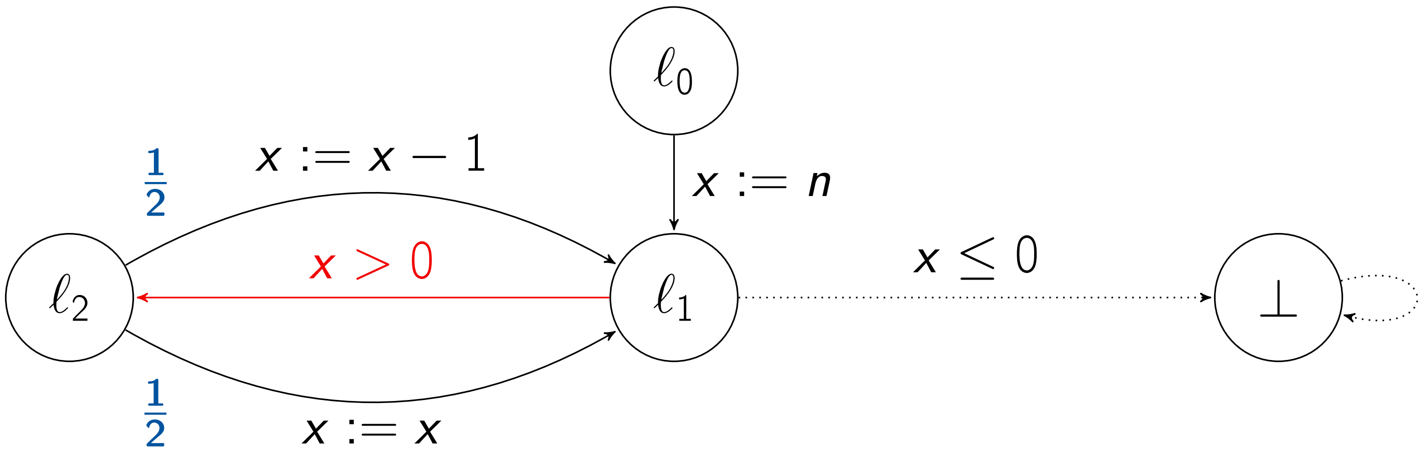
Markov Decision Processes



Example

$$(\ell_0, (0, 1)) \to (\ell_1, (1, 1))$$

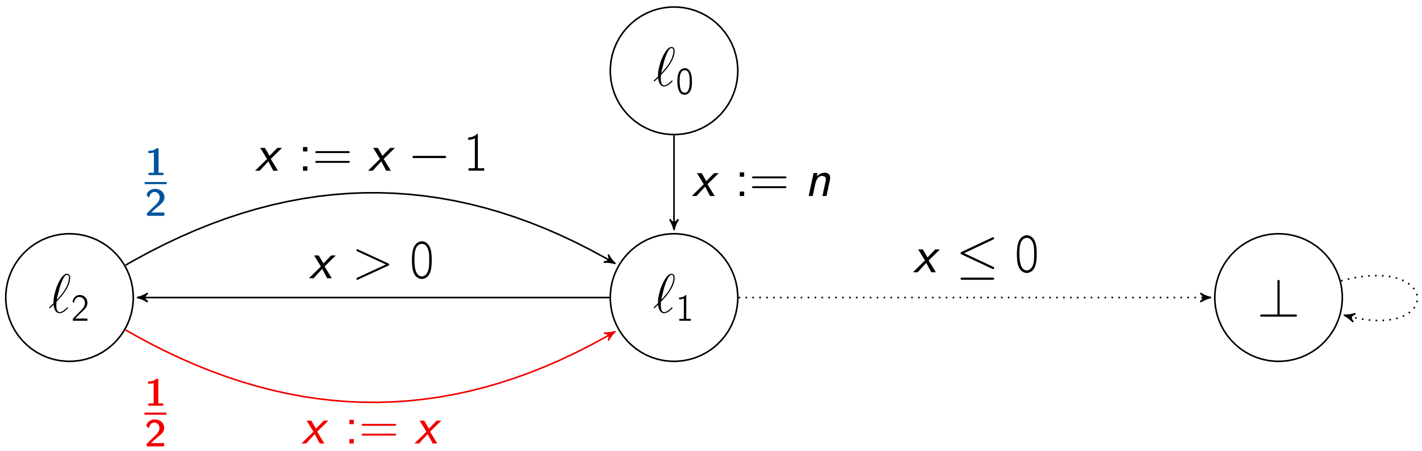
Markov Decision Processes



Example

$$(\ell_0, (0, 1)) \rightarrow (\ell_1, (1, 1)) \rightarrow (\ell_2, (1, 1))$$

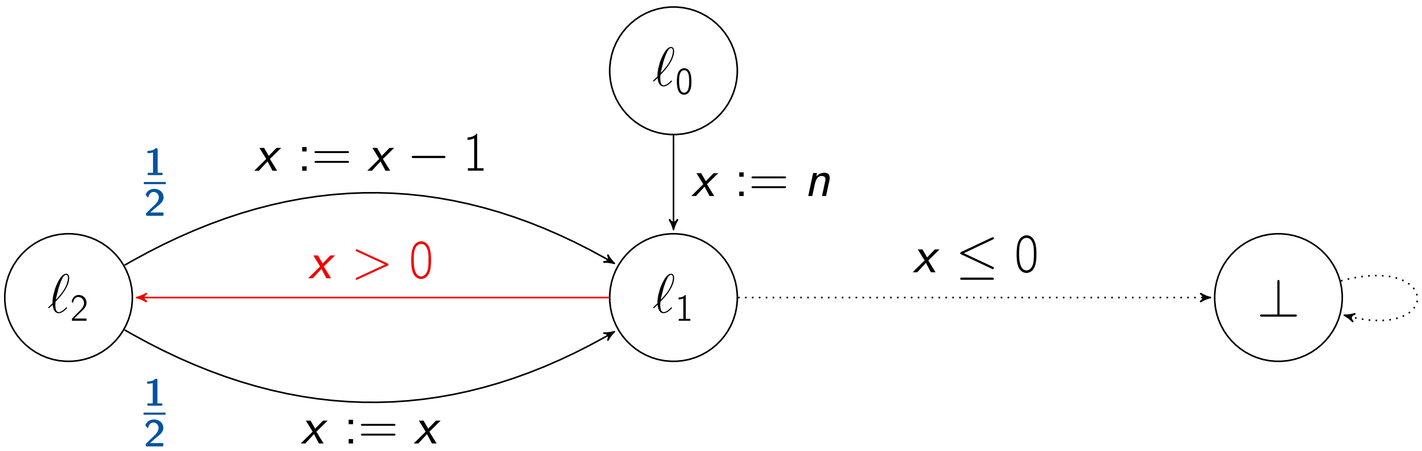
Markov Decision Processes



Example

$$(\ell_0, (0, 1)) \rightarrow (\ell_1, (1, 1)) \rightarrow (\ell_2, (1, 1)) \xrightarrow{\frac{1}{2}} (\ell_1, (1, 1))$$

Markov Decision Processes



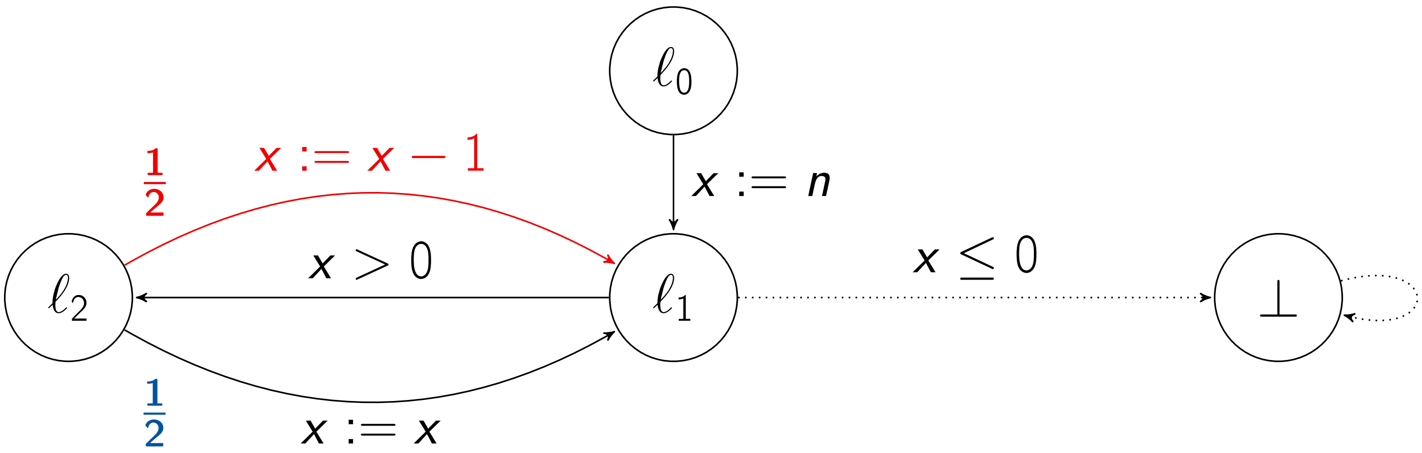
Example

$$(\ell_0, (0,1)) \rightarrow (\ell_1, (1,1)) \rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (1,1))$$

$$\rightarrow (\ell_2, (1,1))$$

Semantics of Probabilistic Programs

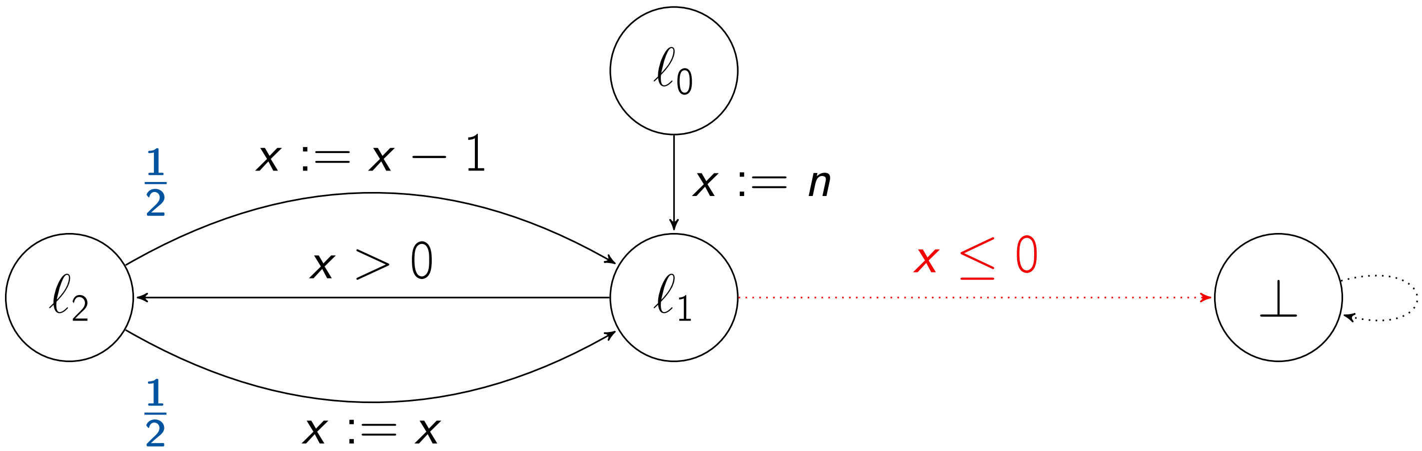
Markov Decision Processes



Example

$$
\begin{aligned}
(\ell_0, (0,1)) \rightarrow (\ell_1, (1,1)) \rightarrow & (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (1,1)) \\
\rightarrow & (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (0,1))
\end{aligned}
$$

Markov Decision Processes



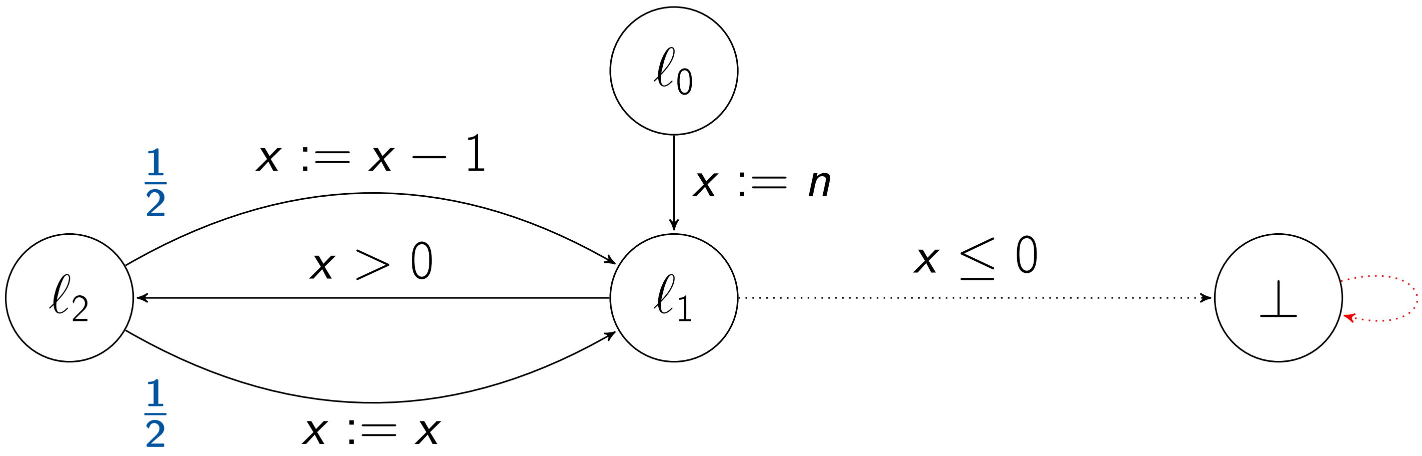
Example

$$(\ell_0, (0,1)) \rightarrow (\ell_1, (1,1)) \rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (1,1))$$

$$\rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (0,1)) \rightarrow (\bot, (0,1))$$

Semantics of Probabilistic Programs

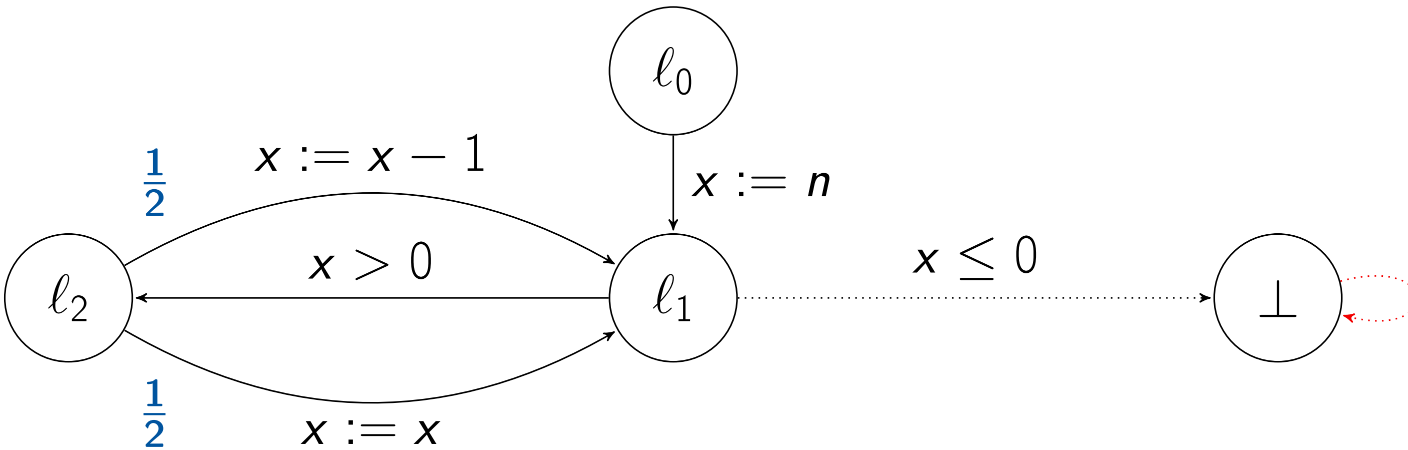
Markov Decision Processes



Example

$$
\begin{aligned}
(\ell_0, (0,1)) \rightarrow (\ell_1, (1,1)) &\rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (1,1)) \\
&\rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (0,1)) \rightarrow (\perp, (0,1)) \\
&\rightarrow (\perp, (0,1))
\end{aligned}
$$

Markov Decision Processes



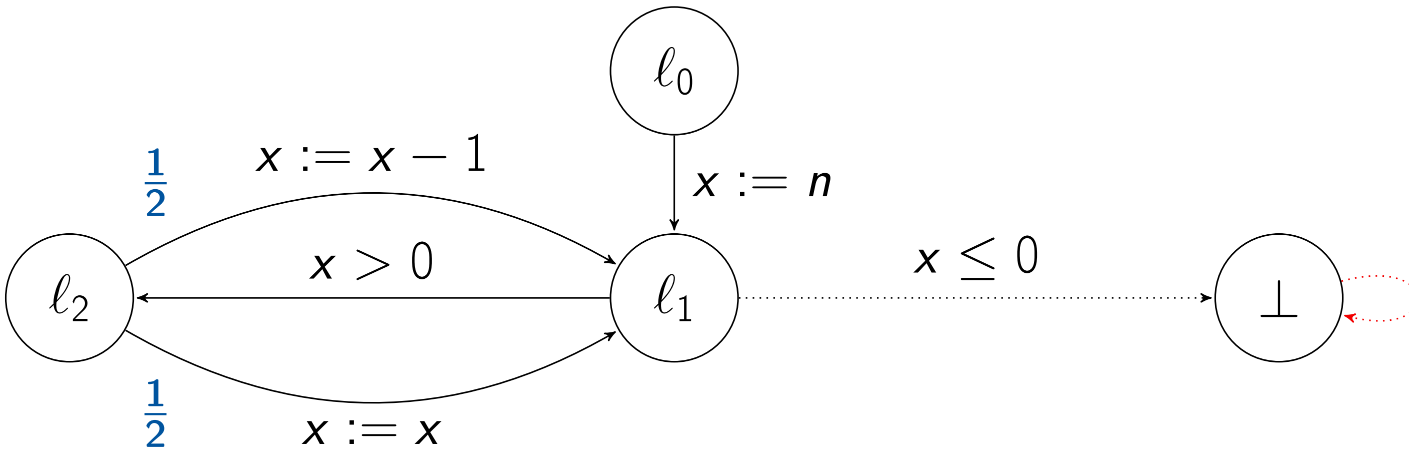
Example

$$
\begin{aligned}
(\ell_0, (0,1)) \rightarrow (\ell_1, (1,1)) &\rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (1,1)) \\
&\rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (0,1)) \rightarrow (\bot, (0,1)) \\
&\rightarrow (\bot, (0,1)) \rightarrow (\bot, (0,1))
\end{aligned}
$$

UnRAVeL RWTH AACHEN UNIVERSITY

Semantics of Probabilistic Programs

Markov Decision Processes

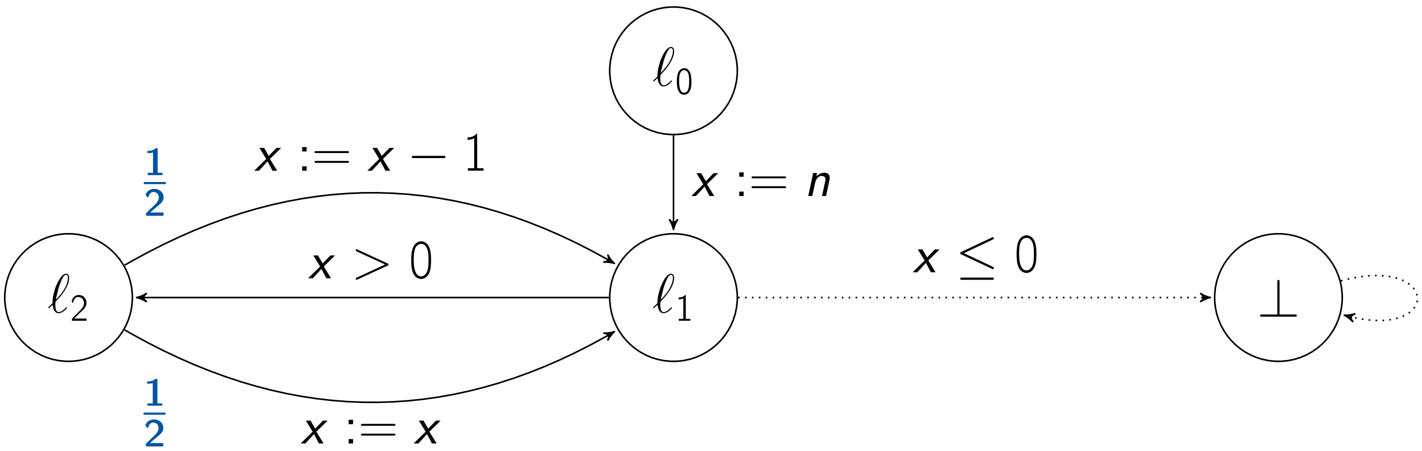


Example

$$(\ell_0, (0,1)) \to (\ell_1, (1,1)) \to (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (1,1))$$
$$\to (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (0,1)) \to (\bot, (0,1))$$
$$\to (\bot, (0,1)) \to (\bot, (0,1)) \to \ldots$$

Semantics of Probabilistic Programs

Markov Decision Processes

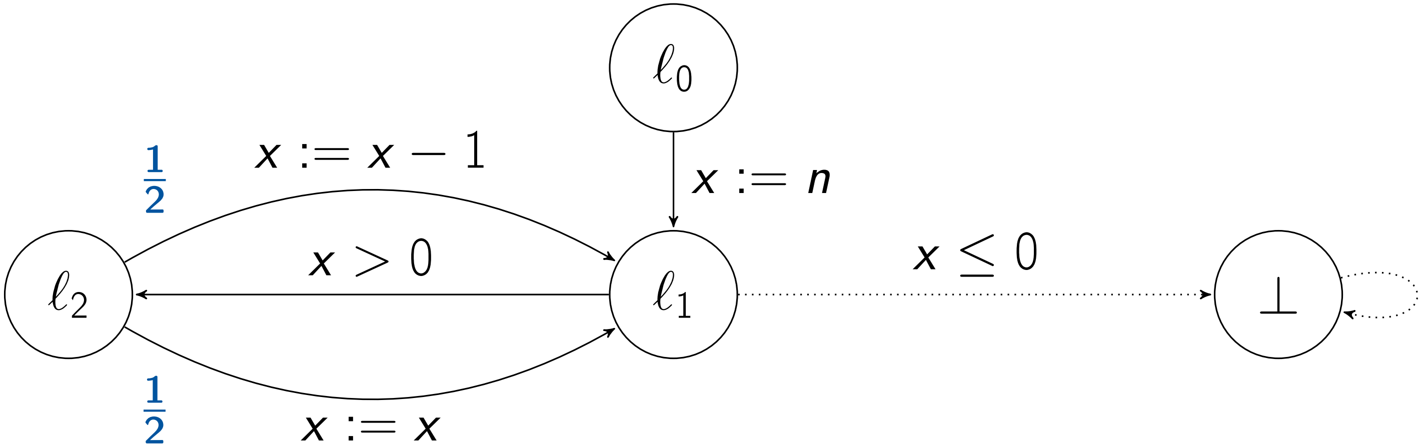


Example

$$\mathbb{P}(\{(\ell_0,(0,1)) \to (\ell_1,(1,1)) \to (\ell_2,(1,1)) \xrightarrow{\frac{1}{2}} (\ell_1,(1,1)) \to (\ell_2,(1,1)) \xrightarrow{\frac{1}{2}} (\ell_1,(0,1)) \to (\bot,(0,1)) \to (\bot,(0,1)) \to (\bot,(0,1)) \to \ldots\})$$

Semantics of Probabilistic Programs

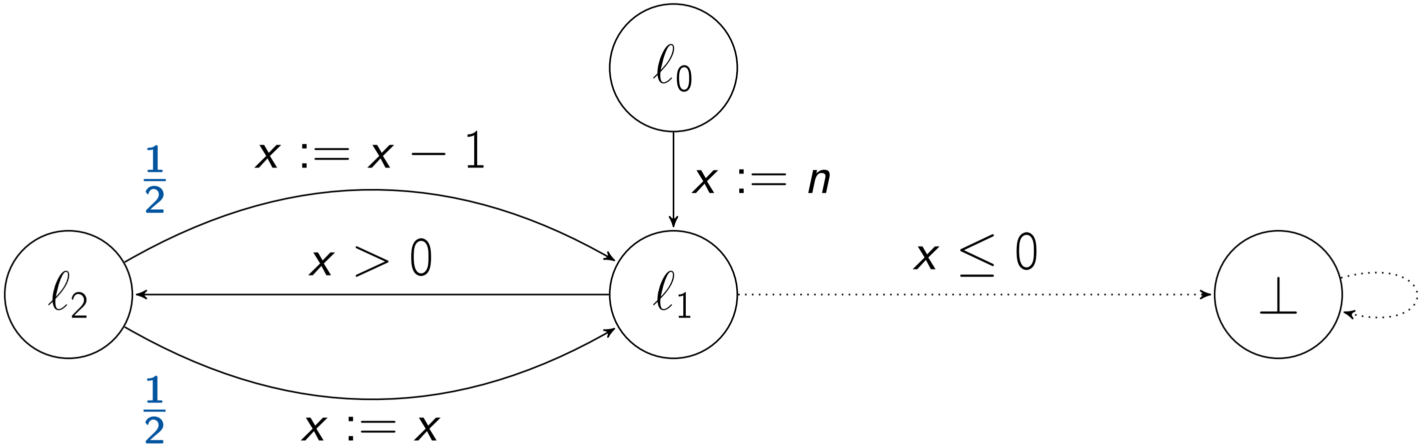
Markov Decision Processes



Example

$$\mathbb{P}(\{(\ell_0,(0,1)) \to (\ell_1,(1,1)) \to (\ell_2,(1,1)) \xrightarrow{\frac{1}{2}} (\ell_1,(1,1))$$
$$\to (\ell_2,(1,1)) \xrightarrow{\frac{1}{2}} (\ell_1,(0,1)) \to (\perp,(0,1))$$
$$\to (\perp,(0,1)) \to (\perp,(0,1)) \to \ldots\}) = \frac{1}{2}\cdot\frac{1}{2}$$

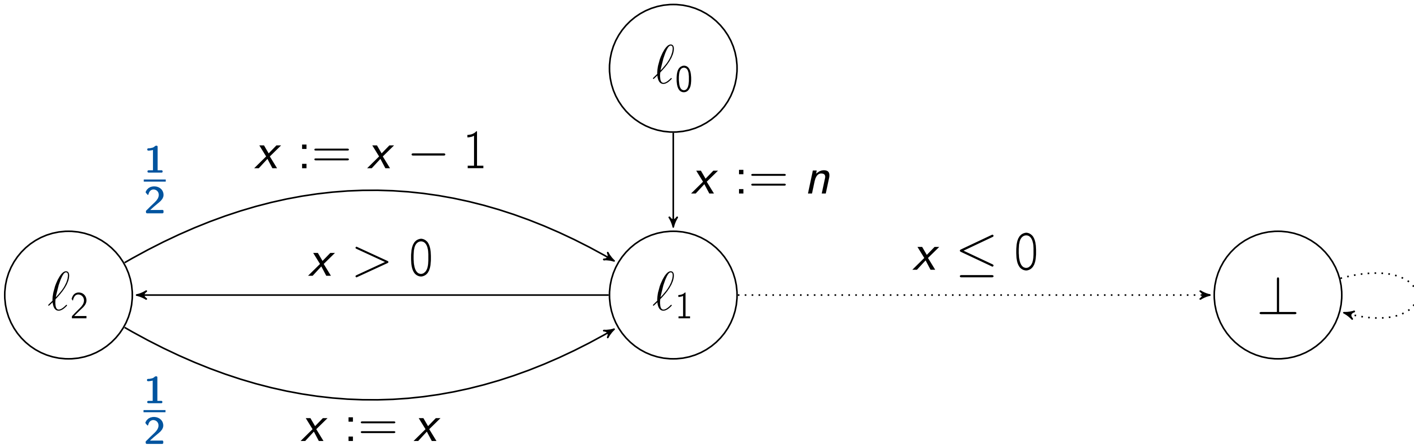
Markov Decision Processes



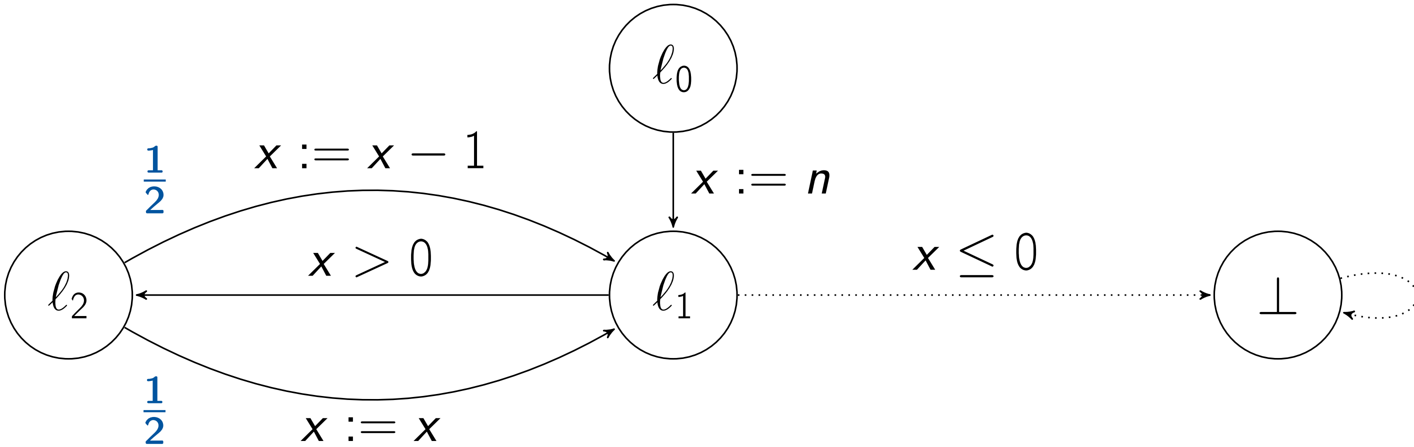
Example

$$\mathbb{P}(\{(\ell_0, (0,1)) \rightarrow (\ell_1, (1,1)) \rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (1,1)) \rightarrow (\ell_2, (1,1)) \xrightarrow{\frac{1}{2}} (\ell_1, (0,1)) \rightarrow (\bot, (0,1)) \rightarrow (\bot, (0,1)) \rightarrow (\bot, (0,1)) \rightarrow \ldots\}) = \frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$$

Markov Decision Processes



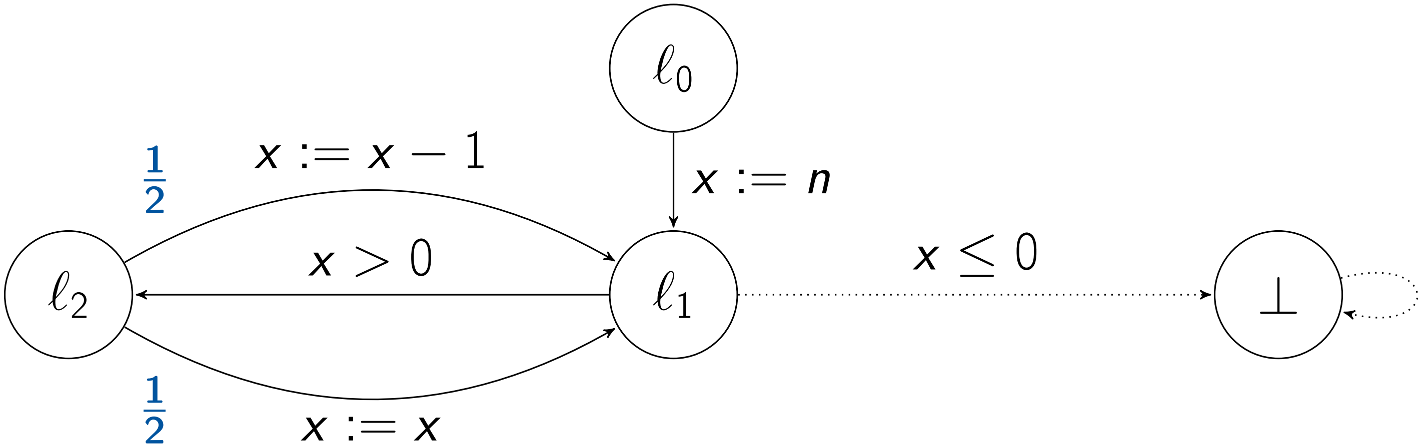
Markov Decision Processes



Definition (Underlying Process)

Semantics of Probabilistic Programs

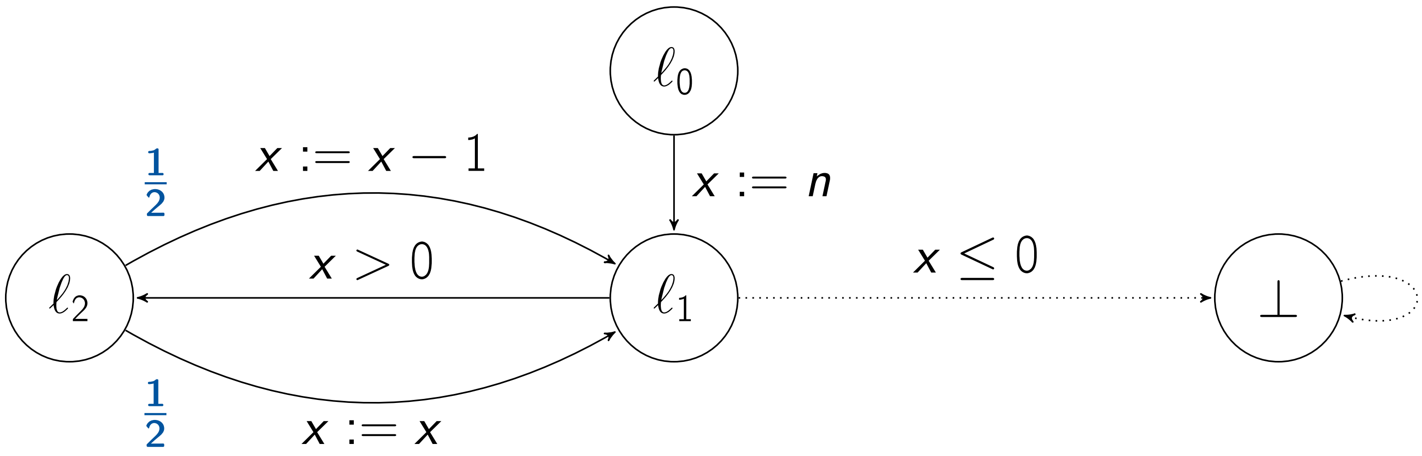
Markov Decision Processes



Definition (Underlying Process)

- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.

Markov Decision Processes

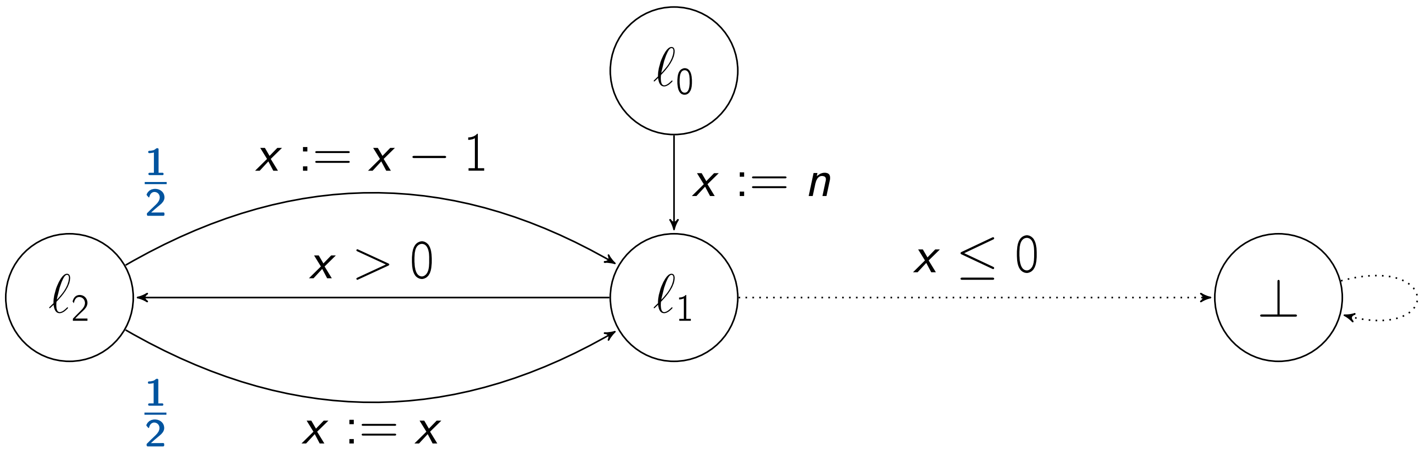


Definition (Underlying Process)

- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
- $\Omega := \{\omega \mid \omega \text{ is a run}\} \subseteq \left(\mathbb{N} \to \mathcal{L} \times \mathbb{Z}^2\right)$,

Semantics of Probabilistic Programs

Markov Decision Processes

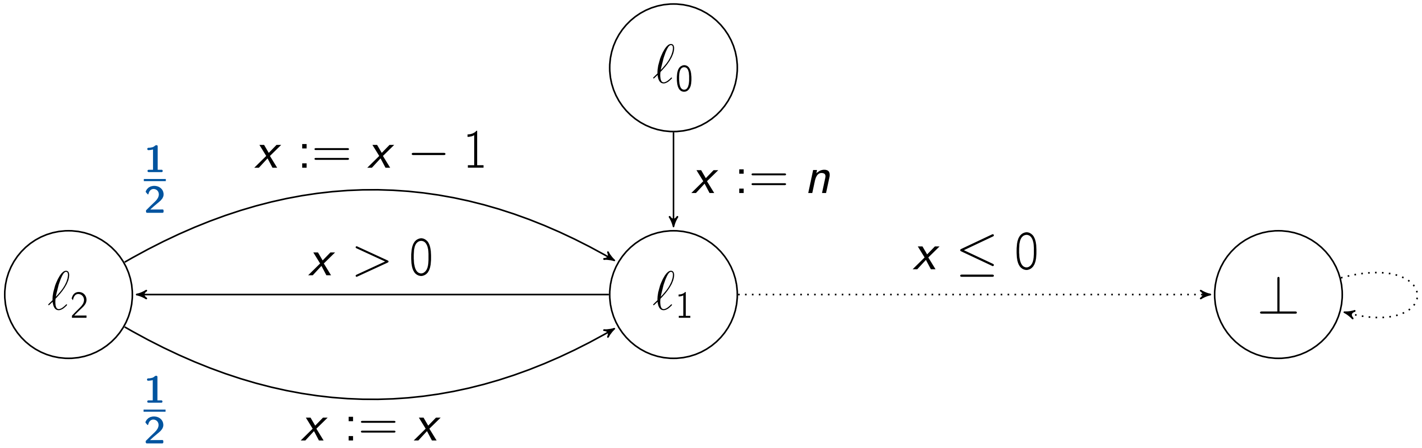


Definition (Underlying Process)

- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
- $\Omega := \{\omega \mid \omega \text{ is a run}\} \subseteq \left(\mathbb{N} \to \mathcal{L} \times \mathbb{Z}^2\right)$,
- $\mathcal{F} := \sigma(\{\textit{cylinder sets}\})$.

Semantics of Probabilistic Programs

Markov Decision Processes

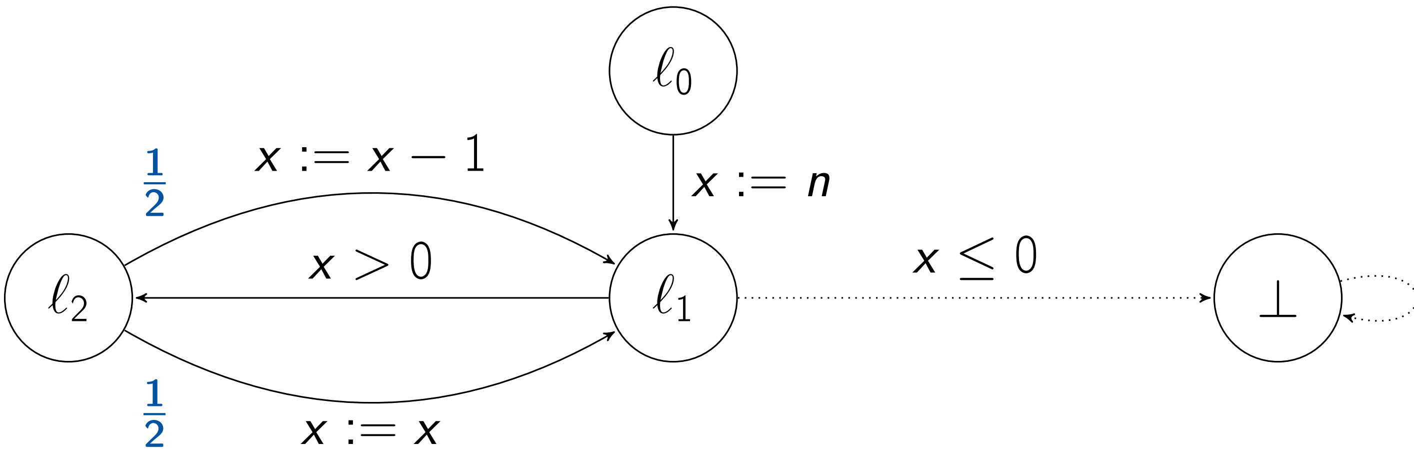


Definition (Underlying Process)

- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
- $\Omega := \{\omega \mid \omega \text{ is a run}\} \subseteq \left(\mathbb{N} \to \mathcal{L} \times \mathbb{Z}^2\right)$,
- $\mathcal{F} := \sigma(\{\textit{cylinder sets}\})$.
- $\mathbb{P} : \mathcal{F} \to [0, 1]$ the induced probability measure.

Semantics of Probabilistic Programs

Markov Decision Processes



Definition (Underlying Process)

- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
- $\Omega := \{\omega \mid \omega \text{ is a run}\} \subseteq \left(\mathbb{N} \to \mathcal{L} \times \mathbb{Z}^2\right)$,
- $\mathcal{F} := \sigma(\{\mathit{cylinder\ sets}\})$.
- $\mathbb{P} : \mathcal{F} \to [0, 1]$ the induced probability measure.
- $\mathit{Term} : \Omega \to \mathbb{N}, \omega \mapsto \min\{k \mid \omega_n = (\bot, \dots) \text{ for all } n \geq k\}$.

Markov Decision Process

Definition (underlying process)

- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
- $\Omega := \{\omega \mid \omega \text{ is a run}\} \subseteq \left(\mathbb{N} \to \mathcal{L} \times \mathbb{Z}^2\right)$
- $\mathcal{F} := \sigma\left(\mathsf{C}_i \mid i \in \mathbb{N}\right)$.
- $\mathbb{P} : \mathcal{F} \to [0, 1]$ the induced probability measure.
- $\mathit{Term} : \Omega \to \mathbb{N}, \omega \mapsto \min\left\{k \mid \omega_n = (\perp, \ldots) \text{ for all } n \geq k\right\}$.

Markov Decision Process

Definition (underlying process)

- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
- $\Omega := \{\omega \mid \omega \text{ is a run}\} \subseteq \left(\mathbb{N} \to \mathcal{L} \times \mathbb{Z}^2\right)$
- $\mathcal{F} := \sigma\left(\mathbf{C}_i \mid i \in \mathbb{N}\right)$.
- $\mathbb{P} : \mathcal{F} \to [0, 1]$ the induced probability measure.
- $\mathit{Term} : \Omega \to \mathbb{N}, \omega \mapsto \min\left\{k \mid \omega_n = (\bot, \dots) \text{ for all } n \geq k\right\}$.

Objective

Approximating $\mathbb{E}(\mathit{Term})$ by a stochastic process $(X_i)_{i \in \mathbb{N}}$ with „nice“ properties.

Markov Decision Process

Definition (underlying process)

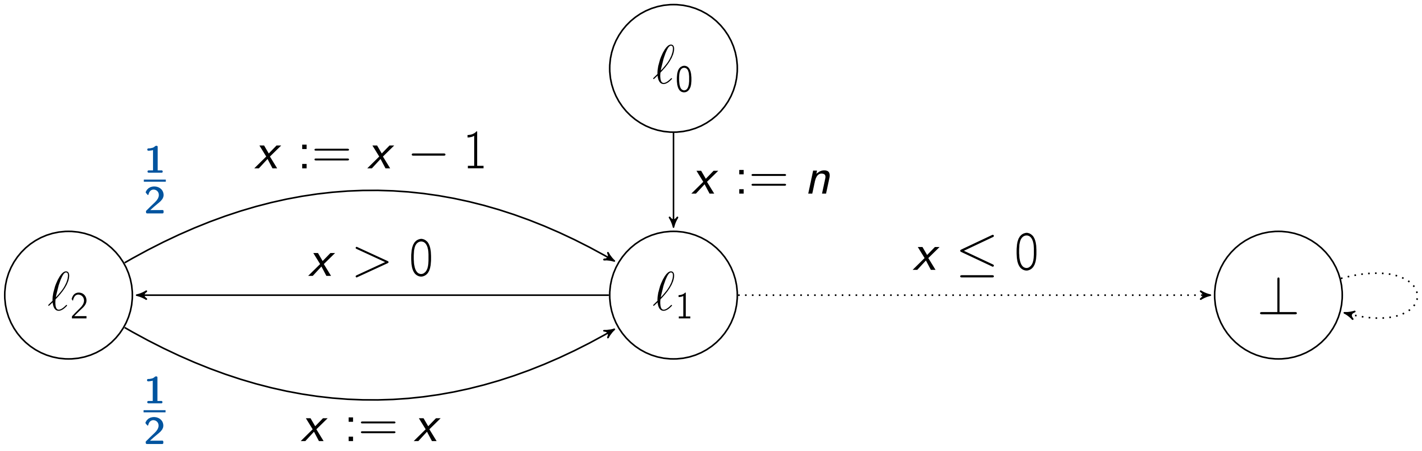
- $(\Omega, \mathcal{F}, \mathbb{P})$ is a probability space.
- $\Omega := \{\omega \mid \omega \text{ is a run}\} \subseteq \left(\mathbb{N} \to \mathcal{L} \times \mathbb{Z}^2\right)$
- $\mathcal{F} := \sigma\left(\mathbf{C}_i \mid i \in \mathbb{N}\right)$.
- $\mathbb{P} : \mathcal{F} \to [0, 1]$ the induced probability measure.
- $\mathit{Term} : \Omega \to \mathbb{N}, \omega \mapsto \min\{k \mid \omega_n = (\bot, \dots) \text{ for all } n \geq k\}$.

Objective

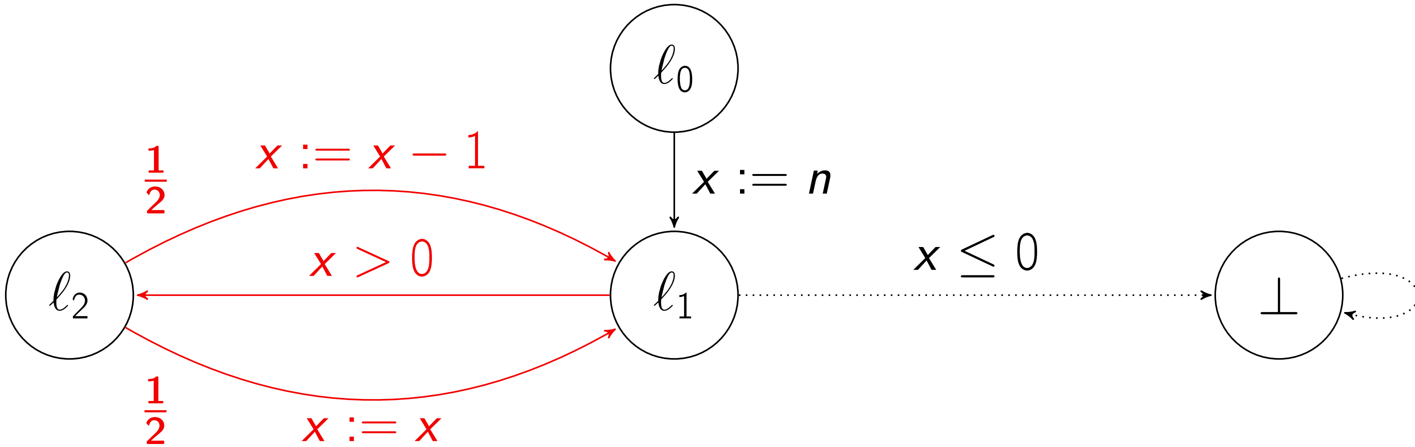
Approximating $\mathbb{E}(\mathit{Term})$ by a stochastic process $(X_i)_{i \in \mathbb{N}}$ with „nice" properties.

„nice" $\leftrightarrow$ decreasing in expectation.

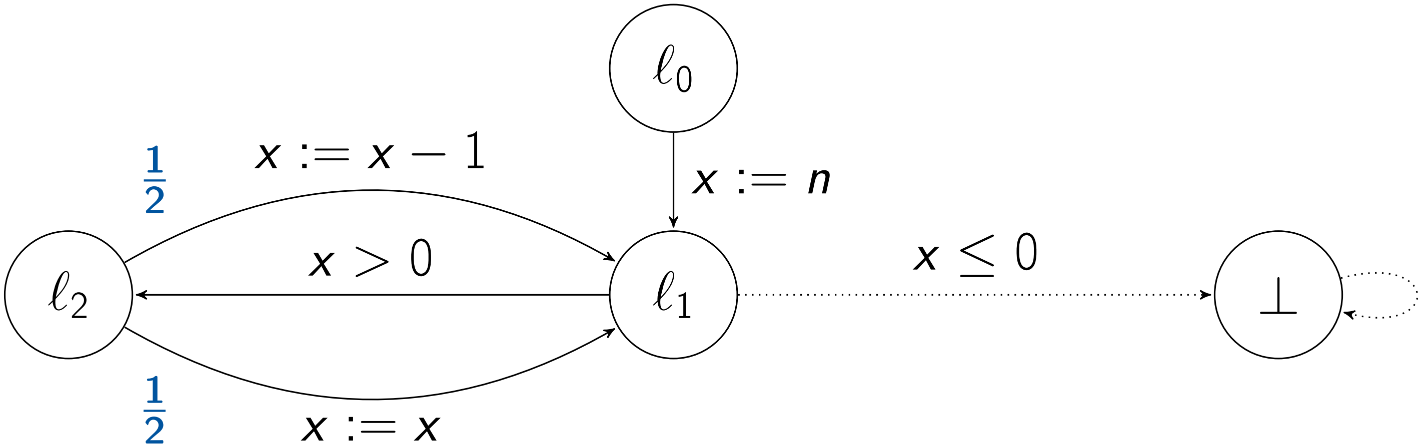
Probabilistic Ranking Functions



Probabilistic Ranking Functions

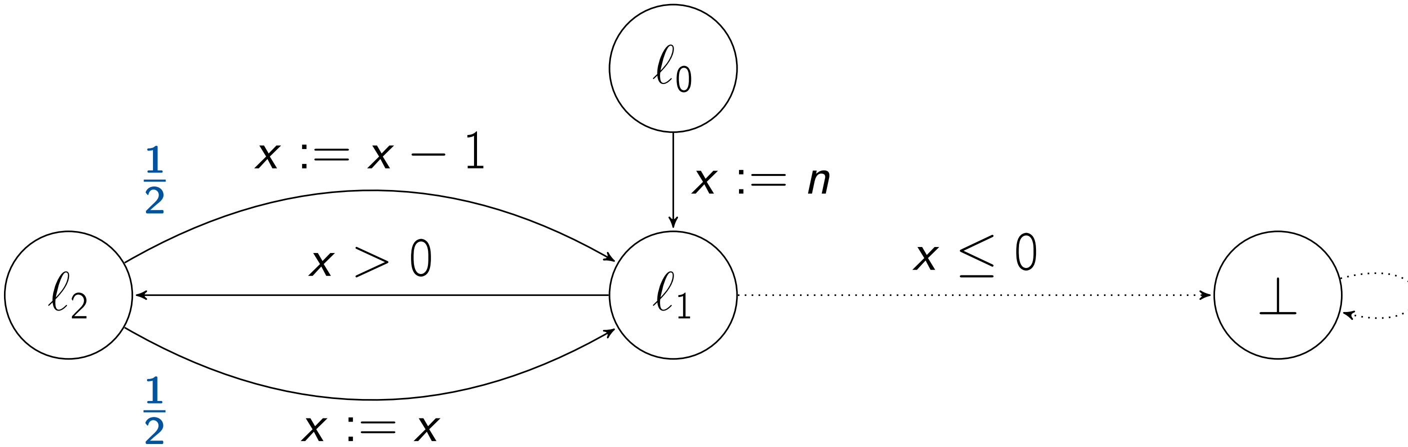


Probabilistic Ranking Functions



Example

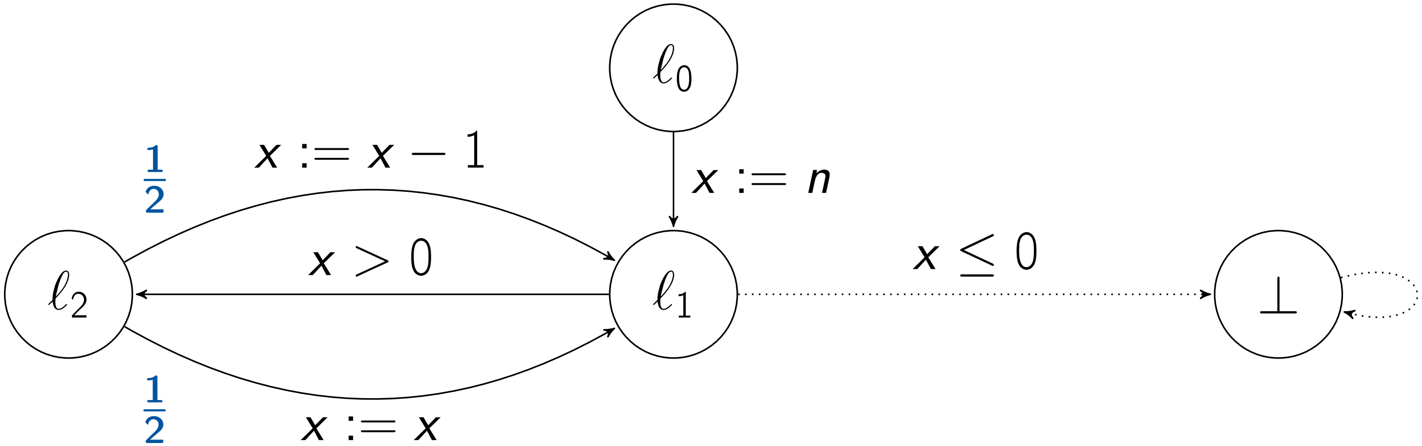
Probabilistic Ranking Functions



Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

Probabilistic Ranking Functions

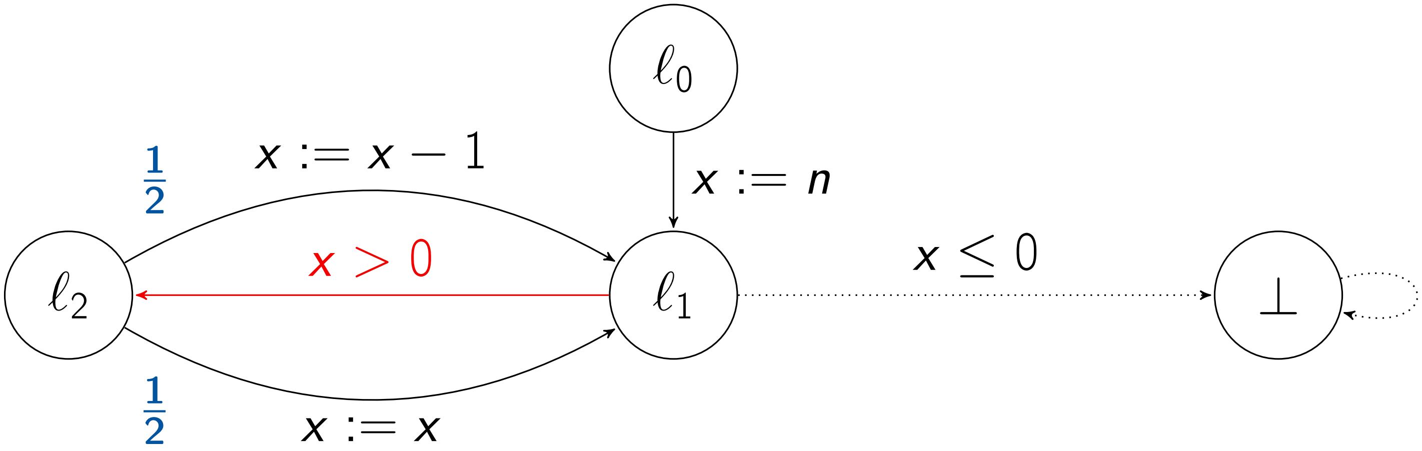


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x$$

Probabilistic Ranking Functions

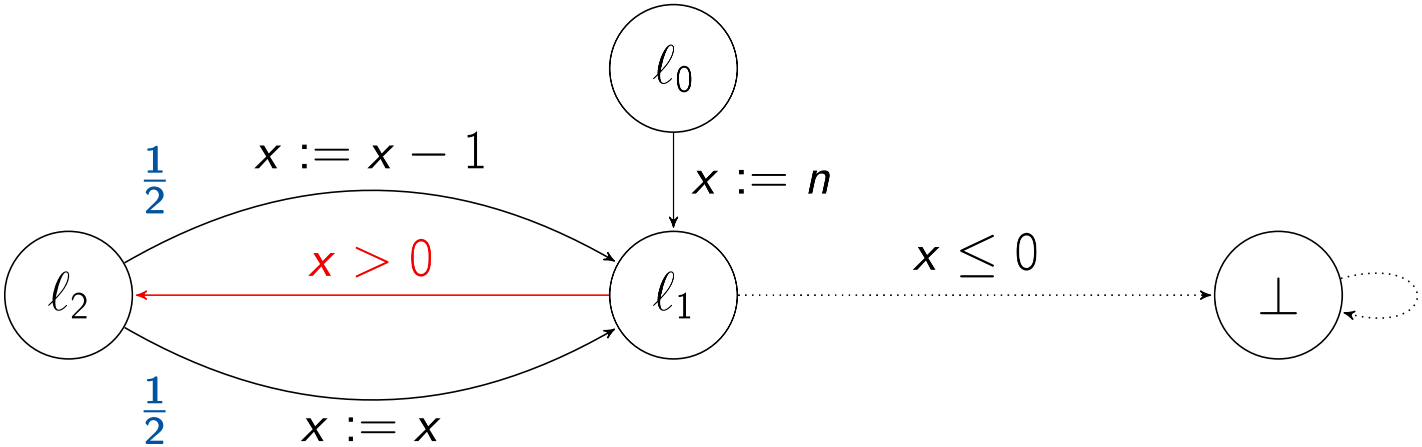


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x$$

Probabilistic Ranking Functions

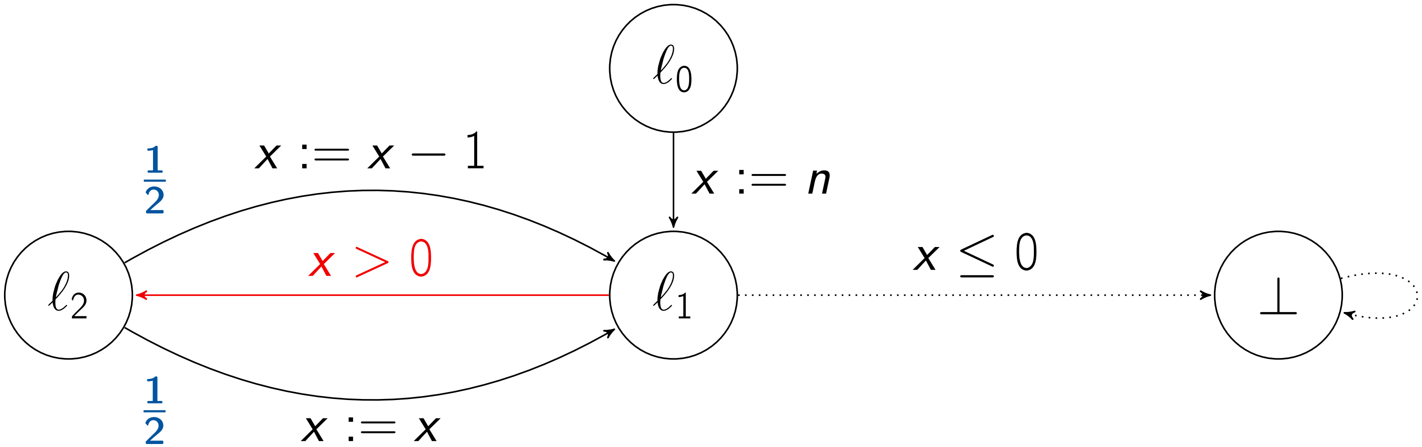


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x \geq 0$$

Probabilistic Ranking Functions

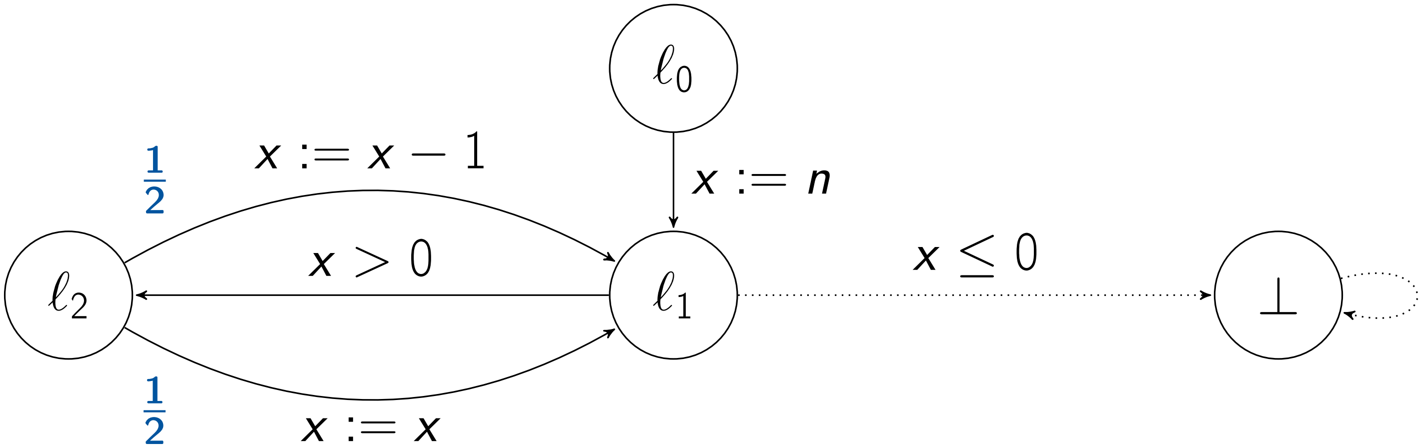


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x \geq 0 \Rightarrow \mathbb{E}(R)(x, n) \geq 0$$

Probabilistic Ranking Functions

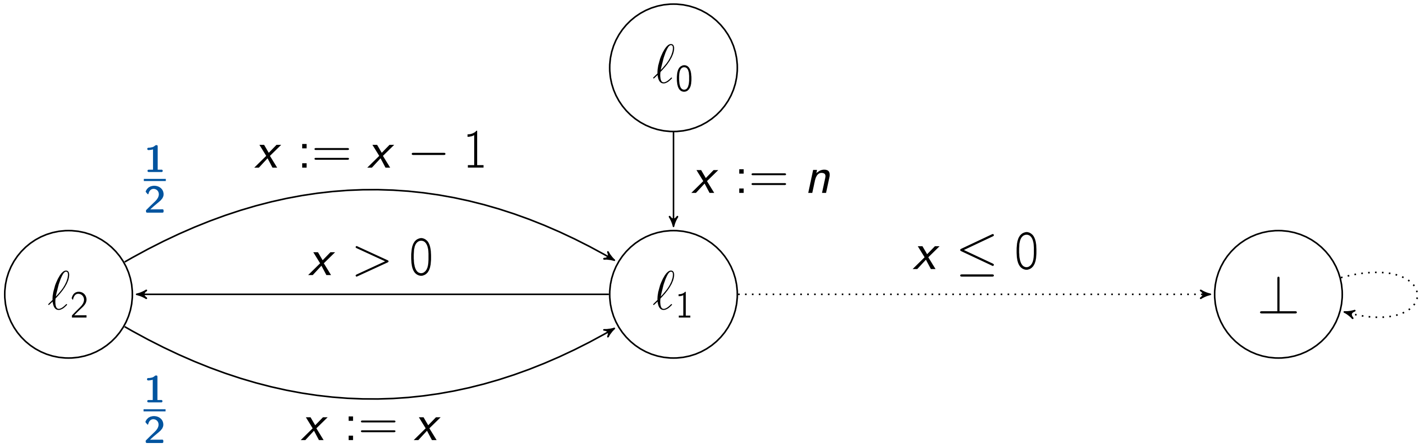


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x \geq 0 \Rightarrow \mathbb{E}(R)(x, n) \geq 0$$

Probabilistic Ranking Functions

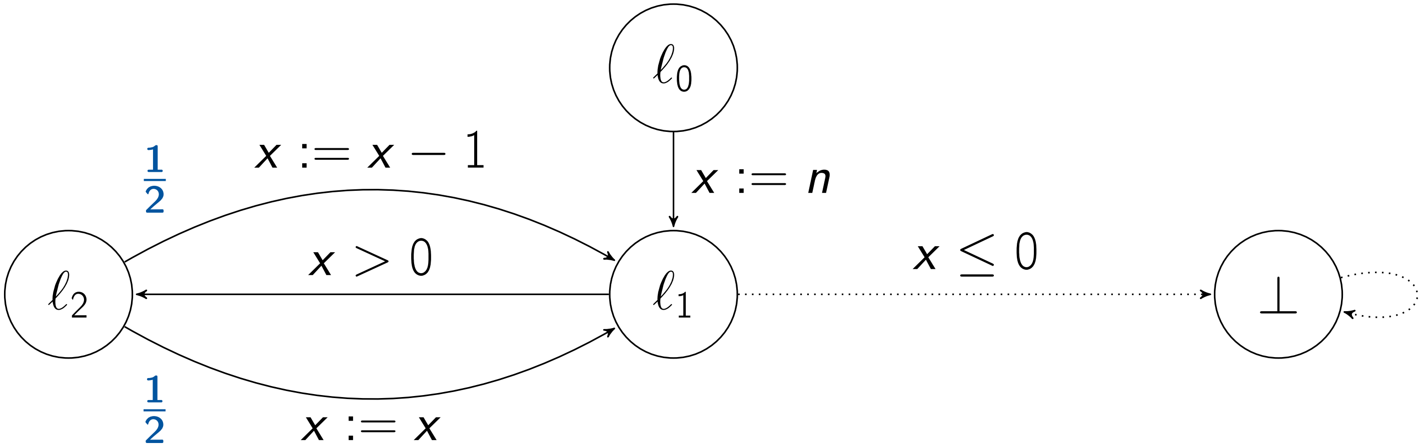


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$R((x, n)) = 2 \cdot x$

Probabilistic Ranking Functions

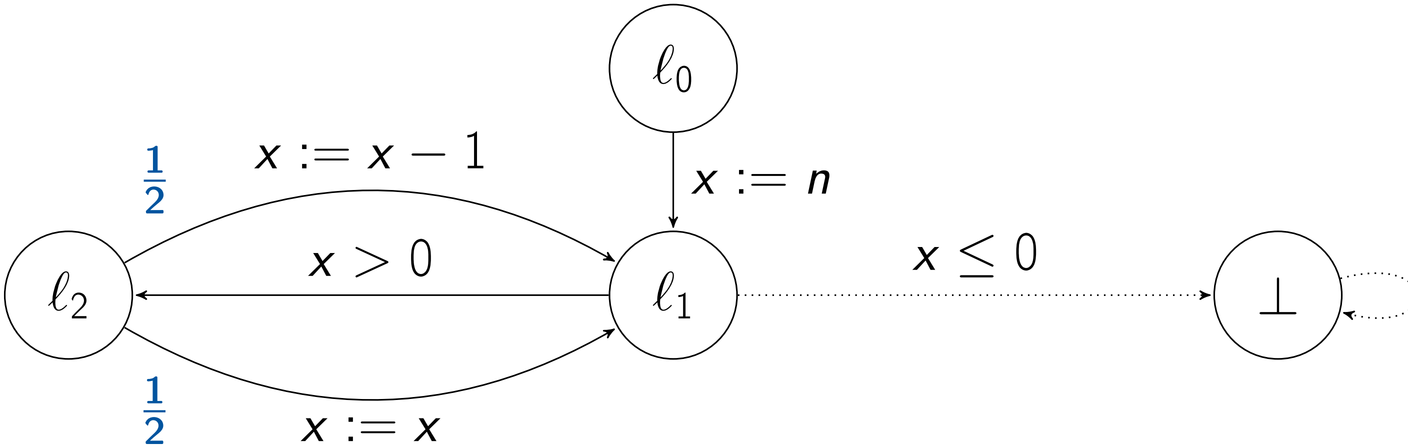


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$R((x, n)) = 2 \cdot x > 2 \cdot x - 1$

Probabilistic Ranking Functions



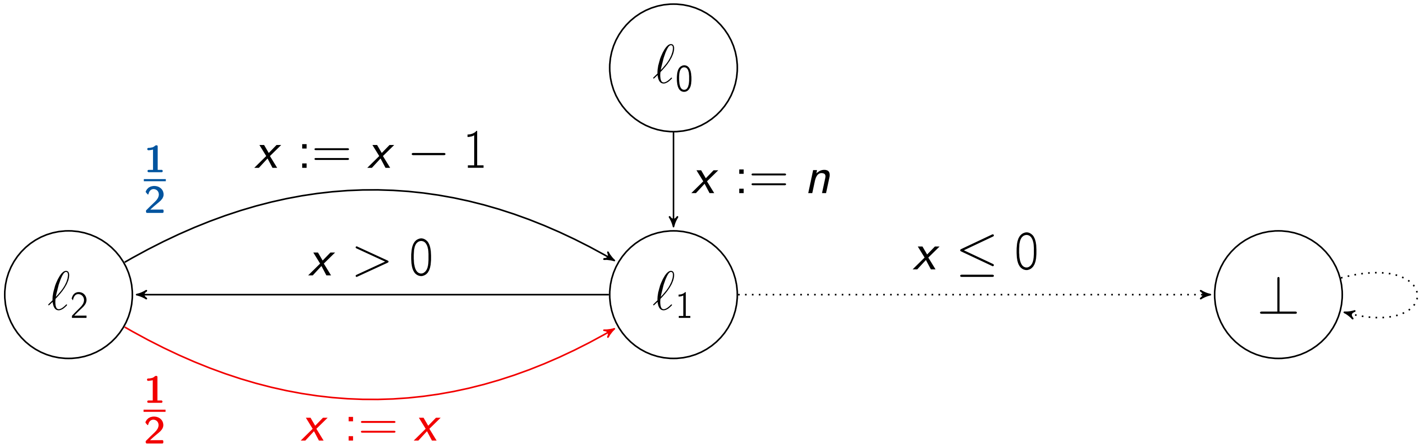
Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x > 2 \cdot x - 1 = x + x - 1$$

Semantics of Probabilistic Programs

Probabilistic Ranking Functions



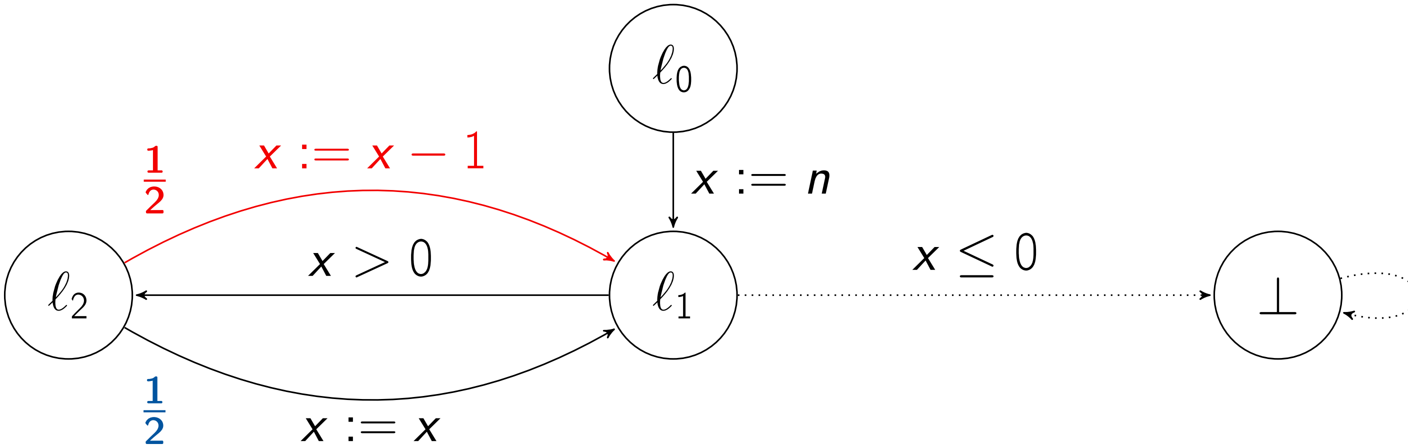
Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x > 2 \cdot x - 1 = x + x - 1 = \frac{1}{2} R(x)$$

Semantics of Probabilistic Programs

Probabilistic Ranking Functions



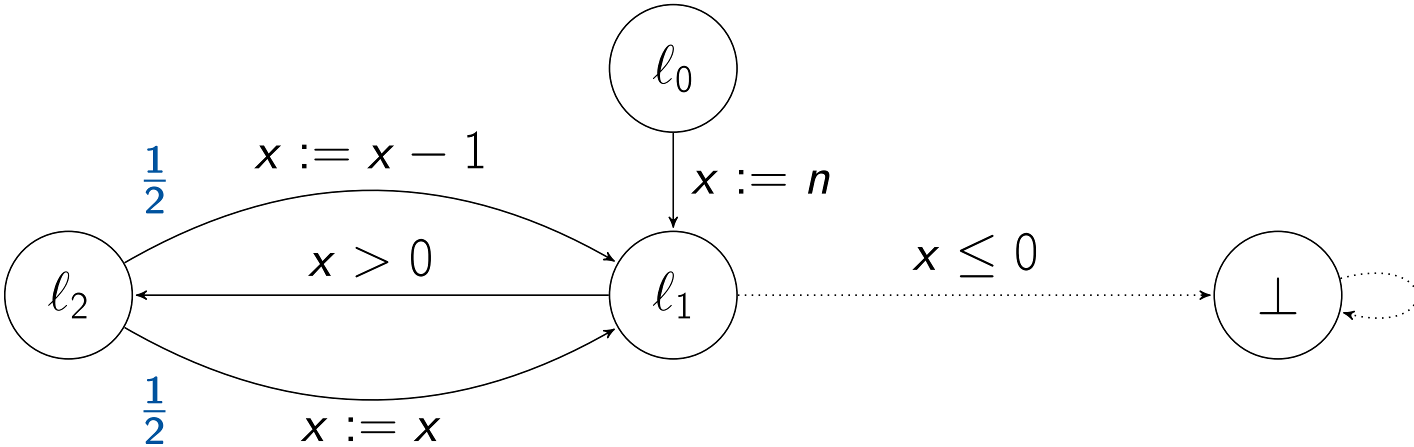
Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x > 2 \cdot x - 1 = x + x - 1 = \frac{1}{2}R(x) + \frac{1}{2} \cdot R(x - 1)$$

Semantics of Probabilistic Programs

Probabilistic Ranking Functions

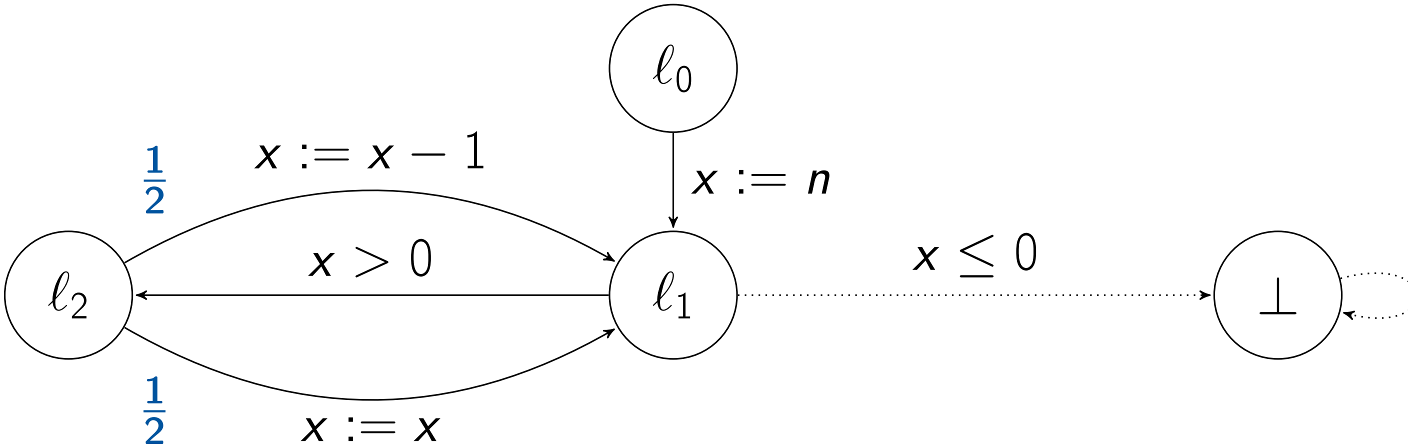


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$R((x, n)) = 2 \cdot x > 2 \cdot x - 1 = x + x - 1 = \frac{1}{2}R(x) + \frac{1}{2} \cdot R(x - 1) = \mathbb{E}(R)(x, n)$$

Probabilistic Ranking Functions

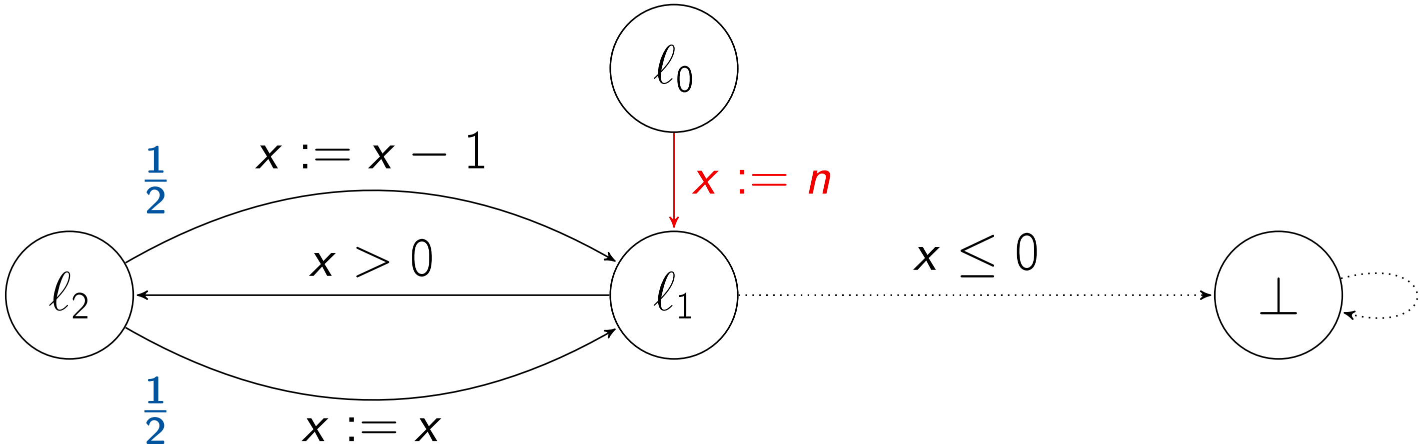


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$\mathbb{E}(\mathit{Term})$$

Probabilistic Ranking Functions

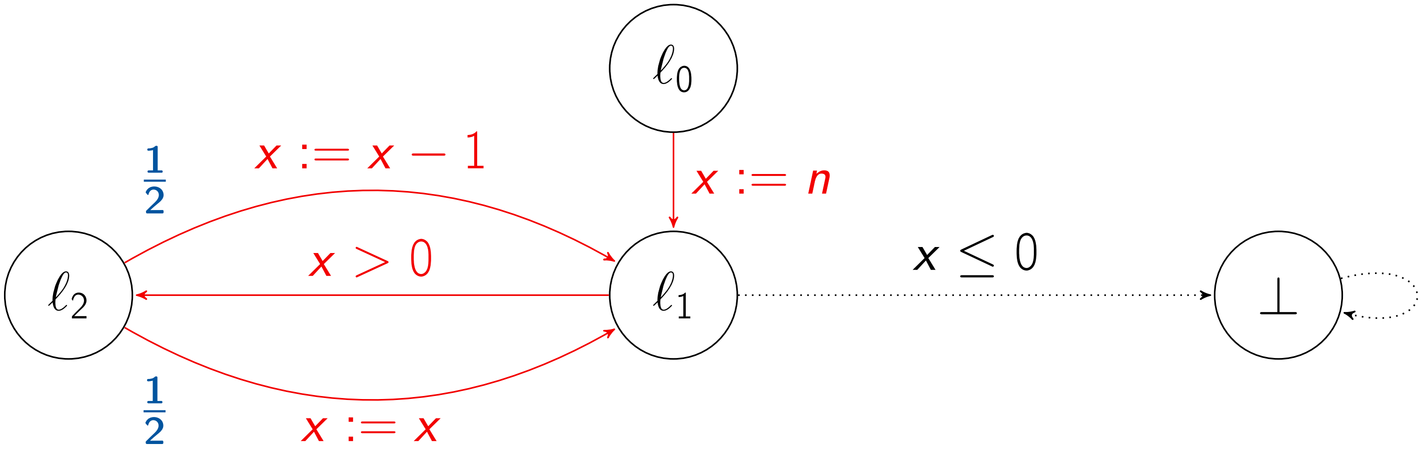


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$\mathbb{E}(\mathit{Term}) \leq 1$$

Probabilistic Ranking Functions

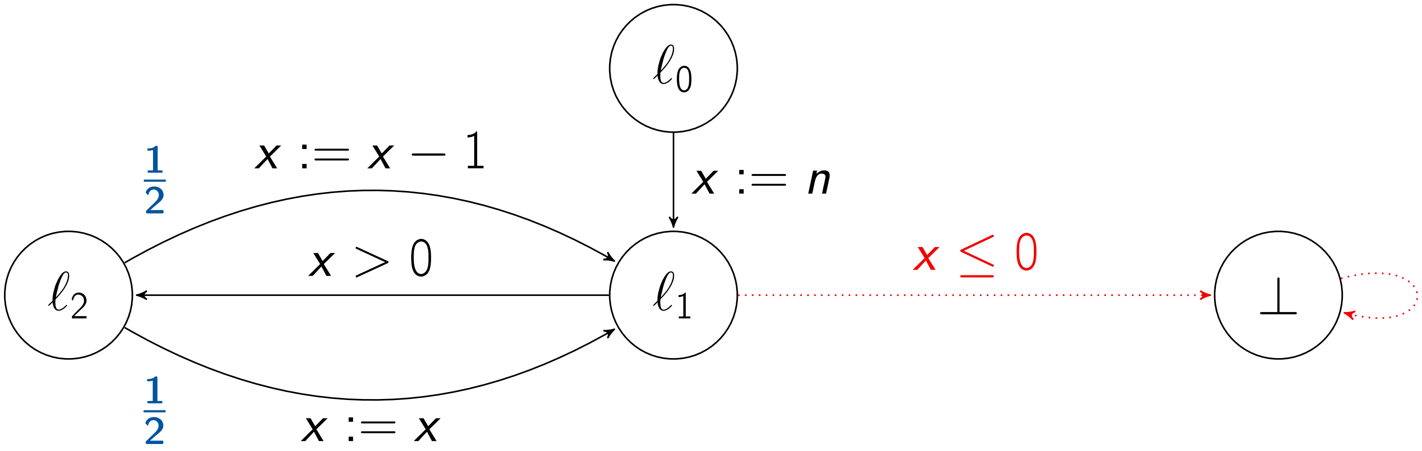


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$\mathbb{E}(\mathit{Term}) \leq 1 + \mathbb{E}(R)(n, n)$$

Probabilistic Ranking Functions

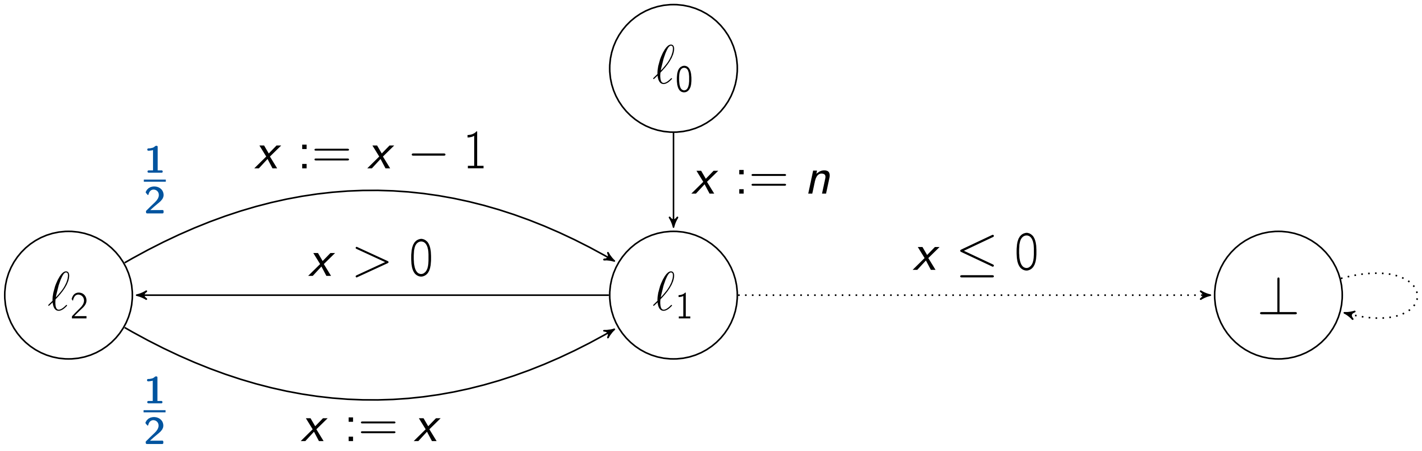


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$\mathbb{E}(\mathit{Term}) \leq 1 + \mathbb{E}(R)(n, n) + 0$$

Probabilistic Ranking Functions

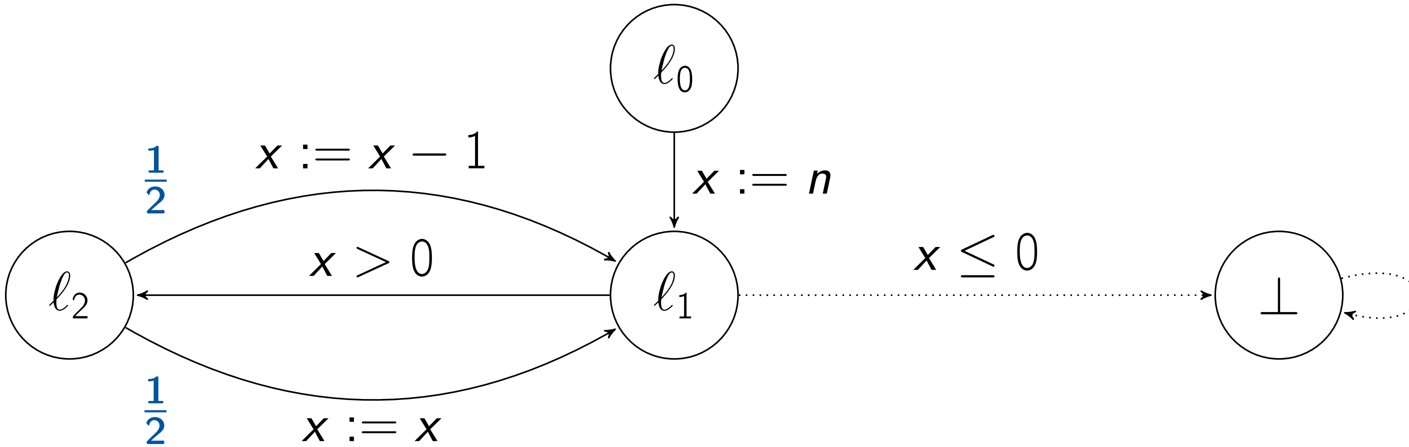


Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

$$\mathbb{E}(\mathit{Term}) \leq 1 + \mathbb{E}(R)(n, n) + 0 = 2n + 1$$

Probabilistic Ranking Functions



Example

$$R : \mathbb{Z}^2 \to \mathbb{Z}, (x, n) \mapsto 2 \cdot x$$

R can be generated automatically.

Outline

Modularity with Expected Size

Example

$x := n$

while $x > 0$ **do**

$\quad x := \frac{1}{2}\langle x-1\rangle + \frac{1}{2}\langle x\rangle$

My Work

Modularity with Expected Size

Example

$x := n$

$y = 0$

while $x > 0$ do

$\quad y := y + x$

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

Modularity with Expected Size

Example

$x := n$

$y = 0$

while $x > 0$ do

$\quad y := y + x$

$\quad x := \frac{1}{2}\langle x - 1 \rangle + \frac{1}{2}\langle x \rangle$

while $y > 0$ do

$\quad y := y - 1$

My Work

Modularity with Expected Size

Example

$x := n$

$y = 0$

while $x > 0$ do

$\quad y := y + x$

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

$\rightarrow \mathbb{E}(y) \in \mathcal{O}(n^2)$

while $y > 0$ do

$\quad y := y - 1$

My Work

Modularity with Expected Size

Example

$x := n$

$y = 0$

while $x > 0$ do

$\quad y := y + x$

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

$\rightarrow \mathbb{E}(y) \in \mathcal{O}(n^2)$

while $y > 0$ do

$\quad y := y - 1$

$\rightarrow$ Expectation of y after the first loop is at most quadratic in n.

My Work

Modularity with Expected Size

Example

$x := n$

$y = 0$

while $x > 0$ **do**

$\quad y := y + x$

$\quad x := \frac{1}{2}\langle x - 1 \rangle + \frac{1}{2}\langle x \rangle$

$\rightarrow \mathbb{E}(y) \in \mathcal{O}(n^2)$

while $y > 0$ **do**

$\quad y := y - 1$

$\rightarrow$ Expectation of y after the first loop is at most quadratic in n.

$\rightarrow$ Expected runtime of the second loop is linear in y.

My Work

Modularity with Expected Size

Example

$x := n$

$y = 0$

while $x > 0$ **do**

| $y := y + x$

| $x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

$\rightarrow \mathbb{E}(y) \in \mathcal{O}(n^2)$

while $y > 0$ **do**

| $y := y - 1$

$\rightarrow$ Expectation of y after the first loop is at most quadratic in n.

$\rightarrow$ Expected runtime of the second loop is linear in y.

$\rightarrow$ Expected runtime of the program is at most quadratic.

My Work

Modularity with Expected Size

Example

$x := n$

$y = 0$

while $x > 0$ **do**

  $y := y + x$

  $x := \frac{1}{2}\langle x-1\rangle + \frac{1}{2}\langle x\rangle$

$\rightarrow \mathbb{E}(y) \in \mathcal{O}(n^2)$

while $y > 0$ **do**

  $y := y - 1$

→ Expectation of y after the first loop is at most quadratic in n.

→ Expected runtime of the second loop is linear in y.

→ Expected runtime of the program is at most quadratic.

→ Modular approach in „Koat“ [ACM–TOPLAS 2016] uses size-approximations.

My Work

Modularity with Expected Size

Example

$x := n$

$y = 0$

while $x > 0$ **do**

$\quad y := y + x$

$\quad x := \frac{1}{2}\langle x - 1\rangle + \frac{1}{2}\langle x\rangle$

$\rightarrow \mathbb{E}(y) \in \mathcal{O}(n^2)$

while $y > 0$ **do**

$\quad y := y - 1$

$\rightarrow$ Expectation of y after the first loop is at most quadratic in n.

$\rightarrow$ Expected runtime of the second loop is linear in y.

$\rightarrow$ Expected runtime of the program is at most quadratic.

$\rightarrow$ Modular approach in „Koat“ [ACM–TOPLAS 2016] uses size-approximations.

$\rightarrow$ Expected size is needed for generalising the modular approach.

My Work

Lower bounds

My Work

Lower bounds

Example

$x := 1$
$k := 0$
while $x = 1$ **do**
  $x := \frac{1}{2}\langle 0 \rangle + \frac{1}{2}\langle x \rangle$
  $k := k + 1$

My Work

Lower bounds

Example

$x := 1$
$k := 0$
while $x = 1$ **do**
  $x := \frac{1}{2}\langle 0 \rangle + \frac{1}{2}\langle x \rangle$
  $k := k + 1$

$\rightarrow$ $\mathbb{E}(\mathit{Term})$ can be constructed as a **least fixpoint** of a certain function.

My Work

Lower bounds

Example

$x := 1$
$k := 0$
while $x = 1$ **do**
  $x := \frac{1}{2}\langle 0 \rangle + \frac{1}{2}\langle x \rangle$
  $k := k + 1$

$\rightarrow$ $\mathbb{E}(\mathit{Term})$ can be constructed as a least fixpoint of a certain function.

$\rightarrow$ Underapproximation of $\mathbb{E}(\mathit{Term})$ is more difficult than overapproximation.

My Work

Lower bounds

Example

$x := 1$
$k := 0$
while $x = 1$ **do**
  $x := \frac{1}{2}\langle 0 \rangle + \frac{1}{2}\langle x \rangle$
  $k := k + 1$

$\rightarrow$ $\mathbb{E}(\mathit{Term})$ can be constructed as a least fixpoint of a certain function.

$\rightarrow$ Underapproximation of $\mathbb{E}(\mathit{Term})$ is more difficult than overapproximation.

$\rightarrow$ Important to infer tight bounds.

Conclusion

Conclusion

- Hot research topic at the moment.

Conclusion

- Hot research topic at the moment.
- Applications in analysing stochastic processes.

Conclusion

- Hot research topic at the moment.
- Applications in analysing stochastic processes.
- Increase modularity of fully automatic approaches.

Conclusion

- Hot research topic at the moment.
- Applications in analysing stochastic processes.
- Increase modularity of fully automatic approaches.
- Theory of expected size has to be developed.

Conclusion

- Hot research topic at the moment.
- Applications in analysing stochastic processes.
- Increase modularity of fully automatic approaches.
- Theory of expected size has to be developed.
- Lower bounds have to be considered.

References

Thank You

[1] Sheshansh Agrawal, Krishnendu Chatterjee, and Petr Novotný. Lexicographic ranking supermartingales: an efficient approach to termination of probabilistic programs. *POPL*, 2018.

[2] Olivier Bournez and Florent Garnier. Proving positive almost sure termination. *RTA*, 2005.

[3] Benjamin Lucien Kaminski and Joost-Pieter Katoen. On the hardness of almost–sure termination. *MFCS*, 2015.

[4] Benjamin Lucien Kaminski, Joost-Pieter Katoen, Christoph Matheja, and Federico Olmedo. Weakest precondition reasoning for expected run-times of probabilistic programs. *ESOP*, 2016.