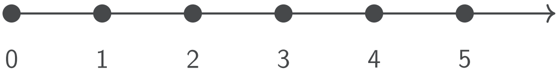
Automatically Analyzing Termination and Expected Runtime Complexity of Probabilistic Term Rewriting

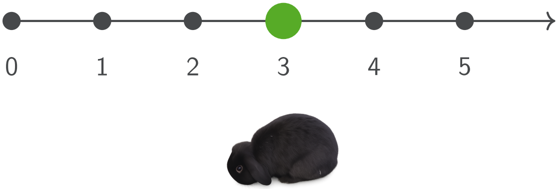
Jan-Christoph Kassing
RWTH Aachen University
27.05.2026

LuFG Informatik II

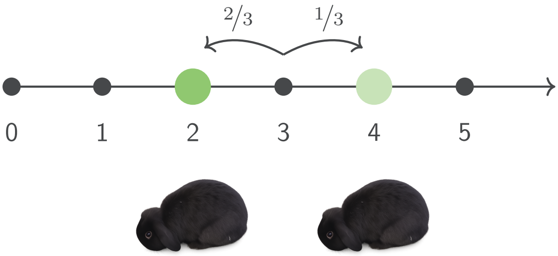
Random Walk



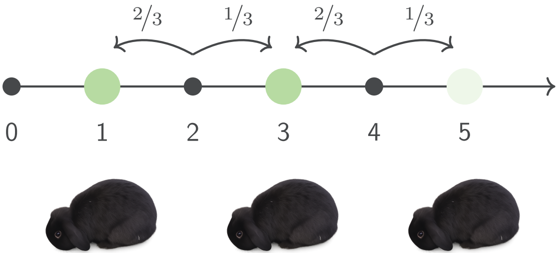
Random Walk



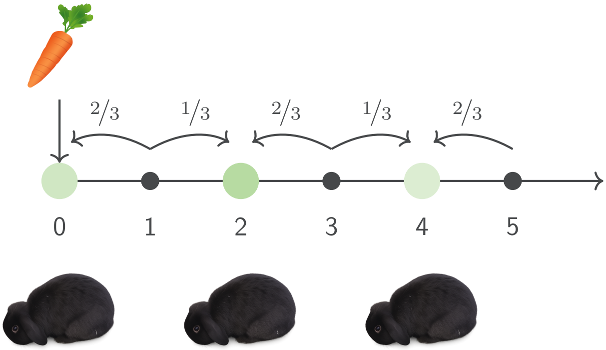
Random Walk



Random Walk

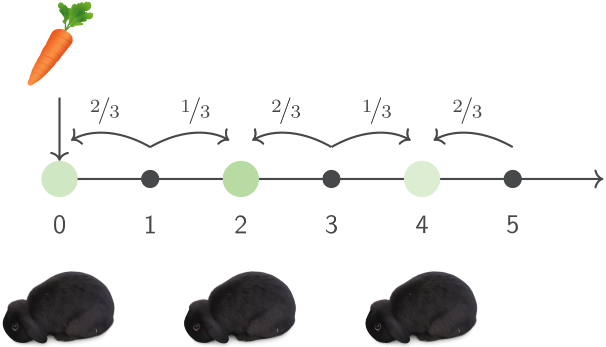


Random Walk



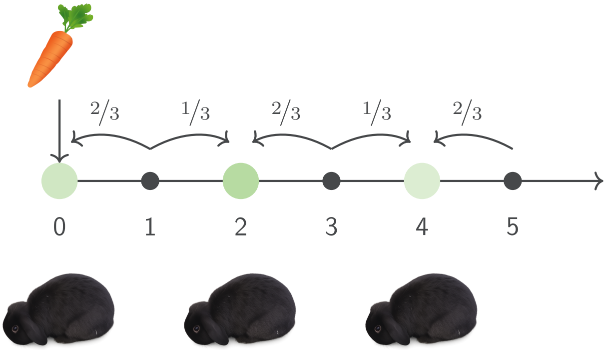
Random Walk

$x \leftarrow 3$
while $x > 0$ **do**
$\quad x \leftarrow x - 1 \ \oplus_{2/3} \ x \leftarrow x + 1;$



Random Walk

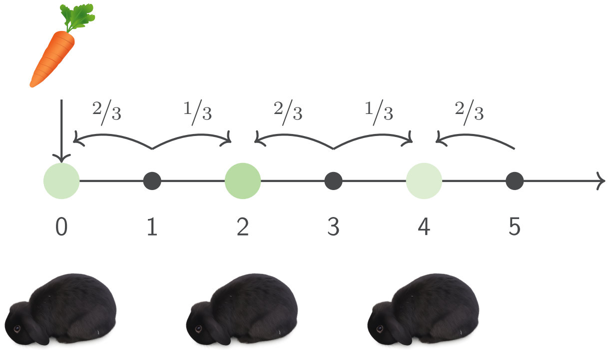
$x \leftarrow 3$
while $x > 0$ **do**
$\quad x \leftarrow x - 1 \oplus_{2/3} x \leftarrow x + 1;$



- Does the bunny (program) always reach the carrot (terminate)?

Random Walk

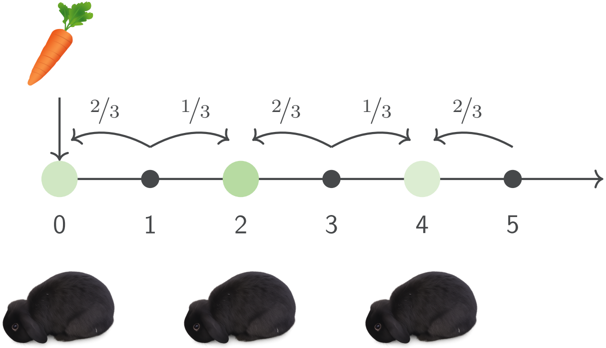
$$x \leftarrow 3$$
while $x > 0$ **do**
$$x \leftarrow x - 1 \;\oplus_{2/3}\; x \leftarrow x + 1;$$



- Does the bunny (program) always reach the carrot (terminate)?
- What is the probability of reaching the carrot (probability of termination)?

Random Walk

$x \leftarrow 3$
while $x > 0$ **do**
$\quad x \leftarrow x - 1 \;\oplus_{2/3}\; x \leftarrow x + 1;$



- Does the bunny (program) always reach the carrot (terminate)?
- What is the probability of reaching the carrot (probability of termination)?
- What is the expected number of steps it takes to reach the carrot (expected runtime)?

Two Important Questions

Two Important Questions

1. What are applications of probabilistic programs? Why are they interesting?

Two Important Questions

1. What are applications of probabilistic programs? Why are they interesting?

2. How to automatically analyze probabilistic programs?

Two Important Questions

1. What are applications of probabilistic programs? Why are they interesting?

2. How to automatically analyze probabilistic programs?

Probabilistic Quicksort

Order Natural Numbers

Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Probabilistic Quicksort

Order Natural Numbers

Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.

Probabilistic Quicksort

Order Natural Numbers

Given: $x_1, \dots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.

3	2	1	5	4

Probabilistic Quicksort

Order Natural Numbers

Given: $x_1, \dots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.

3	2	1	5	4

Probabilistic Quicksort

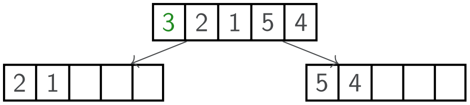
Order Natural Numbers

Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.



Probabilistic Quicksort

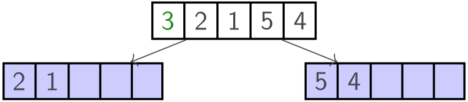
Order Natural Numbers

Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.



Probabilistic Quicksort

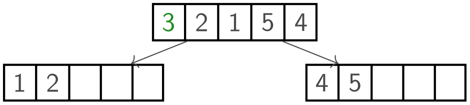
Order Natural Numbers

Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.



Probabilistic Quicksort

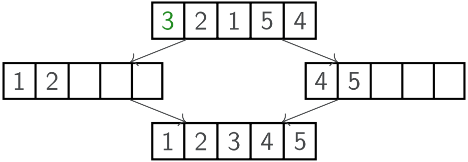
Order Natural Numbers

Given: $x_1, \dots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.



Probabilistic Quicksort

Order Natural Numbers

Given: $x_1, \dots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.

$\Longrightarrow$ Average runtime in $\mathcal{O}(n \cdot \log(n))$.

Probabilistic Quicksort

Order Natural Numbers

Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.

$\Longrightarrow$ Average runtime in $\mathcal{O}(n \cdot \log(n))$.
$\Longrightarrow$ Worst-Case runtime in $\mathcal{O}(n^2)$.

Probabilistic Quicksort

Order Natural Numbers

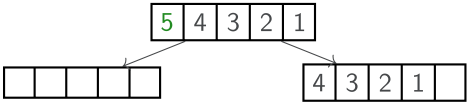
Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Deterministic Algorithm:

- Pick x_1 as *pivot element* and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.

$\Longrightarrow$ Average runtime in $\mathcal{O}(n \cdot \log(n))$.
$\Longrightarrow$ Worst-Case runtime in $\mathcal{O}(n^2)$.



Probabilistic Quicksort

Order Natural Numbers

Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Las Vegas Algorithm:

- Pick pivot element at random and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.

Probabilistic Quicksort

Order Natural Numbers

Given: $x_1, \ldots, x_n \in \mathbb{N}$

Problem: Sort the elements in ascending order.

Las Vegas Algorithm:

- Pick pivot element at random and split the list into $L = \{x_i \mid x_i \leq x_1\}$ and $R = \{x_i \mid x_i > x_1\}$.
- Solve the problem for L and R recursively.
- Combine the two solutions and place x_1 in between.

$\Longrightarrow$ Expected runtime in $\mathcal{O}(n \cdot \log(n))$.

Two Important Questions

1. What are applications of probabilistic programs? Why are they interesting?

2. How to automatically analyze probabilistic programs?

Two Important Questions

1. What are applications of probabilistic programs? Why are they interesting?

2. How to automatically analyze probabilistic programs?

Termination and Complexity Analysis for Ordinary Programs

Termination and Complexity Analysis for Ordinary Programs

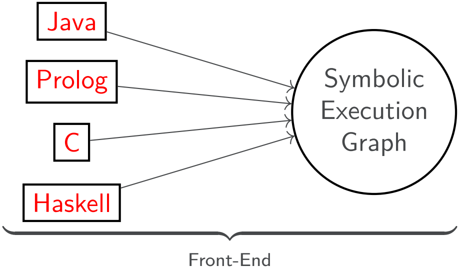
Java

Prolog

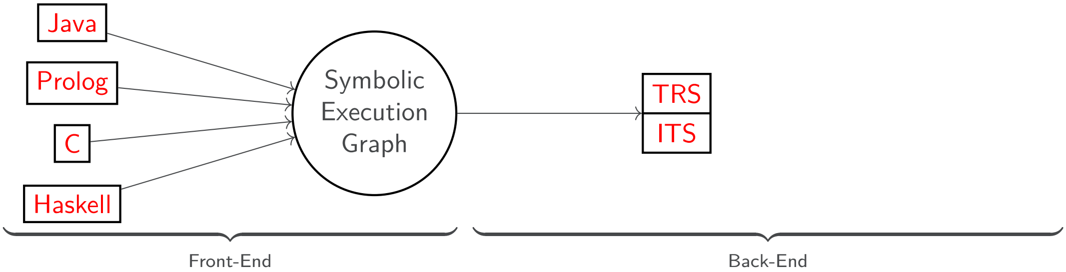
C

Haskell

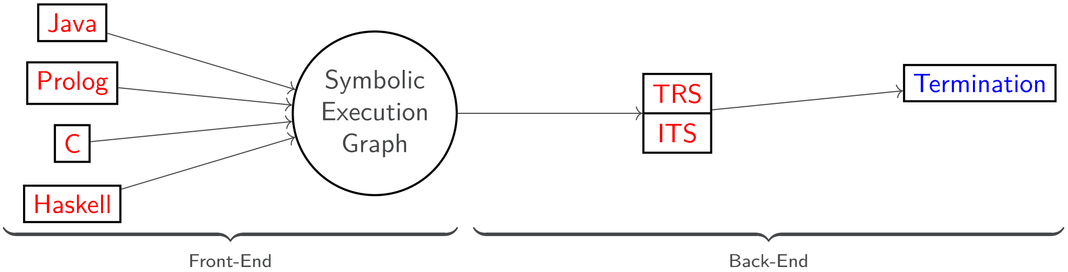
Termination and Complexity Analysis for Ordinary Programs



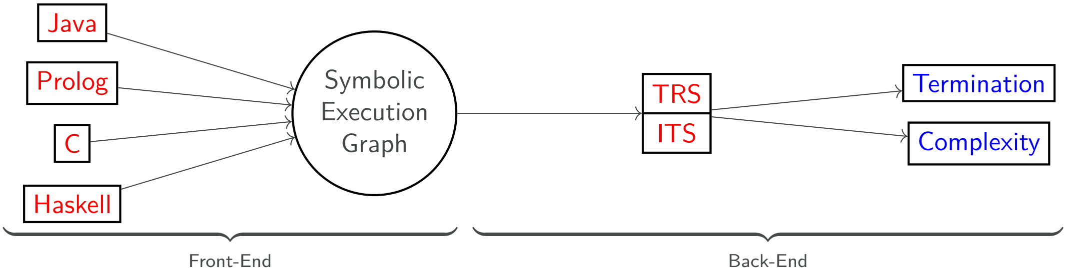
Termination and Complexity Analysis for Ordinary Programs



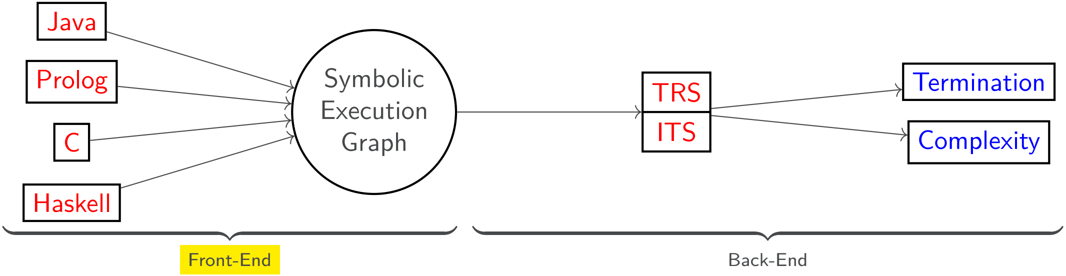
Termination and Complexity Analysis for Ordinary Programs



Termination and Complexity Analysis for Ordinary Programs

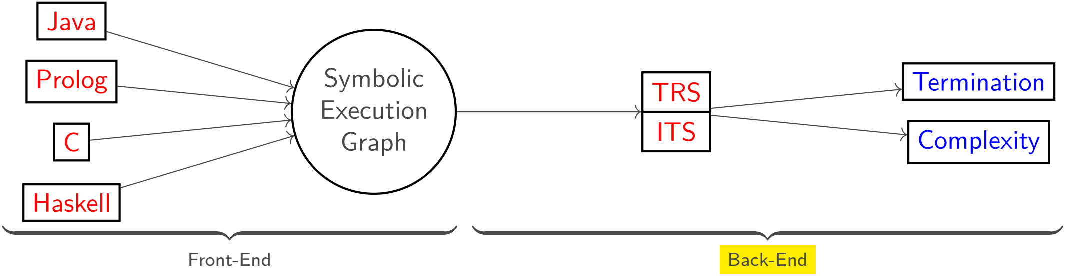


Termination and Complexity Analysis for Ordinary Programs



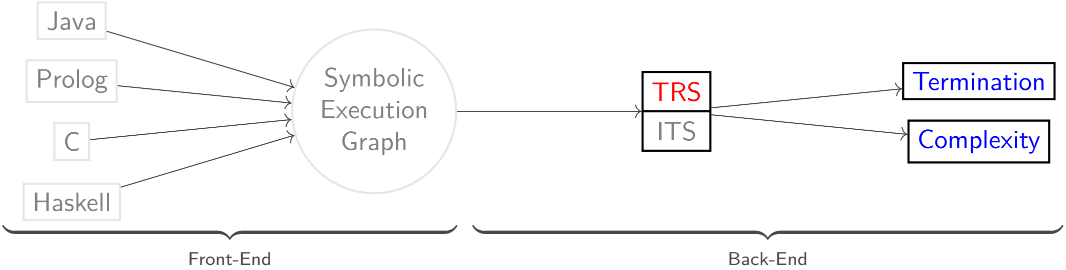
- language-specific features when generating symbolic execution graph

Termination and Complexity Analysis for Ordinary Programs



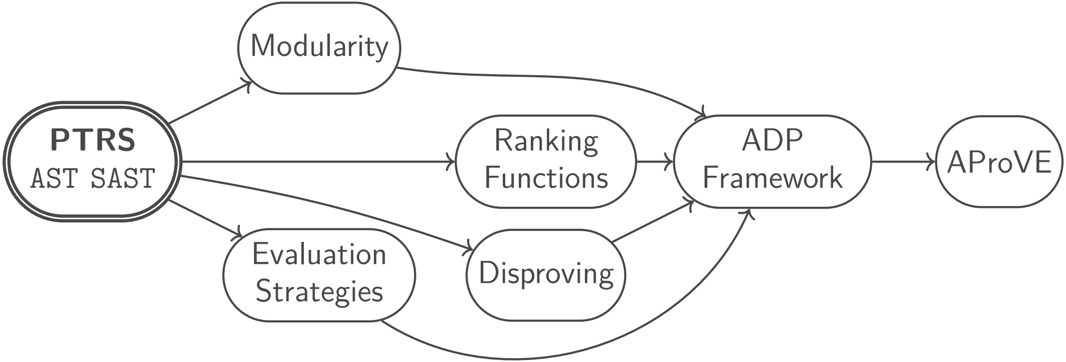
- language-specific features when generating symbolic execution graph
- back-end analyzes Term Rewrite Systems and/or Integer Transition Systems

Termination and Complexity Analysis for Ordinary Programs

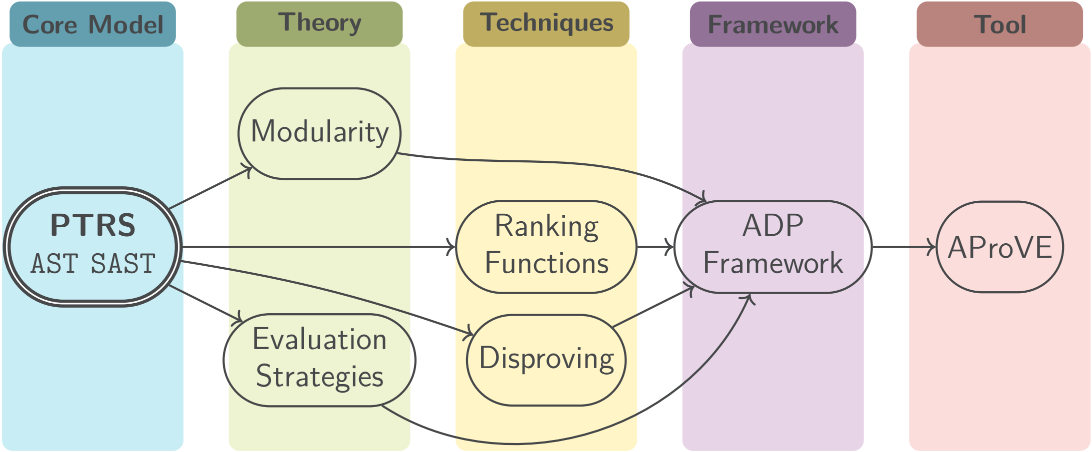


- language-specific features when generating symbolic execution graph
- back-end analyzes Term Rewrite Systems and/or Integer Transition Systems

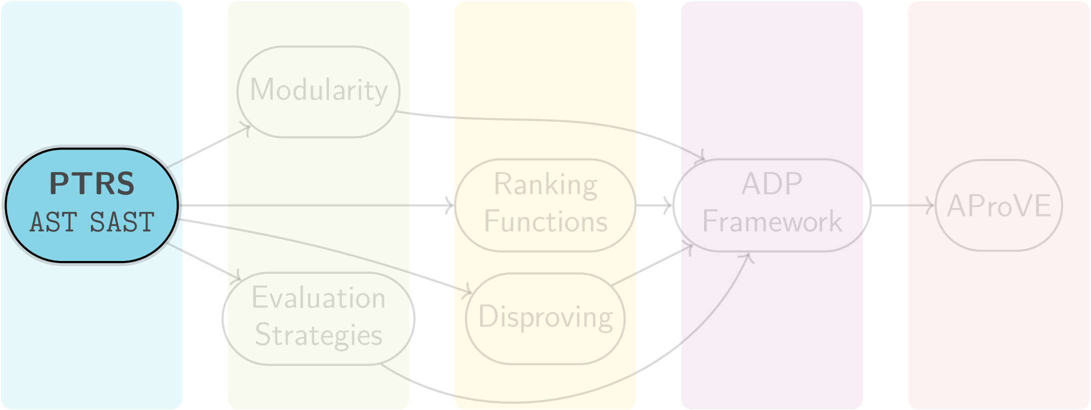
How to Automatically Analyze Probabilistic Programs?



How to Automatically Analyze Probabilistic Programs?



Probabilistic Term Rewriting



Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $$\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(\mathtt{0} : \mathtt{x}) \right\}$$

Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$:

$$\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(0 : \mathtt{x}) \right\}$$

Distribution: $\mu = \{^{p_1}t_1, \dots, {}^{p_k}t_k\}$ with $p_1 + \dots + p_k = 1$

Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathtt{0} : \mathtt{x}) \right\}$

Distribution: $\mu = \{^{p_1} t_1, \ldots, ^{p_k} t_k\}$ with $p_1 + \cdots + p_k = 1$

$\mathfrak{T}$:

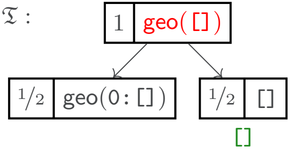
1	geo([])

Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(\mathsf{0} : \mathtt{x}) \right\}$

Distribution: $\mu = \{^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

$\mathfrak{T}$:

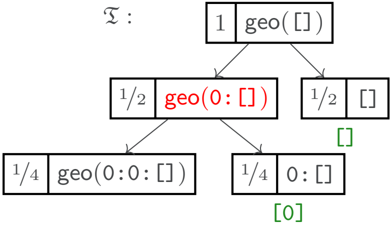


Probabilistic Term Rewrite Systems

$\mathcal{R}_{\text{geo}}$: $\text{geo}(x) \to \left\{ {}^{1/2}x, {}^{1/2}\text{geo}(0 : x) \right\}$

Distribution: $\mu = \{{}^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

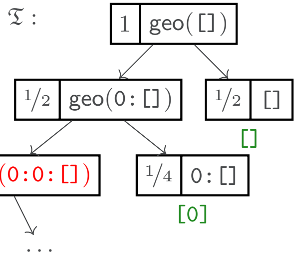
$\mathfrak{T}$:



Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(0 : \mathtt{x}) \right\}$

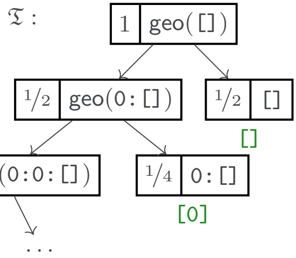
Distribution: $\mu = \{ {}^{p_1}t_1, \dots, {}^{p_k}t_k \}$ with $p_1 + \dots + p_k = 1$



Probabilistic Term Rewrite Systems

$\mathcal{R}_{\text{geo}}$: $\text{geo}(x) \to \left\{ {}^{1/2}x,\ {}^{1/2}\text{geo}(0 : x) \right\}$

Distribution: $\mu = \{{}^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$



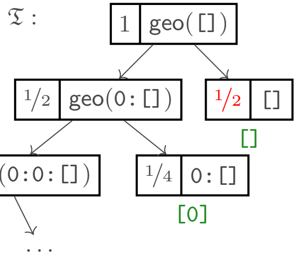
Probability of Termination:

$|\mathfrak{T}|$

Probabilistic Term Rewrite Systems

$\mathcal{R}_{\text{geo}}$: $\quad \text{geo}(x) \to \left\{ {}^{1/2}x, {}^{1/2}\text{geo}(0:x) \right\}$

Distribution: $\mu = \{^{p_1}t_1, \dots, ^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$



Probability of Termination:

$$|\mathfrak{T}| = 1/2 +$$

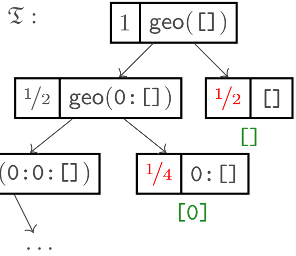
Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(0:\mathtt{x}) \right\}$

Distribution: $\mu = \{^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

Probability of Termination:

$$|\mathfrak{T}| = 1/2 + 1/4 +$$



Probabilistic Term Rewrite Systems

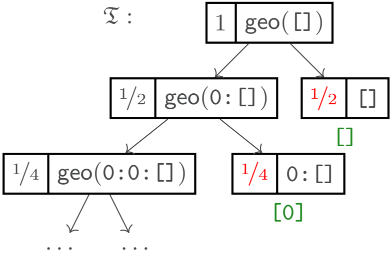
$\mathcal{R}_{\mathsf{geo}}$: $$\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(0 : \mathtt{x}) \right\}$$

Distribution: $\mu = \{{}^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

Probability of Termination:

$$|\mathfrak{T}| = 1/2 + 1/4 + 1/8 + \ldots$$

$\mathfrak{T}$:



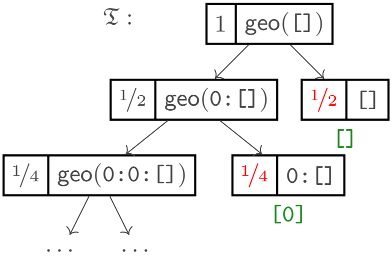
Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathtt{0}:\mathtt{x}) \right\}$

Distribution: $\mu = \{^{p_1}t_1, \ldots, ^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

Probability of Termination:

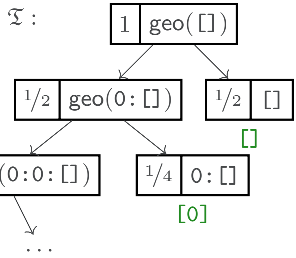
$$|\mathfrak{T}| = 1/2 + 1/4 + 1/8 + \ldots = \sum_{n=0}^{\infty} \frac{1}{2^{n+1}} = 1$$



Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathtt{0}:\mathtt{x}) \right\}$

Distribution: $\mu = \{^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$



Probability of Termination:

$$|\mathfrak{T}| = 1/2 + 1/4 + 1/8 + \ldots = \sum_{n=0}^{\infty} \frac{1}{2^{n+1}} = 1$$

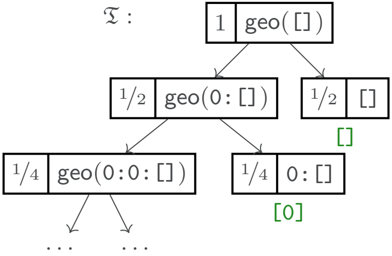
Almost-Sure Termination (AST)

PTRS $\mathcal{R}$ is AST if $|\mathfrak{T}| = 1$ for every $\mathfrak{T}$.

Probabilistic Term Rewrite Systems

$\mathcal{R}_{\text{geo}}$: $\quad \mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathtt{0}:\mathtt{x}) \right\}$ $\quad$ AST

Distribution: $\mu = \{^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$



Probability of Termination:

$$|\mathfrak{T}| = {}^{1}/{}_{2} + {}^{1}/{}_{4} + {}^{1}/{}_{8} + \ldots = \sum_{n=0}^{\infty} \frac{1}{2^{n+1}} = 1$$

Almost-Sure Termination (AST)

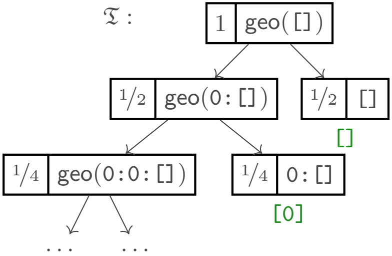
PTRS $\mathcal{R}$ is AST if $|\mathfrak{T}| = 1$ for every $\mathfrak{T}$.

Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(\mathtt{0}:\mathtt{x})\}$

Distribution: $\mu = \{{}^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

$\mathfrak{T}$:



Probabilistic Term Rewrite Systems

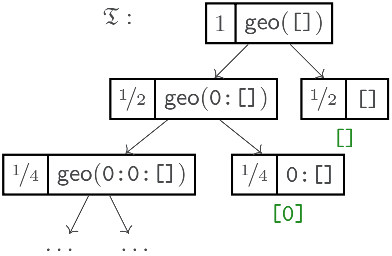
$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(0 : \mathtt{x}) \right\}$

Distribution: $\mu = \{{}^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

Expected Derivation Length:

$\mathrm{edl}(\mathfrak{T})$

$\mathfrak{T}$:



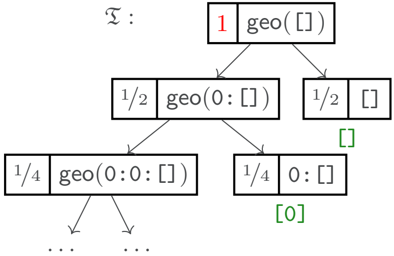
Probabilistic Term Rewrite Systems

$\mathcal{R}_{\text{geo}}$: $\quad \text{geo}(\text{x}) \to \left\{ {}^{1/2}\text{x}, {}^{1/2}\text{geo}(0:\text{x}) \right\}$

Distribution: $\mu = \{{}^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

Expected Derivation Length:

$$\text{edl}(\mathfrak{T}) = 1 +$$



Probabilistic Term Rewrite Systems

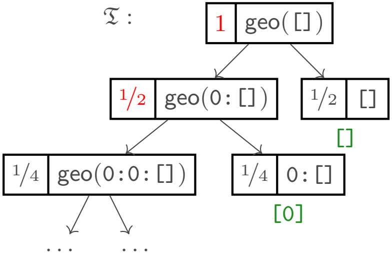
$\mathcal{R}_{\text{geo}}$: $\quad \mathsf{geo}(\mathsf{x}) \to \left\{ {}^{1/2}\mathsf{x},\ {}^{1/2}\mathsf{geo}(\mathsf{0}:\mathsf{x}) \right\}$

Distribution: $\mu = \{^{p_1}t_1, \dots, {}^{p_k}t_k\}$ with $p_1 + \dots + p_k = 1$

Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + {}^1\!/_2 +$$

$\mathfrak{T}$:



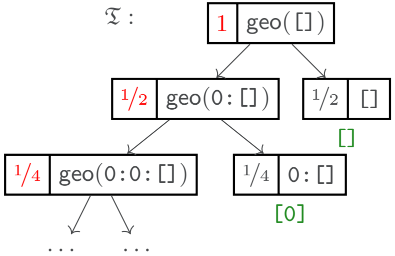
Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(0 : \mathtt{x}) \right\}$

Distribution: $\mu = \{^{p_1} t_1, \ldots, {}^{p_k} t_k\}$ with $p_1 + \cdots + p_k = 1$

Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + 1/2 + 1/4 + \ldots$$



Probabilistic Term Rewriting Systems

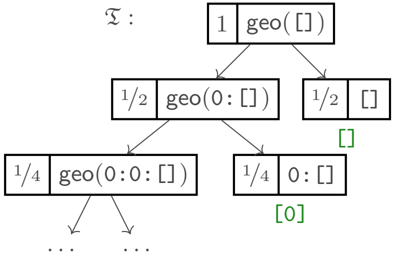
$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(\mathtt{0} : \mathtt{x}) \right\}$

Distribution: $\mu = \{ {}^{p_1} t_1, \ldots, {}^{p_k} t_k \}$ with $p_1 + \cdots + p_k = 1$

Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + {}^{1}\!/\!{}_{2} + {}^{1}\!/\!{}_{4} + \ldots = \sum_{n=0}^{\infty} ({}^{1}\!/\!{}_{2})^n = 2$$

$\mathfrak{T}$:



Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\quad \mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(0:\mathtt{x}) \right\}$

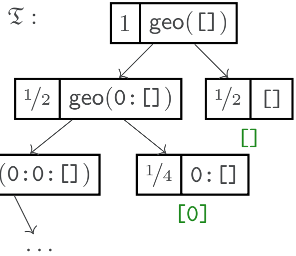
Distribution: $\mu = \{{}^{p_1}t_1, \dots, {}^{p_k}t_k\}$ with $p_1 + \dots + p_k = 1$

Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + 1/2 + 1/4 + \dots = \sum_{n=0}^{\infty} (1/2)^n = 2$$

Expected Derivation Height:

$$\mathrm{edh}_{\mathcal{R}_{\mathsf{geo}}}(\mathsf{geo}([\,])) = \sup\{\mathrm{edl}(\mathfrak{T}) \mid \mathfrak{T} \text{ starts with } \mathsf{geo}([\,])\}$$



Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(0:\mathtt{x}) \right\}$

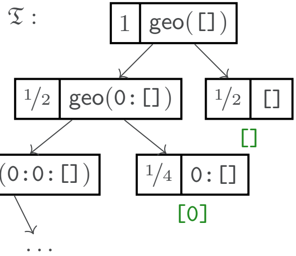
Distribution: $\mu = \{ {}^{p_1} t_1, \dots, {}^{p_k} t_k \}$ with $p_1 + \dots + p_k = 1$

Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + {}^{1}\!/_{2} + {}^{1}\!/_{4} + \dots = \sum_{n=0}^{\infty} ({}^{1}\!/_{2})^n = 2$$

Expected Derivation Height:

$$\mathrm{edh}_{\mathcal{R}_{\mathsf{geo}}}(\mathsf{geo}([\,])) = \sup\{\mathrm{edl}(\mathfrak{T}) \mid \mathfrak{T} \text{ starts with } \mathsf{geo}([\,])\}$$



Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(0:\mathtt{x}) \right\}$

Distribution: $\mu = \{{}^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

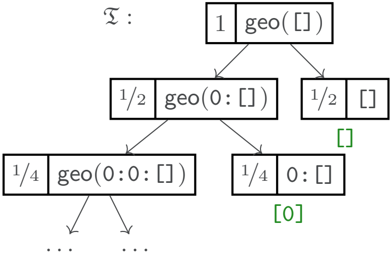
Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + {}^{1}/{}_{2} + {}^{1}/{}_{4} + \ldots = \sum_{n=0}^{\infty} ({}^{1}/{}_{2})^n = 2$$

Expected Derivation Height:

$$\mathrm{edh}_{\mathcal{R}_{\mathsf{geo}}}(\mathsf{geo}([\,])) = \sup\{\mathrm{edl}(\mathfrak{T}) \mid \mathfrak{T} \text{ starts with } \mathsf{geo}([\,])\} = 2$$

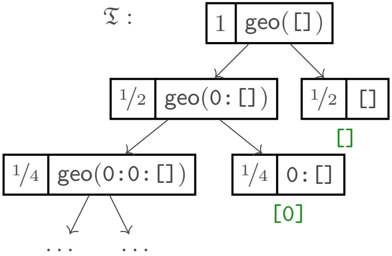
$\mathfrak{T}$:



Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(0:\mathtt{x}) \right\}$

Distribution: $\mu = \{^{p_1}t_1, \dots, {}^{p_k}t_k\}$ with $p_1 + \dots + p_k = 1$



Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + 1/2 + 1/4 + \dots = \sum_{n=0}^{\infty} (1/2)^n = 2$$

Expected Derivation Height:

$$\mathrm{edh}_{\mathcal{R}_{\mathsf{geo}}}(\mathsf{geo}([\,])) = \sup\{\mathrm{edl}(\mathfrak{T}) \mid \mathfrak{T} \text{ starts with } \mathsf{geo}([\,])\} = 2$$

Strong Almost-Sure Termination (SAST)[ADY20]

PTRS $\mathcal{R}$ is SAST if $\mathrm{edh}_{\mathcal{R}}(t)$ is finite for every start term t.

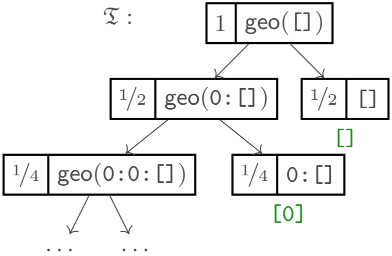
[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Probabilistic Term Rewrite Systems

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \left\{{}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(\mathtt{0}:\mathtt{x})\right\}$ SAST

Distribution: $\mu = \{{}^{p_1}t_1, \ldots, {}^{p_k}t_k\}$ with $p_1 + \cdots + p_k = 1$

$\mathfrak{T}$:



Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + {}^{1}/{}_{2} + {}^{1}/{}_{4} + \ldots = \sum_{n=0}^{\infty} ({}^{1}/{}_{2})^n = 2$$

Expected Derivation Height:

$$\mathrm{edh}_{\mathcal{R}_{\mathsf{geo}}}(\mathsf{geo}([\,])) = \sup\{\mathrm{edl}(\mathfrak{T}) \mid \mathfrak{T} \text{ starts with } \mathsf{geo}([\,])\} = 2$$

Strong Almost-Sure Termination (SAST)[ADY20]

PTRS $\mathcal{R}$ is SAST if $\mathrm{edh}_{\mathcal{R}}(t)$ is finite for every start term t.

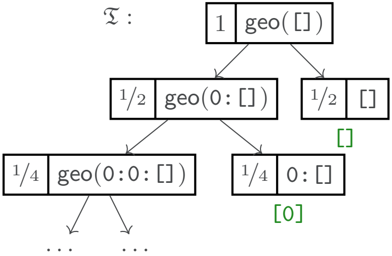
[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Probabilistic Term Rewrite Systems

$\mathcal{R}_{\text{geo}}$: $\mathtt{geo(x)} \to \left\{ {}^{1/2}\mathtt{x},\ {}^{1/2}\mathtt{geo(0:x)} \right\}$ SAST

Distribution: $\mu = \{^{p_1}t_1, \dots, {}^{p_k}t_k\}$ with $p_1 + \dots + p_k = 1$

$\mathfrak{T}$:



Expected Derivation Length:

$$\mathrm{edl}(\mathfrak{T}) = 1 + {}^1\!/_2 + {}^1\!/_4 + \dots = \sum_{n=0}^{\infty} ({}^1\!/_2)^n = 2$$

Expected Derivation Height:

$$\mathrm{edh}_{\mathcal{R}_{\text{geo}}}(\mathtt{geo([])}) = \sup\{\mathrm{edl}(\mathfrak{T}) \mid \mathfrak{T} \text{ starts with } \mathtt{geo([])}\} = 2$$

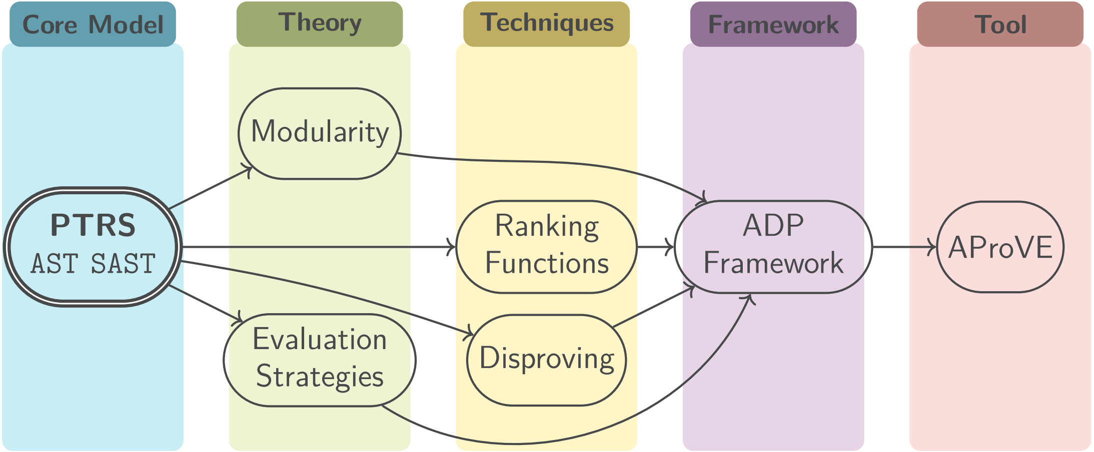
Strong Almost-Sure Termination (SAST)[ADY20]

PTRS $\mathcal{R}$ is SAST if $\mathrm{edh}_{\mathcal{R}}(t)$ is finite for every start term t.

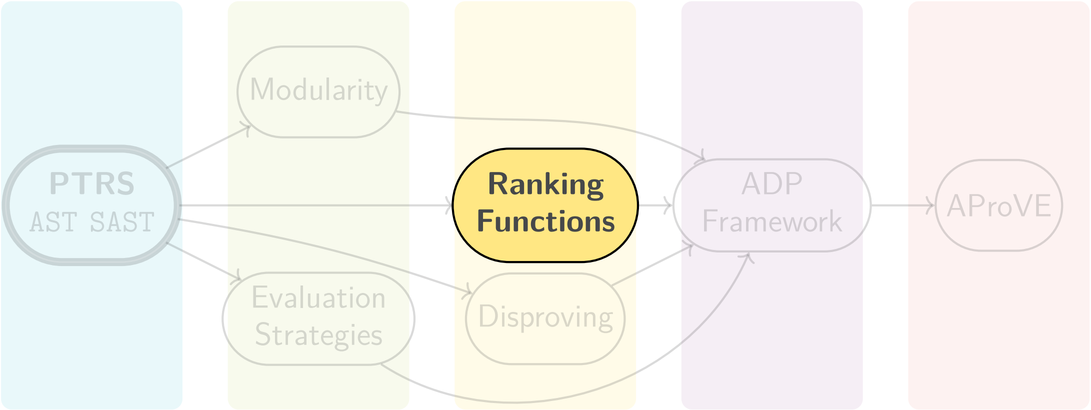
SAST $\Rightarrow$ AST and AST $\not\Rightarrow$ SAST

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

How to Automatically Analyze Probabilistic Programs?



Ranking Functions



Proving Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathsf{0} : \mathtt{x})\}$

Proving Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathtt{0} : \mathtt{x})\}$

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathtt{0}:\mathtt{x})\}$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathsf{0}:\mathtt{x})\}$

$$Pol_{\mathsf{geo}}(x) = x + 2 \qquad Pol_{\mathsf{0}} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\text{geo}}$: $\text{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\text{geo}(0 : \mathtt{x})\}$

$$Pol_{\text{geo}}(x) = x + 2 \qquad Pol_0 = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is AST if for all $\ell \to \{{}^{p_1}r_1, \ldots, {}^{p_k}r_k\} \in \mathcal{R}$:

- $Pol(\ell) > Pol(r_j)$ for some $1 \leq j \leq k$
- $Pol(\ell) \geq p_1 \cdot Pol(r_1) + \ldots + p_k \cdot Pol(r_k)$

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(0:\mathtt{x})\}$

$$Pol_{\mathsf{geo}}(x) = x + 2 \qquad Pol_{0} = 0 \qquad Pol_{:}(x,y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is AST if for all $\ell \to \{{}^{p_1}r_1, \ldots, {}^{p_k}r_k\} \in \mathcal{R}$:

- $Pol(\ell) > Pol(r_j)$ for some $1 \leq j \leq k$
- $Pol(\ell) \geq p_1 \cdot Pol(r_1) + \ldots + p_k \cdot Pol(r_k)$

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\text{geo}}$: $Pol(\text{geo}(\mathtt{x})) > Pol(\mathtt{x})$

$$Pol_{\text{geo}}(x) = x + 2 \qquad Pol_{0} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is AST if for all $\ell \to \{^{p_1} r_1, \ldots, {}^{p_k} r_k\} \in \mathcal{R}$:

- $Pol(\ell) > Pol(r_j)$ for some $1 \leq j \leq k$
- $Pol(\ell) \geq p_1 \cdot Pol(r_1) + \ldots + p_k \cdot Pol(r_k)$

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $x + 2 > x$

$$Pol_{\mathsf{geo}}(x) = x + 2 \qquad Pol_{\mathsf{0}} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is AST if for all $\ell \to \{{}^{p_1} r_1, \ldots, {}^{p_k} r_k\} \in \mathcal{R}$:

- $Pol(\ell) > Pol(r_j)$ for some $1 \leq j \leq k$
- $Pol(\ell) \geq p_1 \cdot Pol(r_1) + \ldots + p_k \cdot Pol(r_k)$

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $Pol(\mathsf{geo}(\mathtt{x})) \geq \mathbb{E}_{Pol}(\{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(0:\mathtt{x})\})$

$$Pol_{\mathsf{geo}}(x) = x + 2 \qquad Pol_{0} = 0 \qquad Pol_{:}(x,y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is AST if for all $\ell \to \{{}^{p_1}r_1, \ldots, {}^{p_k}r_k\} \in \mathcal{R}$:

- $Pol(\ell) > Pol(r_j)$ for some $1 \leq j \leq k$
- $Pol(\ell) \geq p_1 \cdot Pol(r_1) + \ldots + p_k \cdot Pol(r_k)$

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\text{geo}}$: $Pol(\text{geo}(\mathtt{x})) \geq {}^1\!/\!_2 \cdot Pol(\mathtt{x}) + {}^1\!/\!_2 \cdot Pol(\text{geo}(\mathtt{0}:\mathtt{x}))$

$$Pol_{\text{geo}}(x) = x + 2 \qquad Pol_{\mathtt{0}} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is AST if for all $\ell \to \{^{p_1}r_1, \ldots, ^{p_k}r_k\} \in \mathcal{R}$:

- $Pol(\ell) > Pol(r_j)$ for some $1 \leq j \leq k$
- $Pol(\ell) \geq p_1 \cdot Pol(r_1) + \ldots + p_k \cdot Pol(r_k)$

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $x + 2 \geq x + {}^3\!/_2$

$$Pol_{\mathsf{geo}}(x) = x + 2 \qquad Pol_{\mathsf{0}} = 0 \qquad Pol_{\mathsf{:}}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is AST if for all $\ell \to \{^{p_1}r_1, \ldots, {}^{p_k}r_k\} \in \mathcal{R}$:

- $Pol(\ell) > Pol(r_j)$ for some $1 \leq j \leq k$
- $Pol(\ell) \geq p_1 \cdot Pol(r_1) + \ldots + p_k \cdot Pol(r_k)$

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Almost-Sure Termination

$\mathcal{R}_{\text{geo}}$: $x + 2 \geq x + 3/2$ $\Longrightarrow$ proves AST

$$Pol_{\text{geo}}(x) = x + 2 \qquad Pol_{0} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving AST [KG23][MMKK]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is AST if for all $\ell \to \{^{p_1}r_1, \ldots, ^{p_k}r_k\} \in \mathcal{R}$:

- $Pol(\ell) > Pol(r_j)$ for some $1 \leq j \leq k$
- $Pol(\ell) \geq p_1 \cdot Pol(r_1) + \ldots + p_k \cdot Pol(r_k)$

[MMKK] McIver et al. A new proof rule for almost-sure termination (POPL 2017)
[KG23] J.-C. Kassing and J. Giesl. Proving Almost-Sure Innermost Termination [...] (CADE 2023)

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $$\mathsf{geo}(\mathtt{x}) \to \{ {}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathsf{0} : \mathtt{x}) \}$$

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x},\ {}^{1/2}\mathsf{geo}(0:\mathtt{x})\}$

Theorem: Proving SAST[ADY20]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathsf{0}:\mathtt{x})\}$

$$Pol_{\mathsf{geo}}(\mathtt{x}) = x + 2 \qquad Pol_{\mathsf{0}} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving SAST[ADY20]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathtt{0}:\mathtt{x})\}$

$$Pol_{\mathsf{geo}}(\mathtt{x}) = x + 2 \qquad Pol_{\mathtt{0}} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving SAST[ADY20]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is SAST if for all rules $\ell \to \mu$: $Pol(\ell) > \mathbb{E}_{Pol}(\mu) = \sum_{1 \leq j \leq k} p_j \cdot Pol(r_j)$ for $\mu = \{{}^{p_1}r_1, \ldots, {}^{p_k}r_k\}$.

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $Pol(\mathsf{geo}(\mathtt{x})) > \mathbb{E}_{Pol}(\{^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathsf{0}:\mathtt{x})\})$

$$Pol_{\mathsf{geo}}(\mathtt{x}) = x + 2 \qquad Pol_{\mathsf{0}} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving SAST[ADY20]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is SAST if for all rules $\ell \to \mu$: $Pol(\ell) > \mathbb{E}_{Pol}(\mu) = \sum_{1 \leq j \leq k} p_j \cdot Pol(r_j)$ for $\mu = \{^{p_1}r_1, \ldots, {}^{p_k}r_k\}$.

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\quad Pol(\mathsf{geo}(\mathtt{x})) > {}^{1}\!/\!{}_{2} \cdot Pol(\mathtt{x}) + {}^{1}\!/\!{}_{2} \cdot Pol(\mathsf{geo}(\mathtt{0}:\mathtt{x}))$

$$Pol_{\mathsf{geo}}(\mathtt{x}) = x + 2 \qquad Pol_{\mathtt{0}} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{\mathtt{[]}} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving SAST[ADY20]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is SAST if for all rules $\ell \to \mu$: $Pol(\ell) > \mathbb{E}_{Pol}(\mu) = \sum_{1 \leq j \leq k} p_j \cdot Pol(r_j)$ for $\mu = \{{}^{p_1}r_1, \ldots, {}^{p_k}r_k\}$.

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\text{geo}}$: $x + 2 > x + {}^3\!/_2$

$$Pol_{\text{geo}}(\mathrm{x}) = x + 2 \qquad Pol_{0} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving SAST[ADY20]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is SAST if for all rules $\ell \to \mu$: $Pol(\ell) > \mathbb{E}_{Pol}(\mu) = \sum_{1 \leq j \leq k} p_j \cdot Pol(r_j)$ for $\mu = \{{}^{p_1}r_1, \ldots, {}^{p_k}r_k\}$.

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\mathsf{geo}}$: $\mathsf{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{geo}(\mathtt{0}:\mathtt{x})\}$ $\Longrightarrow$ proves SAST

$$Pol_{\mathsf{geo}}(\mathtt{x}) = x + 2 \qquad Pol_{\mathtt{0}} = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

Theorem: Proving SAST[ADY20]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is SAST if for all rules $\ell \to \mu$: $Pol(\ell) > \mathbb{E}_{Pol}(\mu) = \sum_{1 \leq j \leq k} p_j \cdot Pol(r_j)$ for $\mu = \{{}^{p_1} r_1, \ldots, {}^{p_k} r_k\}$.

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

Proving Strong Almost-Sure Termination

$\mathcal{R}_{\text{geo}}$: $\text{geo}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\text{geo}(0:\mathtt{x})\}$ $\Longrightarrow$ proves SAST

$$Pol_{\text{geo}}(\mathtt{x}) = x + 2 \qquad Pol_0 = 0 \qquad Pol_{:}(x, y) = x + y + 1 \qquad Pol_{[]} = 0$$

Multilinear Polynomials

$x \cdot y$ is multilinear but x^2 is not.

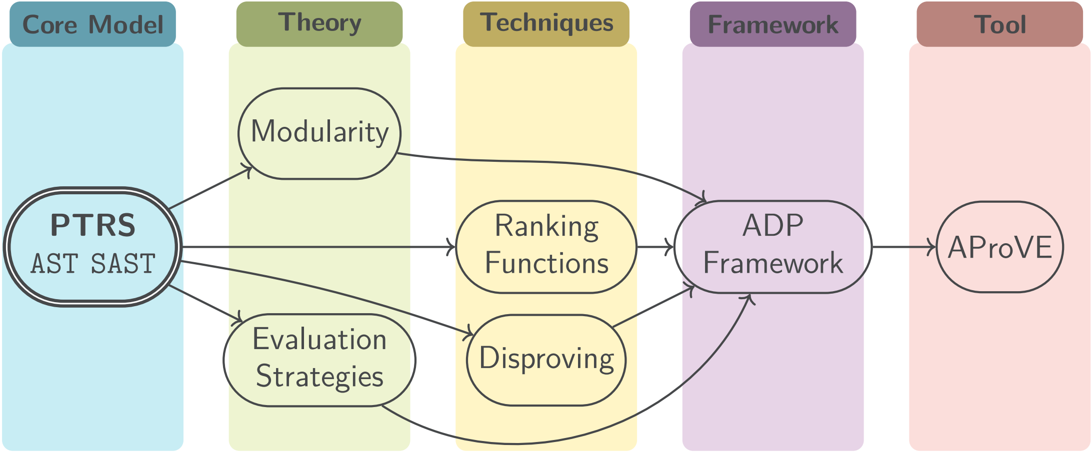
Theorem: Proving SAST[ADY20]

Let Pol be a natural & monotonic & multilinear polynomial interpretation.
$\mathcal{R}$ is SAST if for all rules $\ell \to \mu$: $Pol(\ell) > \mathbb{E}_{Pol}(\mu) = \sum_{1 \leq j \leq k} p_j \cdot Pol(r_j)$ for $\mu = \{{}^{p_1}r_1, \ldots, {}^{p_k}r_k\}$.

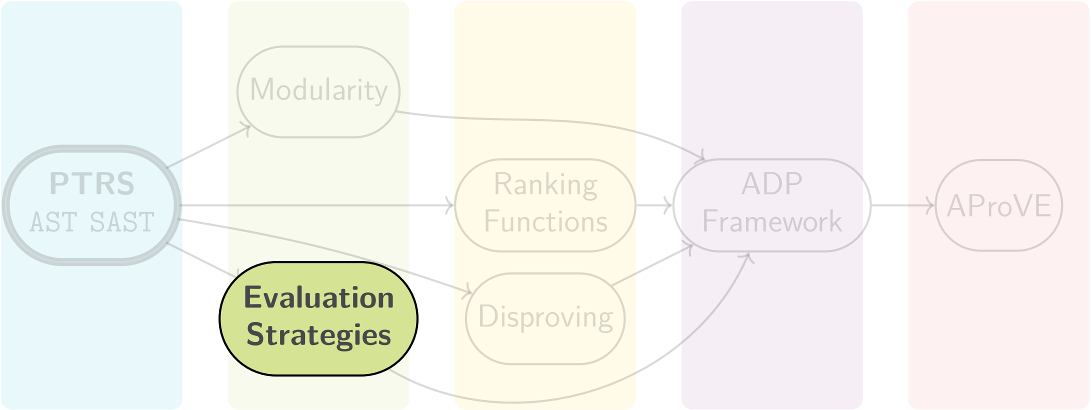
Goal: Infer expected complexity from the highest degree of Pol.

[ADY20] M. Avanzini, U. Dal Lago, and A. Yamada. On Probabilistic Term Rewriting (SCICO 2020)

How to Automatically Analyze Probabilistic Programs?



Evaluation Strategies



Innermost Termination vs. Termination

$$\mathcal{R}: \quad \begin{array}{rcl} \mathsf{f}(\mathsf{g}(\texttt{[]})) & \to & \mathsf{f}(\mathsf{g}(\texttt{[]})) \\ \mathsf{g}(\texttt{[]}) & \to & \texttt{[]} \end{array}$$

Innermost Termination vs. Termination

$\mathcal{R}$:

$$\begin{aligned} \mathsf{f}(\mathsf{g}(\texttt{[]})) &\to \mathsf{f}(\mathsf{g}(\texttt{[]})) \\ \mathsf{g}(\texttt{[]}) &\to \texttt{[]} \end{aligned}$$

Terminating? No:

Innermost Termination vs. Termination

$\mathcal{R}$:
$$\begin{aligned} \mathsf{f}(\mathsf{g}([\,])) &\to \mathsf{f}(\mathsf{g}([\,])) \\ \mathsf{g}([\,]) &\to [\,] \end{aligned}$$

Terminating? No:

$$\mathsf{f}(\mathsf{g}([\,]))$$

Innermost Termination vs. Termination

$\mathcal{R}$:

$$\mathsf{f}(\mathsf{g}(\mathsf{[]})) \to \mathsf{f}(\mathsf{g}(\mathsf{[]}))$$
$$\mathsf{g}(\mathsf{[]}) \to \mathsf{[]}$$

Terminating? **No**:

$$\mathsf{f}(\mathsf{g}(\mathsf{[]})) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}(\mathsf{[]}))$$

Innermost Termination vs. Termination

$\mathcal{R}$:

$$\begin{aligned} \mathsf{f}(\mathsf{g}(\texttt{[]})) &\to \mathsf{f}(\mathsf{g}(\texttt{[]})) \\ \mathsf{g}(\texttt{[]}) &\to \texttt{[]} \end{aligned}$$

Terminating? No:

$$\mathsf{f}(\mathsf{g}(\texttt{[]})) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}(\texttt{[]})) \to_{\mathcal{R}} \ldots$$

Innermost Termination vs. Termination

$\mathcal{R}$:
$$\begin{aligned} \mathsf{f}(\mathsf{g}([\,])) &\to \mathsf{f}(\mathsf{g}([\,])) \\ \mathsf{g}([\,]) &\to [\,] \end{aligned}$$

Terminating? No:

$$\mathsf{f}(\mathsf{g}([\,])) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}([\,])) \to_{\mathcal{R}} \ldots$$

Innermost Terminating? Yes:

Innermost Termination vs. Termination

$\mathcal{R}$:
$$\begin{aligned} \mathsf{f}(\mathsf{g}([\,])) &\to \mathsf{f}(\mathsf{g}([\,])) \\ \mathsf{g}([\,]) &\to [\,] \end{aligned}$$

Terminating? No:

$$\mathsf{f}(\mathsf{g}([\,])) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}([\,])) \to_{\mathcal{R}} \ldots$$

Innermost Terminating? Yes:

$$\mathsf{f}(\mathsf{g}([\,]))$$

Innermost Termination vs. Termination

$\mathcal{R}$:

$$
\begin{aligned}
\mathsf{f}(\mathsf{g}([\,])) &\to \mathsf{f}(\mathsf{g}([\,]))\\
\mathsf{g}([\,]) &\to [\,]
\end{aligned}
$$

Terminating? No:

$$\mathsf{f}(\mathsf{g}([\,])) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}([\,])) \to_{\mathcal{R}} \dots$$

Innermost Terminating? Yes:

$$\mathsf{f}(\mathsf{g}([\,])) \xrightarrow{\mathrm{i}}_{\mathcal{R}} \mathsf{f}([\,])$$

Innermost Termination vs. Termination

$\mathcal{R}$:

$$\begin{array}{rcl} \mathsf{f}(\mathsf{g}(\texttt{[]})) & \to & \mathsf{f}(\mathsf{g}(\texttt{[]})) \\ \mathsf{g}(\texttt{[]}) & \to & \texttt{[]} \end{array}$$

Terminating? **No**:

$$\mathsf{f}(\mathsf{g}(\texttt{[]})) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}(\texttt{[]})) \to_{\mathcal{R}} \ldots$$

Innermost Terminating? **Yes**:

$$\mathsf{f}(\mathsf{g}(\texttt{[]})) \xrightarrow{\mathsf{i}}_{\mathcal{R}} \mathsf{f}(\texttt{[]}) \leftarrow \text{normal form}$$

Innermost Termination vs. Termination

$\mathcal{R}$:

$$\begin{aligned} \mathsf{f}(\mathsf{g}([])) &\to \mathsf{f}(\mathsf{g}([])) \\ \mathsf{g}([]) &\to [] \end{aligned}$$

Terminating? No:

$$\mathsf{f}(\mathsf{g}([])) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}([])) \to_{\mathcal{R}} \dots$$

Innermost Terminating? Yes:

$$\mathsf{f}(\mathsf{g}([])) \xrightarrow{\mathsf{i}}_{\mathcal{R}} \mathsf{f}([]) \leftarrow \text{normal form}$$

$$\text{Termination} \implies \text{Innermost Termination}$$

Innermost Termination vs. Termination

$\mathcal{R}$:

$$\begin{array}{rcl} \mathsf{f}(\mathsf{g}([\,])) & \to & \mathsf{f}(\mathsf{g}([\,])) \\ \mathsf{g}([\,]) & \to & [\,] \end{array}$$

Terminating? No:

$$\mathsf{f}(\mathsf{g}([\,])) \to_{\mathcal{R}} \mathsf{f}(\mathsf{g}([\,])) \to_{\mathcal{R}} \dots$$

Innermost Terminating? Yes:

$$\mathsf{f}(\mathsf{g}([\,])) \xrightarrow{\mathsf{i}}_{\mathcal{R}} \mathsf{f}([\,]) \leftarrow \text{normal form}$$

$$\text{Termination} \implies \text{Innermost Termination}$$

Goal: Decidable Conditions s.t. Innermost AST $\implies$ AST and Innermost SAST $\implies$ SAST

Innermost Termination vs. Termination

Required Conditions

Innermost Termination vs. Termination

Required Conditions

- **Non-overlapping:** No left-hand sides overlap

Innermost Termination vs. Termination

Required Conditions

- **Non-overlapping:** No left-hand sides overlap

  $\mathtt{f}(\mathtt{[]}) \to \ldots$ and $\mathtt{f}(\mathtt{[]}) \to \ldots$ overlap

Innermost Termination vs. Termination

Required Conditions

- **Non-overlapping:** No left-hand sides overlap

  $\texttt{f}(\texttt{[]}) \to \ldots$ and $\texttt{f}(\texttt{[]}) \to \ldots$ overlap $\qquad$ $\texttt{f}(\texttt{[]}) \to \ldots$ and $\texttt{f}(\texttt{x:xs}) \to \ldots$ do not overlap

Innermost Termination vs. Termination

Required Conditions

- **Non-overlapping:** No left-hand sides overlap

  $\texttt{f([])} \to \ldots$ and $\texttt{f([])} \to \ldots$ overlap $\qquad$ $\texttt{f([])} \to \ldots$ and $\texttt{f(x:xs)} \to \ldots$ do not overlap

- **Linear:** Each variable at most once in a left-hand side and in each term of the right-hand side

Innermost Termination vs. Termination

Required Conditions

- **Non-overlapping:** No left-hand sides overlap

  $\mathtt{f}(\mathtt{[]}) \to \dots$ and $\mathtt{f}(\mathtt{[]}) \to \dots$ overlap $\mathtt{f}(\mathtt{[]}) \to \dots$ and $\mathtt{f}(\mathtt{x:xs}) \to \dots$ do not overlap

- **Linear:** Each variable at most once in a left-hand side and in each term of the right-hand side

  $\mathrm{g}(x, x) \to \dots$ not linear

Innermost Termination vs. Termination

Required Conditions

- **Non-overlapping:** No left-hand sides overlap

  $\mathtt{f([])} \to \dots$ and $\mathtt{f([])} \to \dots$ overlap $\mathtt{f([])} \to \dots$ and $\mathtt{f(x{:}xs)} \to \dots$ do not overlap

- **Linear:** Each variable at most once in a left-hand side and in each term of the right-hand side

  $\mathsf{g}(x, x) \to \dots$ not linear $\mathsf{f}(x) \to \{^1 \mathsf{g}(x, x)\}$ not linear

Innermost Termination vs. Termination

Required Conditions

- **Non-overlapping:** No left-hand sides overlap

  $\texttt{f([])} \to \ldots$ and $\texttt{f([])} \to \ldots$ overlap $\qquad$ $\texttt{f([])} \to \ldots$ and $\texttt{f(x:xs)} \to \ldots$ do not overlap

- **Linear:** Each variable at most once in a left-hand side and in each term of the right-hand side

  $\mathsf{g}(x, x) \to \ldots$ not linear $\qquad$ $\mathsf{f}(x) \to \{^1\mathsf{g}(x, x)\}$ not linear $\qquad$ $\mathsf{f}(x) \to \{^{1/2}\mathsf{f}(x), {}^{1/2}\mathsf{h}(x)\}$ linear

Innermost Termination vs. Termination

Required Conditions

- **Non-overlapping:** No left-hand sides overlap

  $\mathsf{f}(\texttt{[]}) \to \dots$ and $\mathsf{f}(\texttt{[]}) \to \dots$ overlap $\qquad$ $\mathsf{f}(\texttt{[]}) \to \dots$ and $\mathsf{f}(\texttt{x:xs}) \to \dots$ do not overlap

- **Linear:** Each variable at most once in a left-hand side and in each term of the right-hand side

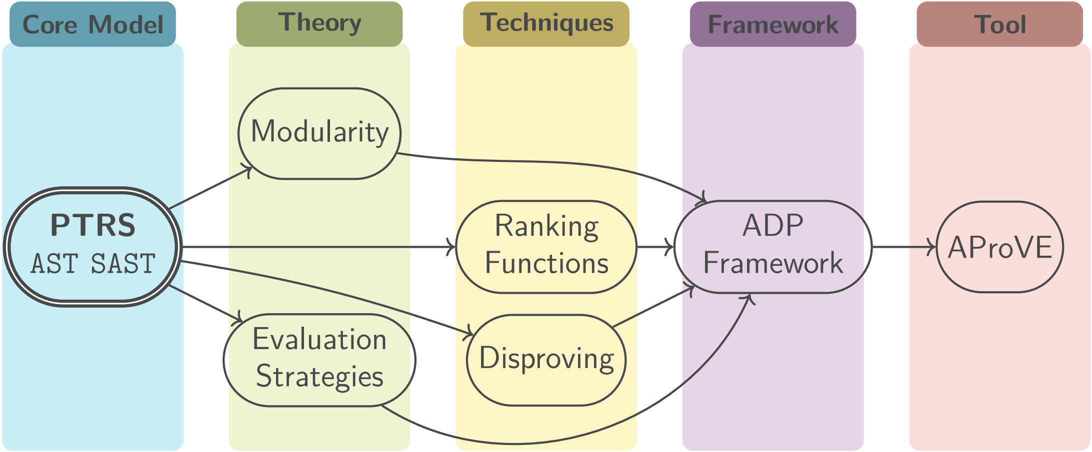
  $\mathsf{g}(x,x) \to \dots$ not linear $\qquad$ $\mathsf{f}(x) \to \{^1\mathsf{g}(x,x)\}$ not linear $\qquad$ $\mathsf{f}(x) \to \{^{1/2}\mathsf{f}(x), {}^{1/2}\mathsf{h}(x)\}$ linear

Theorem[KFG24][KG25a]

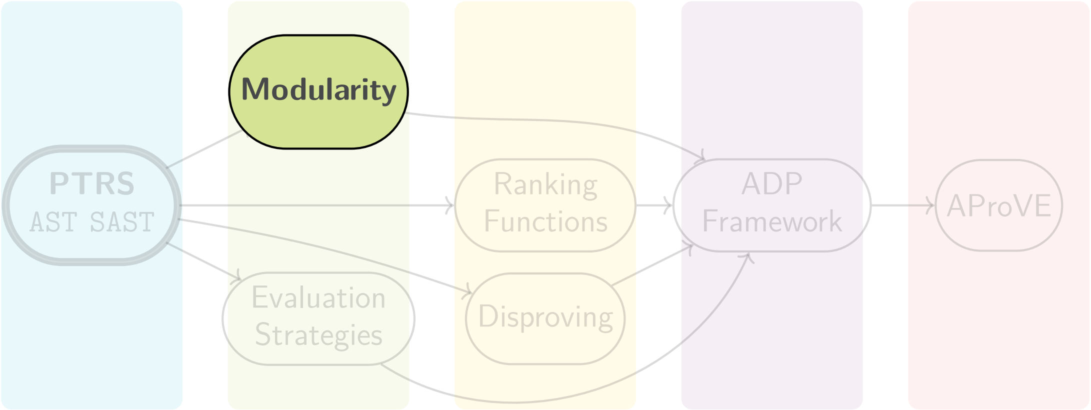
If a PTRS $\mathcal{R}$ is non-overlapping, and linear, then: $\qquad$ $\mathcal{R}$ is AST $\iff$ $\mathcal{R}$ is iAST.

[KG25a] J.-C. Kassing and J. Giesl. From Innermost to Full Probabilistic Term Rewriting: AST, Complexity, and Modularity (LMCS 2025)
[KFG24] J.-C. Kassing, F. Frohn, and J. Giesl. From Innermost to Full Almost-Sure Termination [...] (FoSSaCS 2024)

How to Automatically Analyze Probabilistic Programs?



Modularity Results



Modularity

Modularity

Imperative Programs:

Modularity

Imperative Programs:

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$

Modularity

Imperative Programs:

$$\begin{array}{l}\mathcal{R}_1 \text{ has property } \texttt{Prop} \\ \mathcal{R}_2 \text{ has property } \texttt{Prop}\end{array} \quad \Longleftrightarrow$$

Modularity

Imperative Programs:

$$\begin{array}{l} \mathcal{R}_1 \text{ has property } \texttt{Prop} \\ \mathcal{R}_2 \text{ has property } \texttt{Prop} \end{array} \quad \Longleftrightarrow \quad \mathcal{R}_1; \mathcal{R}_2 \;\text{ has property } \texttt{Prop}$$

Modularity

Imperative Programs:

Sequential Execution

↑

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$ $\iff$ $\mathcal{R}_1 ; \mathcal{R}_2$ has property $\texttt{Prop}$

Modularity

Imperative Programs:

Sequential Execution ↑

$\mathcal{R}_1$ has property `Prop`
$\mathcal{R}_2$ has property `Prop` $\iff$ $\mathcal{R}_1 ; \mathcal{R}_2$ has property `Prop`

Term Rewriting:

Modularity

Imperative Programs:

Sequential Execution ↑

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$ $\iff$ $\mathcal{R}_1 ; \mathcal{R}_2$ has property $\texttt{Prop}$

Term Rewriting:

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$

Modularity

Imperative Programs:

Sequential Execution

↑

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$ $\iff$ $\mathcal{R}_1; \mathcal{R}_2$ has property $\texttt{Prop}$

Term Rewriting:

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$ $\iff$

Modularity

Imperative Programs:

Sequential Execution

↑

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$ $\iff$ $\mathcal{R}_1 ; \mathcal{R}_2$ has property $\texttt{Prop}$

Term Rewriting:

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$ $\iff$ $\mathcal{R}_1 \cup \mathcal{R}_2$ has property $\texttt{Prop}$

Modularity

Imperative Programs:

Sequential Execution

$\uparrow$

$\mathcal{R}_1$ has property $\mathtt{Prop}$
$\mathcal{R}_2$ has property $\mathtt{Prop}$ $\Longleftrightarrow$ $\mathcal{R}_1 ; \mathcal{R}_2$ has property $\mathtt{Prop}$

Term Rewriting:

Union of Rule Sets

$\uparrow$

$\mathcal{R}_1$ has property $\mathtt{Prop}$
$\mathcal{R}_2$ has property $\mathtt{Prop}$ $\Longleftrightarrow$ $\mathcal{R}_1 \cup \mathcal{R}_2$ has property $\mathtt{Prop}$

Modularity

Imperative Programs:

Sequential Execution

$\uparrow$

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$ $\iff$ $\mathcal{R}_1; \mathcal{R}_2$ has property $\texttt{Prop}$

Term Rewriting:

Union of Rule Sets

$\uparrow$

$\mathcal{R}_1$ has property $\texttt{Prop}$
$\mathcal{R}_2$ has property $\texttt{Prop}$ $\iff$ $\mathcal{R}_1 \cup \mathcal{R}_2$ has property $\texttt{Prop}$

$\mathcal{R}_{len}$:

$$\begin{aligned} \mathsf{len}(\texttt{[]}) &\to \texttt{0} \\ \mathsf{len}(\texttt{x:xs}) &\to \texttt{1+}\mathsf{len}(\texttt{xs}) \end{aligned}$$

$\mathcal{R}_{add}$:

$$\begin{aligned} \texttt{0+x} &\to \texttt{x} \\ (\texttt{1+x})\texttt{+y} &\to \texttt{x+}(\texttt{1+y}) \end{aligned}$$

Modularity AST

Modularity AST

Disjoint Unions (No common function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(x) \to \{\nicefrac{1}{2}\,x, \nicefrac{1}{2}\,\mathsf{f}(\mathsf{f}(x))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(x) \to \{\nicefrac{1}{2}\,x, \nicefrac{1}{2}\,\mathsf{g}(\mathsf{g}(x))\}$$

Modularity AST

Disjoint Unions (No common function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathrm{x}) \to \{{}^{1/2}\mathrm{x}, {}^{1/2}\mathsf{f}(\mathsf{f}(\mathrm{x}))\}$$

$\mathcal{R}_2$: AST

$$\mathrm{g}(\mathrm{x}) \to \{{}^{1/2}\mathrm{x}, {}^{1/2}\mathrm{g}(\mathrm{g}(\mathrm{x}))\}$$

$$\mathsf{f}(\mathrm{g}(\mathrm{x}))$$

Modularity AST

Disjoint Unions (No common function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathtt{f}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathtt{f}(\mathtt{f}(\mathtt{x}))\}$$

$\mathcal{R}_2$: AST

$$\mathtt{g}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathtt{g}(\mathtt{g}(\mathtt{x}))\}$$

$$\mathtt{f}(\mathtt{g}(\mathtt{x}))$$

Modularity AST

Disjoint Unions (No common function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{f}(\mathsf{f}(\mathtt{x}))\}$$

$\mathcal{R}_2$: AST

$$\mathtt{g}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathtt{g}(\mathtt{g}(\mathtt{x}))\}$$

$$\mathsf{f}(\mathtt{g}(\mathtt{x}))$$

Shared Constructor Systems (No common defined function symbols): Yes

Modularity AST

Disjoint Unions (No common function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{f}(\mathsf{f}(\mathtt{x}))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{g}(\mathsf{g}(\mathtt{x}))\}$$

$$\mathsf{f}(\mathsf{g}(\mathtt{x}))$$

Shared Constructor Systems (No common defined function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathsf{s}(\mathtt{x})) \to \{{}^{1/2}\mathsf{f}(\mathtt{x}), {}^{1/2}\mathsf{f}(\mathsf{s}(\mathsf{s}((\mathtt{x}))))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(0) \to \{{}^{1/2}\mathsf{s}(0), {}^{1/2}\mathsf{s}(\mathsf{g}(\mathsf{g}(0)))\}$$

Modularity AST

Disjoint Unions (No common function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{f}(\mathsf{f}(\mathtt{x}))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{g}(\mathsf{g}(\mathtt{x}))\}$$

$$\mathsf{f}(\mathsf{g}(\mathtt{x}))$$

Shared Constructor Systems (No common defined function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathsf{s}(\mathtt{x})) \to \{{}^{1/2}\mathsf{f}(\mathtt{x}), {}^{1/2}\mathsf{f}(\mathsf{s}(\mathsf{s}((\mathtt{x}))))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(0) \to \{{}^{1/2}\mathsf{s}(0), {}^{1/2}\mathsf{s}(\mathsf{g}(\mathsf{g}(0)))\}$$

$$\{{}^{1}\mathsf{f}(\mathsf{g}(0))\}$$

Modularity AST

Disjoint Unions (No common function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{f}(\mathsf{f}(\mathtt{x}))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{g}(\mathsf{g}(\mathtt{x}))\}$$

$$\mathsf{f}(\mathsf{g}(\mathtt{x}))$$

Shared Constructor Systems (No common defined function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathsf{s}(\mathtt{x})) \to \{{}^{1/2}\mathsf{f}(\mathtt{x}), {}^{1/2}\mathsf{f}(\mathsf{s}(\mathsf{s}((\mathtt{x}))))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(0) \to \{{}^{1/2}\mathsf{s}(0), {}^{1/2}\mathsf{s}(\mathsf{g}(\mathsf{g}(0)))\}$$

$$\{{}^{1}\mathsf{f}(\mathsf{g}(0))\} \xrightarrow{\mathrm{i}}_{\mathcal{R}_2} \{{}^{1/2}\mathsf{f}(\mathsf{s}(0)), {}^{1/2}\mathsf{f}(\mathsf{s}(\mathsf{g}(\mathsf{g}(0))))\}$$

Modularity AST

Disjoint Unions (No common function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{f}(\mathsf{f}(\mathtt{x}))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(\mathtt{x}) \to \{{}^{1/2}\mathtt{x}, {}^{1/2}\mathsf{g}(\mathsf{g}(\mathtt{x}))\}$$

$$\mathsf{f}(\mathsf{g}(\mathtt{x}))$$

Shared Constructor Systems (No common defined function symbols): Yes

$\mathcal{R}_1$: AST

$$\mathsf{f}(\mathsf{s}(\mathtt{x})) \to \{{}^{1/2}\mathsf{f}(\mathtt{x}), {}^{1/2}\mathsf{f}(\mathsf{s}(\mathsf{s}((\mathtt{x}))))\}$$

$\mathcal{R}_2$: AST

$$\mathsf{g}(0) \to \{{}^{1/2}\mathsf{s}(0), {}^{1/2}\mathsf{s}(\mathsf{g}(\mathsf{g}(0)))\}$$

$$\{{}^{1}\mathsf{f}(\mathsf{g}(0))\} \xrightarrow{\mathrm{i}}_{\mathcal{R}_2} \{{}^{1/2}\mathsf{f}(\mathsf{s}(0)), {}^{1/2}\mathsf{f}(\mathsf{s}(\mathsf{g}(\mathsf{g}(0))))\} \xrightarrow{\mathrm{i}}_{\mathcal{R}_1} \ldots$$

Modularity SAST

Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

$$\begin{aligned} \mathsf{f}(\mathsf{c}(x,y)) &\to \{{}^{1}\mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(0) &\to \{{}^{1}0\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(x) &\to \{{}^{1/2}\mathsf{g}(\mathsf{d}(x)), {}^{1/2}x\} \\ \mathsf{d}(x) &\to \{{}^{1}\mathsf{c}(x,x)\} \end{aligned}$$

Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

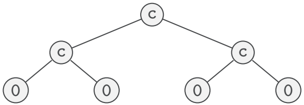
$\mathcal{R}_1$: SAST

$$\begin{aligned} \mathsf{f}(\mathsf{c}(x,y)) &\to \{^1 \mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(0) &\to \{^1 0\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(x) &\to \{^{1/2} \mathsf{g}(\mathsf{d}(x)), {}^{1/2} x\} \\ \mathsf{d}(x) &\to \{^1 \mathsf{c}(x,x)\} \end{aligned}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

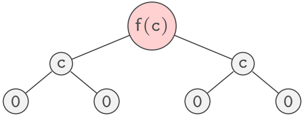
$\mathcal{R}_1$: SAST

$$\begin{aligned} \mathsf{f}(\mathsf{c}(x,y)) &\to \{1\!: \mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(0) &\to \{1\!: 0\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(x) &\to \{1/2\!: \mathsf{g}(\mathsf{d}(x)),\ 1/2\!: x\} \\ \mathsf{d}(x) &\to \{1\!: \mathsf{c}(x,x)\} \end{aligned}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

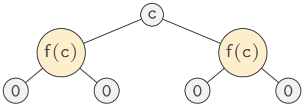
$\mathcal{R}_1$: SAST

$$\begin{aligned} \mathsf{f}(\mathsf{c}(\mathtt{x},\mathtt{y})) &\to \{^1\mathsf{c}(\mathsf{f}(\mathtt{x}),\mathsf{f}(\mathtt{y}))\} \\ \mathsf{f}(\mathsf{0}) &\to \{^1\mathsf{0}\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(\mathtt{x}) &\to \{^{1/2}\mathsf{g}(\mathsf{d}(\mathtt{x})), {}^{1/2}\mathtt{x}\} \\ \mathsf{d}(\mathtt{x}) &\to \{^1\mathsf{c}(\mathtt{x},\mathtt{x})\} \end{aligned}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

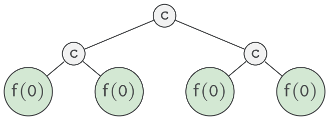
$\mathcal{R}_1$: SAST

$$\begin{aligned} \mathsf{f}(\mathsf{c}(x,y)) &\to \{^1\mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(\mathsf{0}) &\to \{^1\mathsf{0}\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(x) &\to \{^{1/2}\mathsf{g}(\mathsf{d}(x)), {}^{1/2}x\} \\ \mathsf{d}(x) &\to \{^1\mathsf{c}(x,x)\} \end{aligned}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

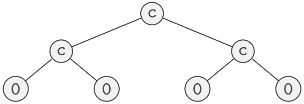
$$\begin{array}{rcl} \mathsf{f}(\mathsf{c}(\mathtt{x},\mathtt{y})) & \to & \{^1 \mathsf{c}(\mathsf{f}(\mathtt{x}), \mathsf{f}(\mathtt{y}))\} \\ \mathsf{f}(\mathsf{0}) & \to & \{^1 \mathsf{0}\} \end{array}$$

$\mathcal{R}_2$: SAST

$$\begin{array}{rcl} \mathsf{g}(\mathtt{x}) & \to & \{^{1/2} \mathsf{g}(\mathsf{d}(\mathtt{x})), {}^{1/2} \mathtt{x}\} \\ \mathsf{d}(\mathtt{x}) & \to & \{^1 \mathsf{c}(\mathtt{x}, \mathtt{x})\} \end{array}$$

$$\{^1 \mathsf{f}(\mathsf{g}(\mathsf{0}))\}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

$$\begin{aligned} \mathsf{f}(\mathsf{c}(\mathtt{x},\mathtt{y})) &\to \{^1\mathsf{c}(\mathsf{f}(\mathtt{x}),\mathsf{f}(\mathtt{y}))\} \\ \mathsf{f}(\mathsf{0}) &\to \{^1\mathsf{0}\} \end{aligned}$$

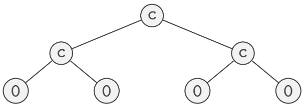
$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(\mathtt{x}) &\to \{^{1/2}\mathsf{g}(\mathsf{d}(\mathtt{x})), {}^{1/2}\mathtt{x}\} \\ \mathsf{d}(\mathtt{x}) &\to \{^1\mathsf{c}(\mathtt{x},\mathtt{x})\} \end{aligned}$$

$$\{^1\mathsf{f}(\mathsf{g}(\mathsf{0}))\}$$

$$\to_{\mathcal{R}_2}^k \quad \{\ldots, {}^{(1/2)^k}\mathsf{f}(\mathsf{d}^{k-1}(\mathsf{0})), \ldots\}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

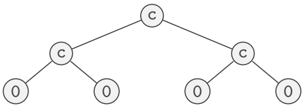
$$\begin{array}{rcl} \mathsf{f}(\mathsf{c}(x,y)) & \to & \{^1 \mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(\mathsf{0}) & \to & \{^1 \mathsf{0}\} \end{array}$$

$\mathcal{R}_2$: SAST

$$\begin{array}{rcl} \mathsf{g}(x) & \to & \{^{1/2} \mathsf{g}(\mathsf{d}(x)), {}^{1/2} x\} \\ \mathsf{d}(x) & \to & \{^1 \mathsf{c}(x,x)\} \end{array}$$

$$\begin{array}{rl} & \{^1 \mathsf{f}(\mathsf{g}(\mathsf{0}))\} \\ \to_{\mathcal{R}_2}^{k} & \{\ldots, {}^{(1/2)^k} \mathsf{f}(\mathsf{d}^{k-1}(\mathsf{0})), \ldots\} \\ \to_{\mathcal{R}_2}^{k} & \{\ldots, {}^{(1/2)^k} \mathsf{f}(\mathsf{c}^{k-1}(\mathsf{0})), \ldots\} \end{array}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

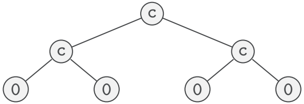
$$\begin{aligned} \mathsf{f}(\mathsf{c}(x,y)) &\to \{1{:}\,\mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(0) &\to \{1{:}\,0\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(x) &\to \{{}^{1/2}\mathsf{g}(\mathsf{d}(x)), {}^{1/2}x\} \\ \mathsf{d}(x) &\to \{{}^{1}\mathsf{c}(x,x)\} \end{aligned}$$

$$\begin{aligned} & \{{}^{1}\mathsf{f}(\mathsf{g}(0))\} \\ \to_{\mathcal{R}_2}^{k} \quad & \{\ldots, {}^{(1/2)^k}\mathsf{f}(\mathsf{d}^{k-1}(0)), \ldots\} \\ \to_{\mathcal{R}_2}^{k} \quad & \{\ldots, {}^{(1/2)^k}\mathsf{f}(\mathsf{c}^{k-1}(0)), \ldots\} \\ \to_{\mathcal{R}_1}^{2^{k-1}-1} \quad & \{\ldots, {}^{(1/2)^k}\mathsf{c}^{k-1}(0) \quad , \ldots\} \end{aligned}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

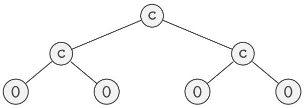
$$\begin{array}{rcl} \mathsf{f}(\mathsf{c}(x,y)) & \to & \{^1\mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(0) & \to & \{^1 0\} \end{array}$$

$\mathcal{R}_2$: SAST

$$\begin{array}{rcl} \mathsf{g}(x) & \to & \{^{1/2}\mathsf{g}(\mathsf{d}(x)), {}^{1/2}x\} \\ \mathsf{d}(x) & \to & \{^1\mathsf{c}(x,x)\} \end{array}$$

$$\begin{array}{rl} & \{^1\mathsf{f}(\mathsf{g}(0))\} \\ \to^k_{\mathcal{R}_2} & \{\ldots, (1/2)^k\,\mathsf{f}(\mathsf{d}^{k-1}(0)), \ldots\} \\ \to^k_{\mathcal{R}_2} & \{\ldots, (1/2)^k\,\mathsf{f}(\mathsf{c}^{k-1}(0)), \ldots\} \\ \to^{2^{k-1}-1}_{\mathcal{R}_1} & \{\ldots, (1/2)^k\,\mathsf{c}^{k-1}(0), \ldots\} \end{array}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



$\mathbb{E}(\vec{\mu})$

Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

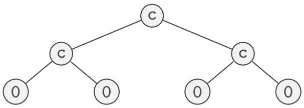
$$\begin{aligned} \mathsf{f}(\mathsf{c}(\mathtt{x},\mathtt{y})) &\to \{^1 \mathsf{c}(\mathsf{f}(\mathtt{x}),\mathsf{f}(\mathtt{y}))\} \\ \mathsf{f}(0) &\to \{^1 0\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(\mathtt{x}) &\to \{^{1/2} \mathsf{g}(\mathsf{d}(\mathtt{x})), {}^{1/2} \mathtt{x}\} \\ \mathsf{d}(\mathtt{x}) &\to \{^1 \mathsf{c}(\mathtt{x},\mathtt{x})\} \end{aligned}$$

$$\begin{aligned} & \{^1 \mathsf{f}(\mathsf{g}(0))\} \\ \to_{\mathcal{R}_2}^{k} \quad & \{\dots, {}^{(1/2)^k} \mathsf{f}(\mathsf{d}^{k-1}(0)), \dots\} \\ \to_{\mathcal{R}_2}^{k} \quad & \{\dots, {}^{(1/2)^k} \mathsf{f}(\mathsf{c}^{k-1}(0)), \dots\} \\ \to_{\mathcal{R}_1}^{2^{k-1}-1} \quad & \{\dots, {}^{(1/2)^k} \mathsf{c}^{k-1}(0) \quad , \dots\} \end{aligned}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



$$\mathbb{E}(\vec{\mu}) \geq \sum_{k=0}^{\infty} \frac{1}{2^{k+1}} \cdot 2^k$$

Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

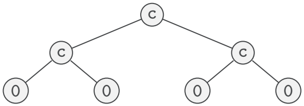
$$\begin{aligned} \mathsf{f}(\mathsf{c}(x,y)) &\to \{^1\mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(\mathsf{0}) &\to \{^1\mathsf{0}\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(x) &\to \{^{1/2}\mathsf{g}(\mathsf{d}(x)), {}^{1/2}x\} \\ \mathsf{d}(x) &\to \{^1\mathsf{c}(x,x)\} \end{aligned}$$

$$\begin{aligned} & \{^1\mathsf{f}(\mathsf{g}(\mathsf{0}))\} \\ \to_{\mathcal{R}_2}^{k} \quad & \{\ldots, {}^{(1/2)^k}\mathsf{f}(\mathsf{d}^{k-1}(\mathsf{0})), \ldots\} \\ \to_{\mathcal{R}_2}^{k} \quad & \{\ldots, {}^{(1/2)^k}\mathsf{f}(\mathsf{c}^{k-1}(\mathsf{0})), \ldots\} \\ \to_{\mathcal{R}_1}^{2^{k-1}-1} \quad & \{\ldots, {}^{(1/2)^k}\mathsf{c}^{k-1}(\mathsf{0}), \ldots\} \end{aligned}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



$$\mathbb{E}(\vec{\mu}) \geq \sum_{k=0}^{\infty} \frac{1}{2^{k+1}} \cdot 2^k = \sum_{k=0}^{\infty} 1/2$$

Modularity SAST

Disjoint Unions (No common function symbols): Yes (no details)

Shared Constructor Systems (No common defined function symbols): No

$\mathcal{R}_1$: SAST

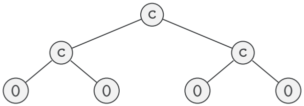
$$\begin{aligned} \mathsf{f}(\mathsf{c}(x,y)) &\to \{{}^1\mathsf{c}(\mathsf{f}(x),\mathsf{f}(y))\} \\ \mathsf{f}(0) &\to \{{}^1 0\} \end{aligned}$$

$\mathcal{R}_2$: SAST

$$\begin{aligned} \mathsf{g}(x) &\to \{{}^{1/2}\mathsf{g}(\mathsf{d}(x)), {}^{1/2}x\} \\ \mathsf{d}(x) &\to \{{}^1\mathsf{c}(x,x)\} \end{aligned}$$

$$\begin{aligned} & \{{}^1\mathsf{f}(\mathsf{g}(0))\} \\ \to_{\mathcal{R}_2}^{k} \; & \{\ldots, {}^{(1/2)^k}\mathsf{f}(\mathsf{d}^{k-1}(0)), \ldots\} \\ \to_{\mathcal{R}_2}^{k} \; & \{\ldots, {}^{(1/2)^k}\mathsf{f}(\mathsf{c}^{k-1}(0)), \ldots\} \\ \to_{\mathcal{R}_1}^{2^{k-1}-1} \; & \{\ldots, {}^{(1/2)^k}\mathsf{c}^{k-1}(0), \ldots\} \end{aligned}$$

$\mathbb{P}[\text{height } n] = (1/2)^{n+1}$



$$\mathbb{E}(\vec{\mu}) \geq \sum_{k=0}^{\infty} \frac{1}{2^{k+1}} \cdot 2^k = \sum_{k=0}^{\infty} 1/2 = \infty$$

Modularity for PTRSs

Theorem (Innermost Modularity)[KG25a]

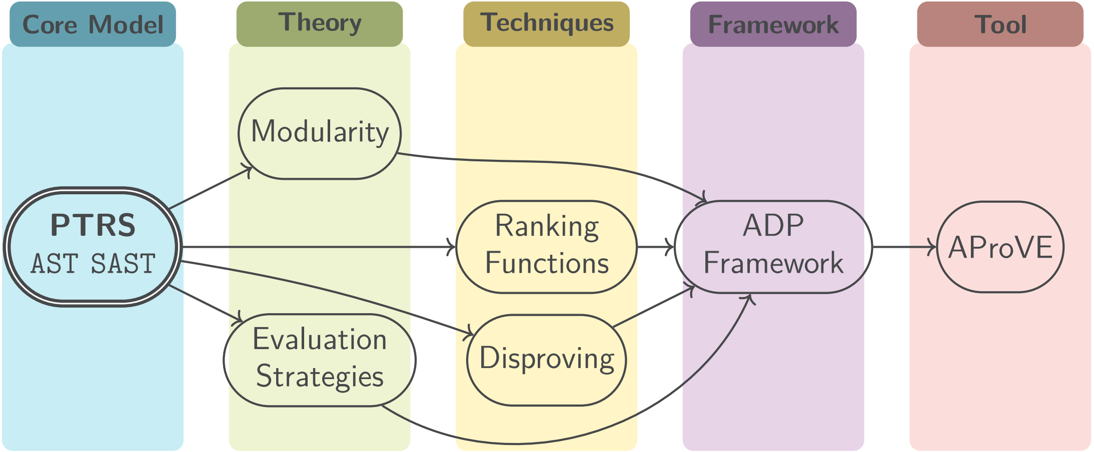
Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be PTRSs.

$$\begin{aligned}
\mathcal{R}_1, \mathcal{R}_2 \text{ are } \mathtt{iAST} &\iff \mathcal{R}_1 \cup \mathcal{R}_2 \text{ is } \mathtt{iAST} \text{ for disjoint unions,}\\
\mathcal{R}_1, \mathcal{R}_2 \text{ are } \mathtt{iAST} &\iff \mathcal{R}_1 \cup \mathcal{R}_2 \text{ is } \mathtt{iAST} \text{ for shared constructor unions,}\\
\mathcal{R}_1, \mathcal{R}_2 \text{ are } \mathtt{iSAST} &\iff \mathcal{R}_1 \cup \mathcal{R}_2 \text{ is } \mathtt{iSAST} \text{ for disjoint unions,}\\
&\ldots
\end{aligned}$$

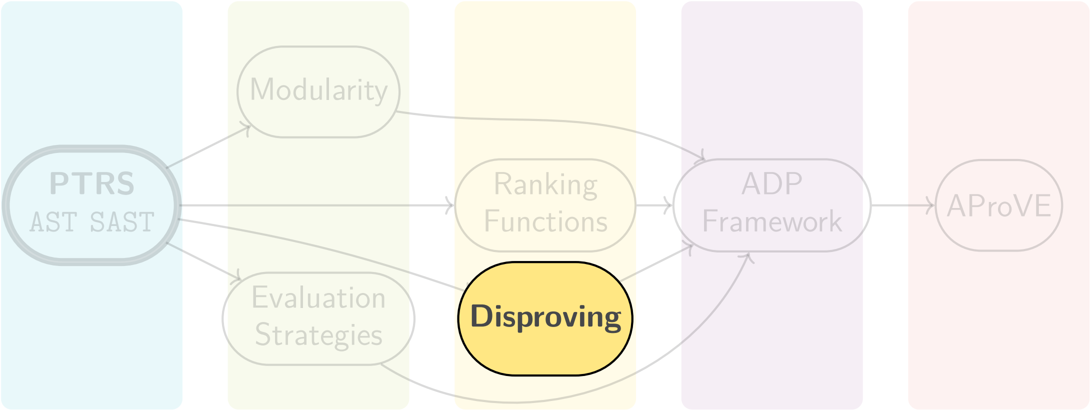
Modularity Results	iAST	iSAST
Disjoint Unions	Yes	Yes
Shared Constructor Unions	Yes	No

[KG25a] J.-C. Kassing and J. Giesl. From Innermost to Full Probabilistic Term Rewriting: AST, Complexity, and Modularity (LMCS 2025)

How to Automatically Analyze Probabilistic Programs?



Disproving


PTRS
AST SAST
Modularity
Evaluation
Strategies
Ranking
Functions
Disproving
ADP
Framework
AProVE

Disproving AST and SAST of a PTRS

Disproving Termination of a TRS:

Disproving AST and SAST of a PTRS

Disproving Termination of a TRS: Find t, context C, and substitution σ s. t.

Disproving AST and SAST of a PTRS

Disproving Termination of a TRS: Find t, context C, and substitution σ s. t.

$$\begin{array}{c} t \\ \downarrow \\ C[t\sigma] \end{array}$$

Disproving AST and SAST of a PTRS

Disproving Termination of a TRS: Find t, context C, and substitution σ s. t.

$$\begin{array}{c} t \\ \downarrow \\ C[t\sigma] \\ \downarrow \\ C[C[t\sigma]\sigma] \\ \downarrow \\ \ldots \end{array}$$

Disproving AST and SAST of a PTRS

Disproving Termination of a TRS: Find t, context C, and substitution σ s. t.

$$t \to C[t\sigma] \to C[C[t\sigma]\sigma] \to \ldots$$

$\mathcal{R}'_{\mathsf{f}}$:

$$\mathsf{f}(0) \to 0$$

$$\mathsf{f}(\mathtt{s}(\mathtt{x})) \to \mathtt{s}(\mathsf{f}(\mathtt{s}(\mathtt{s}(\mathtt{x}))))$$

Disproving AST and SAST of a PTRS

Disproving Termination of a TRS: Find t, context C, and substitution σ s. t.

$$\begin{array}{ccc} t & \qquad & \mathsf{f}(\mathtt{s}(\mathtt{x})) \\ \downarrow & & \downarrow \\ C[t\sigma] & & \mathtt{s}(\mathsf{f}(\mathtt{s}(\mathtt{s}(\mathtt{x})))) \\ \downarrow & & \downarrow \\ C[C[t\sigma]\sigma] & & \mathtt{s}(\mathtt{s}(\mathsf{f}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{x})))))) \\ \downarrow & & \downarrow \\ \dots & & \dots \end{array}$$

$\mathcal{R}'_{\mathsf{f}}$:

$$\begin{aligned} \mathsf{f}(0) &\to 0 \\ \mathsf{f}(\mathtt{s}(\mathtt{x})) &\to \mathtt{s}(\mathsf{f}(\mathtt{s}(\mathtt{s}(\mathtt{x})))) \end{aligned}$$

Disproving AST and SAST of a PTRS

Disproving Termination of a TRS: Find t, context C, and substitution σ s. t.

$$t \to C[t\sigma] \to C[C[t\sigma]\sigma] \to \ldots$$

$$\mathsf{f}(\mathtt{s}(\mathtt{x})) \to \mathtt{s}(\mathsf{f}(\mathtt{s}(\mathtt{s}(\mathtt{x})))) \to \mathtt{s}(\mathtt{s}(\mathsf{f}(\mathtt{s}(\mathtt{s}(\mathtt{s}(\mathtt{x})))))) \to \ldots$$

$\mathcal{R}'_{\mathsf{f}}$:

$$\mathsf{f}(\mathsf{0}) \to \mathsf{0}$$

$$\mathsf{f}(\mathtt{s}(\mathtt{x})) \to \mathtt{s}(\mathsf{f}(\mathtt{s}(\mathtt{s}(\mathtt{x}))))$$

Disproving AST and SAST of a PTRS

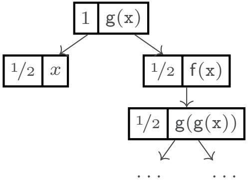
Disproving (S)AST of a PTRS[KNSG26]**:** Try to embed random walks:

[KNSG26] Kassing et al. Disproving (Positive) Almost-Sure Termination [...] via Random Walks (IJCAR 2026)

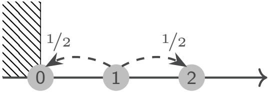
Disproving AST and SAST of a PTRS

Disproving (S)AST of a PTRS[KNSG26]: Try to embed random walks:

$\mathcal{R}$ Computation



Symmetric Random Walk μ



$\mu(-1) = 1/2$

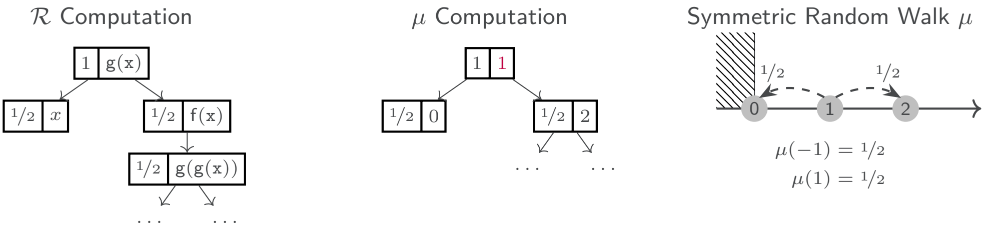
$\mu(1) = 1/2$

- What does it mean to find a random walk?

[KNSG26] Kassing et al. Disproving (Positive) Almost-Sure Termination [...] via Random Walks (IJCAR 2026)

Disproving AST and SAST of a PTRS

Disproving (S)AST of a PTRS[KNSG26]**:** Try to embed random walks:

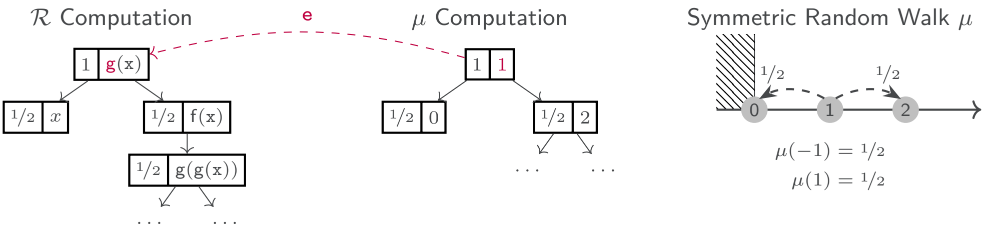


- What does it mean to find a random walk?

[KNSG26] Kassing et al. Disproving (Positive) Almost-Sure Termination [...] via Random Walks (IJCAR 2026)

Disproving AST and SAST of a PTRS

Disproving (S)AST of a PTRS[KNSG26]**:** Try to embed random walks:

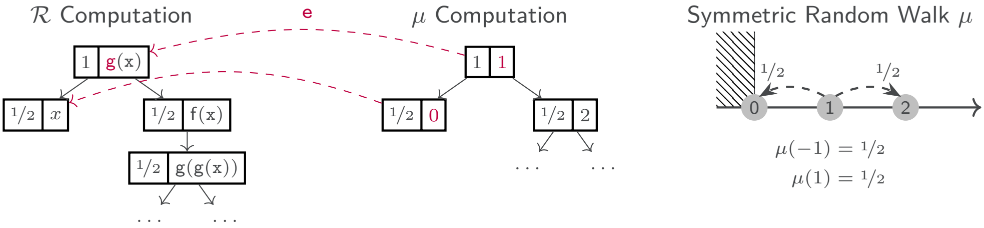


- What does it mean to find a random walk?

[KNSG26] Kassing et al. Disproving (Positive) Almost-Sure Termination [...] via Random Walks (IJCAR 2026)

Disproving AST and SAST of a PTRS

Disproving (S)AST of a PTRS[KNSG26]: Try to embed random walks:

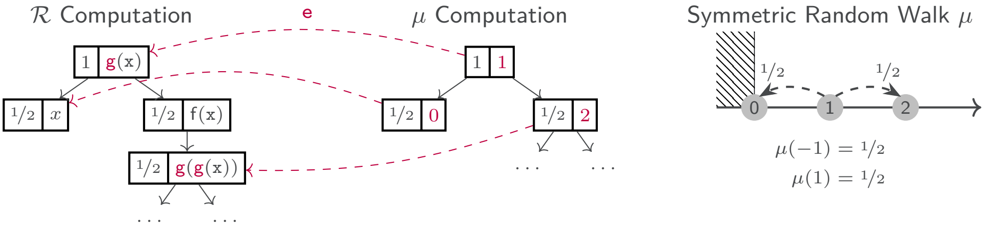


- What does it mean to find a random walk?

[KNSG26] Kassing et al. Disproving (Positive) Almost-Sure Termination [...] via Random Walks (IJCAR 2026)

Disproving AST and SAST of a PTRS

Disproving (S)AST of a PTRS[KNSG26]**:** Try to embed random walks:

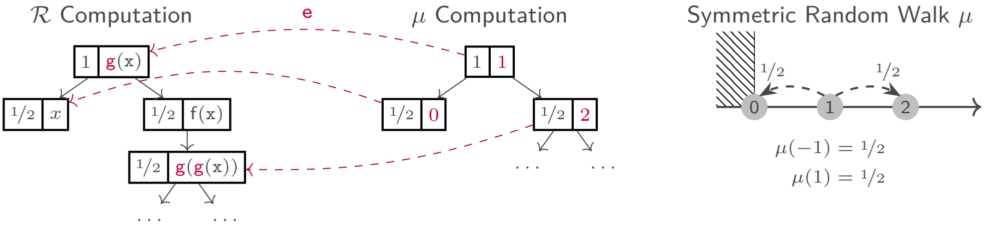


- What does it mean to find a random walk?

[KNSG26] Kassing et al. Disproving (Positive) Almost-Sure Termination [...] via Random Walks (IJCAR 2026)

Disproving AST and SAST of a PTRS

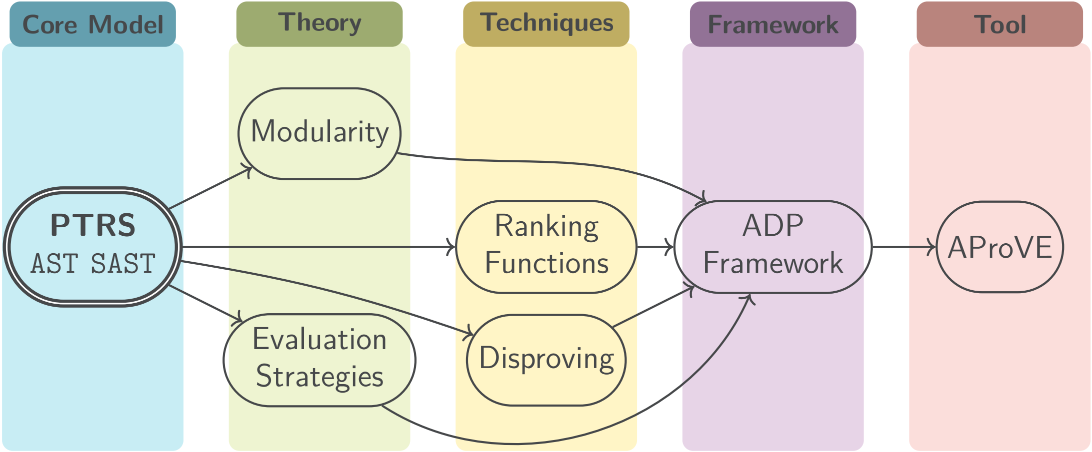
Disproving (S)AST of a PTRS[KNSG26]**:** Try to embed random walks:



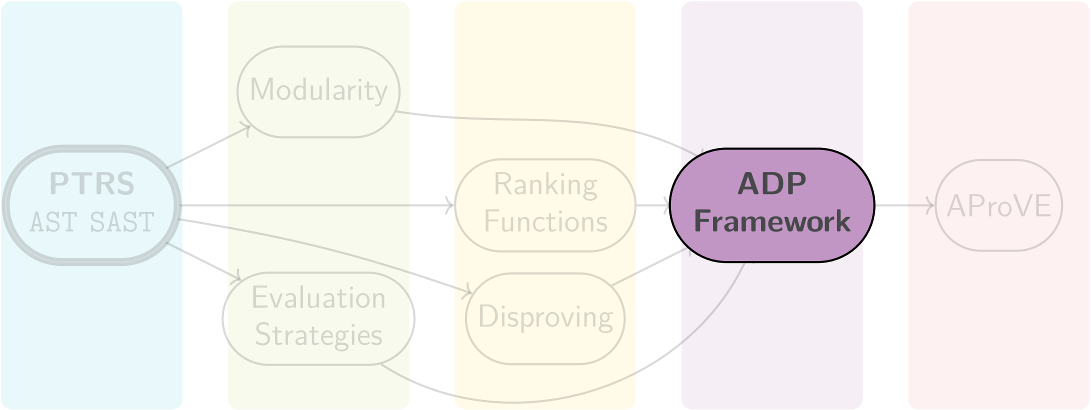
- What does it mean to find a random walk?
  $\rightsquigarrow$ Embedding e from computation of μ to computation of $\mathcal{R}$

[KNSG26] Kassing et al. Disproving (Positive) Almost-Sure Termination [...] via Random Walks (IJCAR 2026)

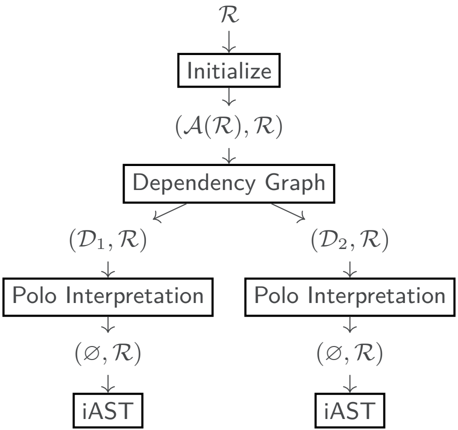
How to Automatically Analyze Probabilistic Programs?



ADP Framework



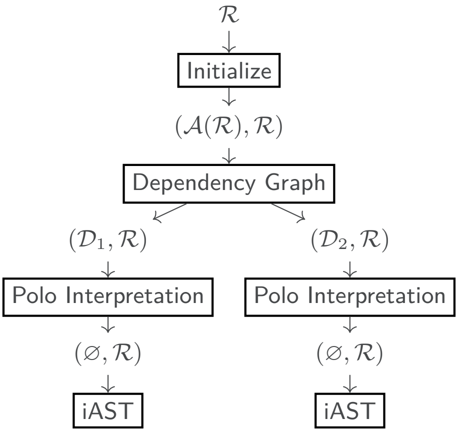
Probabilistic Annotated Dependency Pair Framework



ADP framework for *innermost* AST of PTRSs[KDG24][KG25b]

[KG25b] J.-C. Kassing and J. Giesl. The Annotated Dependency Pair Framework [...] (SCICO 2025)
[KDG24] J.-C. Kassing, S. Dollase, and J. Giesl. A Complete Dependency Pair Framework [...] (FLOPS 2024)

Probabilistic Annotated Dependency Pair Framework



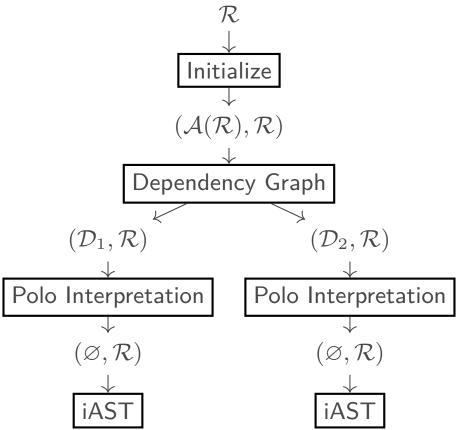
ADP framework for *innermost* AST of PTRSs[KDG24][KG25b]

- allows for modular termination proofs

[KG25b] J.-C. Kassing and J. Giesl. The Annotated Dependency Pair Framework [...] (SCICO 2025)

[KDG24] J.-C. Kassing, S. Dollase, and J. Giesl. A Complete Dependency Pair Framework [...] (FLOPS 2024)

Probabilistic Annotated Dependency Pair Framework

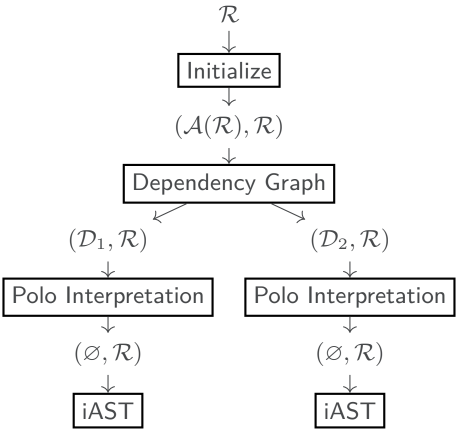


ADP framework for *innermost* `AST` of PTRSs[KDG24][KG25b]

- allows for modular termination proofs
- focus on innermost evaluation

[KG25b] J.-C. Kassing and J. Giesl. The Annotated Dependency Pair Framework [...] (SCICO 2025)
[KDG24] J.-C. Kassing, S. Dollase, and J. Giesl. A Complete Dependency Pair Framework [...] (FLOPS 2024)

Probabilistic Annotated Dependency Pair Framework

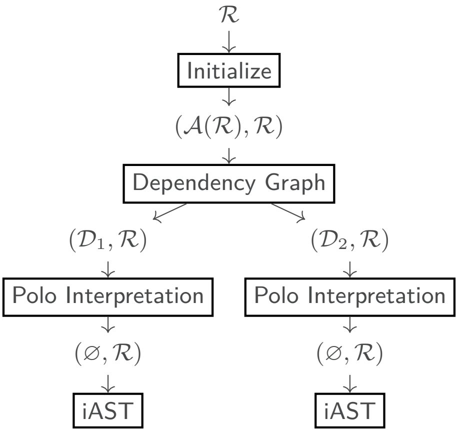


ADP framework for *innermost* AST of PTRSs[KDG24][KG25b]

- allows for modular termination proofs
- focus on innermost evaluation
- developed multiple different processors
  - Dependency Graph Processor (Modularity)
  - Reduction Pair Processor (Ranking Functions)
  - Usable Rules Processor
  - Probability Removal Processor
  - Transformational Processors
  - ...

[KG25b] J.-C. Kassing and J. Giesl. The Annotated Dependency Pair Framework [...] (SCICO 2025)
[KDG24] J.-C. Kassing, S. Dollase, and J. Giesl. A Complete Dependency Pair Framework [...] (FLOPS 2024)

Probabilistic Annotated Dependency Pair Framework



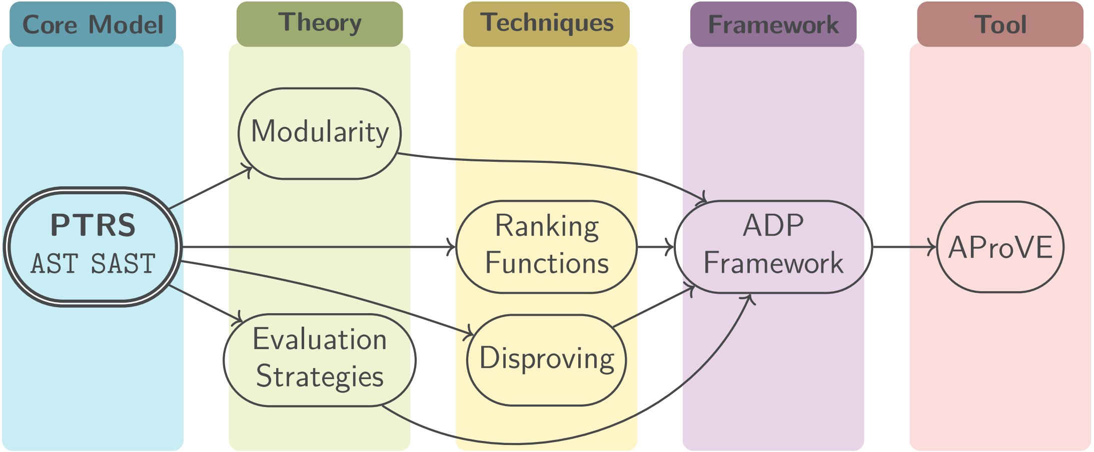
ADP framework for *innermost* AST of PTRSs[KDG24][KG25b]

- allows for modular termination proofs
- focus on innermost evaluation
- developed multiple different processors
  - Dependency Graph Processor (Modularity)
  - Reduction Pair Processor (Ranking Functions)
  - Usable Rules Processor
  - Probability Removal Processor
  - Transformational Processors
  - ...

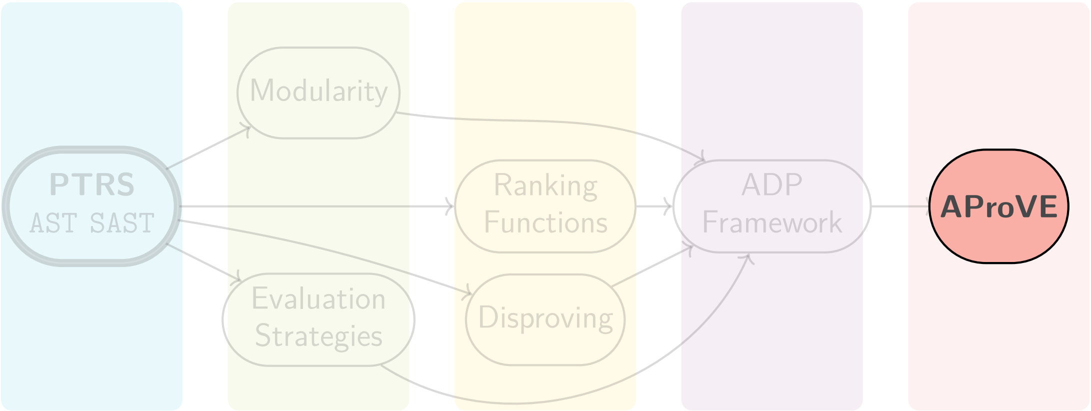
ADP framework for *innermost* SAST[KSG25]

[KSG25] J.-C. Kassing, L. Spitzer, and J. Giesl. Dependency Pairs for Expected Innermost Runtime Complexity [...] (PPDP 2025)
[KG25b] J.-C. Kassing and J. Giesl. The Annotated Dependency Pair Framework [...] (SCICO 2025)
[KDG24] J.-C. Kassing, S. Dollase, and J. Giesl. A Complete Dependency Pair Framework [...] (FLOPS 2024)

How to Automatically Analyze Probabilistic Programs?



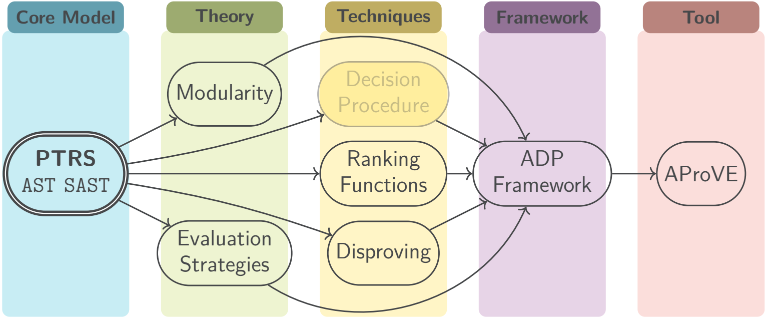
AProVE



PTRS
AST SAST
Modularity
Evaluation
Strategies
Ranking
Functions
Disproving
ADP
Framework
AProVE

AProVE

Conclusion



Annual Termination Competition Results for PTRS

Category	AST	SAST
Innermost Evaluation Strategy	112/141	44/141
Arbitrary Evaluation Strategy	60/141	34/141