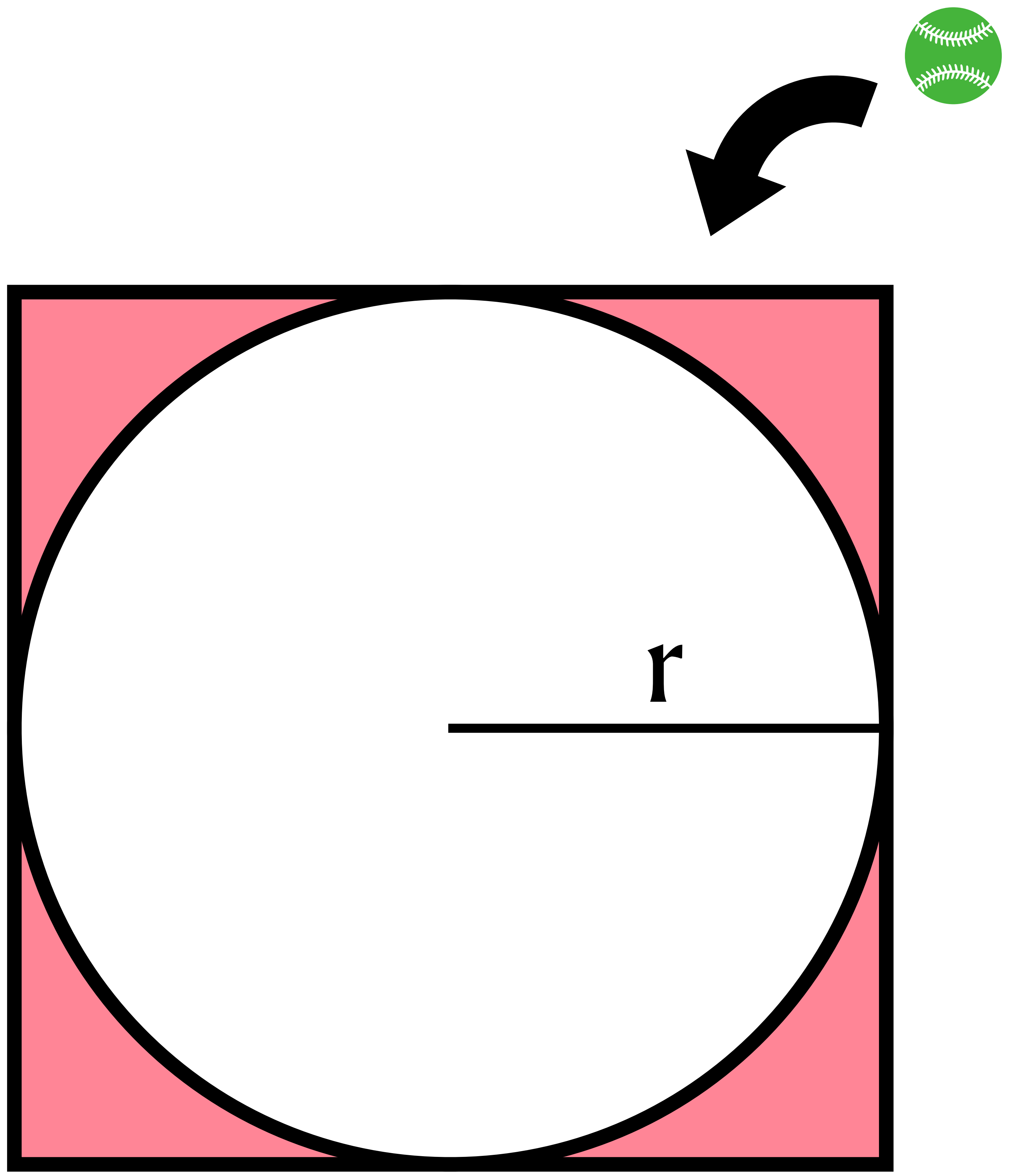
MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

Statistical Reachability Analysis

Seongmin Lee Marcel Böhme
Max Planck Institute for Security and Privacy (MPI-SP)

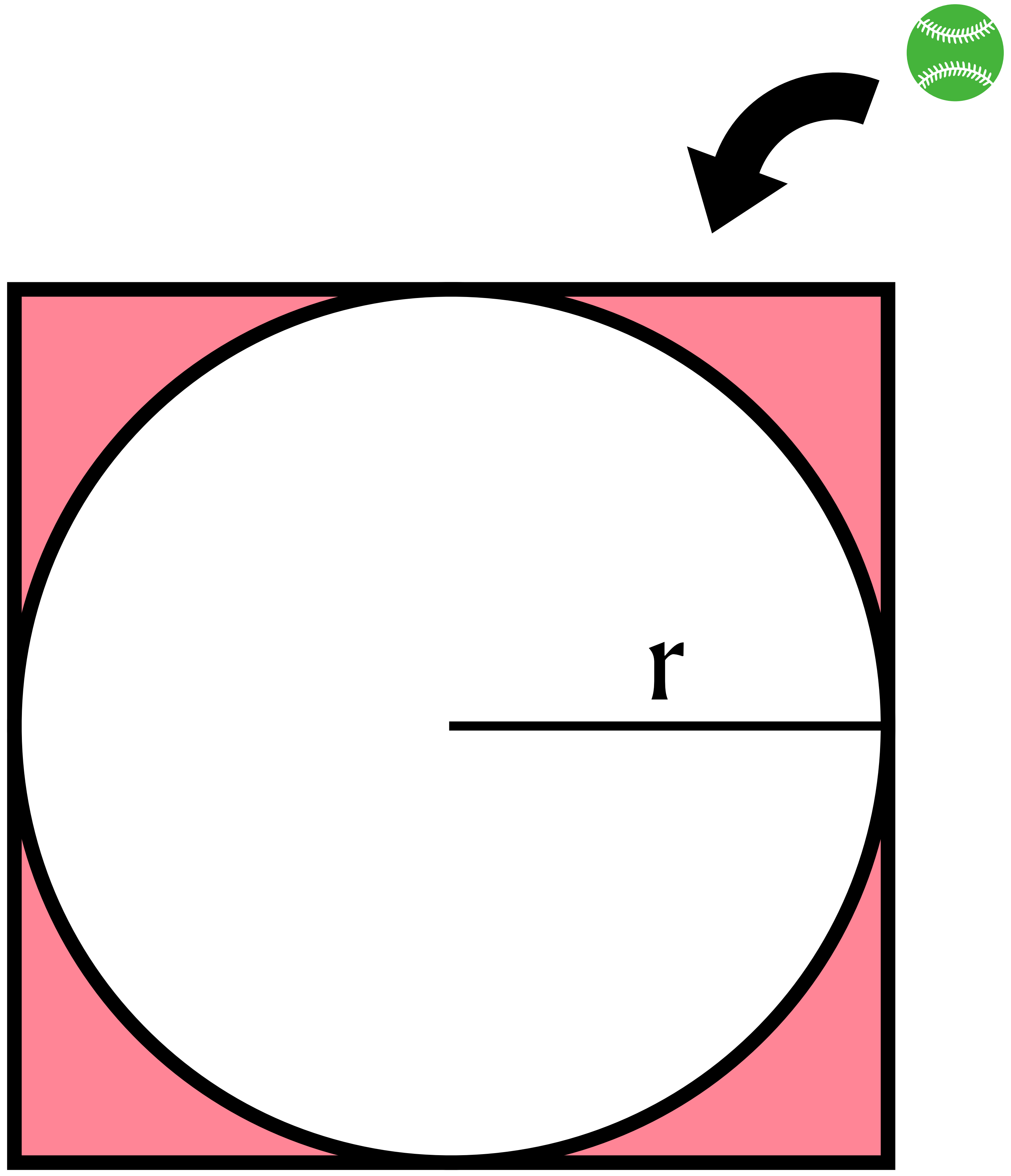
FSE 2023

Q. What is the probability of a thrown ⚾ ball to the 🟥 square dropped not into the ⚪ circle?



$$P(\neg \text{in circle}) = ?$$

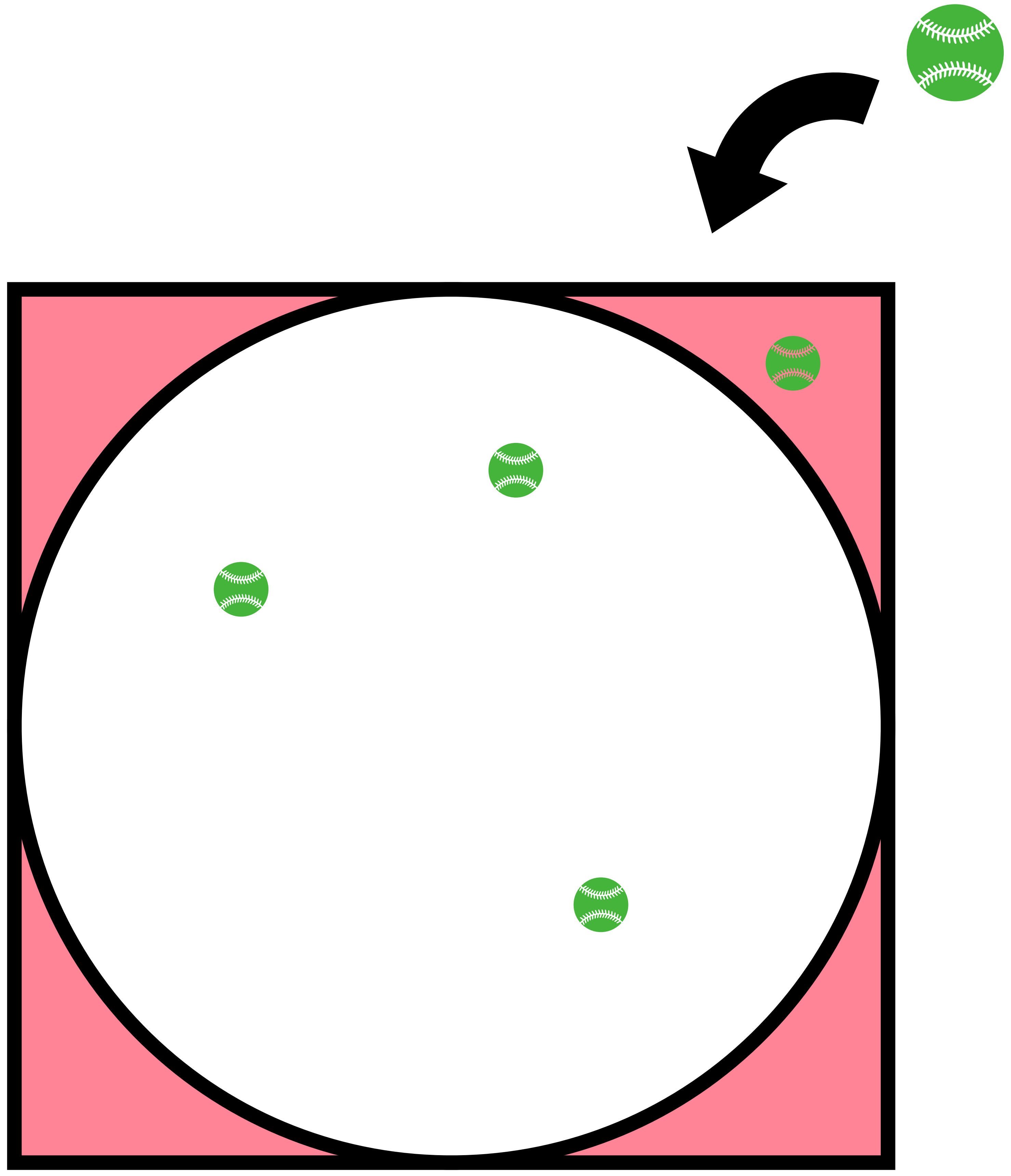
Q. What is the probability of a thrown 🥎 ball to the 🟥 square dropped not into the ⚪ circle?



Analytic approach

$$
\begin{aligned}
&P(\neg \text{in circle}) \\
&= \frac{Area(\text{Square}) - Area(\text{Circle})}{Area(\text{square})} \\
&= \frac{(2r)^2 - \pi r^2}{(2r)^2} \\
&= \frac{4 - \pi}{4} \approx 0.2146\ldots
\end{aligned}
$$

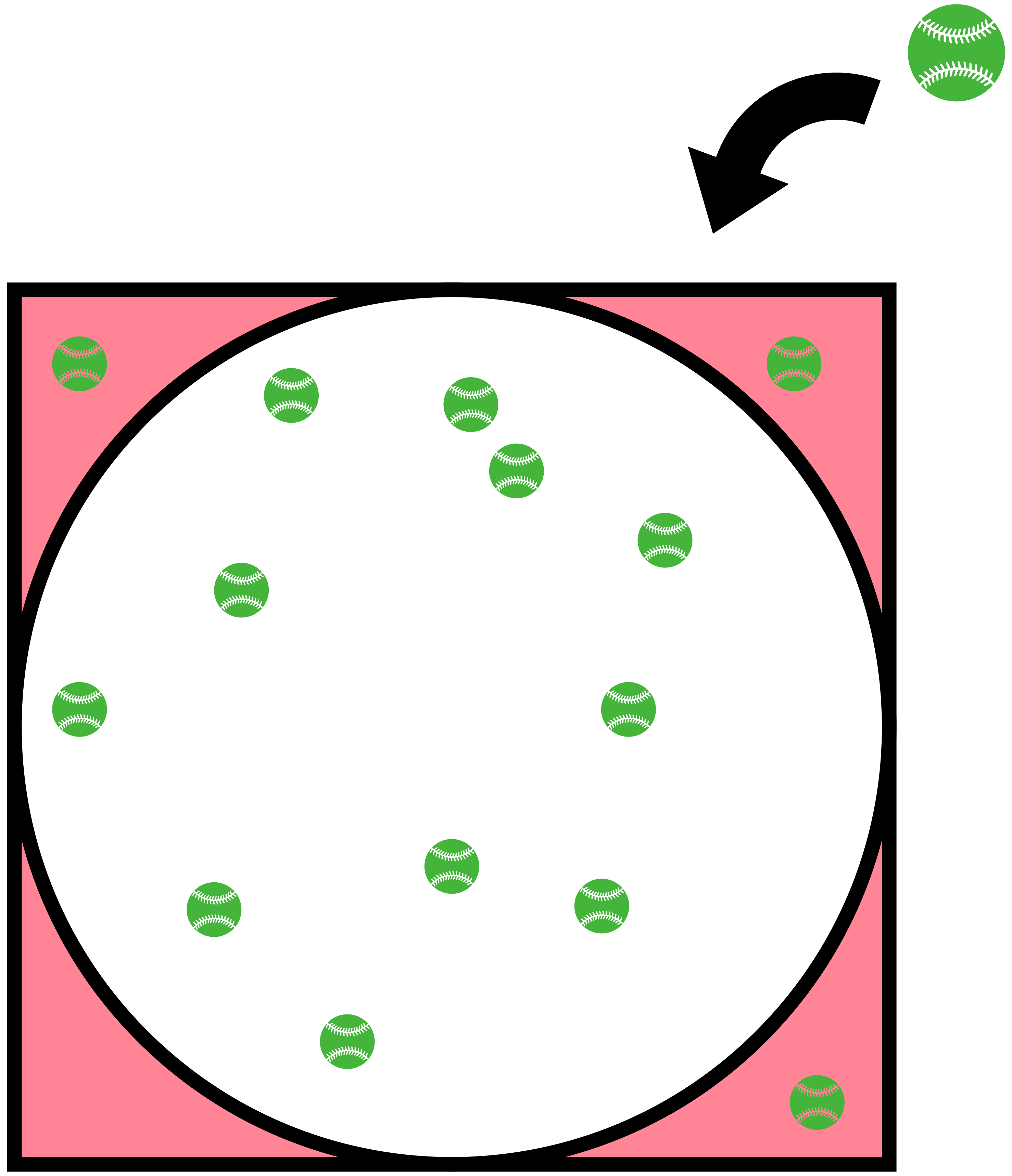
Q. What is the probability of a thrown ball to the square dropped not into the circle?



Statistical approach (e.g., Monte Carlo method)

$$P(\neg\text{in circle}) = \frac{\#\text{ of balls outside the circle}}{\#\text{ of balls thrown}} = \frac{1}{4} = 0.25$$

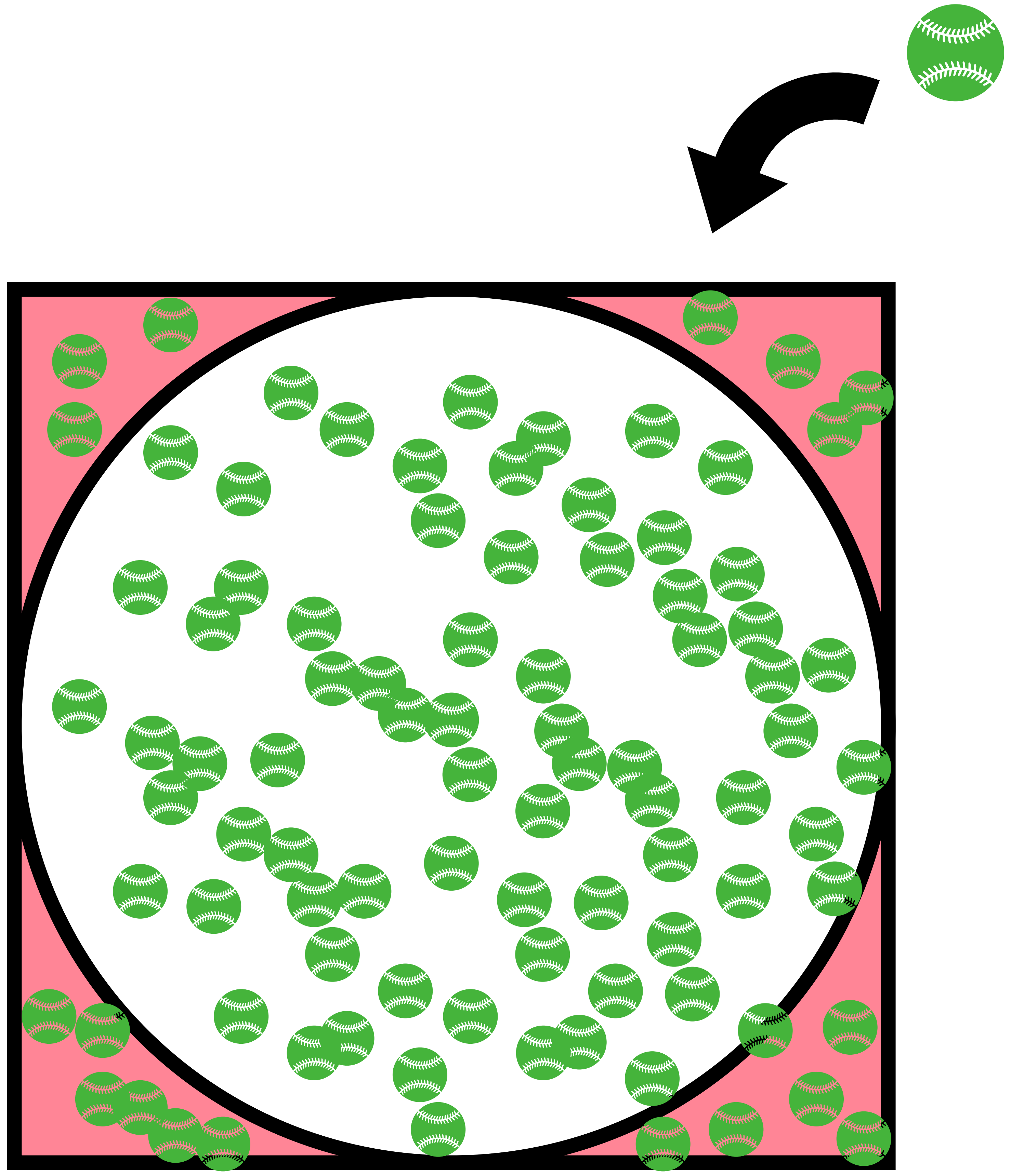
Q. What is the probability of a thrown 🥎 ball to the 🟥 square dropped not into the ⚪ circle?



Statistical approach (e.g., Monte Carlo method)

$$P(\neg \text{in circle}) = \frac{\#\text{ of balls outside the circle}}{\#\text{ of balls thrown}} = \frac{3}{14} \approx 0.2143$$

Q. What is the probability of a thrown 🥎 ball to the 🟥 square dropped not into the 🔴 circle?



Statistical approach (e.g., Monte Carlo method)

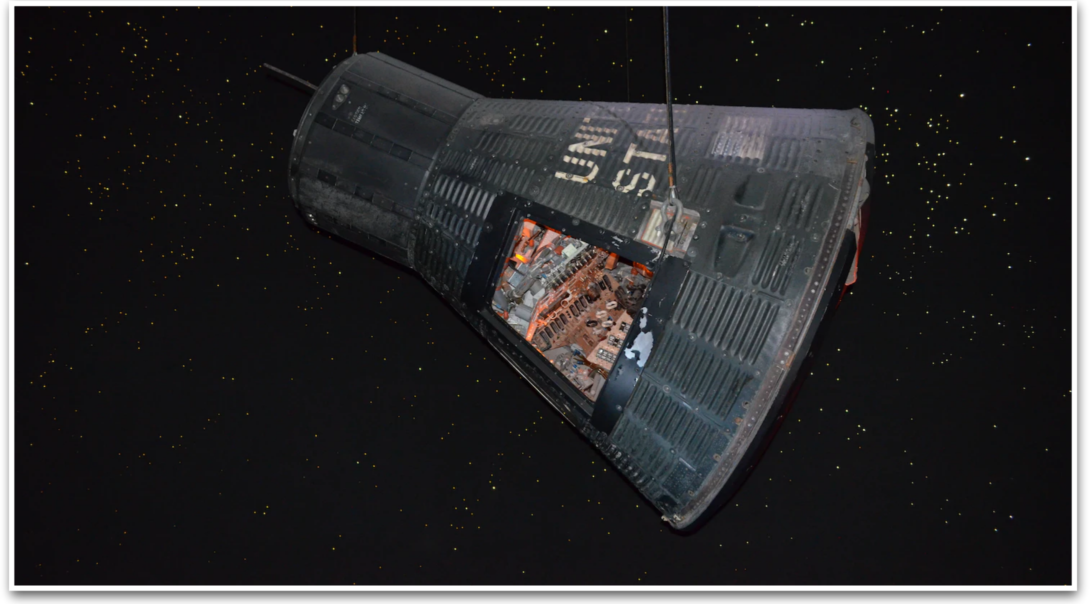
$$P(\neg \text{in circle}) = \frac{\#\text{ of balls outside the circle}}{\#\text{ of balls thrown}} = \frac{65}{303} \approx 0.2145$$

Analytic approach is precise and useful *if we know the exact model.* **However, ...**

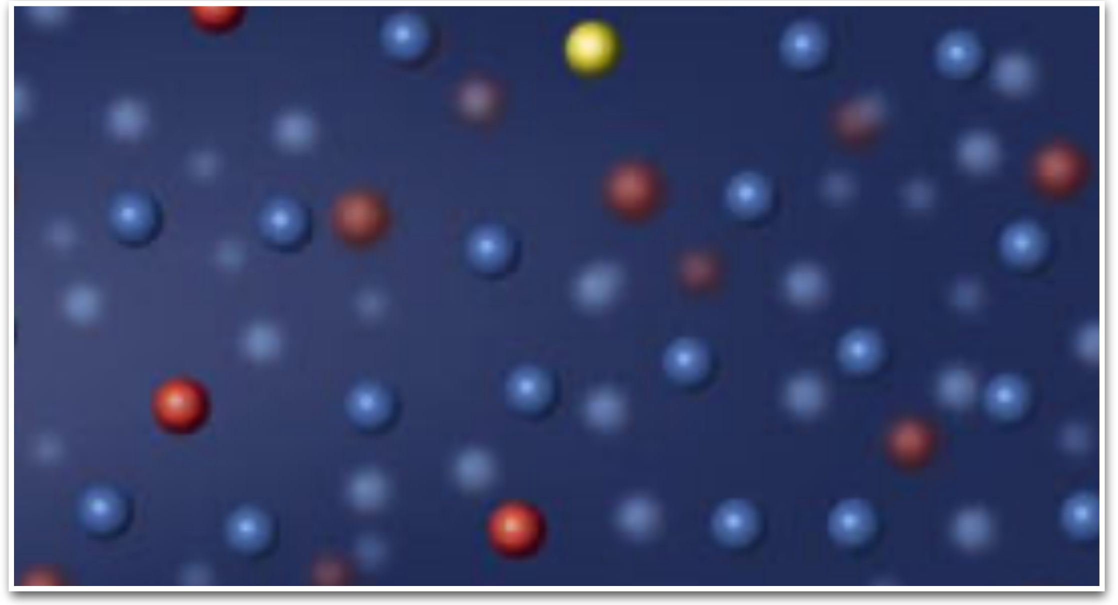
$$
\begin{aligned}
&P(\neg \text{in circle}) \\
&= \frac{Area(\text{Square}) - Area(\text{Circle})}{Area(\text{square})} \\
&= \frac{(2r)^2 - \pi r^2}{(2r)^2} \\
&= \frac{4-\pi}{4} \approx 0.2146...
\end{aligned}
$$

Spacecraft Atmosphere Entry

Spacecraft



Molecules



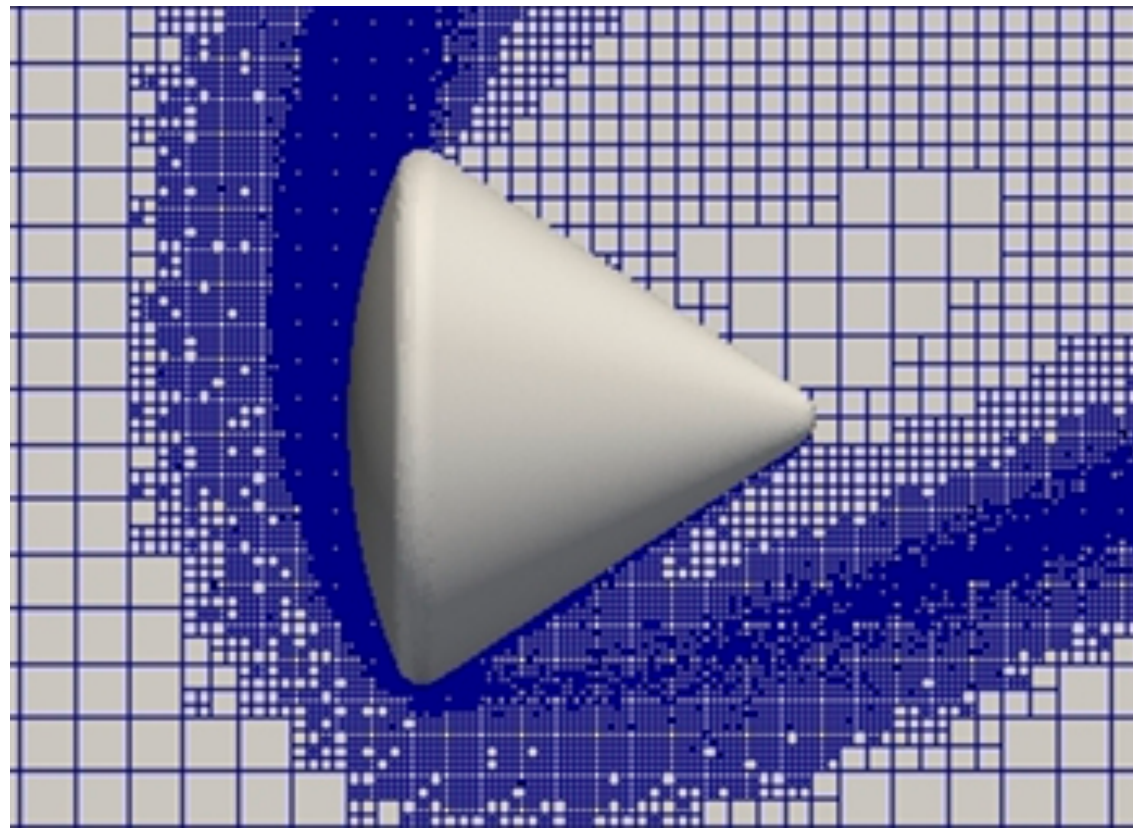
Analytically computing the interaction is nearly ***impossible!***

Analytic approach is precise and useful *if we know the exact model.* **However, ...**

$$
\begin{aligned}
&P(\neg\text{in circle}) \\
&= \frac{Area(\text{Square}) - Area(\text{Circle})}{Area(\text{square})} \\
&= \frac{(2r)^2 - \pi r^2}{(2r)^2} \\
&= \frac{4-\pi}{4} \approx 0.2146...
\end{aligned}
$$

Spacecraft Atmosphere Entry

Instead, a simulation-based ***statistical approach*** works successfully.



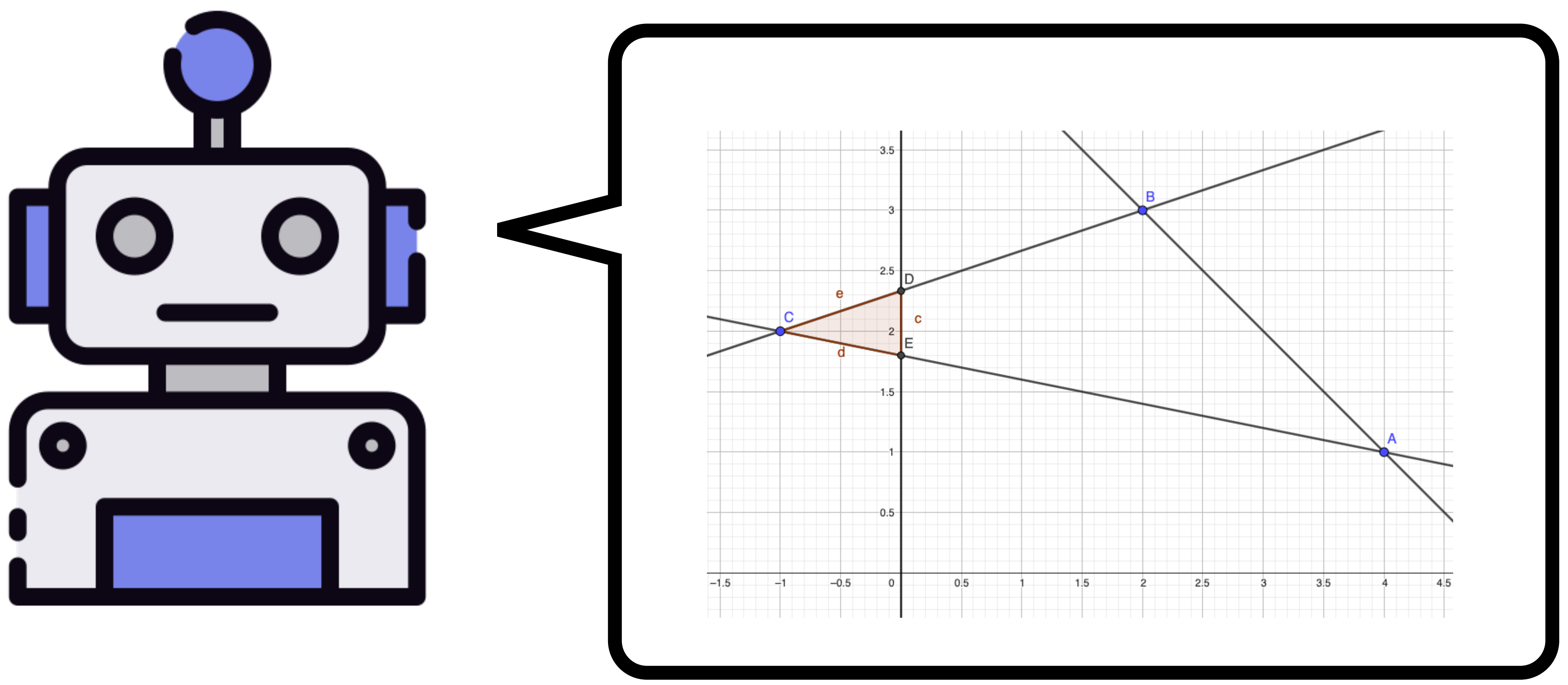
Solution: Direct Simulation Monte Carlo

Program analysis

An analytic approach for program analysis

What is the probability of a failure execution?

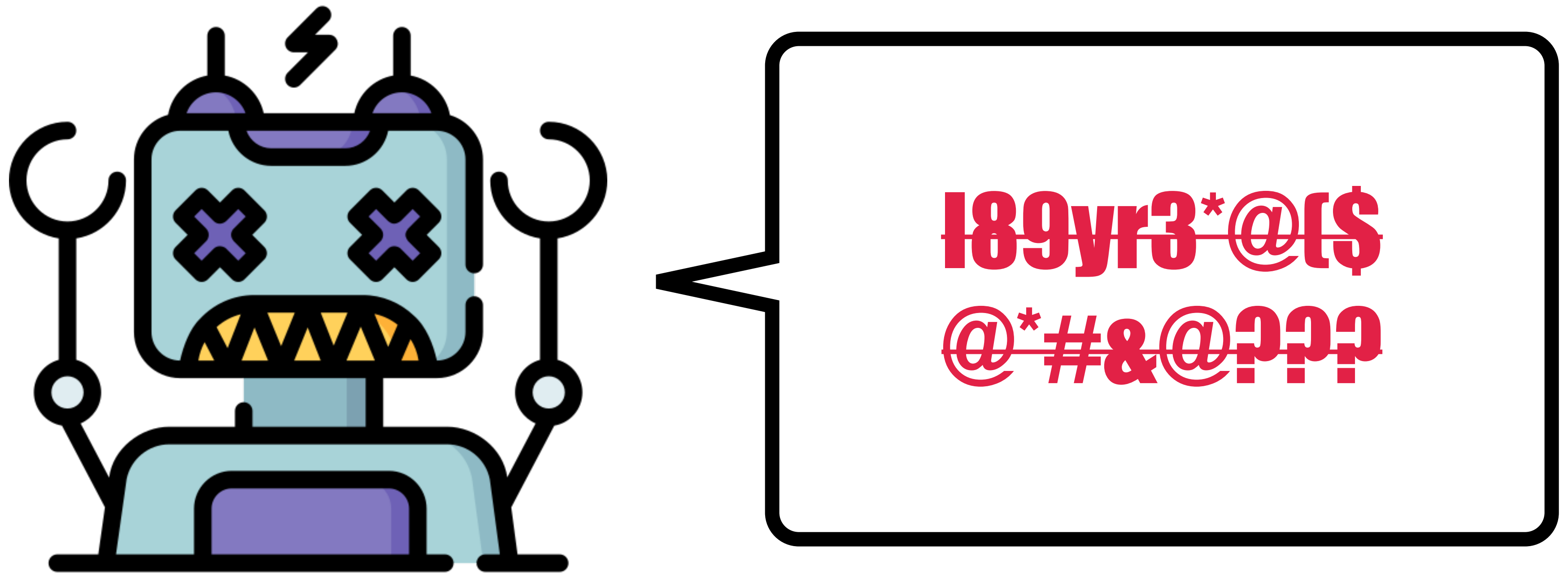
```
def f(x0, x1) {
  if (x0 + 5*x1 - 9 < 0) return;
  if (x0 + x1 -5 > 0) return;
  if (-x0 + 3x1 - 7 > 0) return;
  if (x0 > 0) return;
  assert False
}
f(input() % 5, input() % 5)
```



- Conventional approach
- Based on the formal semantics of the program
- E.g.,
  - Symbolic execution
  - Model checking / Model counting
  - Static analysis

An analytic approach for program analysis

What is the probability of a failure execution?



Analysis of the modern software faces

- an industrial scale *huge code base*
- heterogenous *in-analyzable features*,
  e.g., 3rd party/binary libraries or cross-language
- a nature of *undecidability,*

Statistical Approach for Program Analysis

A statistical method is useful when

Analysis of the modern software faces

- an industrial scale ***huge code base***
- heterogenous ***in-analyzable features,***
  e.g., 3rd party/binary libraries or cross-language
- a nature of ***undecidability,***

Statistical Approach for Program Analysis

A statistical method is useful when

- one wants an *approximation of the quantity,*

Analysis of the modern software faces

- an industrial scale *huge code base*
- heterogenous *in-analyzable features,*
  e.g., 3rd party/binary libraries or cross-language
- a nature of *undecidability,*

Statistical Approach for Program Analysis

A statistical method is useful when

- one wants an *approximation of the quantity,*
- even if the *whole system is unknown,*

Analysis of the modern software faces

- an industrial scale *huge code base*
- heterogenous *in-analyzable features,*
  e.g., 3rd party/binary libraries or cross-language
- a nature of *undecidability,*

Statistical Approach for Program Analysis

A statistical method is useful when

- one wants an *approximation of the quantity,*
- even if the *whole system is unknown,*
- getting the *samples* is convenient.

Analysis of the modern software faces

- an industrial scale *huge code base*
- heterogenous *in-analyzable features,*
  e.g., 3rd party/binary libraries or cross-language
- a nature of *undecidability,*

And,

- modern testing framework (eg. fuzzing) gives
  > 1K executions per sec.

Statistical Approach for Program Analysis

A statistical method is useful when

- one wants an *approximation of the quantity*,
- even if the *whole system is unknown*,
- getting the *samples* is convenient.

Then,

- it performs <u>*regardless of the complexity*</u> of the system.

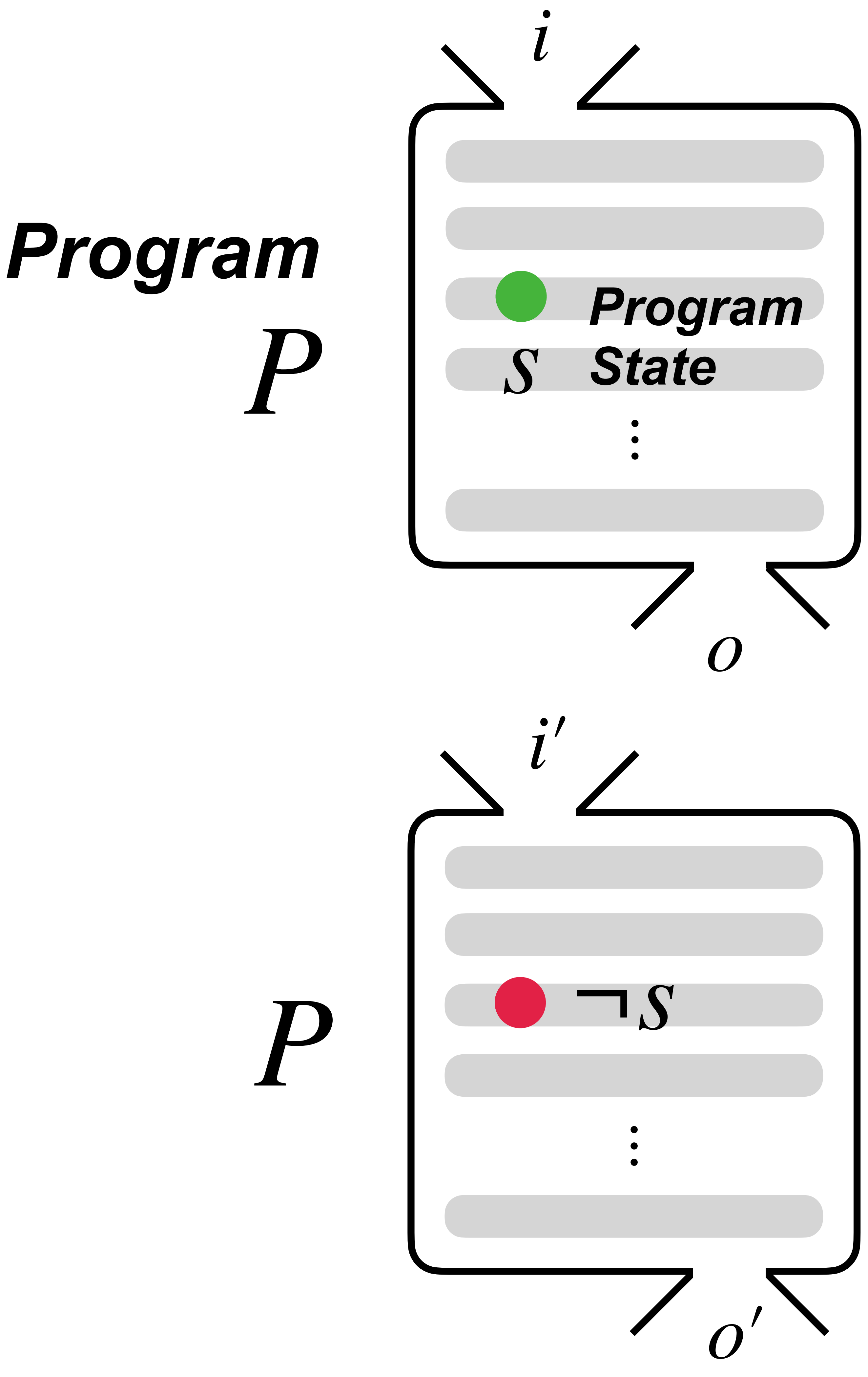
Analysis of the modern software faces

- an industrial scale <u>*huge code base*</u>
- heterogenous <u>*in-analyzable features,*</u>
  e.g., 3rd party/binary libraries or cross-language
- a nature of <u>*undecidability,*</u>

And,

- modern testing framework (eg. fuzzing) gives
  > *1K executions per sec.*

Quantitative Reachability Analysis (QRA)



A *program state* is a property one is interested in that is either reached or unreached, given the program execution.

Quantitative Reachability Analysis (QRA) measures the probability of how likely a certain program state is reached given the workload of the program.

$$\Pr(s) = \sum_{e \in E} \Pr(e) \cdot \mathbf{1}(s \text{ is reached by } e)$$

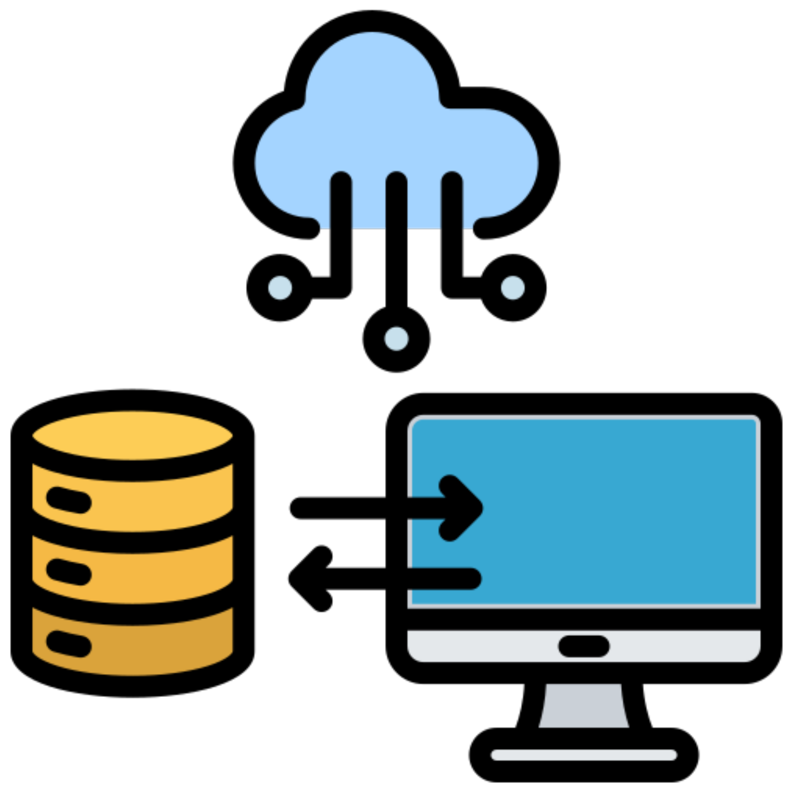
E: workload or execution profile

Quantitative Reachability Analysis (QRA)

Software Testing

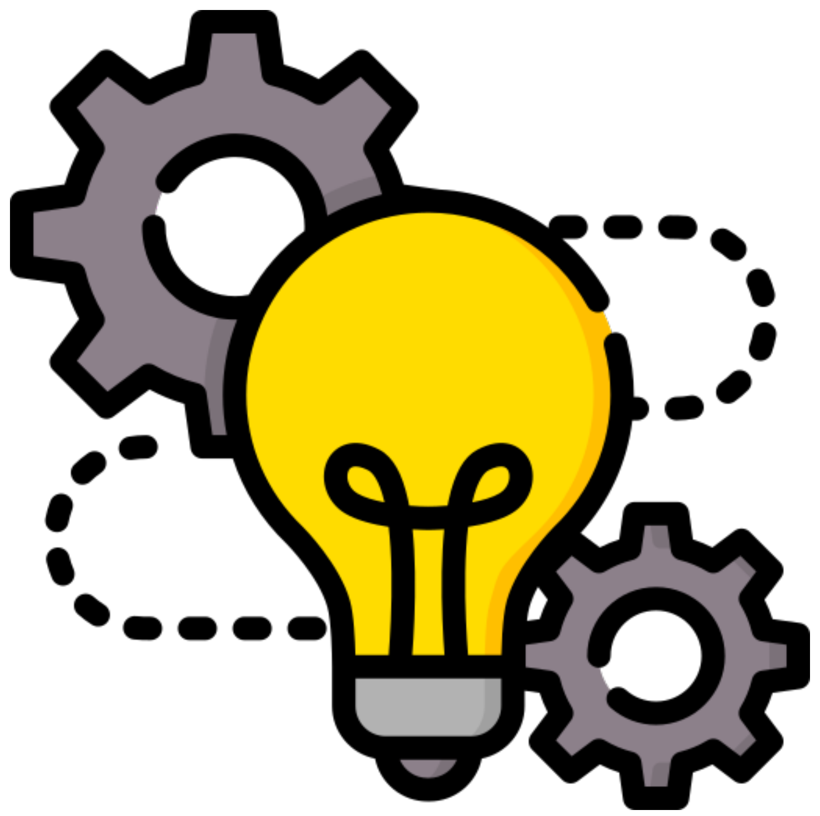
How often the potentially vulnerable method is executed?

Resource Management



How often is the resource requested?

Other SE technology

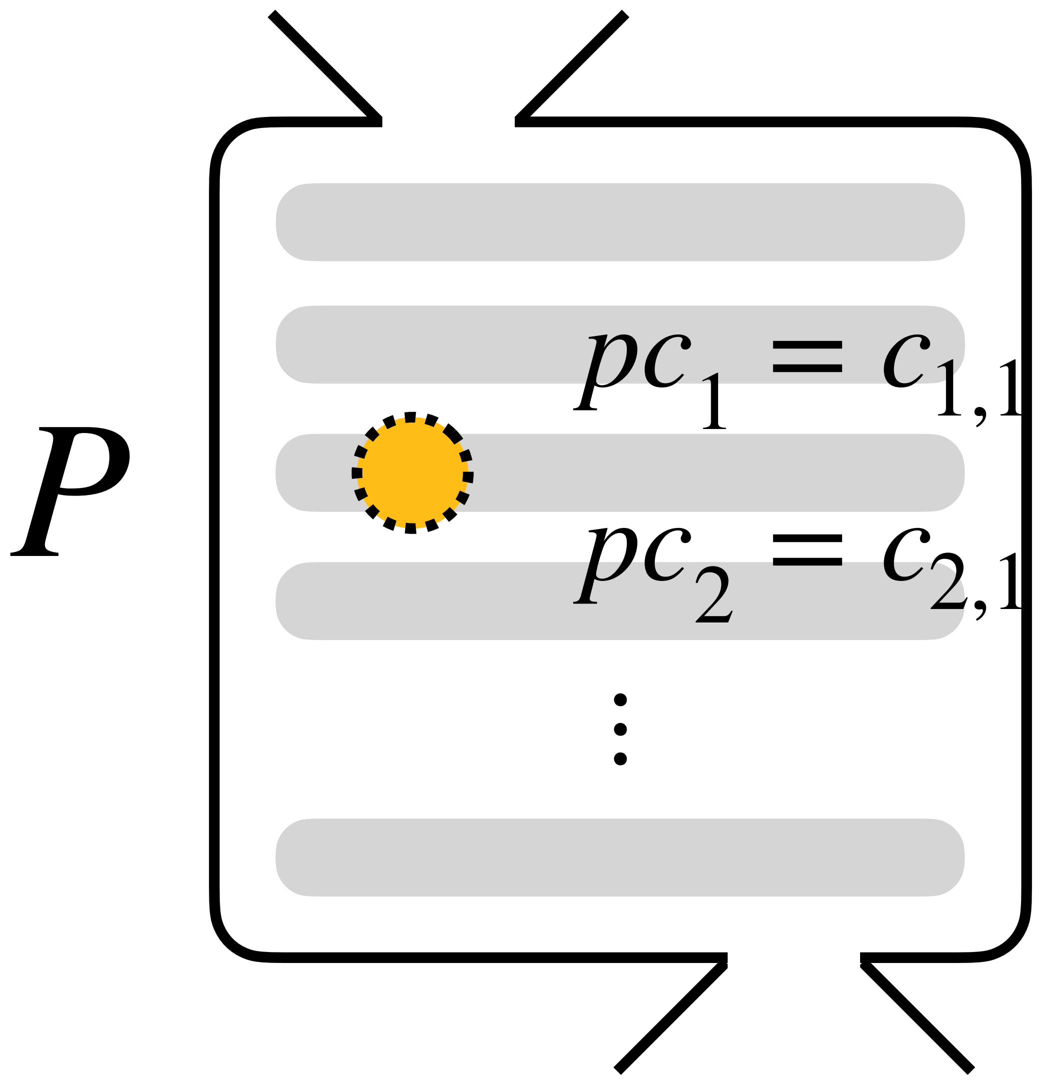


Ensemble testing model of fuzzing & symbolic execution

Existing Method — Analytic Approach

Probabilistic Symbolic Execution (PSE), *Geldenhuys et al. 2012*

$sym_1, sym_2, \cdots$



$pc_1 = c_{1,1} \wedge c_{1,2} \wedge \cdots$

$pc_2 = c_{2,1} \wedge c_{2,2} \wedge \cdots$

$\vdots$

Symbolic Execution

pc_1

pc_2

$\vdots$

Model Counter

$\#(pc_1) = n_1 \qquad \Pr(pc_1) = \dfrac{n_1}{|E|}$

$\#(pc_2) = n_2 \qquad \Pr(pc_2) = \dfrac{n_2}{|E|}$

$\vdots$

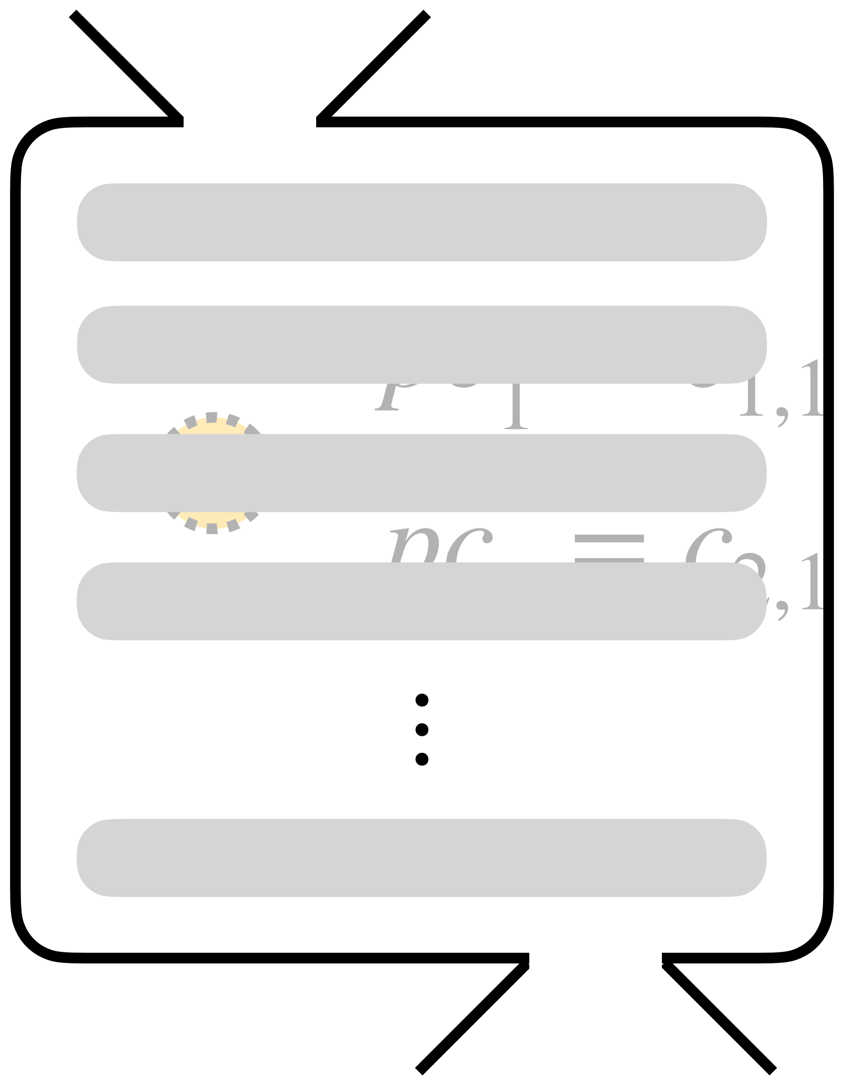
$$\Pr(s) = \Pr(pc_1) + \Pr(pc_2) + \cdots$$

Existing Method — Analytic Approach

Probabilistic Symbolic Execution (PSE), *Geldenhuys et al. 2012*

Limitation 1. Need the system & Model

P



$pc_1 = c_{1,1} \wedge c_{1,2} \wedge \cdots$

$pc_2 = c_{2,1} \wedge c_{2,2} \wedge \cdots$

Symbolic Execution

pc_1

pc_2

$\vdots$

Model Counter

$\#(pc_1) = n_1$ $\quad \Pr(pc_1) = \frac{n_1}{|E|}$

$\#(pc_2) = n_2$ $\quad \Pr(pc_2) = \frac{n_2}{|E|}$

$\vdots$

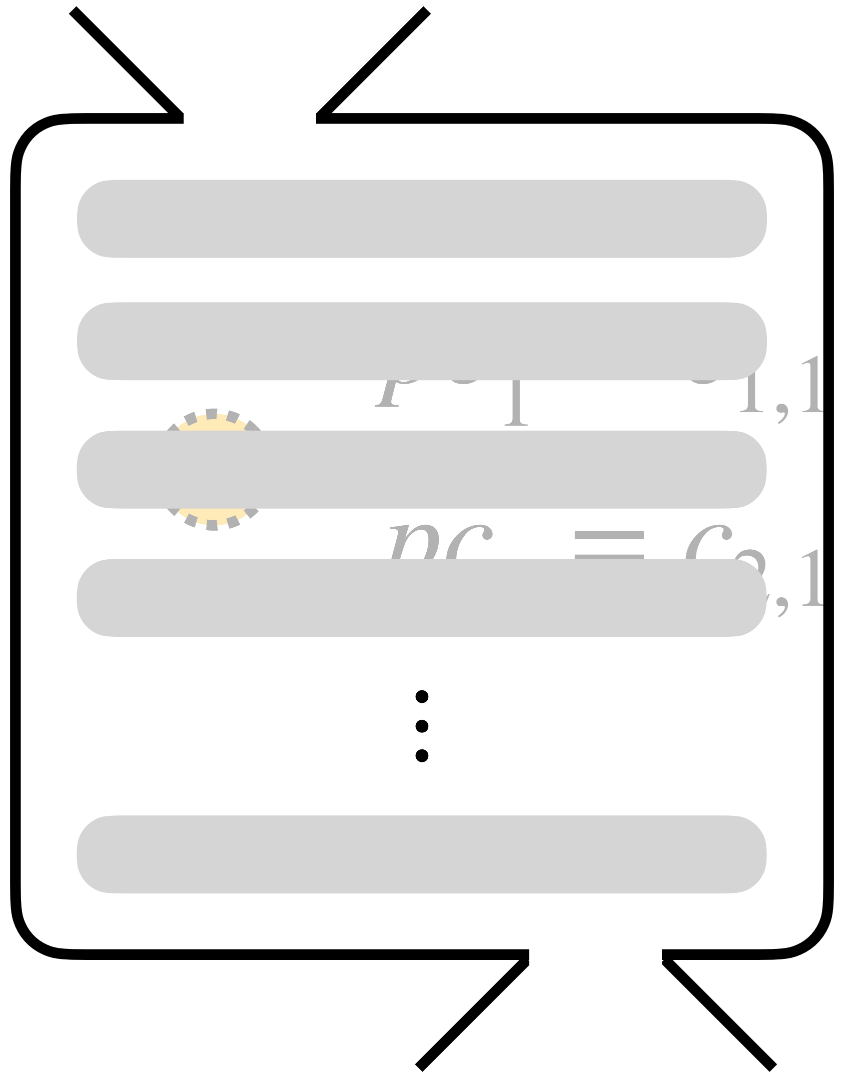
$$\Pr(s) = \Pr(pc_1) + \Pr(pc_2) + \cdots$$

Existing Method — Analytic Approach

Probabilistic Symbolic Execution (PSE), *Geldenhuys et al. 2012*

Limitation 1. Need the system & Model

P



$pc_1 = c_{1,1} \wedge c_{1,2} \wedge \cdots$

$pc_2 = c_{2,1} \wedge c_{2,2} \wedge \cdots$

Symbolic Execution

pc_1

pc_2

$\vdots$

Model Counter

$\#(pc_1) = n_1 \qquad \Pr(pc_1) = \frac{n_1}{|E|}$

$\#(pc_2) = n_2 \qquad \Pr(pc_2) = \frac{n_2}{|E|}$

$\vdots$

$$\Pr(s) = \Pr(pc_1) + \Pr(pc_2) + \cdots$$

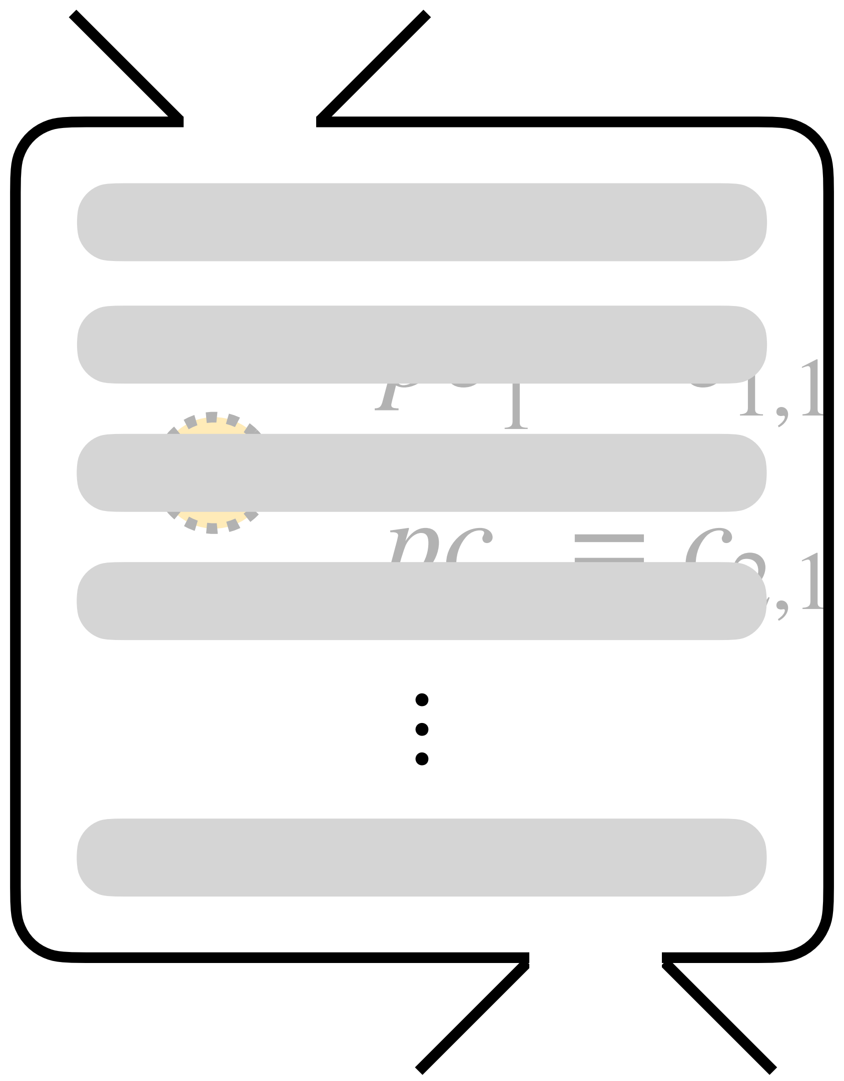
Limitation 2. Path Explosion

Existing Method — Analytic Approach

Probabilistic Symbolic Execution (PSE), *Geldenhuys et al. 2012*

Limitation 1. Need the system & Model

P



$pc_1 = c_{1,1} \wedge c_{1,2} \wedge \cdots$

$pc_2 = c_{2,1} \wedge c_{2,2} \wedge \cdots$

Symbolic Execution

pc_1

pc_2

$\vdots$

Model Counter

$\#(pc_1) = n_1 \qquad \Pr(pc_1) = \frac{n_1}{|E|}$

$\#(pc_2) = n_2 \qquad \Pr(pc_2) = \frac{n_2}{|E|}$

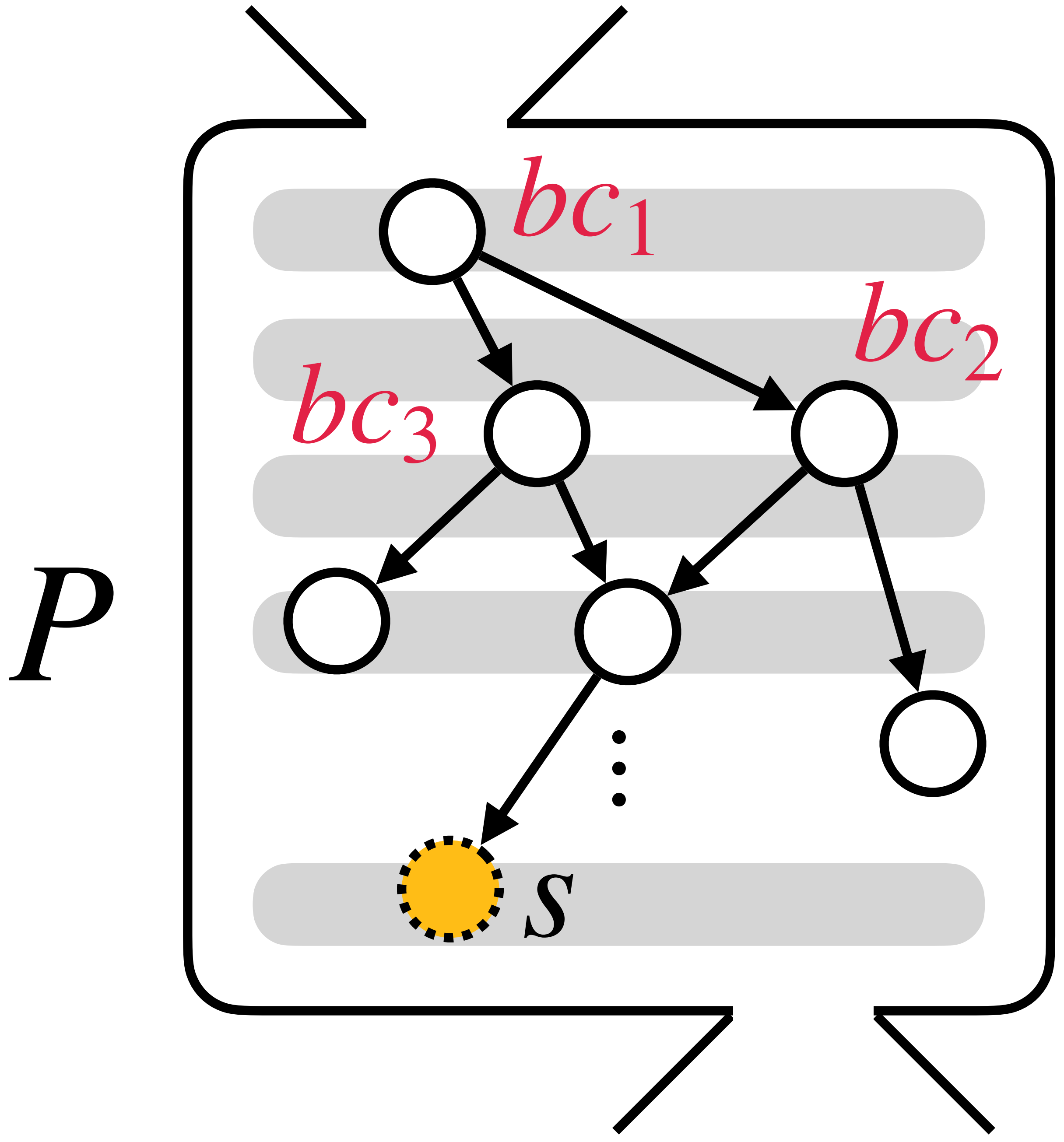
Limitation 3. Capability of Model counting

$$\Pr(s) = \Pr(pc_1) + \Pr(pc_2) + \cdots$$

Limitation 2. Path Explosion

Existing Method — Analytic Approach

PReach: **A Heuristic for Probabilistic Reachability to Identify Hard to Reach Statements,** ***Saha et al., ICSE 2022***



Symbolic Execution

bc_1

bc_2

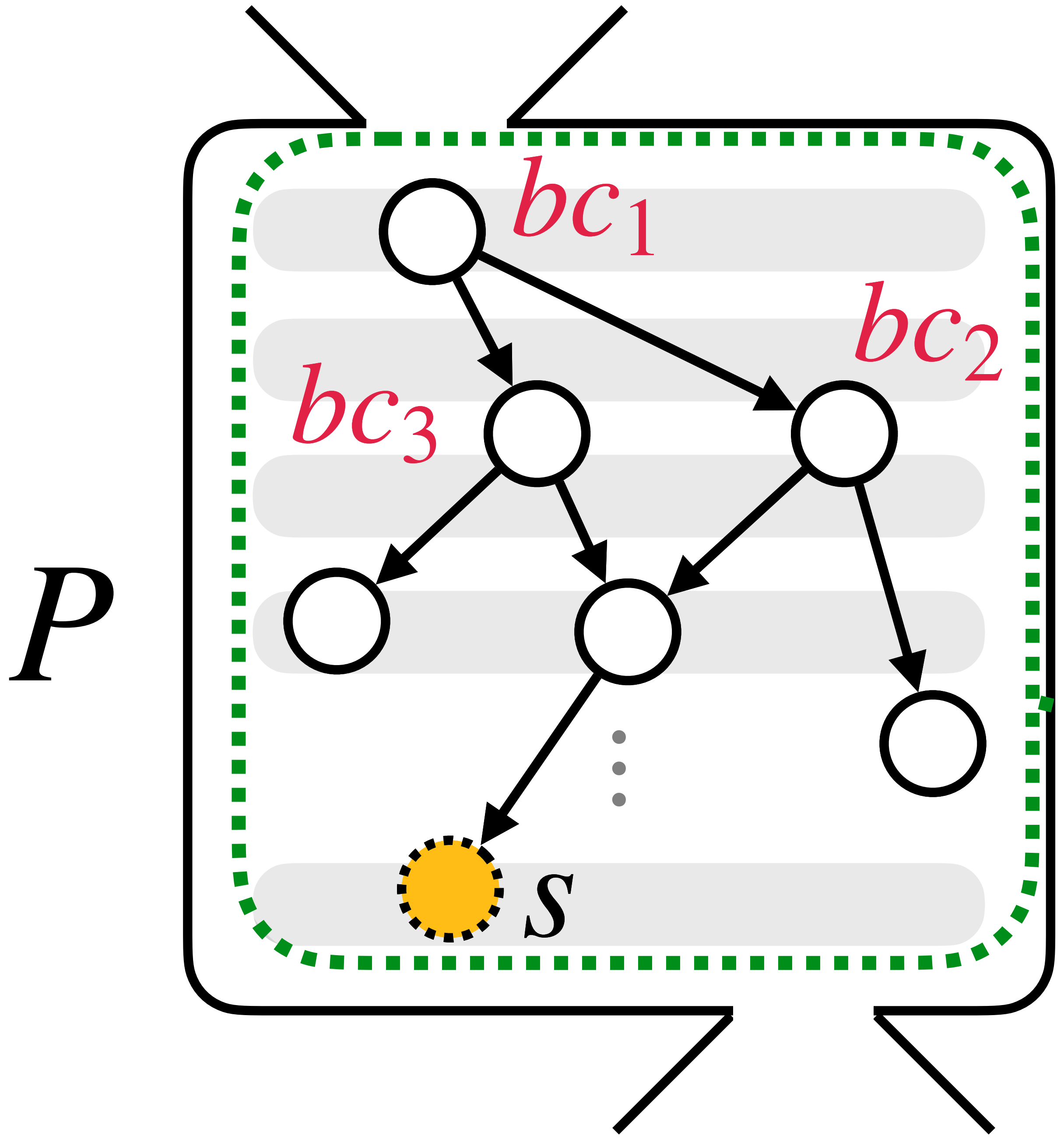
Model Counter

$\#(bc_1) = n_1, \quad \Pr(True_{bc_1}) = \dfrac{n_1}{|O_1|}$

$\#(bc_2) = n_2, \quad \Pr(True_{bc_2}) = \dfrac{n_2}{|O_2|}$

Existing Method — Analytic Approach

PReach: A Heuristic for Probabilistic Reachability to Identify Hard to Reach Statements, *Saha et al., ICSE 2022*



Symbolic Execution

bc_1

bc_2

Model Counter

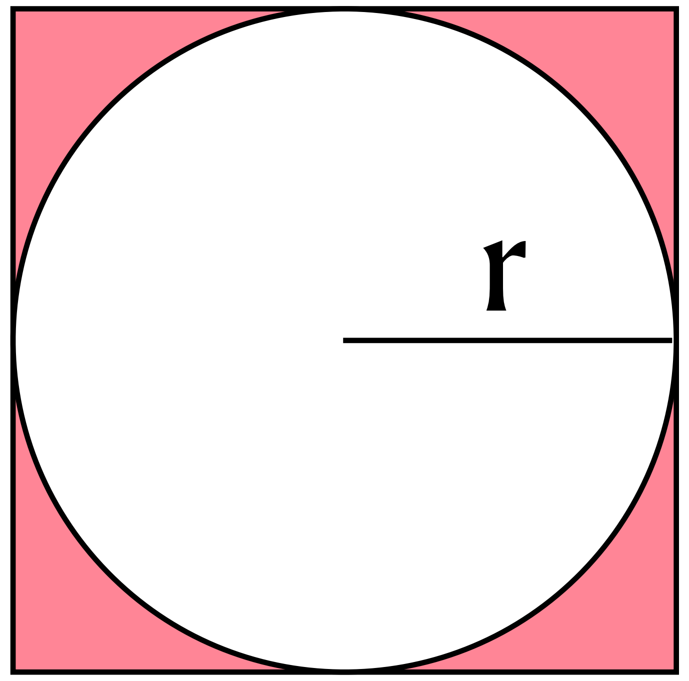
$\#(bc_1) = n_1, \quad \Pr(True_{bc_1}) = \dfrac{n_1}{|O_1|}$

$\#(bc_2) = n_2, \quad \Pr(True_{bc_2}) = \dfrac{n_2}{|O_2|}$

$\Pr(s) =$ Derived by solving DTMC model

Analytic Approach

Q. What is the probability of a thrown ball to the square dropped not into the circle?



$$
\begin{aligned}
&P(\neg\text{in circle})\\
&= \frac{Area(\text{Square}) - Area(\text{Circle})}{Area(\text{square})}\\
&= \frac{(2r)^2 - \pi r^2}{(2r)^2}\\
&= \frac{4-\pi}{4} \approx 0.2146\ldots
\end{aligned}
$$

Q. Quantitative Reachability Analysis?

$sym_1, sym_2, \cdots$

$pc_1 = c_{1,1} \wedge c_{1,2} \wedge \cdots$

$pc_2 = c_{2,1} \wedge c_{2,2} \wedge \cdots$

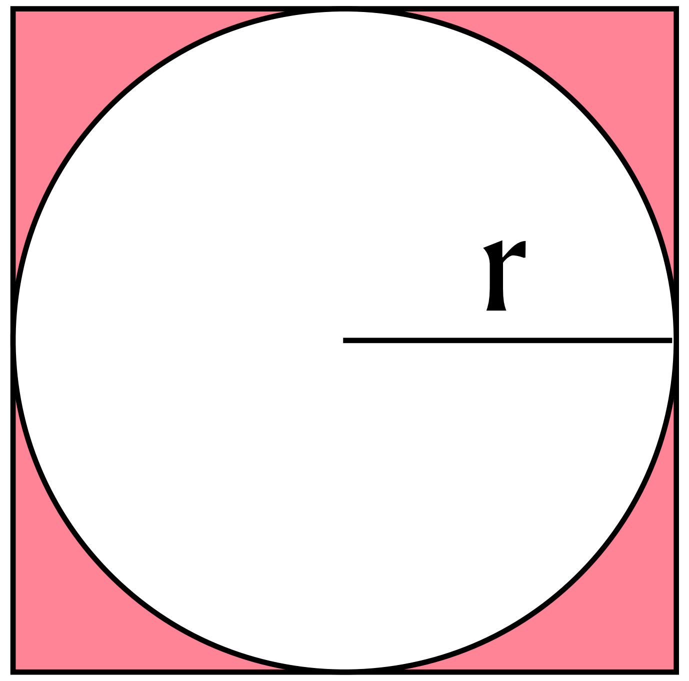
+

Symbolic Execution ***Model Counting***

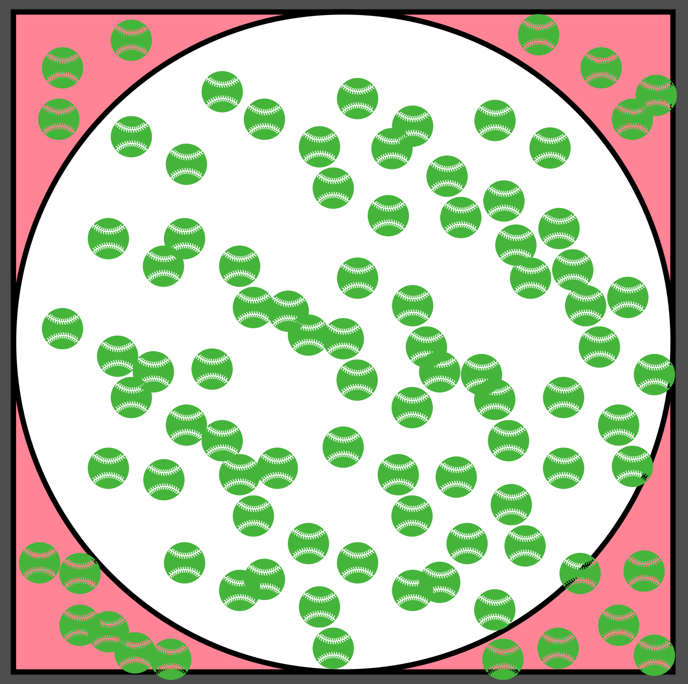
Analytic Approach

Statistical Approach

Q. What is the probability of a thrown ball to the square dropped not into the circle?



$$P(\neg\text{in circle}) = \frac{Area(\text{Square}) - Area(\text{Circle})}{Area(\text{square})} = \frac{(2r)^2 - \pi r^2}{(2r)^2} = \frac{4-\pi}{4} \approx 0.2146...$$



Monte Carlo method

$$P(\neg\text{in circle}) = \frac{\#\text{ of balls outside the circle}}{\#\text{ of balls thrown}} = \frac{65}{303} \approx 0.2145$$

Q. Quantitative Reachability Analysis?

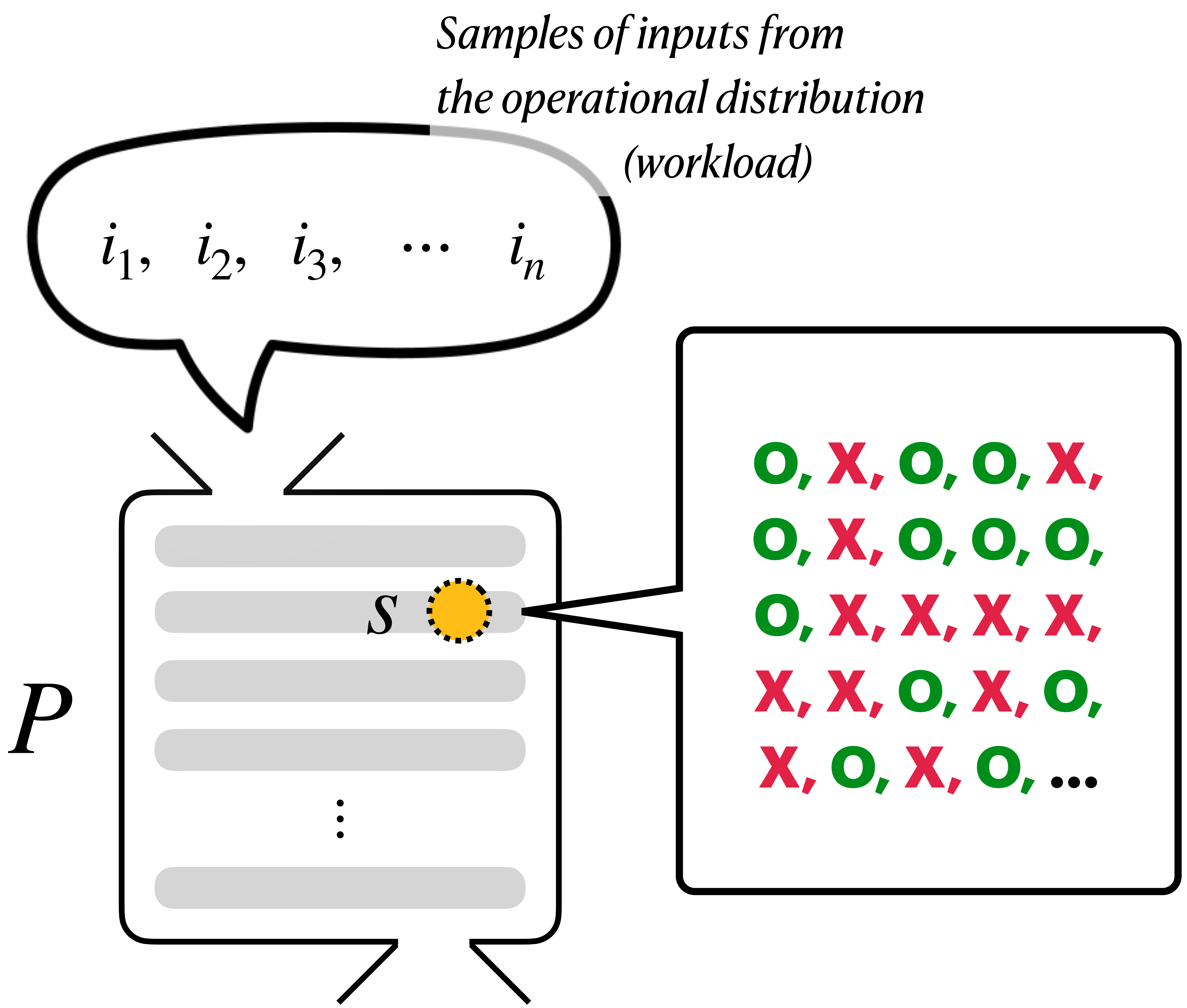
$sym_1, sym_2, \cdots$

$pc_1 = c_{1,1} \wedge c_{1,2} \wedge \cdots$

$pc_2 = c_{2,1} \wedge c_{2,2} \wedge \cdots$

Symbolic Execution + ***Model Counting***

Statistical Reachability Analysis (SRA)



$X_s :=$ the number of o in n samples

$$\hat{\Pr}(s) = \frac{X_s}{n} \overset{n \to \infty}{\Rightarrow} \Pr(s)$$

Empirical Probability

Challenge of SRA : "Rare Program States"

If the state s is rarely observable, i.e., $\Pr(s) \approx 0$,

$$\mathbb{E}\left(\frac{X_s}{n} \,\middle|\, X_s = 0\right) = 0$$

If it is unobserved, the empirical probability underapproximates to zero probability.

Problem of unseen events / Sunrise problem

Existing Estimators for Sunrise Problem

1. Laplace estimator

- *+α count* for every cases

	Case 1	Case 2	Case 3	Total
Count	7	3	0	10
Count + α	7 + α	3 + α	0 + α	10 + 3α
Laplace	(7+α) / (10+3α)	(3+α) / (10+3α)	α / (10 + 3α)	1

— For SRA —

(state s)

$$Lap(s) = \frac{c_s + \alpha}{n + 2\alpha}$$

Two cases for the state: either reached or unreached

2. Good-Turing estimator

- *The probability of seeing an unseen event in the next sample is close to the probability of seeing a singleton event*

$$\Pr(\text{next is unseen}) = \frac{f_1}{n}$$

$$GoTu(s) = \begin{cases} c_s/n, & \text{if } c_s > 0, \\ f_1/n, & \text{otherwise,} \end{cases}$$

If it's seen, empirical probability, otherwise, Good-Turing

Existing Estimators for Sunrise Problem

1. Laplace estimator

- *+α count* for every cases

	Case 1	Case 2	Case 3	Total
Count	7	3	0	10
Count + α	7 + α	3 + α	0 + α	10 + 3α
Laplace	(7+α) / (10+3α)	(3+α) / (10+3α)	α / (10 + 3α)	1

2. Good-Turing estimator

- *The probability of seeing an unseen event in the next sample is close to the probability of seeing a singleton event*

$$\Pr(\text{next is unseen}) = \frac{f_1}{n}$$

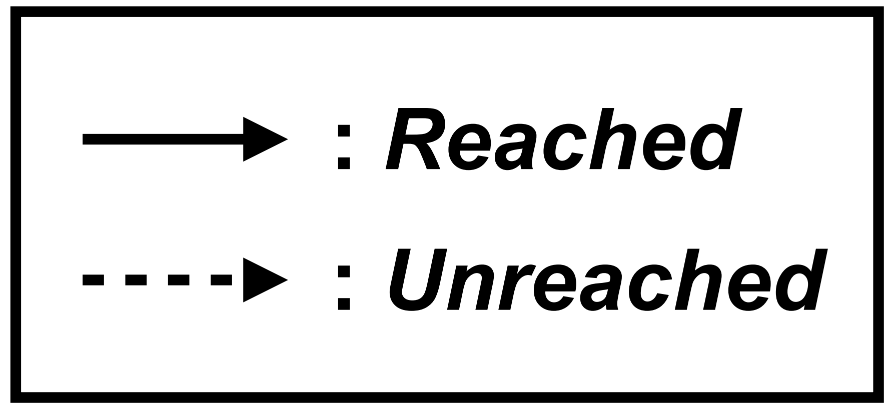
Blackbox estimators

$$Lap(s) = \frac{c_s + \alpha}{n + 2\alpha}$$

Two cases for the state: either reached or unreached

$$GoTu(s) = \begin{cases} c_s/n, & \text{if } c_s > 0, \\ f_1/n, & \text{otherwise,} \end{cases}$$

If it's seen, empirical probability,
otherwise, Good-Turing



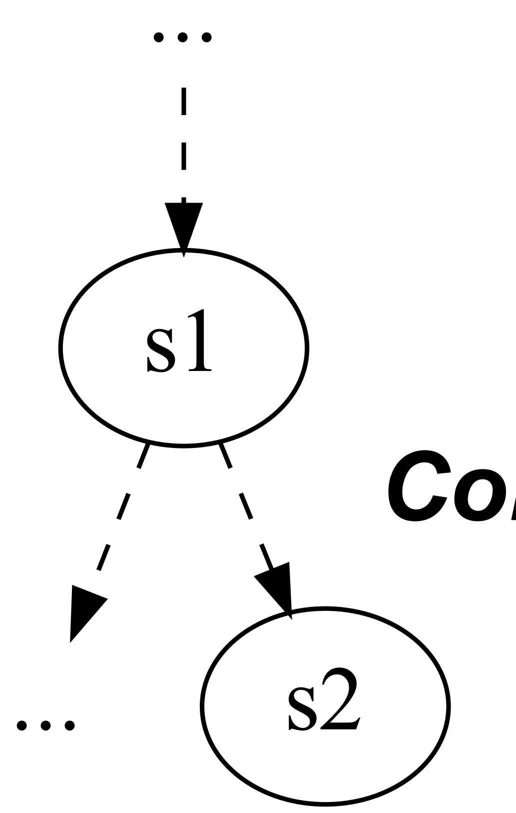
One-step further

Blackbox estimators are awesome, but...

1

Source

```
    ...
s1: if (pred)
s2:   stmt;
    ...
```

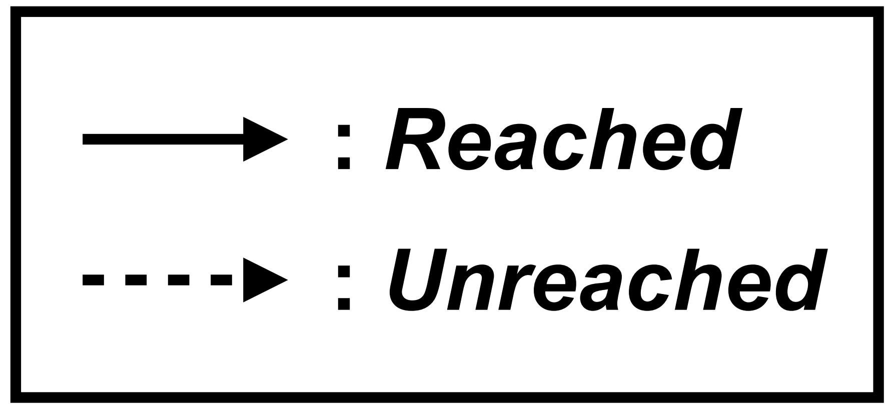


Control-flow

$\Pr(s_1) \geq \Pr(s_2)$; However

$Lap(s_1, O) = Lap(s_2, O)$ and $GoTu(s_1, O) = GoTu(s_2, O)$)

Black-box estimators are entirely unaware of the structural feature of the program.



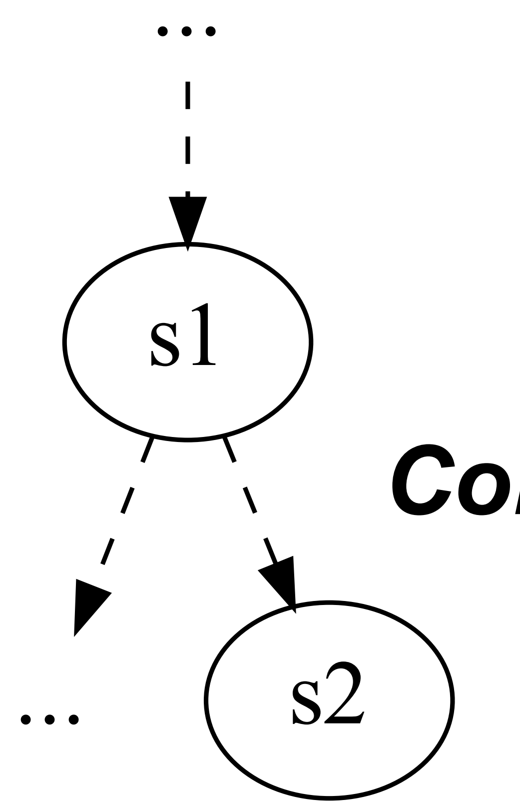
One-step further

Blackbox estimators are awesome, but...

1

Source

```
    ...
s1: if (pred)
s2:   stmt;
    ...
```

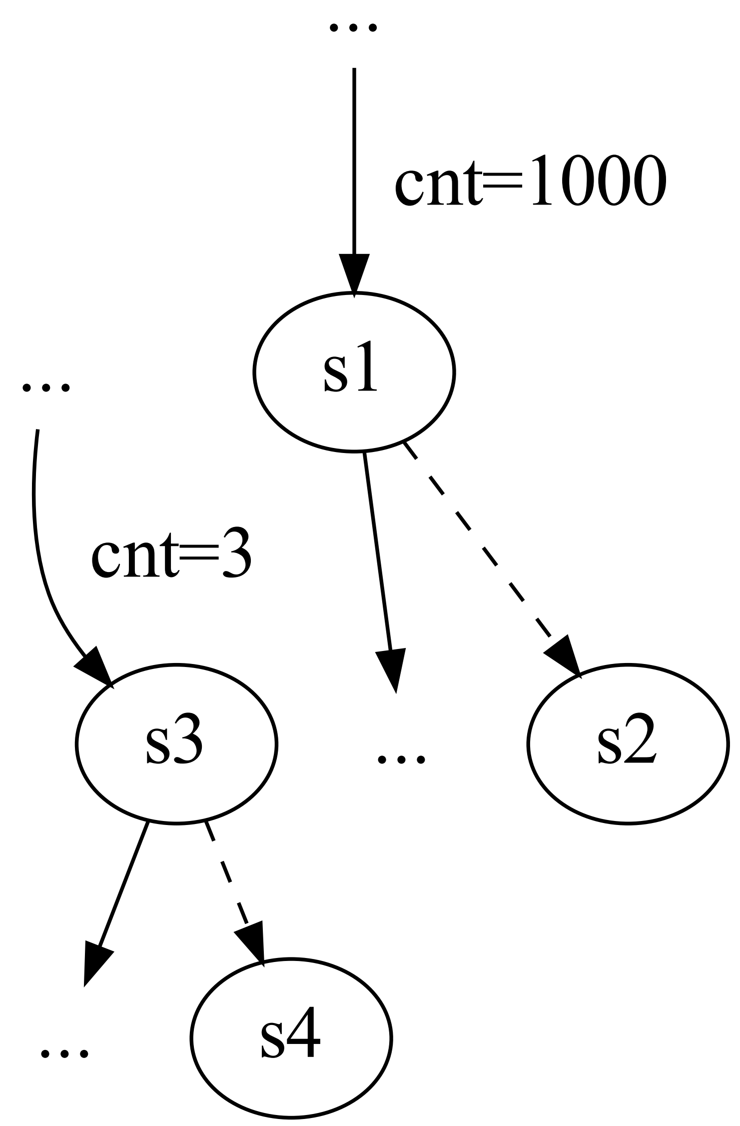


Control-flow

$\Pr(s_1) \geq \Pr(s_2)$; However

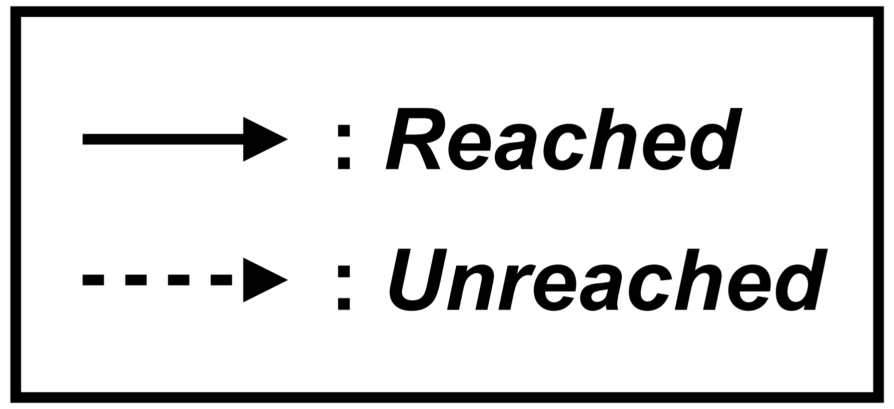
$Lap(s_1, O) = Lap(s_2, O)$ and $GoTu(s_1, O) = GoTu(s_2, O)$)

2



s_2 has larger chances of being reached than s_4

Black-box estimators are entirely unaware of the structural feature of the program.



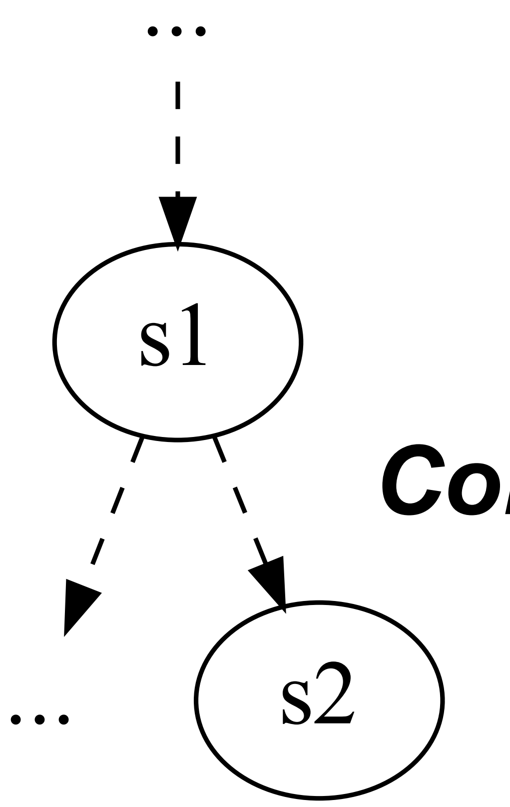
One-step further

Blackbox estimators are awesome, but...

1

Source

```
   ...
s1: if (pred)
s2:   stmt;
   ...
```

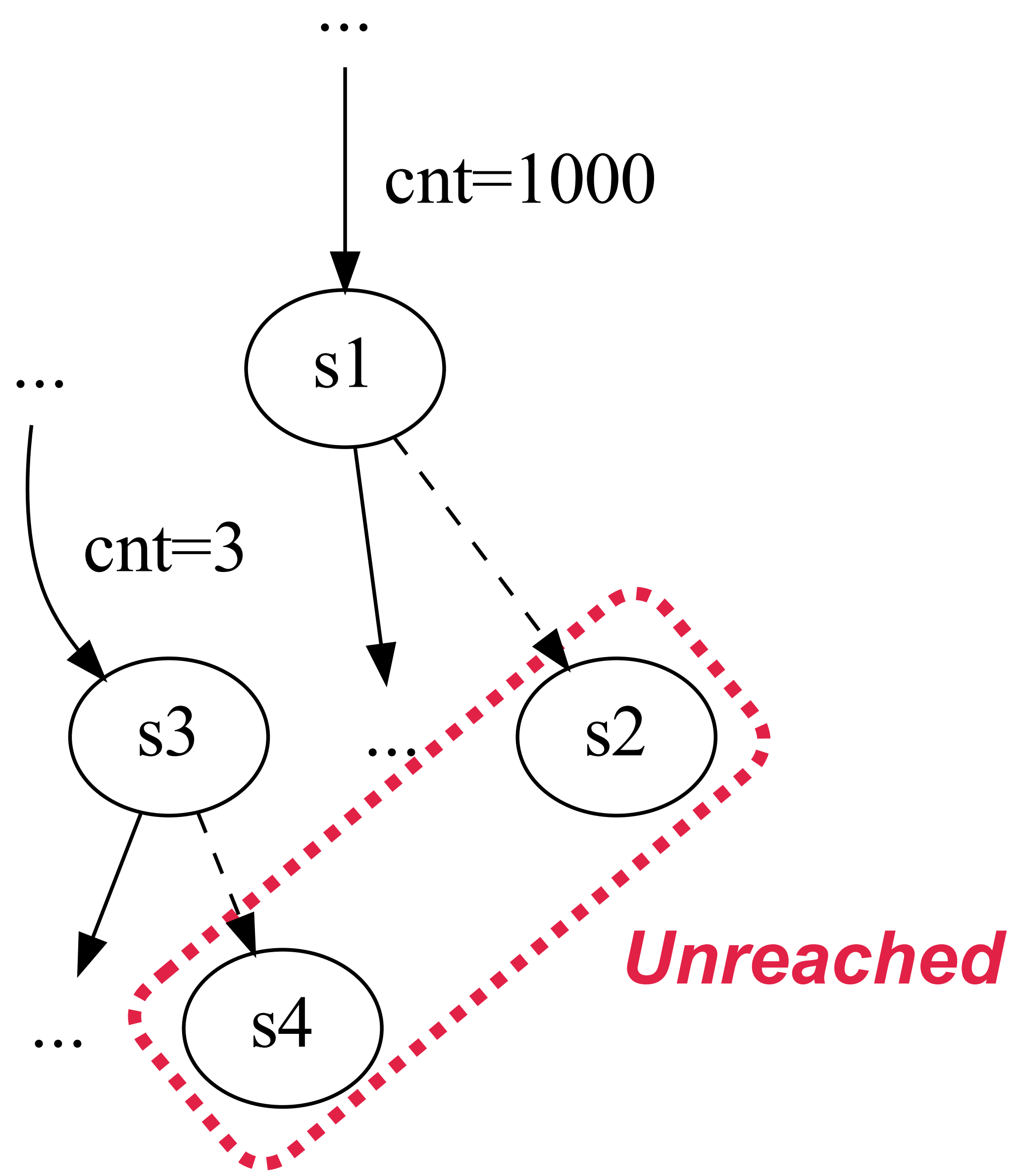


Control-flow

$\Pr(s_1) \geq \Pr(s_2)$; However

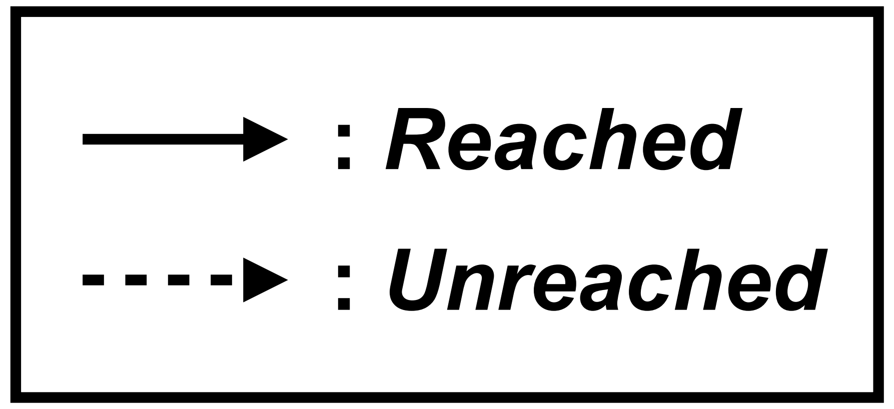
$Lap(s_1, O) = Lap(s_2, O)$ and $GoTu(s_1, O) = GoTu(s_2, O)$)

2



s_2 has larger chances of being reached than s_4

Black-box estimators are entirely unaware of the structural feature of the program.



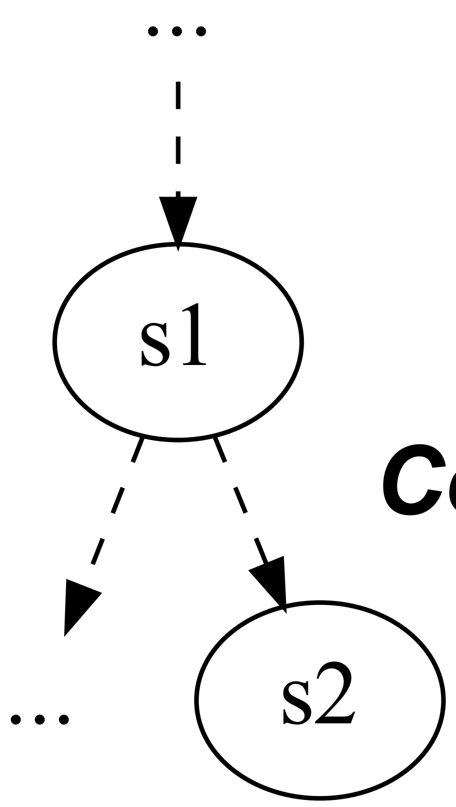
One-step further

Blackbox estimators are awesome, but...

1

Source

```
    ...
s1: if (pred)
s2:    stmt;
    ...
```

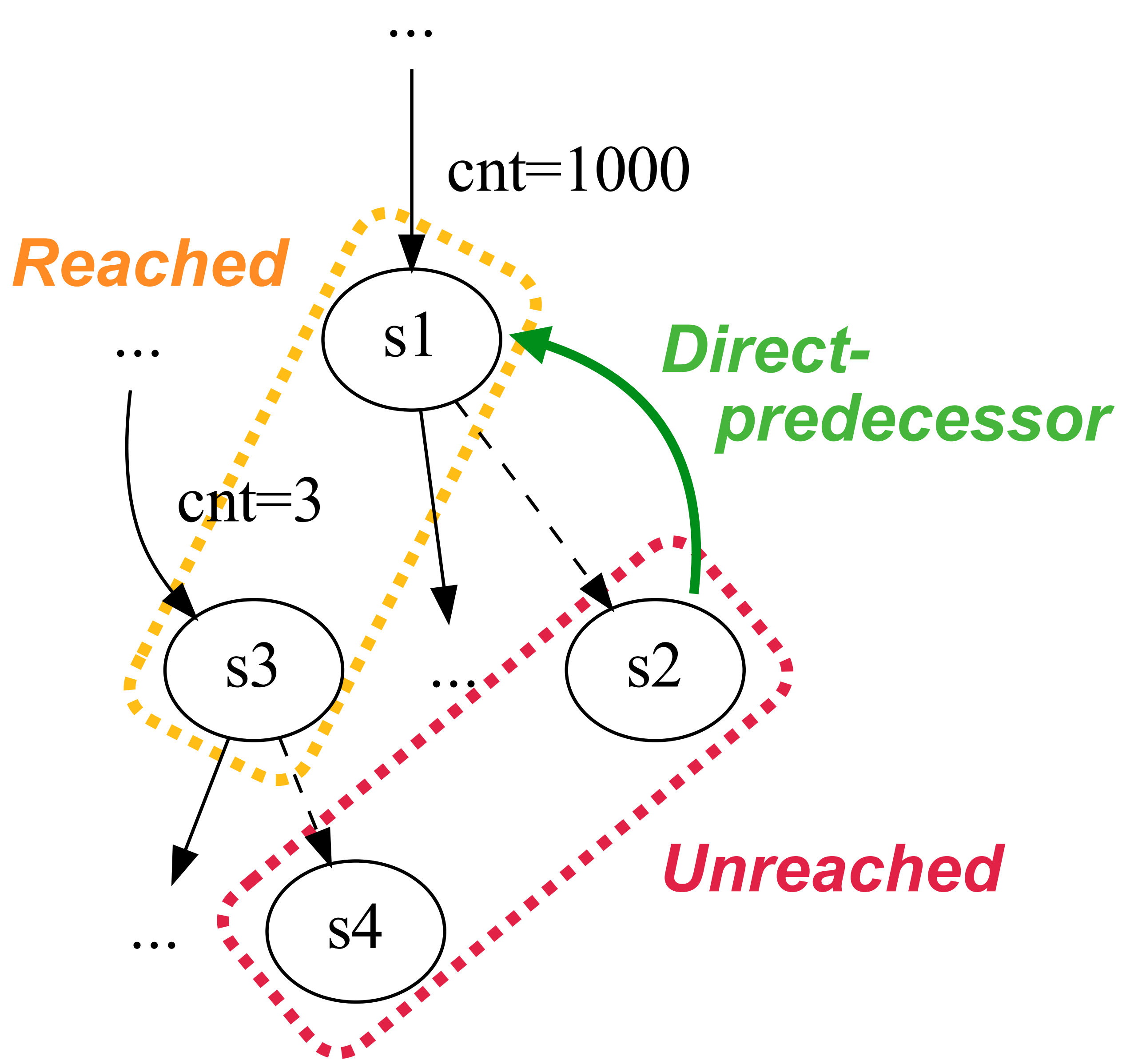


Control-flow

$\Pr(s_1) \geq \Pr(s_2)$; However

$Lap(s_1, O) = Lap(s_2, O)$ and $GoTu(s_1, O) = GoTu(s_2, O)$)

2



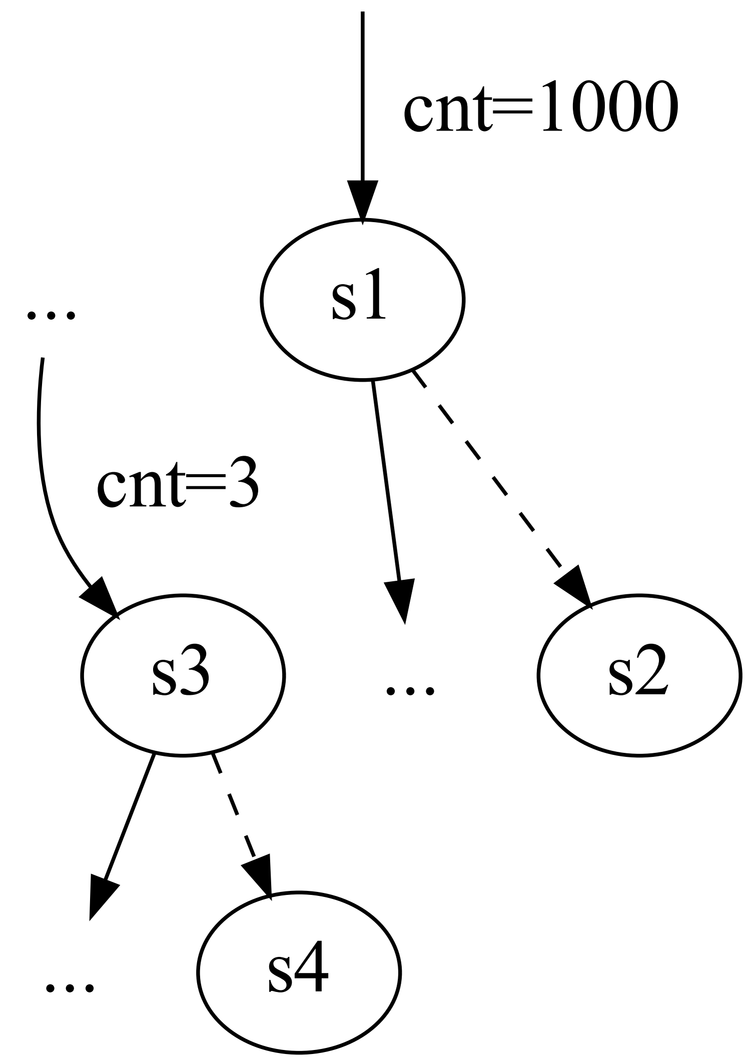
s_2 has larger chances of being reached than s_4

Black-box estimators are entirely unaware of the structural feature of the program.

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



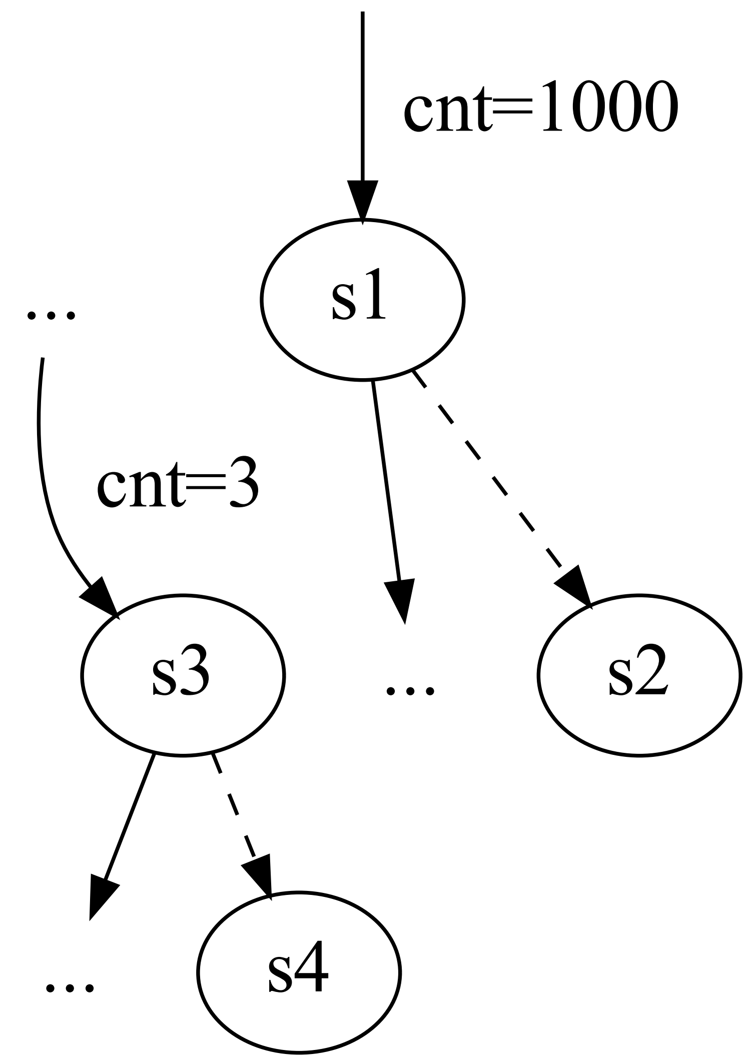
Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Previous (Laplace):

$$\Pr(s_4) = \Pr(s_2) =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

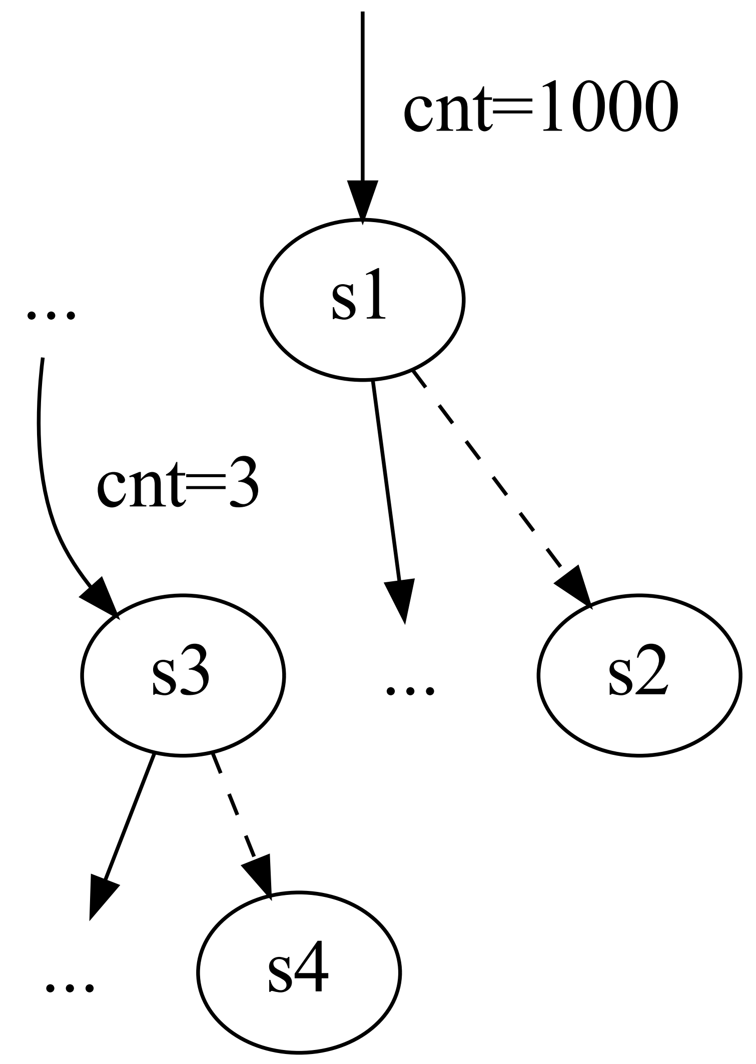
Total sample: 1000



Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

$$\cancel{\Pr(s_4) = \Pr(s_2)} =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

$$\Pr(s_4) = \Pr(s_3) \times \frac{\alpha}{3 + 2 \times \alpha}$$

$$=_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

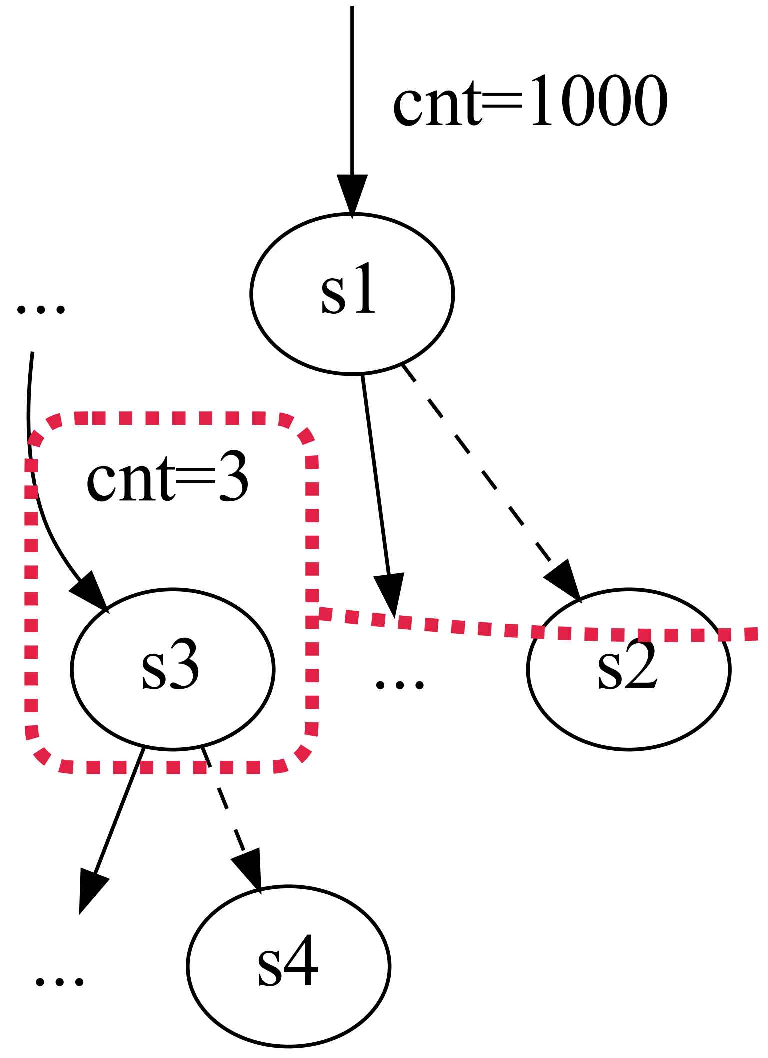
$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha}$$

$$=_{\alpha=2} \quad 1 \times \frac{2}{1{,}004} \approx 0.0020$$

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

$$\cancel{\Pr(s_4) = \Pr(s_2)} =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

Pr(Pred)

$$\Pr(s_4) = \boxed{\Pr(s_3)} \times \frac{\alpha}{3 + 2 \times \alpha}$$

$$=_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

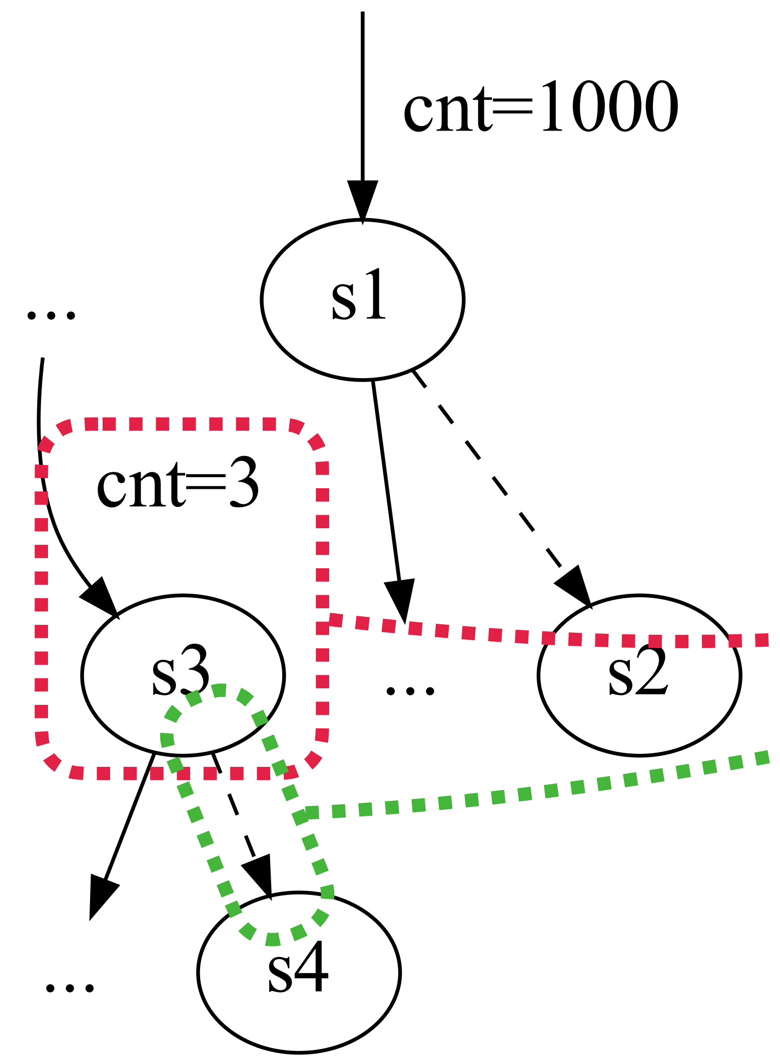
$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha}$$

$$=_{\alpha=2} \quad 1 \times \frac{2}{1{,}004} \approx 0.0020$$

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

$$\Pr(s_4) = \Pr(s_2) =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

Pr(Pred) **Lap(Pred→Next)**

$$\Pr(s_4) = \boxed{\Pr(s_3)} \times \boxed{\frac{\alpha}{3 + 2 \times \alpha}}$$

$$=_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

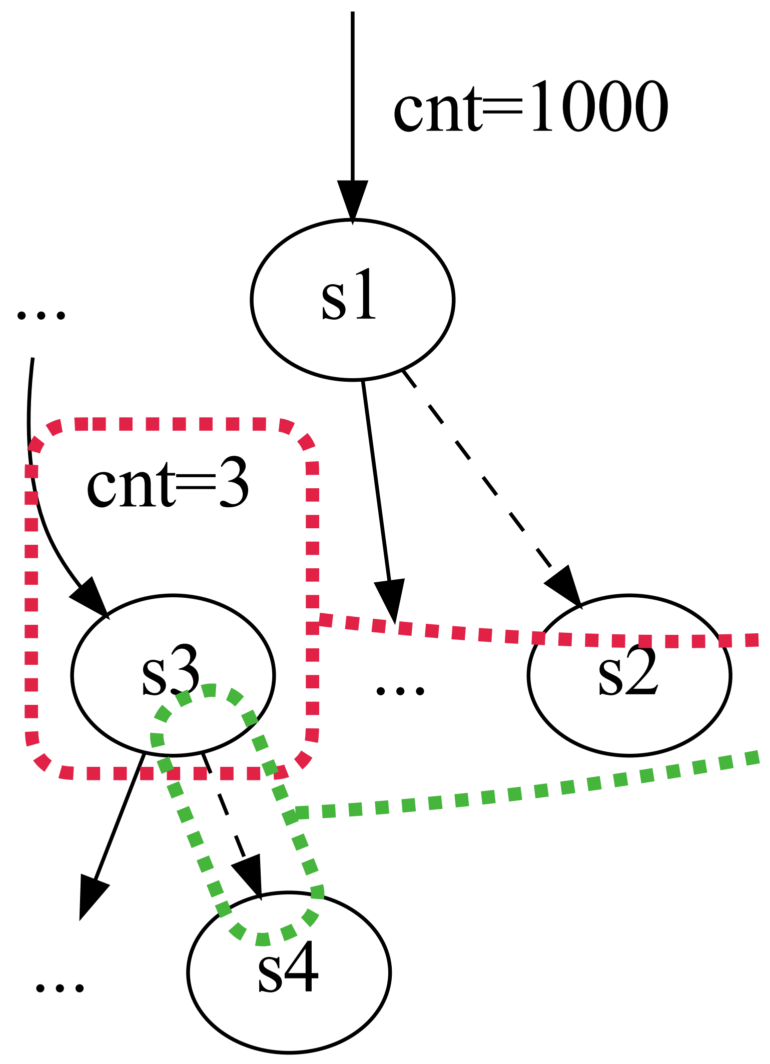
$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha}$$

$$=_{\alpha=2} \quad 1 \times \frac{2}{1{,}004} \approx 0.0020$$

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

$$\Pr(s_4) = \Pr(s_2) =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

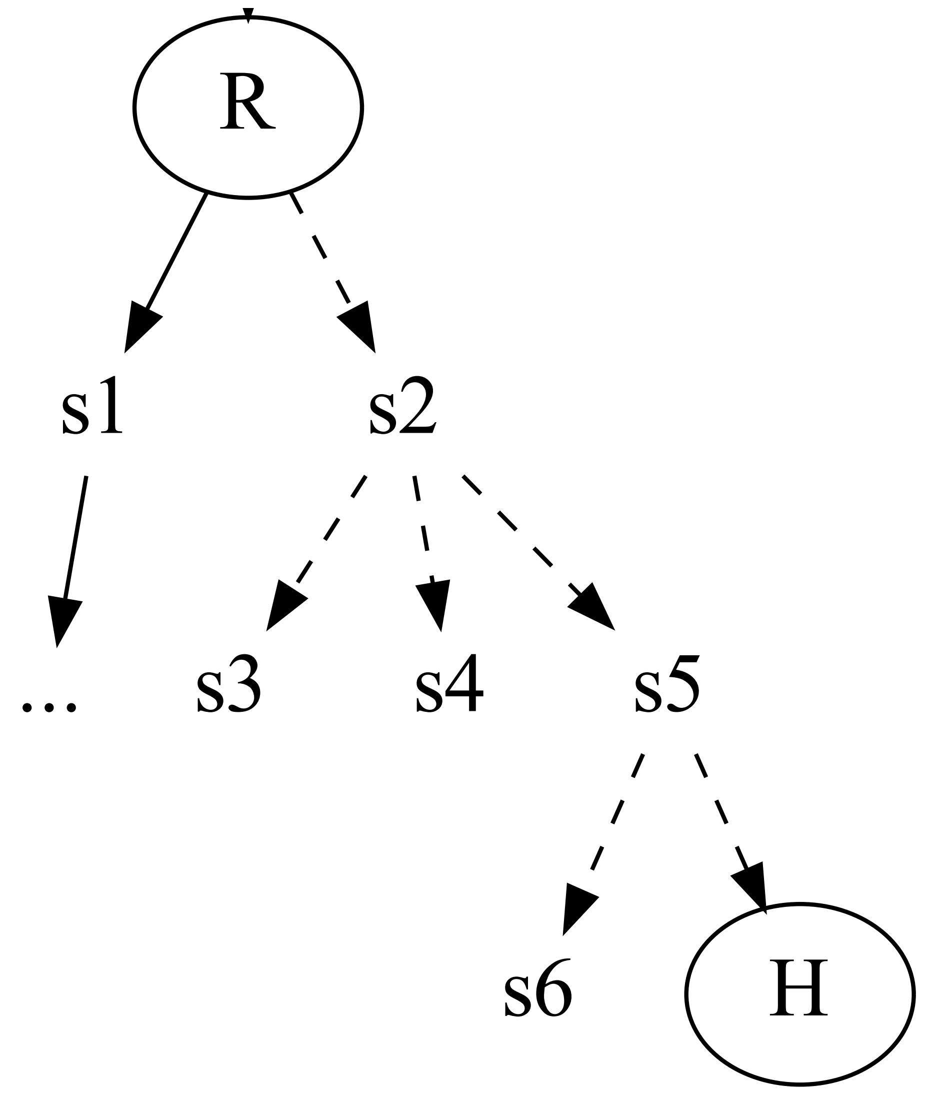
Pr(Pred) **Lap(Pred→Next)**

$$\Pr(s_4) = \boxed{\Pr(s_3)} \times \boxed{\frac{\alpha}{3 + 2 \times \alpha}}$$

$$=_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha}$$

$$=_{\alpha=2} \quad 1 \times \frac{2}{1{,}004} \approx 0.0020$$

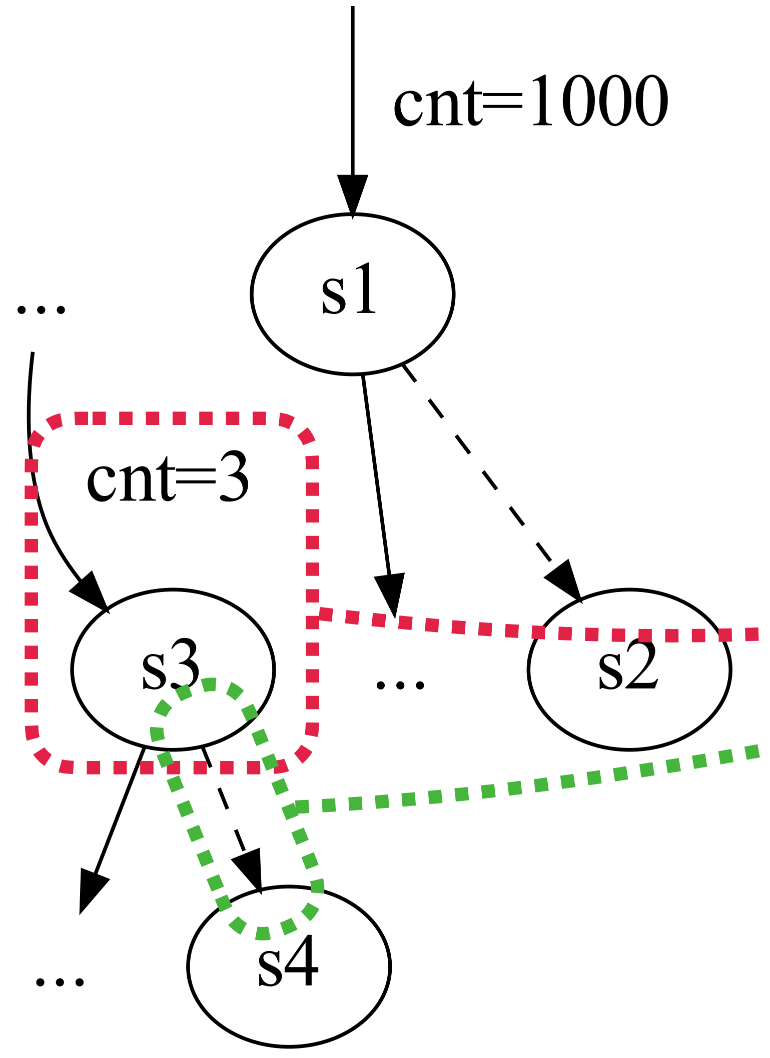


$$\Pr(H) = \Pr(R) \times \frac{\alpha}{\#(R) + 2 \times \alpha} \times \frac{1}{3} \times \frac{1}{2}$$

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

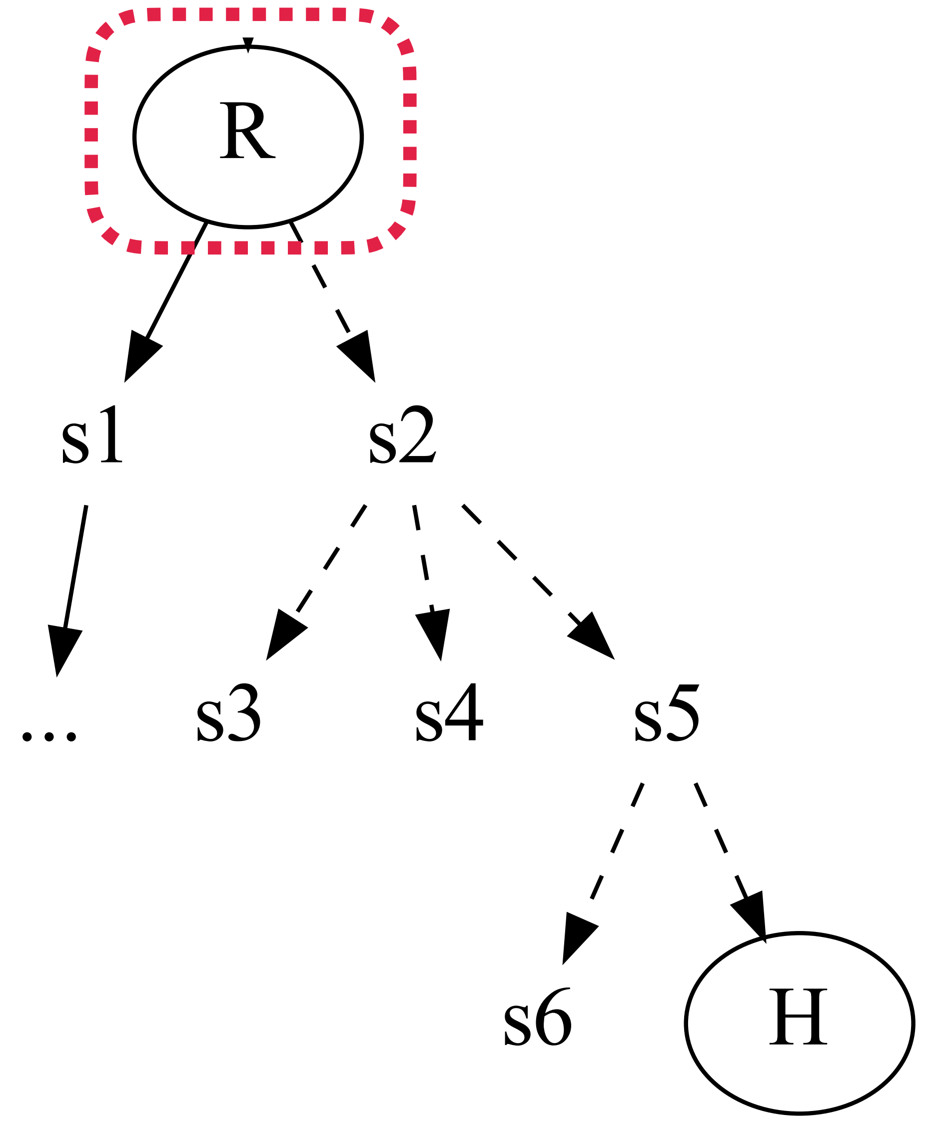
$$\Pr(s_4) = \Pr(s_2) =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

Pr(Pred) **Lap(Pred→Next)**

$$\Pr(s_4) = \Pr(s_3) \times \frac{\alpha}{3 + 2 \times \alpha} =_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha} =_{\alpha=2} 1 \times \frac{2}{1{,}004} \approx 0.0020$$

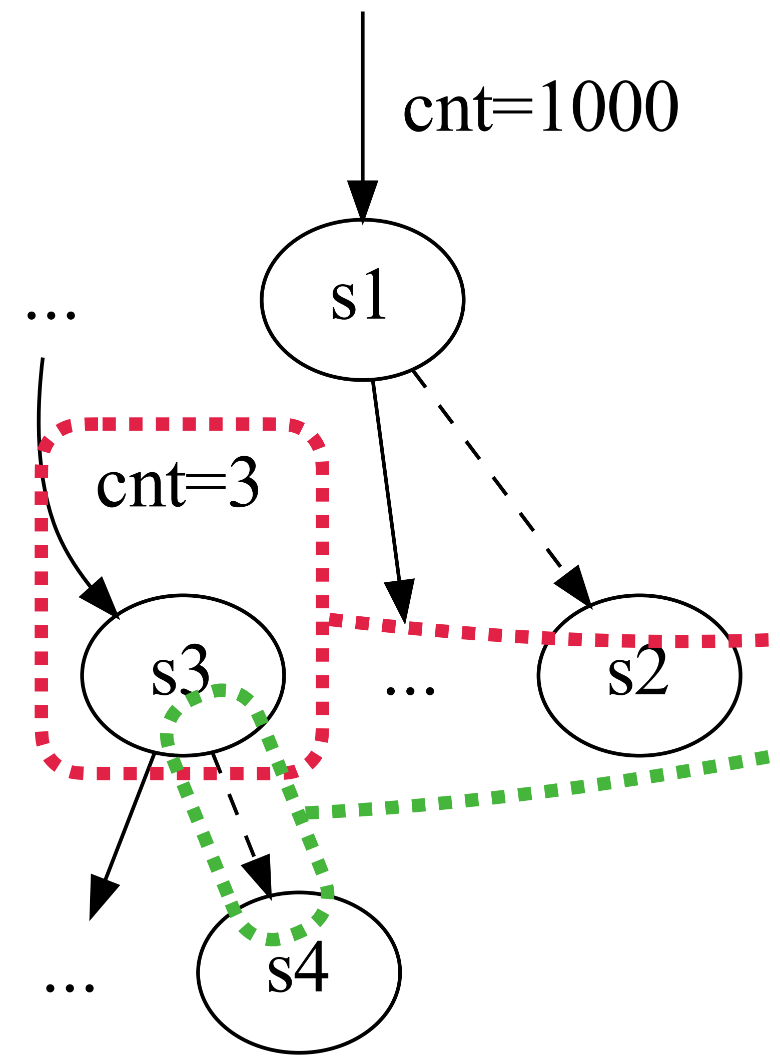


$$\Pr(H) = \Pr(R) \times \frac{\alpha}{\#(R) + 2 \times \alpha} \times \frac{1}{3} \times \frac{1}{2}$$

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

$$\mathrm{Pr}(s_4) = \mathrm{Pr}(s_2) =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

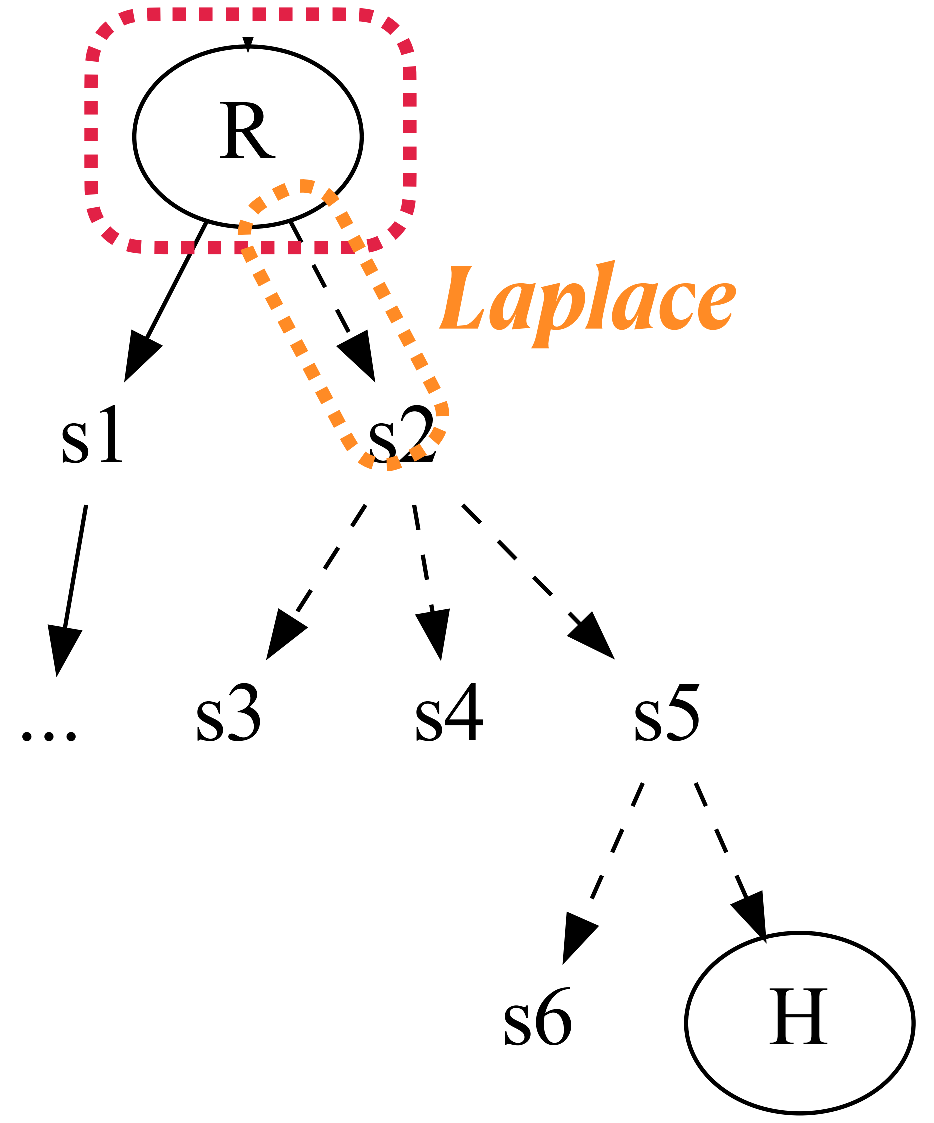
Pr(Pred) **Lap(Pred→Next)**

$$\mathrm{Pr}(s_4) = \boxed{\mathrm{Pr}(s_3)} \times \boxed{\frac{\alpha}{3 + 2 \times \alpha}}$$

$$=_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

$$\mathrm{Pr}(s_2) = \mathrm{Pr}(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha}$$

$$=_{\alpha=2} \quad 1 \times \frac{2}{1{,}004} \approx 0.0020$$

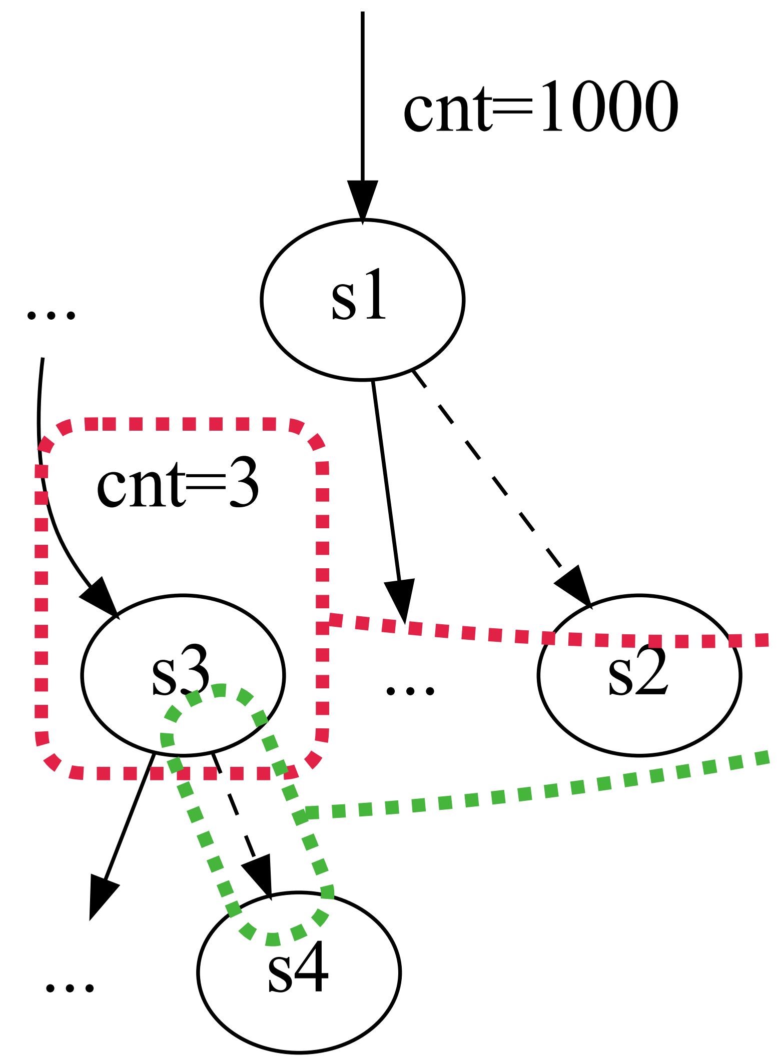


$$\mathrm{Pr}(H) = \boxed{\mathrm{Pr}(R)} \times \boxed{\frac{\alpha}{\#(R) + 2 \times \alpha}} \times \frac{1}{3} \times \frac{1}{2}$$

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

$$\Pr(s_4) = \Pr(s_2) =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

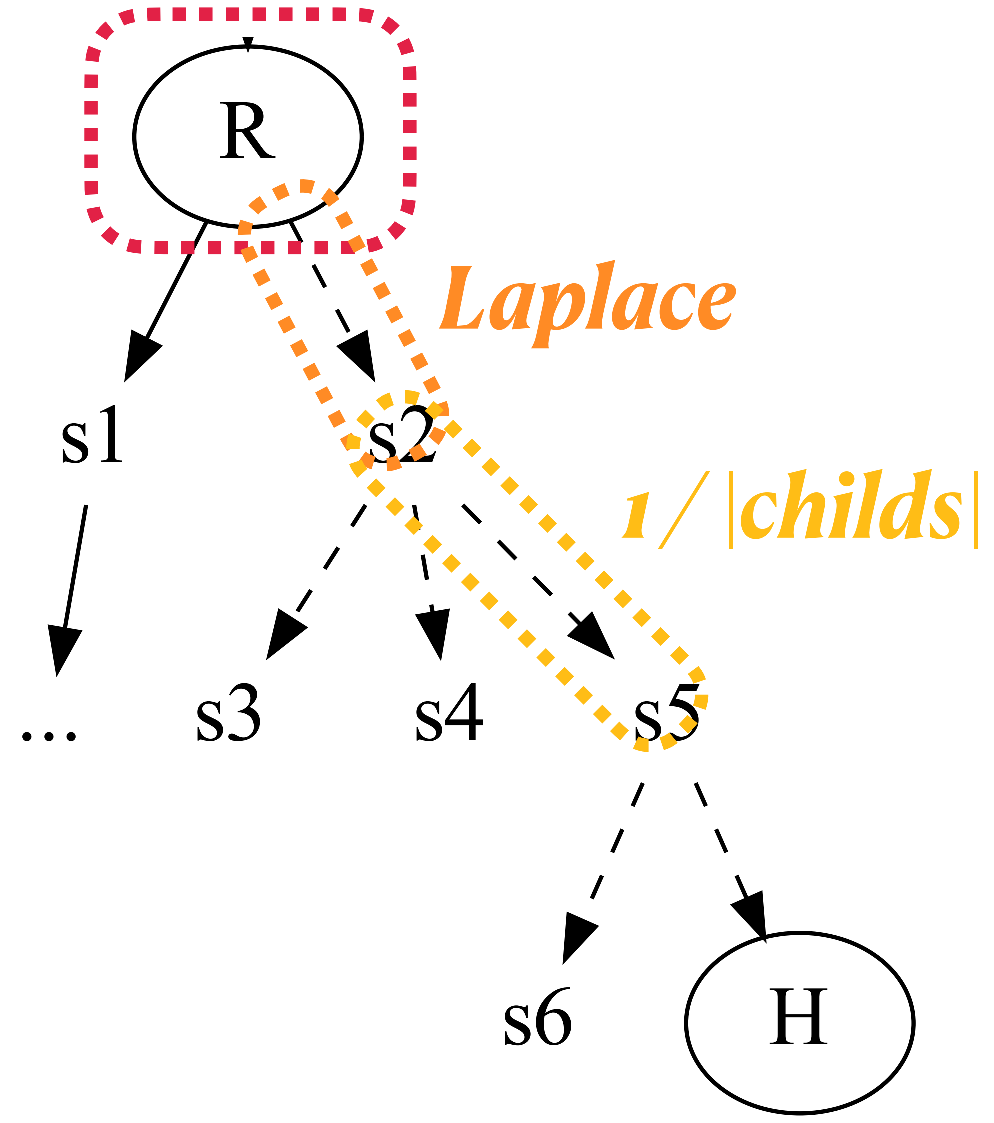
Pr(Pred) **Lap(Pred→Next)**

$$\Pr(s_4) = \boxed{\Pr(s_3)} \times \boxed{\frac{\alpha}{3 + 2 \times \alpha}}$$

$$=_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha}$$

$$=_{\alpha=2} \quad 1 \times \frac{2}{1{,}004} \approx 0.0020$$

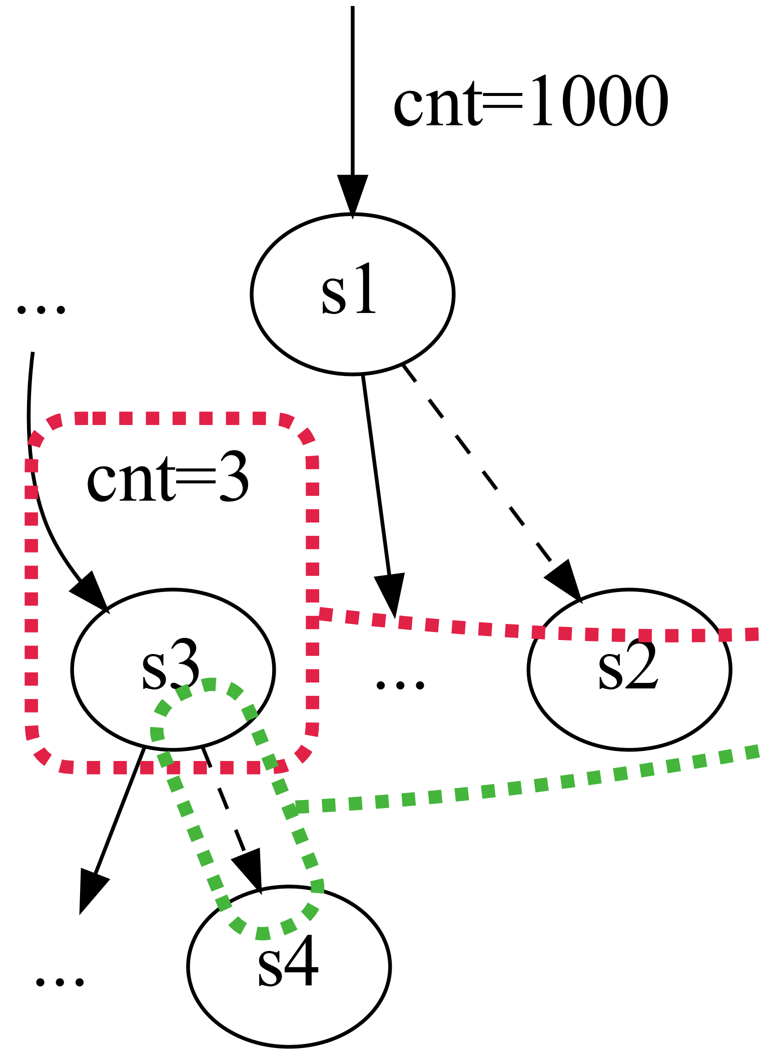


$$\Pr(H) = \boxed{\Pr(R)} \times \boxed{\frac{\alpha}{\#(R) + 2 \times \alpha}} \times \boxed{\frac{1}{3}} \times \frac{1}{2}$$

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

$$\cancel{\Pr(s_4) = \Pr(s_2)} =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

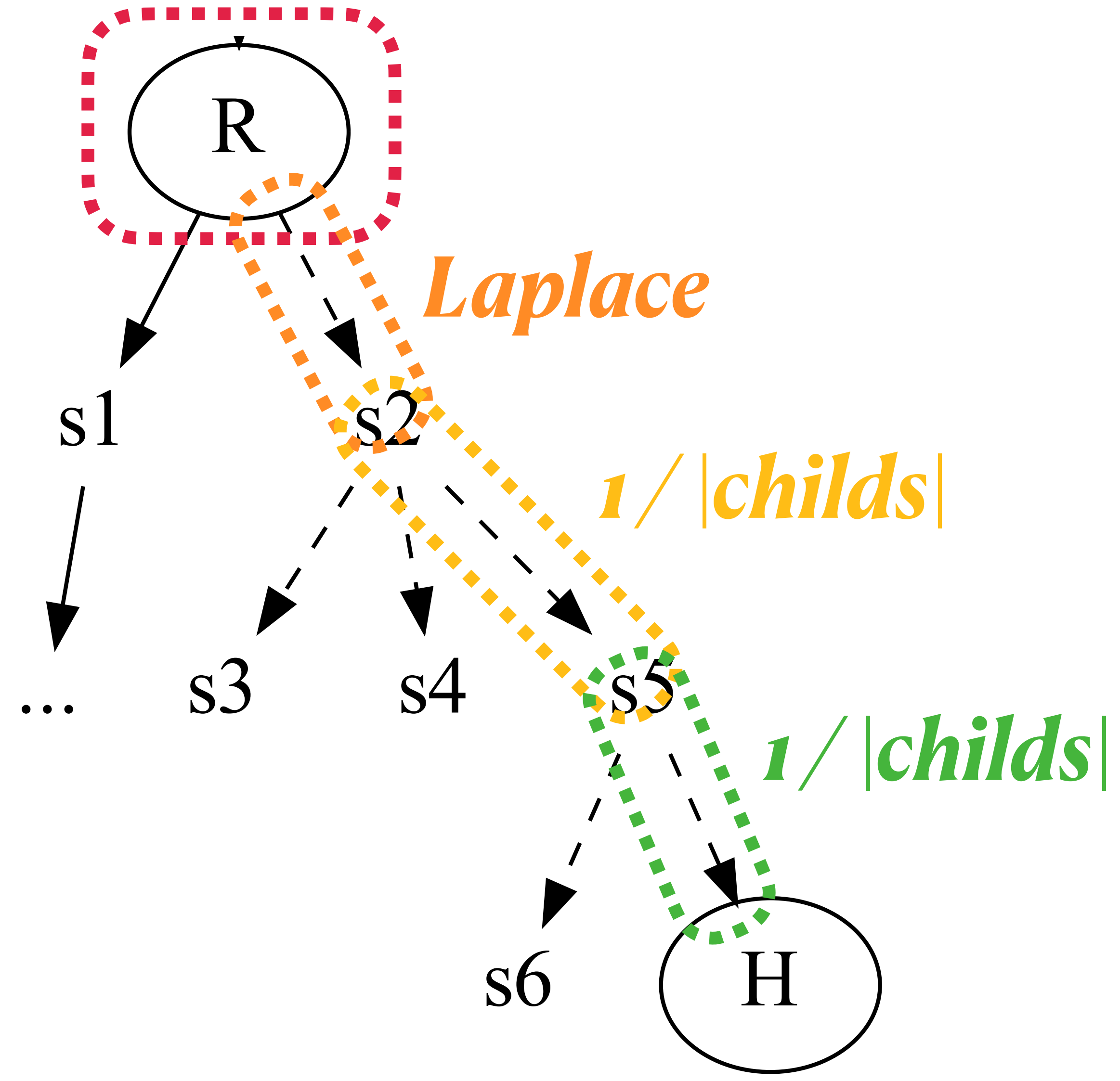
Pr(Pred) **Lap(Pred→Next)**

$$\Pr(s_4) = \boxed{\Pr(s_3)} \times \boxed{\frac{\alpha}{3 + 2 \times \alpha}}$$

$$=_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha}$$

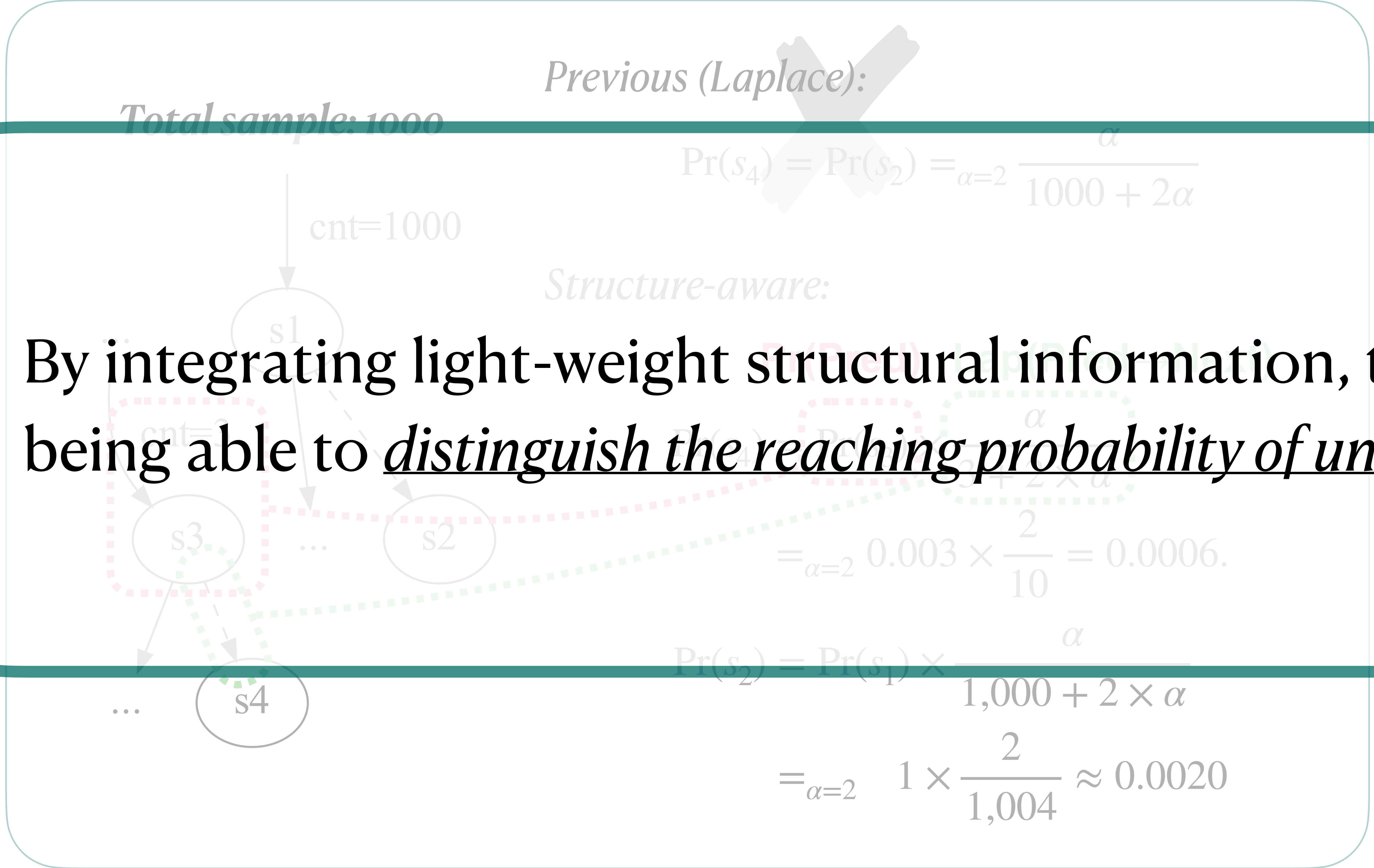
$$=_{\alpha=2} \quad 1 \times \frac{2}{1{,}004} \approx 0.0020$$

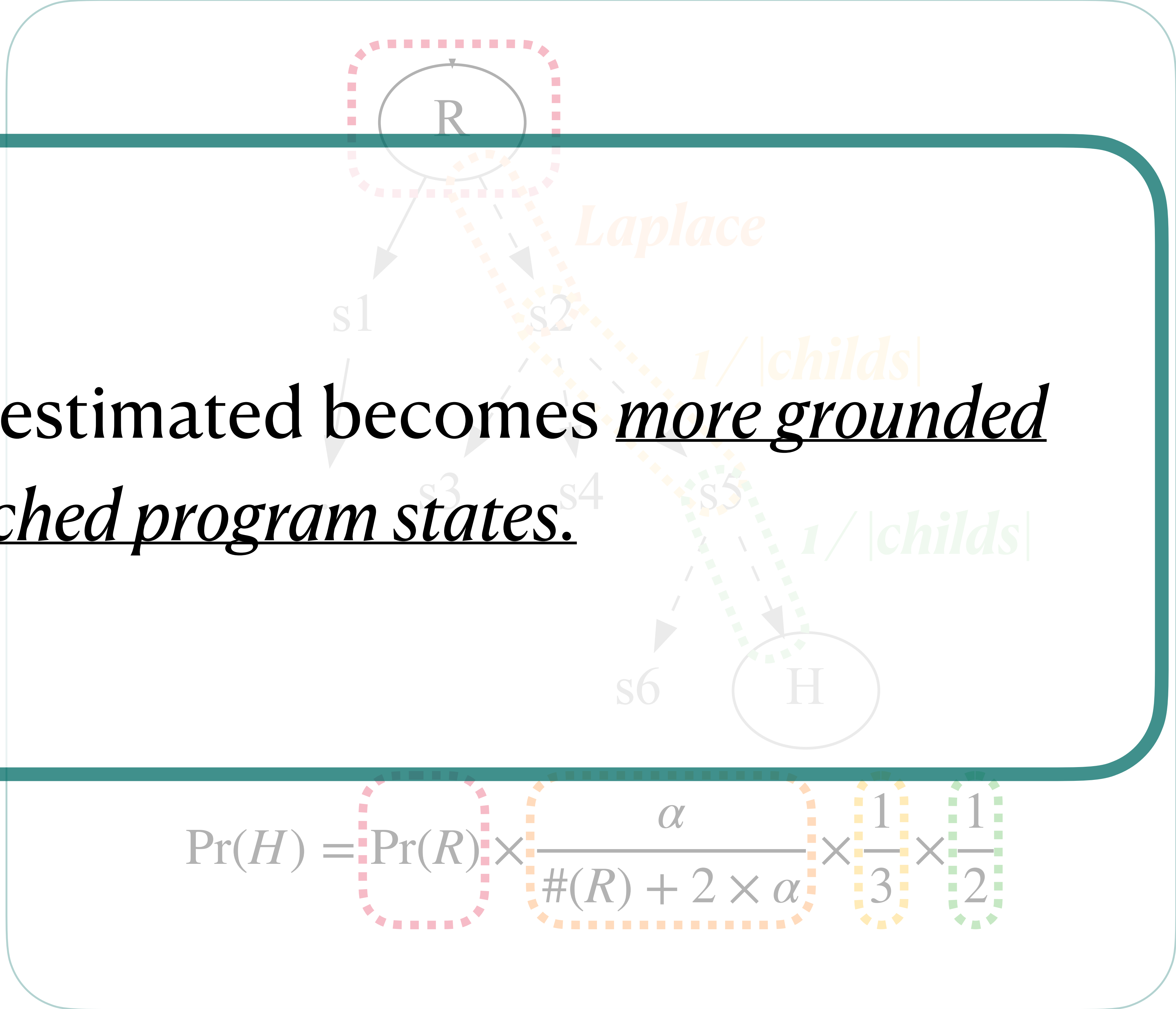


$$\Pr(H) = \boxed{\Pr(R)} \times \boxed{\frac{\alpha}{\#(R) + 2 \times \alpha}} \times \boxed{\frac{1}{3}} \times \boxed{\frac{1}{2}}$$

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.





By integrating light-weight structural information, the estimated becomes *more grounded* being able to *distinguish the reaching probability of unreached program states.*

Evaluation

RQ 1. Statistical method vs. Analytic method for QRA

RQ 2. Blackbox estimator vs. Structure-aware estimator

Evaluation 1: Statistical vs Analytic

- Analytic method: PSE, PReach (SOTA)
- Subjects: Programs used in PReach
  - Target statement: Assertion
- Metric: Accuracy / Estimation time
  - For SRA, 'estimation time' is the time taken until the estimate gets close enough.

jpf-regress. (26)	ExMIT-T, Exe1-F, Exe2-F, Exe4-F, Exe6-F, Exe8-F, Exe10-F, Exe10-T, Exe12-F, Exe12-T, Exe13-T, Exe14-T, Exe15-T, Exe18-F, Exe19-T, Exe20-F, Exe20-T, Exe26-F, Exe27-F, FNEG-T, LCMP-T, Simple-F, Simple-T, Suzette-F, Suzette-T, Assign-T
jbmc-regress. (4)	assert3, if_icmp1, switch1, Token2
algorithms (2)	InsertSort2, RBTree1

Evaluation 1: Statistical vs Analytic

Program	GT	Esti(PSE)	T(PSE)	Esti(PR)	T(PR)	Esti(Lap)	T(Lap)
ExMIT-T	~0	4.7E-10 (O)	.866s	7.6E-06 (O)	14.9s	1.0E-06 (O)	0.044s
Exe1-F	0.49	NL (X)	-	0.500 (O)	13.5s	0.489 (O)	0.006s
Exe2-F	0.2	NL (X)	-	0.125 (X)	14.6s	0.199 (O)	0.003s
Exe4-F	0.25	NL (X)	-	0.125 (X)	14.7s	0.248 (O)	0.014s
Exe6-F	1.0	NL (X)	-	2.3E-10 (X)	14.8s	0.990 (O)	0.001s
Exe8-F	0.3	NL (X)	-	0.500 (X)	14.7s	0.300 (O)	0.005s
Exe10-F	0.25	NL (X)	-	0.250 (O)	14.5s	0.250 (O)	0.005s
Exe10-T	~0	NL (X)	-	1.2E-10 (O)	14.5s	1.0E-06 (O)	0.085s
Exe12-F	0.5	0.500 (O)	.934s	0.500 (O)	14.6s	0.501 (O)	0.004s
Exe12-T	0.375	0.250 (X)	.966s	0.375 (O)	14.6s	0.376 (O)	0.007s
Exe13-T	~0	0 (O)	.909s	5.0E-11 (O)	13.7s	1.0E-06 (O)	0.087s
Exe14-T	0.25	0.5 (X)	.860s	0.25 (O)	11.9s	0.251 (O)	0.018s
Exe15-T	0.25	0.125 (X)	.910s	0.25 (O)	13.1s	0.251 (O)	0.011s
Exe18-F	0.5	NL (X)	-	0.500 (O)	14.5s	0.502 (O)	0.011s
Exe19-T	0.25	0.375 (X)	.950s	0.245 (O)	14.5s	0.251 (O)	0.015s
Exe20-F	0.25	NL (X)	-	0.125 (X)	13.6s	0.249 (O)	0.008s
Exe20-T	0.5	0.500 (O)	.903s	0.5 (O)	14.5s	0.500 (O)	0.008s
Exe26-F	0.5	NL (X)	-	0.245 (X)	14.7s	0.500 (O)	0.006s
Exe27-F	0.5	0.500 (O)	.849s	0.500 (O)	14.7s	0.500 (O)	0.004s
FNEG-T	0	0 (O)	.850s	0.25 (X)	14.5s	1.0E-06 (O)	0.045s
LCMP-T	0	0 (O)	.832s	0.5 (X)	14.9s	1.0E-06 (O)	0.044s
Simple-F	0	0 (O)	.854s	TO (X)	-	1.0E-06 (O)	0.048s
Simple-T	0	0 (O)	.844s	TO (X)	-	1.0E-06 (O)	0.047s
Suzette-F	0.25	0.250 (O)	.910s	4.7E-10 (X)	13.8s	0.249 (O)	0.030s
Suzette-T	~0	2.6E-9 (O)	.926s	2.6E-09 (O)	14.4s	1.0E-06 (O)	0.084s
Assign-T	0	0 (O)	.841s	0.25 (X)	14.6s	1.0E-06 (O)	0.045s
InsertSort2	2.1E-02	TO (X)	-	2.5E-11 (X)	15.8s	2.1E-02 (O)	4,904s
RBTree1	0.125	TO (X)	-	DTMC (X)	14.4s	0.124 (O)	0.002s
assert3	~0	4.7E-10 (O)	.847s	2.3E-10 (O)	10.6s	1.0E-06 (O)	0.044s
if_icmp1	0	0 (O)	.856s	5.0E-11 (O)	10.5s	1.0E-06 (O)	0.045s
switch1	~0	2.8-09 (O)	1.03s	0.0 (O)	11.9s	1.0E-06 (O)	0.044s
Token2	4.8E-04	NL (X)	-	TO (X)	-	5.2E-04 (O)	0.545s

Successful estimation

[Accuracy]

PSE: 15 / 32 **PReach: 17 / 32** **<u>SRA: 32 / 32</u>**

[Time]

PSE: < 1s **PReach: < 1m** **<u>SRA: ~ 0.01s</u>**

Evaluation 2 : Structure-aware Estimator

- Aim: Is the *structural information* useful to better estimate the reaching probability of the *unreached state*?

- Subjects: 5 subjects from Siemens suite + 5 Open-source C libraries
  - Run greybox fuzzing to choose the target *hard-to-reach* statement

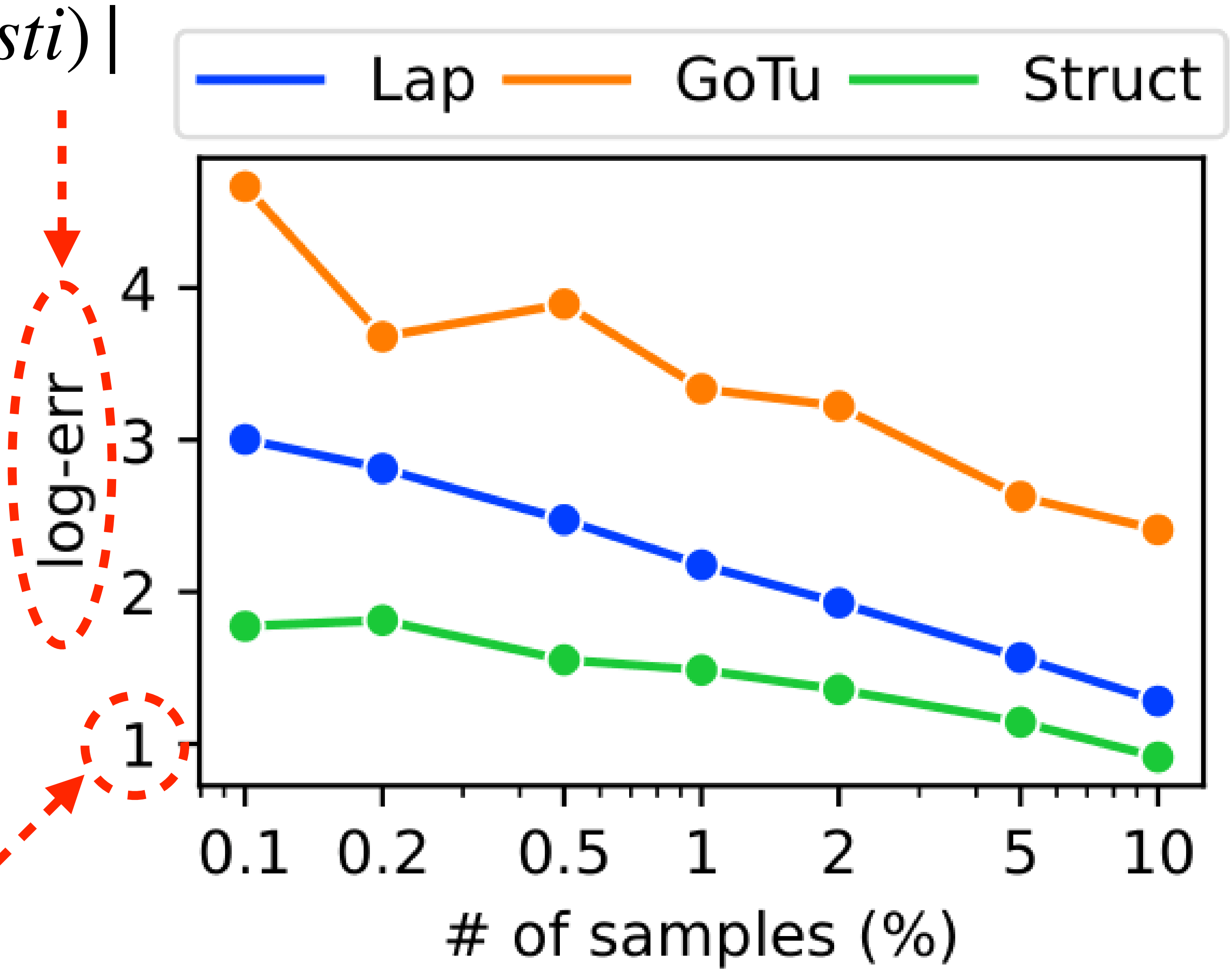
- Evaluation setting:

Program	NCLOC	# Func	# BB	GT
tcas	146	9	63	5.37E-04
schedule2	332	17	138	3.99E-04
totinfo	349	7	132	9.2E-04
printtokens2	438	19	198	7.82E-03
replace	534	21	228	2.73E-04
gif2png*	988	27	700	2.95E-04
jsoncpp	7,251	1,328	5,938	2.28E-03
jasper*	17,385	720	14,417	2.48E-04
readelf	22,347	477	18,578	1.99E-07
freetype2	44,686	1,635	27,521	8.25E-08

Evaluation 2: Structure-aware

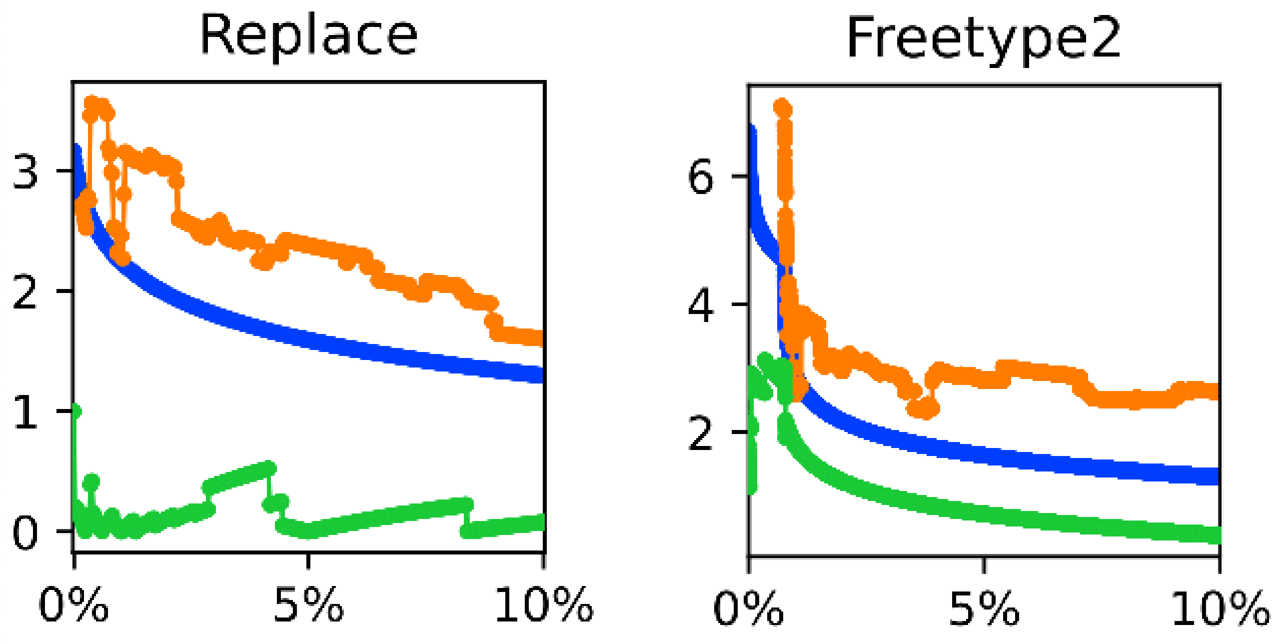
$|\log(GT) - \log(esti)|$

Sample size vs. Bias Curve



log-bias < 1 means one order of magnitude difference

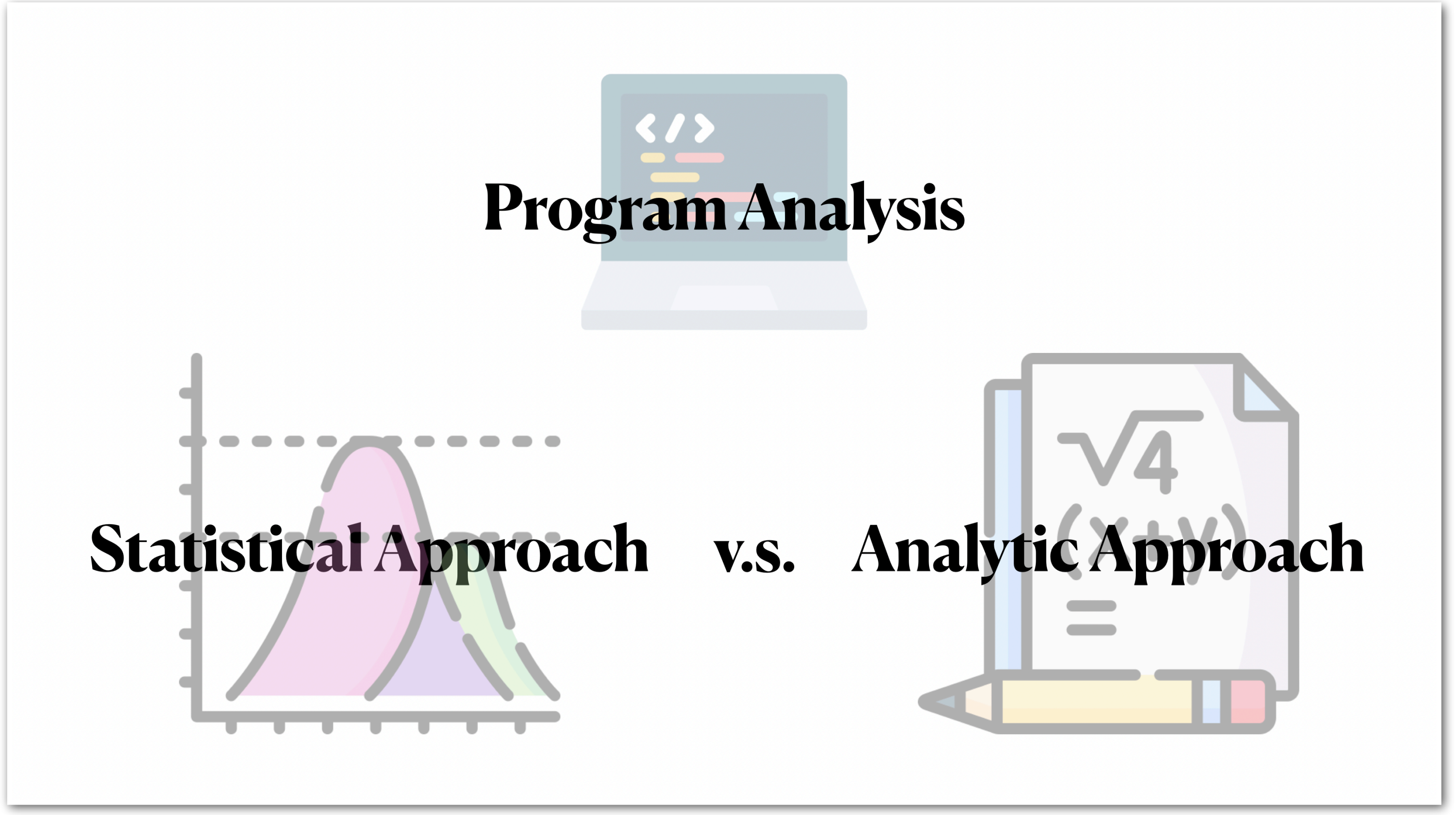
Individual cases



- The ***structure-aware estimator*** performed ***significantly better*** than the blackbox estimators.

log-bias

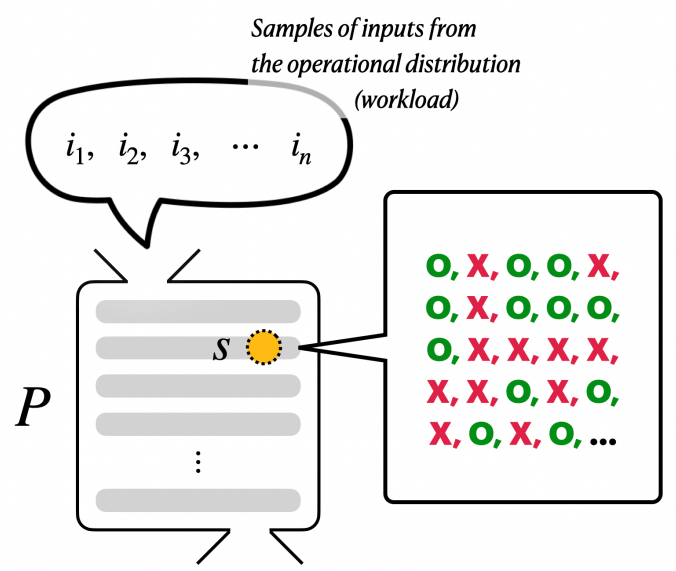
Sample size	*Laplace*	*Good-Turing*	*Struct*
10 %	1.28	2.41	**0.91**
0.01 %	3.00	4.67	**1.77**


Program Analysis
Statistical Approach
v.s.
Analytic Approach

Program Analysis

Statistical Approach v.s. **Analytic Approach**

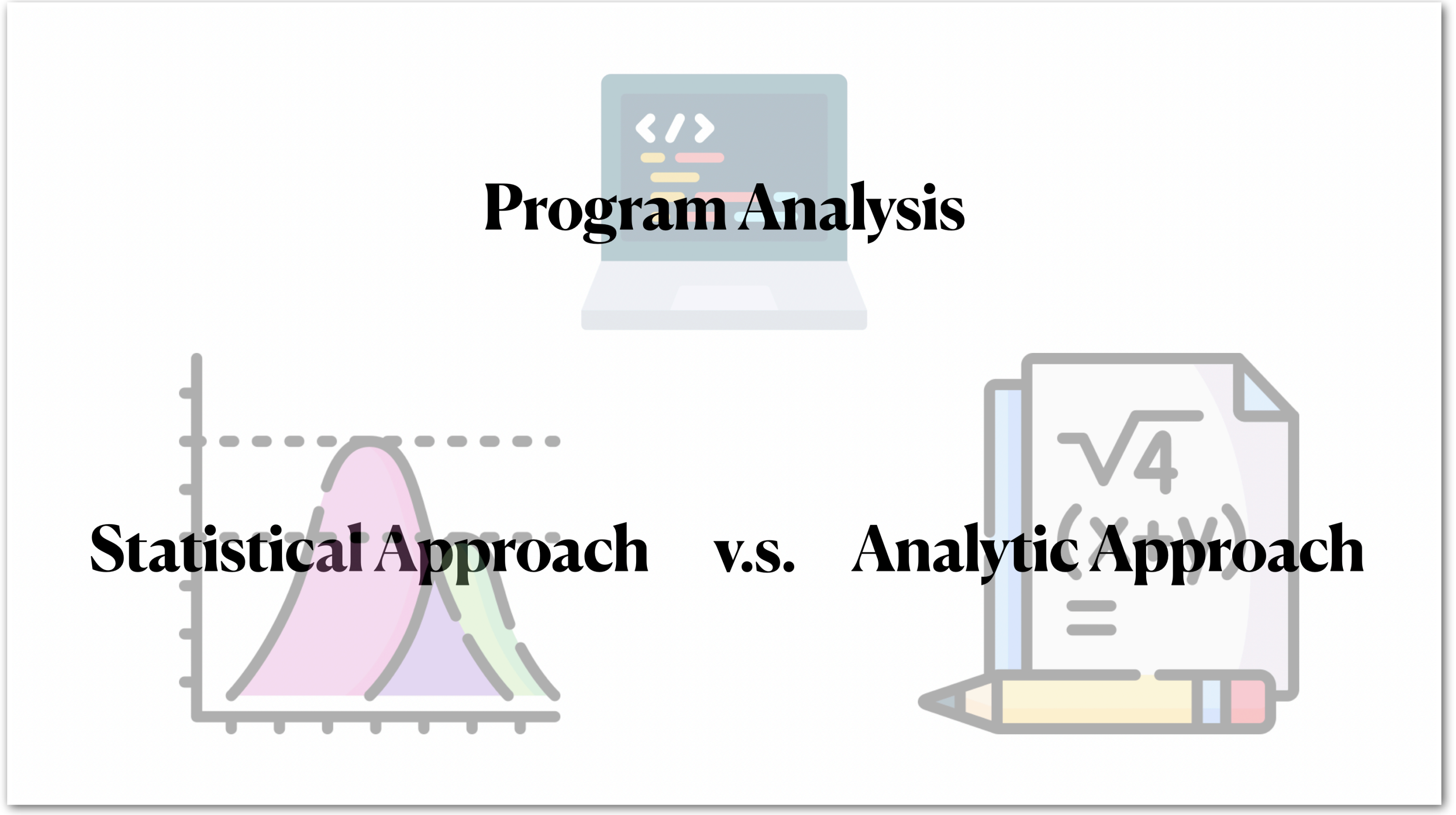
Statistical Reachability Analysis (SRA)

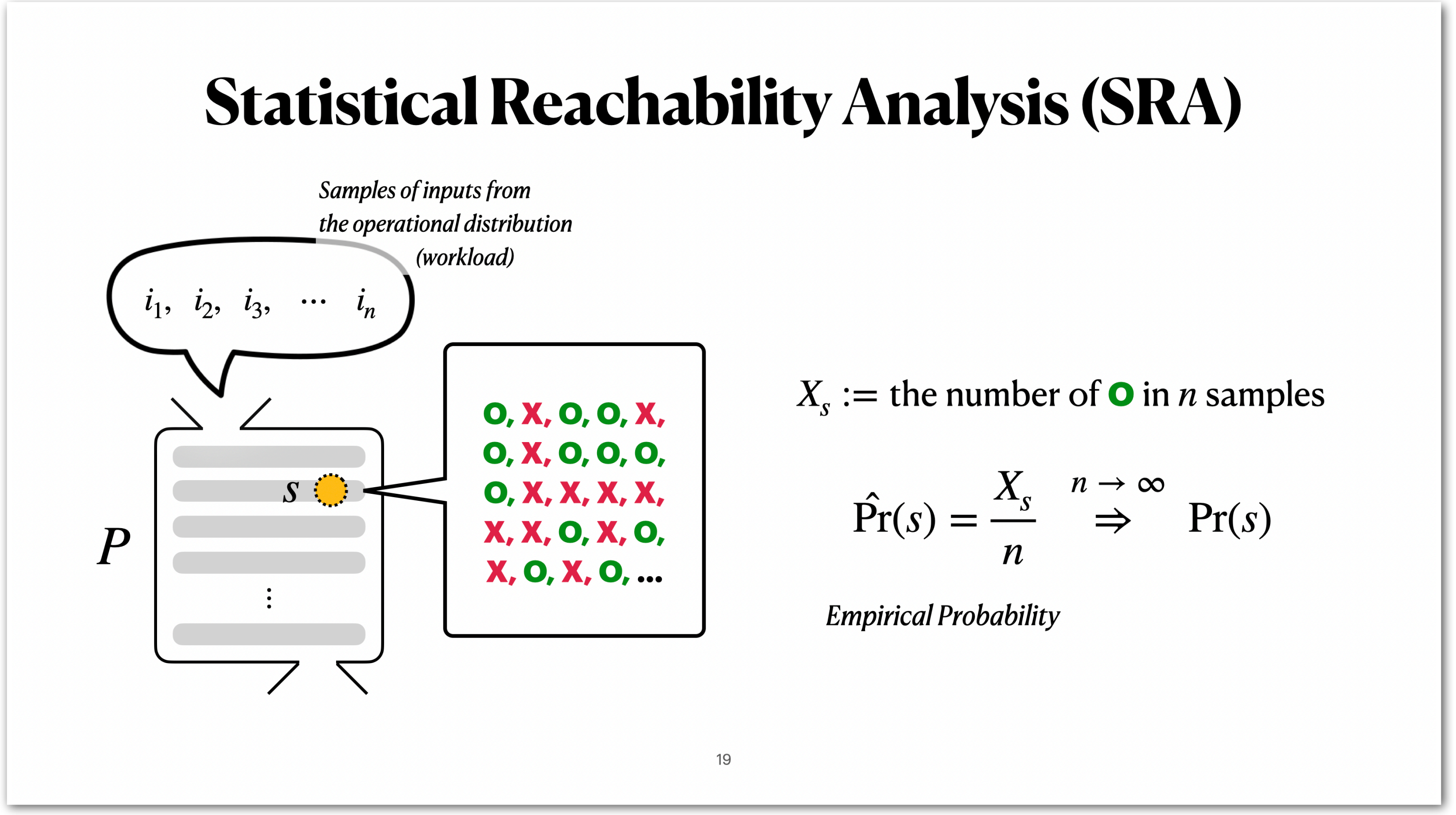


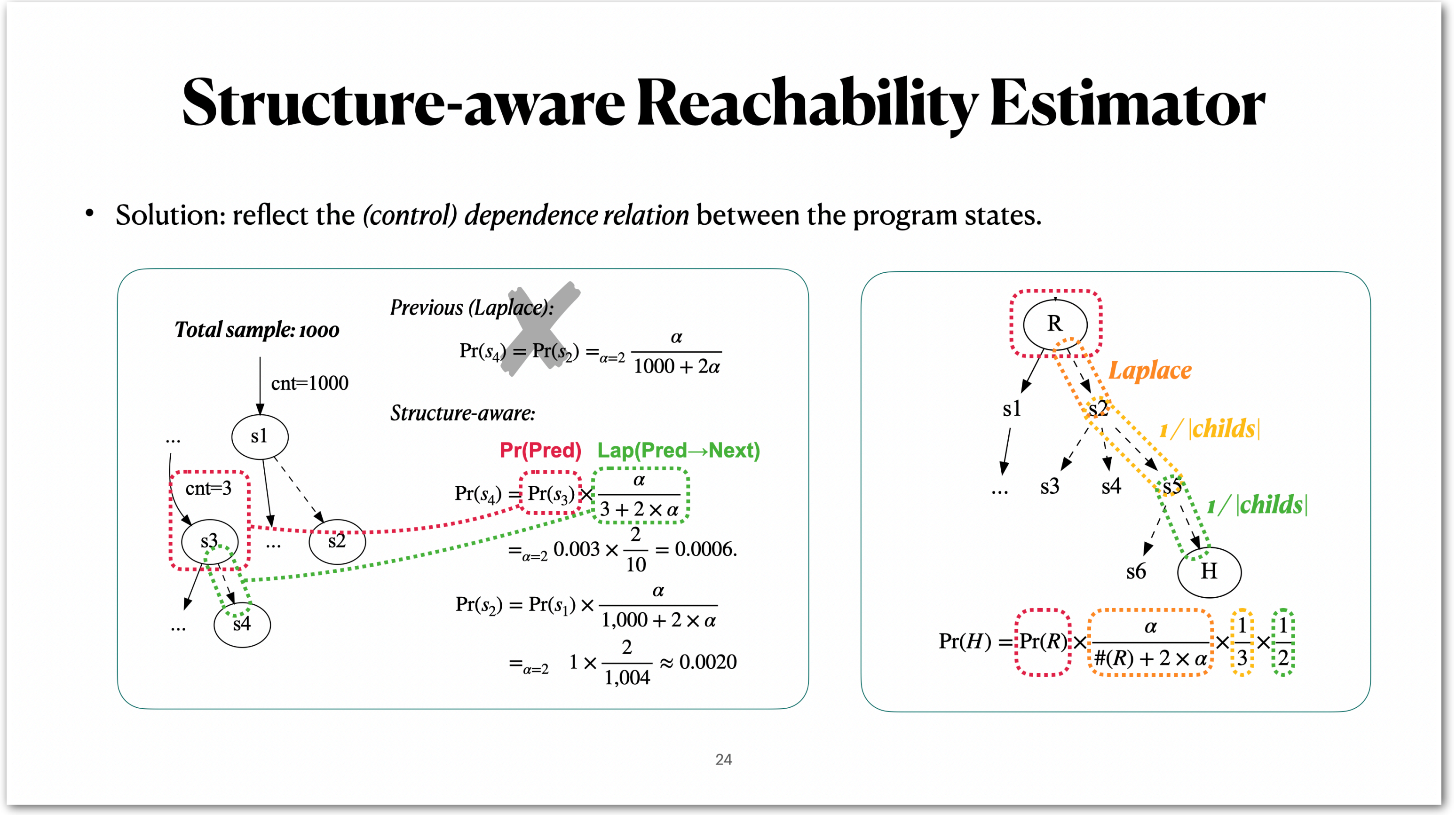
$X_s :=$ the number of O in n samples

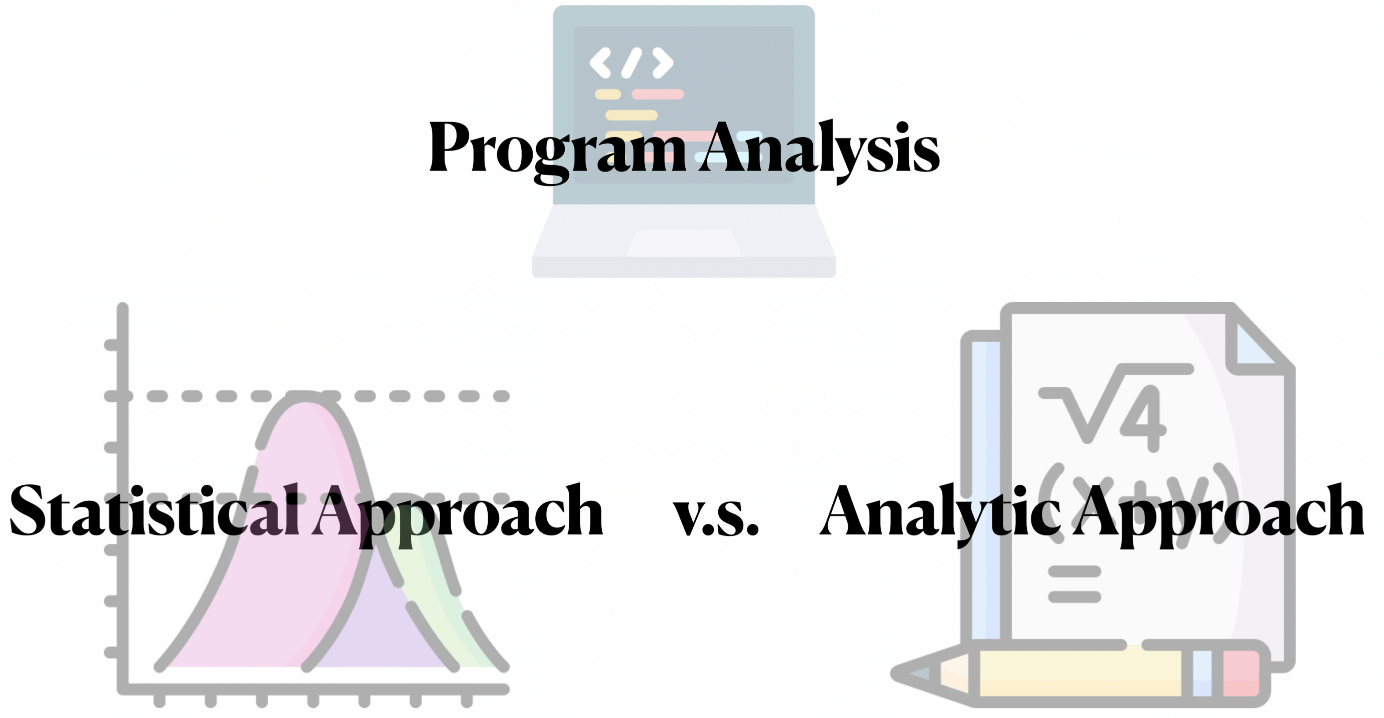
$$\hat{\Pr}(s) = \frac{X_s}{n} \overset{n \to \infty}{\Rightarrow} \Pr(s)$$

Empirical Probability


Program Analysis
Statistical Approach
v.s.
Analytic Approach


Statistical Reachability Analysis (SRA)
Samples of inputs from the operational distribution (workload)
$i_1, i_2, i_3, \cdots i_n$
P
s
O, X, O, O, X,
O, X, O, O, O,
O, X, X, X, X,
X, X, O, X, O,
X, O, X, O, ...
X_s := the number of O in n samples
$\hat{\Pr}(s) = \frac{X_s}{n} \overset{n \to \infty}{\Rightarrow} \Pr(s)$
Empirical Probability
19

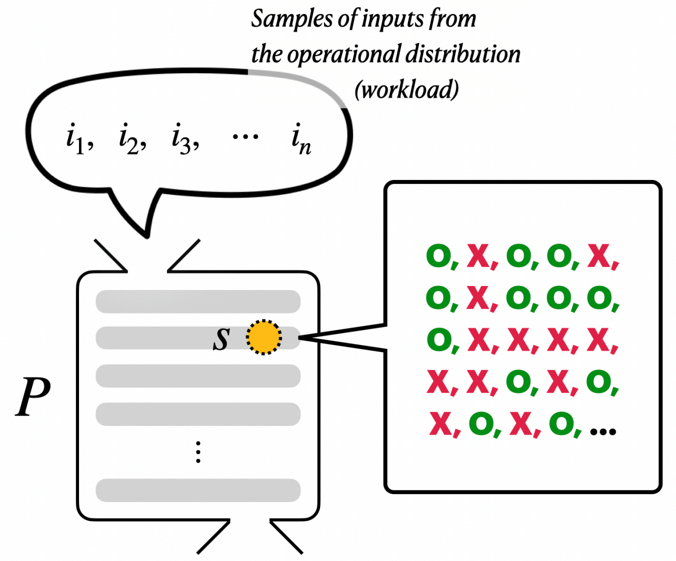

Structure-aware Reachability Estimator
• Solution: reflect the *(control) dependence relation* between the program states.
Total sample: 1000
cnt=1000
s1
cnt=3
s3
s2
s4
Previous (Laplace):
$\Pr(s_4) = \Pr(s_2) =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$
Structure-aware:
Pr(Pred)
Lap(Pred→Next)
$\Pr(s_4) = \Pr(s_3) \times \frac{\alpha}{3 + 2 \times \alpha}$
$=_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$
$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha}$
$=_{\alpha=2} 1 \times \frac{2}{1{,}004} \approx 0.0020$
R
Laplace
s1
s2
1/|childs|
s3
s4
s5
1/|childs|
s6
H
$\Pr(H) = \Pr(R) \times \frac{\alpha}{\#(R) + 2 \times \alpha} \times \frac{1}{3} \times \frac{1}{2}$
24



Program Analysis

Statistical Approach v.s. **Analytic Approach**

Statistical Reachability Analysis (SRA)



X_s := the number of **O** in n samples

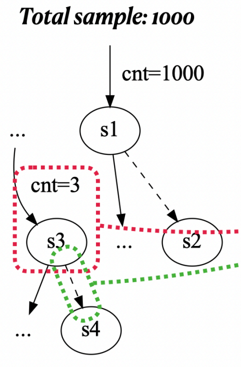
$$\hat{\Pr}(s) = \frac{X_s}{n} \overset{n \to \infty}{\Rightarrow} \Pr(s)$$

Empirical Probability

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.

Total sample: 1000



Previous (Laplace):

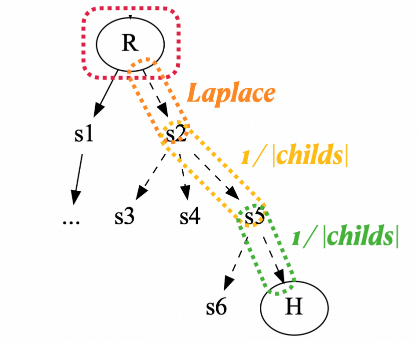
$$\cancel{\Pr(s_4) = \Pr(s_2)} =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

Pr(Pred) **Lap(Pred→Next)**

$$\Pr(s_4) = \boxed{\Pr(s_3)} \times \boxed{\frac{\alpha}{3 + 2 \times \alpha}} =_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha} =_{\alpha=2} 1 \times \frac{2}{1{,}004} \approx 0.0020$$



$$\Pr(H) = \boxed{\Pr(R)} \times \boxed{\frac{\alpha}{\#(R) + 2 \times \alpha}} \times \boxed{\frac{1}{3}} \times \boxed{\frac{1}{2}}$$

Evaluation 1: Statistical vs Analytic

Program	GT	Esti(PSE)	T(PSE)	Esti(PR)	T(PR)	Esti(Lap)	T(Lap)
ExMIT-T	~0	4.7E-10 (O)	.866s	7.6E-06 (O)	14.9s	1.0E-06 (O)	0.044s
Exe1-F	0.49	NL (X)	-	0.500 (O)	13.5s	0.489 (O)	0.006s
Exe2-F	0.2	NL (X)	-	0.125 (X)	14.6s	0.199 (O)	0.003s
Exe4-F	0.25	NL (X)	-	0.125 (X)	14.7s	0.248 (O)	0.014s
Exe6-F	1.0	NL (X)	-	2.3E-10 (X)	14.8s	0.990 (O)	0.001s
Exe8-F	0.3	NL (X)	-	0.500 (X)	14.7s	0.300 (O)	0.005s
Exe10-F	0.25	NL (X)	-	0.250 (O)	14.5s	0.250 (O)	0.005s
Exe10-T	~0	NL (X)	-	1.2E-10 (O)	14.5s	1.0E-06 (O)	0.085s
Exe12-F	0.5	0.500 (O)	.934s	0.500 (O)	14.6s	0.501 (O)	0.004s
Exe12-T	0.375	0.250 (X)	.966s	0.375 (O)	14.6s	0.376 (O)	0.007s
Exe13-T	~0	0 (O)	.909s	5.0E-11 (O)	13.7s	1.0E-06 (O)	0.087s
Exe14-T	0.25	0.5 (X)	.860s	0.25 (O)	11.9s	0.251 (O)	0.018s
Exe15-T	0.25	0.125 (X)	.910s	0.25 (O)	13.1s	0.251 (O)	0.011s
Exe18-F	0.5	NL (X)	-	0.500 (O)	14.5s	0.502 (O)	0.011s
Exe19-T	0.25	0.375 (X)	.950s	0.245 (O)	14.5s	0.251 (O)	0.015s
Exe20-F	0.25	NL (X)	-	0.125 (X)	13.6s	0.249 (O)	0.008s
Exe20-T	0.5	0.500 (O)	.903s	0.5 (O)	14.5s	0.500 (O)	0.008s
Exe26-F	0.5	NL (X)	-	0.245 (X)	14.7s	0.500 (O)	0.006s
Exe27-F	0.5	0.500 (O)	.849s	0.500 (O)	14.7s	0.500 (O)	0.004s
FNEG-T	0	0 (O)	.850s	0.25 (X)	14.5s	1.0E-06 (O)	0.045s
LCMP-T	0	0 (O)	.832s	0.5 (X)	14.9s	1.0E-06 (O)	0.044s
Simple-F	0	0 (O)	.854s	TO (X)	-	1.0E-06 (O)	0.048s
Simple-T	0	0 (O)	.844s	TO (X)	-	1.0E-06 (O)	0.047s
Suzette-F	0.25	0.250 (O)	.910s	4.7E-10 (X)	13.8s	0.249 (O)	0.030s
Suzette-T	~0	2.6E-9 (O)	.926s	2.6E-09 (O)	14.4s	1.0E-06 (O)	0.084s
Assign-T	0	0 (O)	.841s	0.25 (X)	14.6s	1.0E-06 (O)	0.045s
InsertSort2	2.1E-02	TO (X)	-	2.5E-11 (X)	15.8s	2.1E-02 (O)	4,904s
RBTree1	0.125	TO (X)	-	DTMC (X)	14.4s	0.124 (O)	0.002s
assert3	~0	4.7E-10 (O)	.847s	2.3E-10 (O)	10.6s	1.0E-06 (O)	0.044s
if_icmp1	0	0 (O)	.856s	5.0E-11 (O)	10.5s	1.0E-06 (O)	0.045s
switch1	~0	2.8-09 (O)	1.03s	0.0 (O)	11.9s	1.0E-06 (O)	0.044s
Token2	4.8E-04	NL (X)	-	TO (X)	-	5.2E-04 (O)	0.545s

Successful estimation

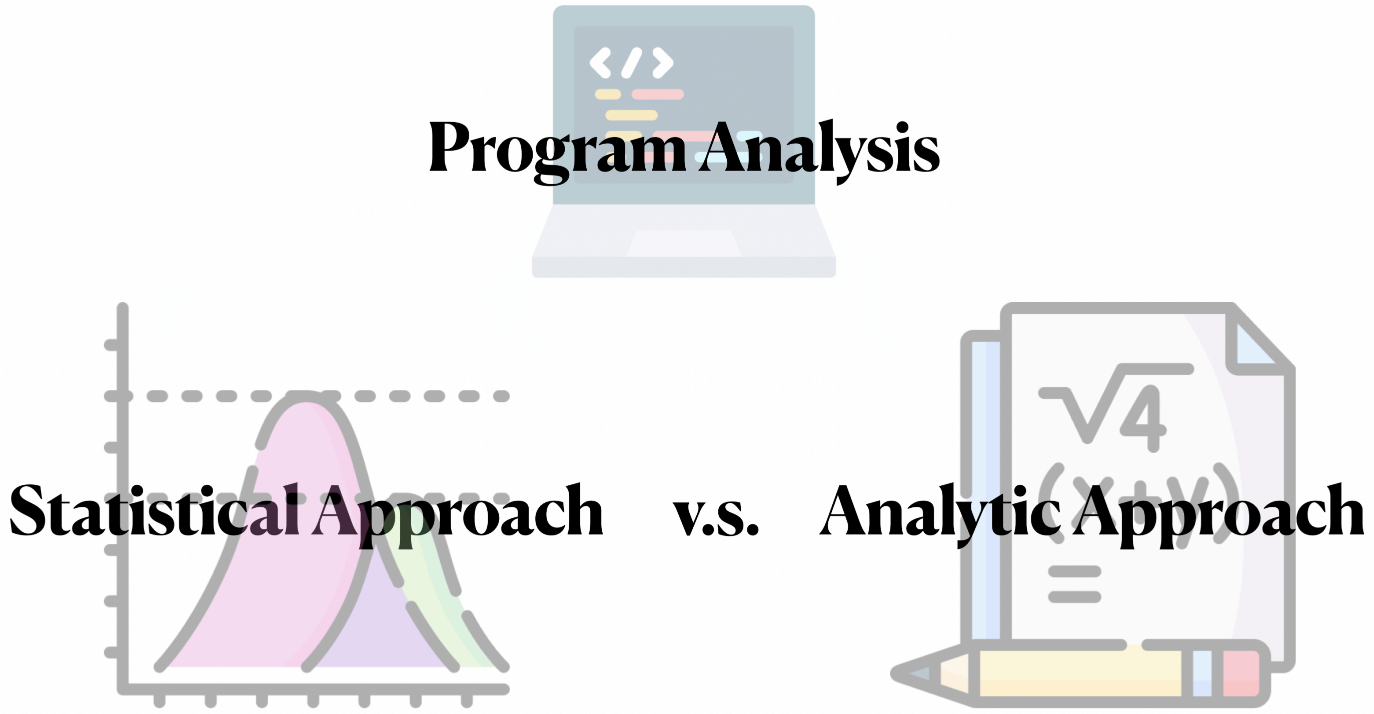
[Accuracy]

PSE: 15 / 32　PReach: 17 / 32　<u>SRA: 32 / 32</u>

[Time]

PSE: < 1s　PReach: < 1m　<u>SRA: ~ 0.01s</u>

MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

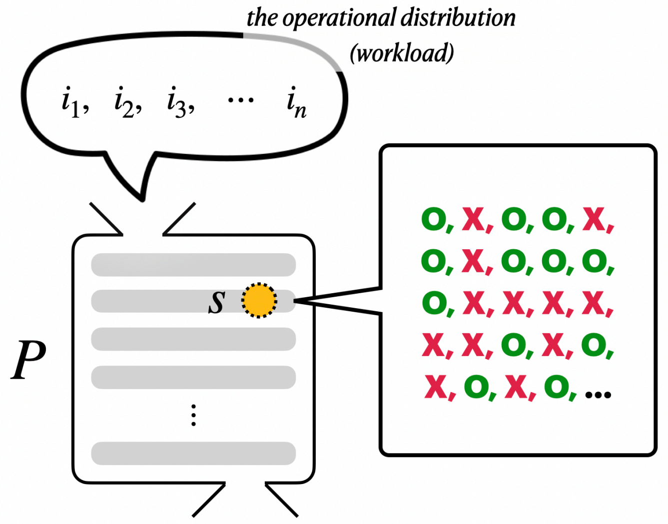


Program Analysis

Statistical Approach v.s. Analytic Approach

Statistical Reachability Analysis (SRA)

Samples of inputs from the operational distribution (workload)



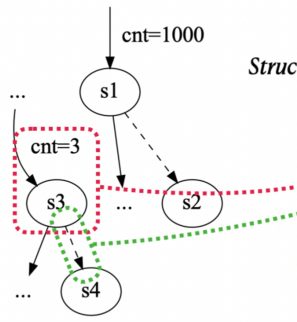
X_s := the number of **O** in n samples

$$\hat{\Pr}(s) = \frac{X_s}{n} \overset{n \to \infty}{\Rightarrow} \Pr(s)$$

Empirical Probability

Structure-aware Reachability Estimator

- Solution: reflect the *(control) dependence relation* between the program states.



Previous (Laplace):

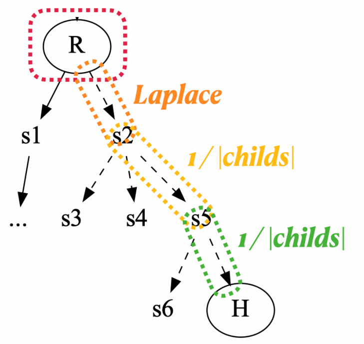
$$\cancel{\Pr(s_4) = \Pr(s_2)} =_{\alpha=2} \frac{\alpha}{1000 + 2\alpha}$$

Structure-aware:

Pr(Pred) Lap(Pred→Next)

$$\Pr(s_4) = \Pr(s_3) \times \frac{\alpha}{3 + 2 \times \alpha} =_{\alpha=2} 0.003 \times \frac{2}{10} = 0.0006.$$

$$\Pr(s_2) = \Pr(s_1) \times \frac{\alpha}{1{,}000 + 2 \times \alpha} =_{\alpha=2} 1 \times \frac{2}{1{,}004} \approx 0.0020$$



$$\Pr(H) = \Pr(R) \times \frac{\alpha}{\#(R) + 2 \times \alpha} \times \frac{1}{3} \times \frac{1}{2}$$

Evaluation 1: Statistical vs Analytic

Program	GT	Esti(PSE)	T(PSE)	Esti(PR)	T(PR)	Esti(Lap)	T(Lap)
ExMIT-T	~0	4.7E-10 (O)	.866s	7.6E-06 (O)	14.9s	1.0E-06 (O)	0.044s
Exe1-F	0.49	NL (X)	-	0.500 (O)	13.5s	0.489 (O)	0.006s
Exe2-F	0.2	NL (X)	-	0.125 (X)	14.6s	0.199 (O)	0.003s
Exe4-F	0.25	NL (X)	-	0.125 (X)	14.7s	0.248 (O)	0.014s
Exe6-F	1.0	NL (X)	-	2.3E-10 (X)	14.8s	0.990 (O)	0.001s
Exe8-F	0.3	NL (X)	-	0.500 (X)	14.7s	0.300 (O)	0.005s
Exe10-F	0.25	NL (X)	-	0.250 (O)	14.5s	0.250 (O)	0.005s
Exe10-T	~0	NL (X)	-	1.2E-10 (O)	14.5s	1.0E-06 (O)	0.085s
Exe12-F	0.5	0.500 (O)	.934s	0.500 (O)	14.6s	0.501 (O)	0.004s
Exe12-T	0.375	0.250 (X)	.966s	0.375 (O)	14.6s	0.376 (O)	0.007s
Exe13-T	~0	0 (O)	.909s	5.0E-11 (O)	13.7s	1.0E-06 (O)	0.087s
Exe14-T	0.25	0.5 (X)	.860s	0.25 (O)	11.9s	0.251 (O)	0.018s
Exe15-T	0.25	0.125 (X)	.910s	0.25 (O)	13.1s	0.251 (O)	0.011s
Exe18-F	0.5	NL (X)	-	0.500 (O)	14.5s	0.502 (O)	0.011s
Exe19-T	0.25	0.375 (X)	.950s	0.245 (O)	14.5s	0.251 (O)	0.015s
Exe20-F	0.25	NL (X)	-	0.125 (X)	13.6s	0.249 (O)	0.008s
Exe20-T	0.5	0.500 (O)	.903s	0.5 (O)	14.5s	0.500 (O)	0.008s
Exe26-F	0.5	NL (X)	-	0.245 (X)	14.7s	0.500 (O)	0.006s
Exe27-F	0.5	0.500 (O)	.849s	0.500 (O)	14.7s	0.500 (O)	0.004s
FNEG-T	0	0 (O)	.850s	0.25 (X)	14.5s	1.0E-06 (O)	0.045s
LCMP-T	0	0 (O)	.832s	0.5 (X)	14.9s	1.0E-06 (O)	0.044s
Simple-F	0	0 (O)	.854s	TO (X)	-	1.0E-06 (O)	0.048s
Simple-T	0	0 (O)	.844s	TO (X)	-	1.0E-06 (O)	0.047s
Suzette-F	0.25	0.250 (O)	.910s	4.7E-10 (X)	13.8s	0.249 (O)	0.030s
Suzette-T	~0	2.6E-9 (O)	.926s	2.6E-09 (O)	14.4s	1.0E-06 (O)	0.084s
Assign-T	0	0 (O)	.841s	0.25 (X)	14.6s	1.0E-06 (O)	0.045s
InsertSort2	2.1E-02	TO (X)	-	2.5E-11 (X)	15.8s	2.1E-02 (O)	4,904s
RBTree1	0.125	TO (X)	-	DTMC (X)	14.4s	0.124 (O)	0.002s
assert3	~0	4.7E-10 (O)	.847s	2.3E-10 (O)	10.6s	1.0E-06 (O)	0.044s
if_icmp1	0	0 (O)	.856s	5.0E-11 (O)	10.5s	1.0E-06 (O)	0.045s
switch1	~0	2.8-09 (O)	1.03s	0.0 (O)	11.9s	1.0E-06 (O)	0.044s
Token2	4.8E-04	NL (X)	-	TO (X)	-	5.2E-04 (O)	0.545s

Successful estimation

[Accuracy]

PSE: 15 / 32 PReach: 17 / 32 **SRA: 32 / 32**

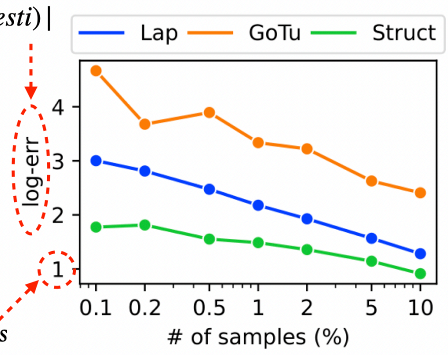
[Time]

PSE: < 1s PReach: < 1m **SRA: ~ 0.01s**

Evaluation 2: Structure-aware

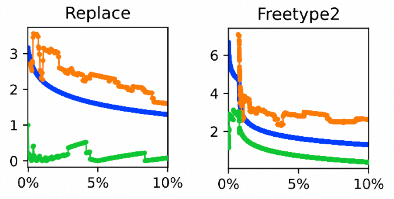
$|\log(GT) - \log(esti)|$

Sample size vs. Bias Curve



log-bias < 1 means one order of magnitude difference

Individual cases

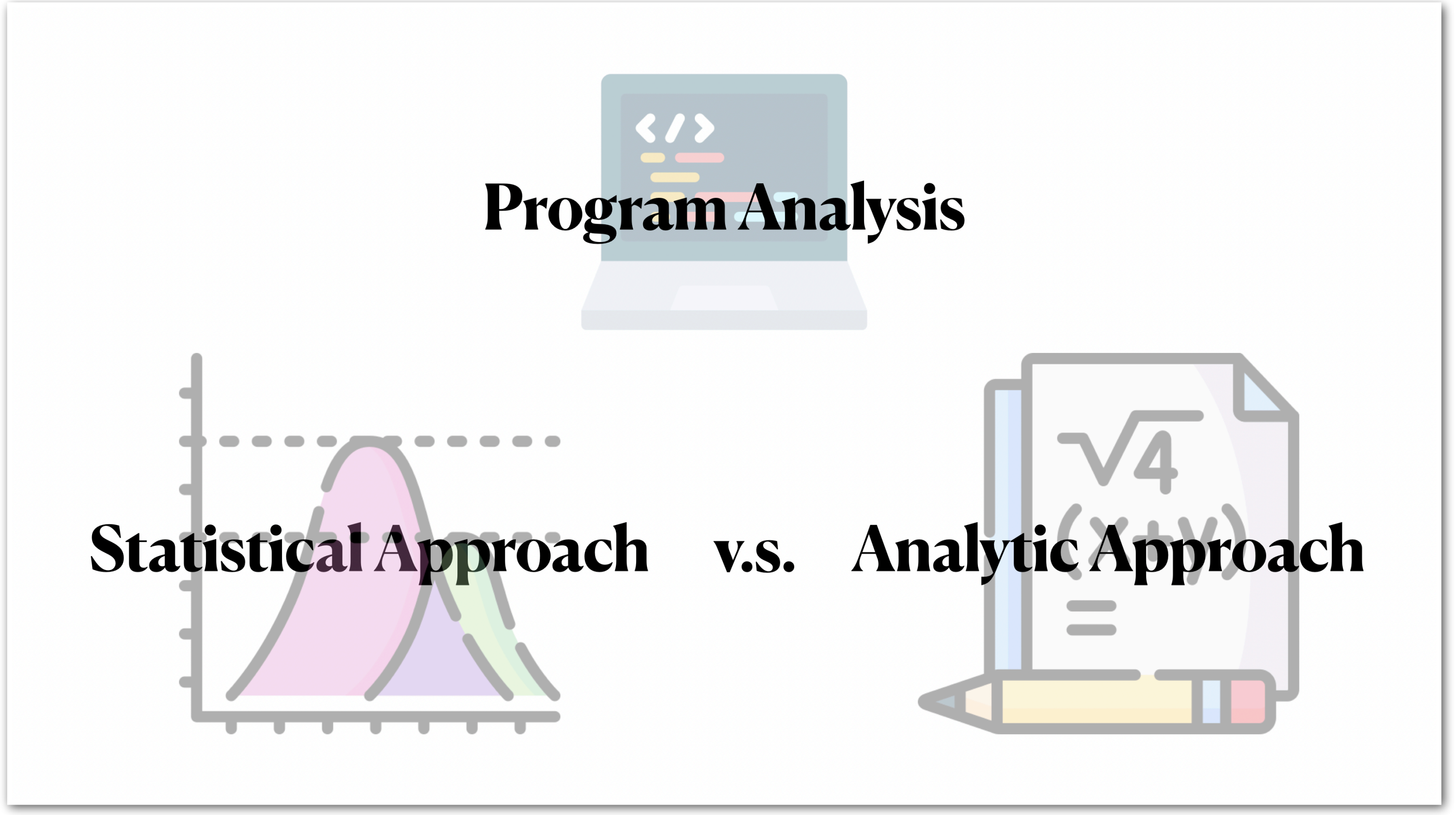


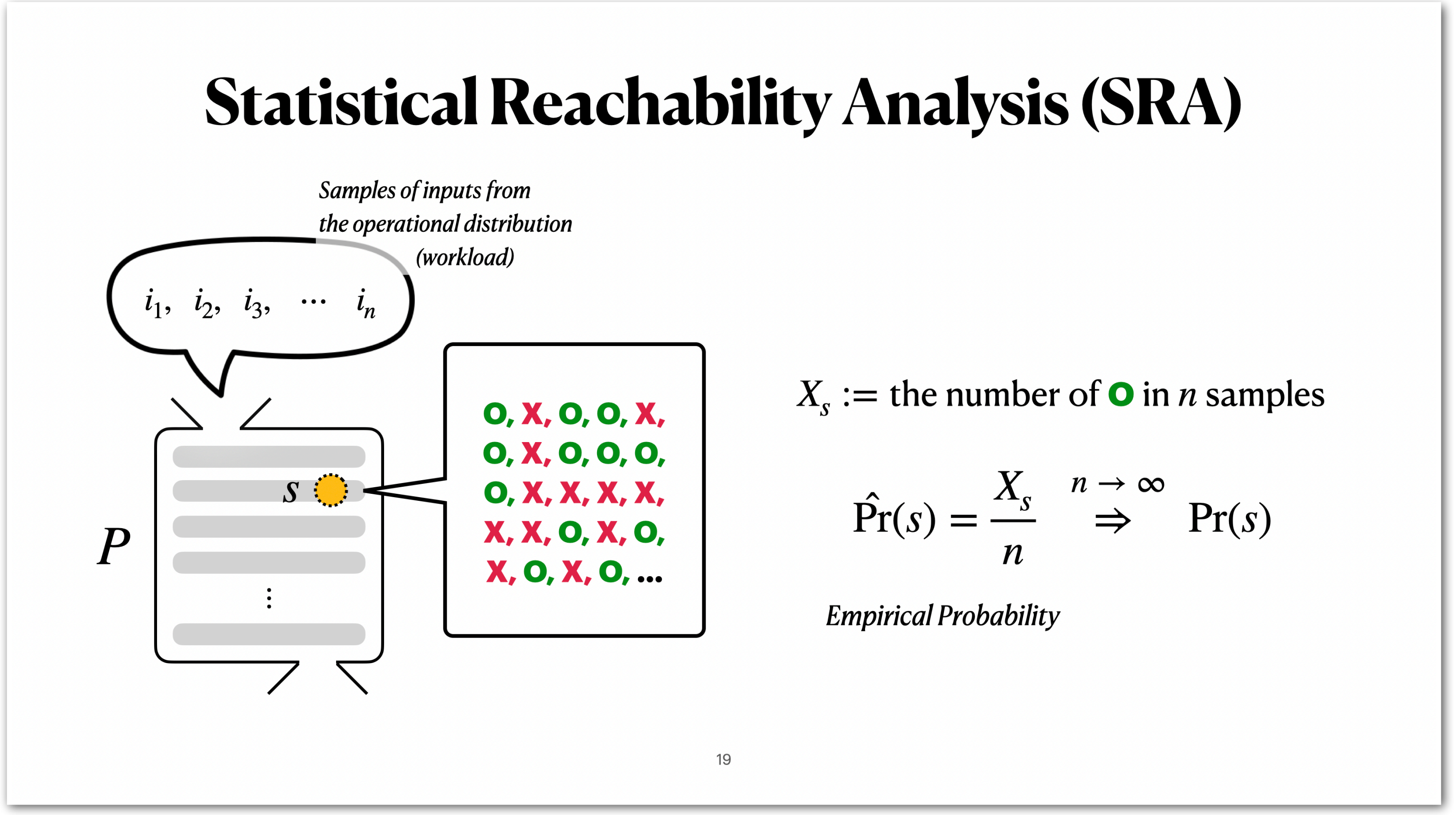
- The ***structure-aware estimator*** performed ***significantly better*** than the blackbox estimators.

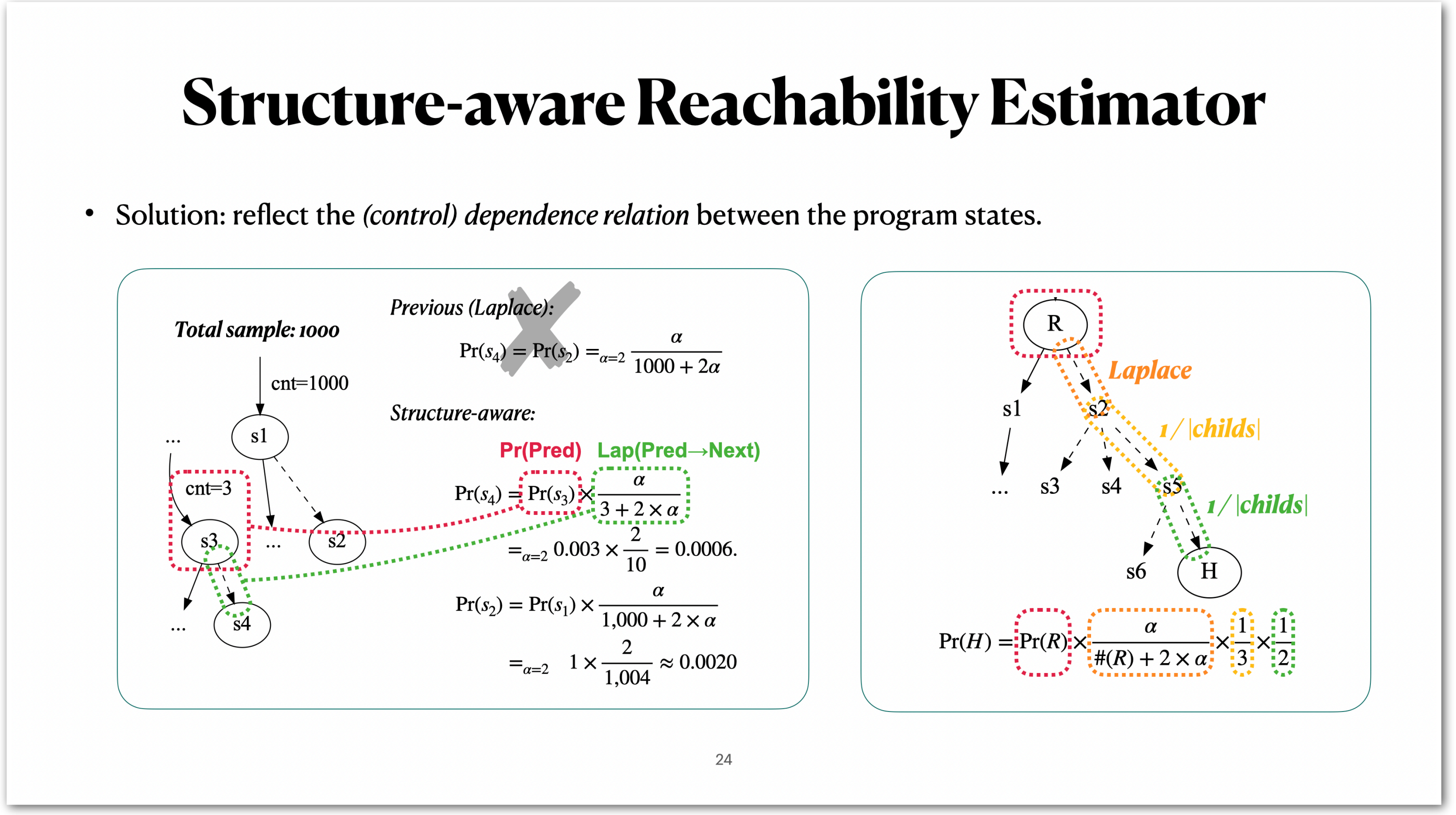
log-bias

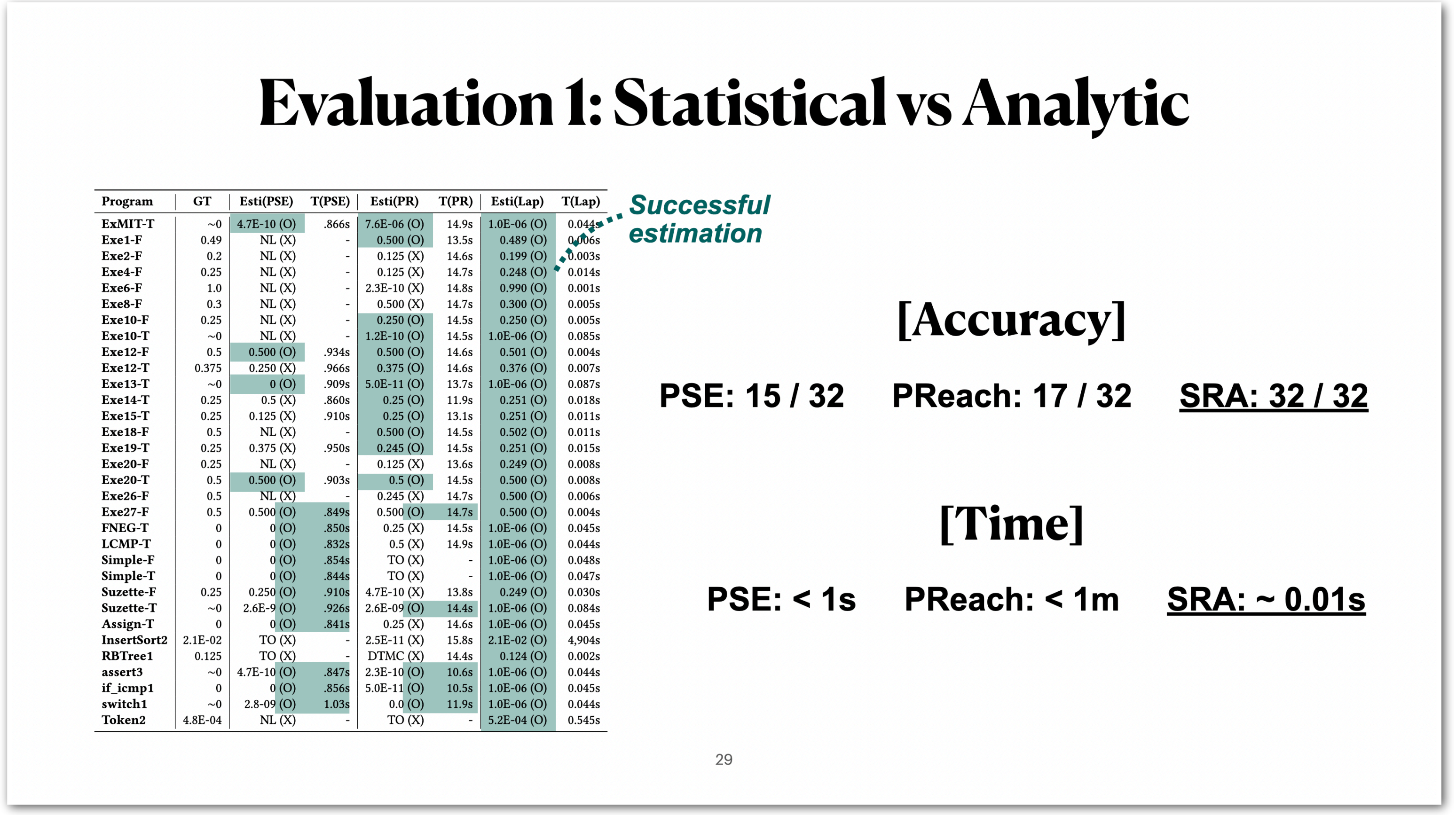
Sample size	Laplace	Good-Turing	Struct
10 %	1.28	2.41	**0.91**
0.01 %	3.00	4.67	**1.77**

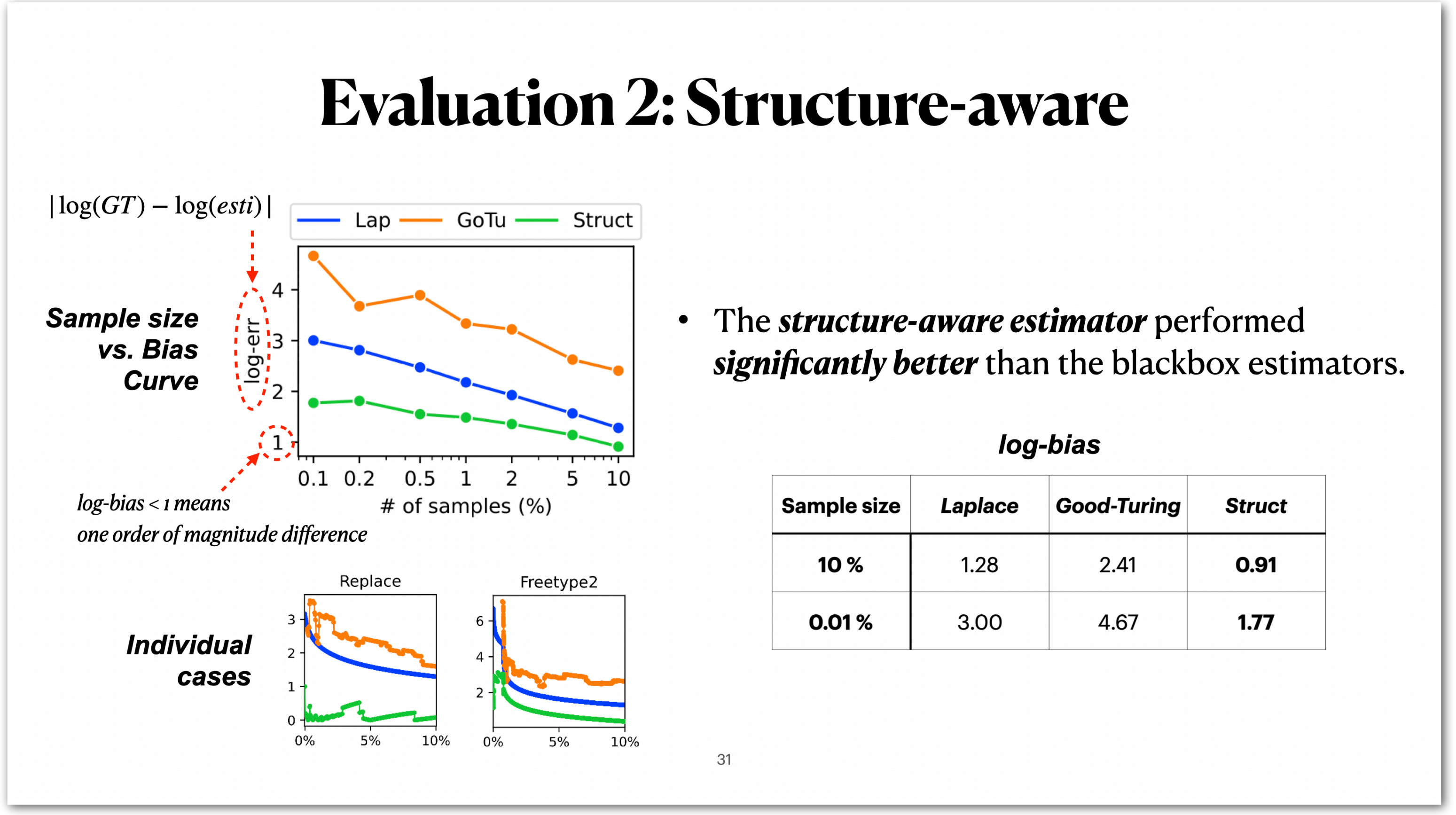
MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY











Dr. Seongmin Lee

🏠 https://nimgnoeseel.github.io/

Dr. Marcel Böhme

MPI-SP Software Security

🏠 https://mpi-softsec.github.io/