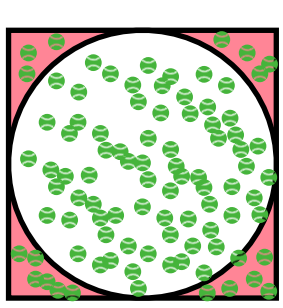
Statistical Reachability Analysis

MAX PLANCK INSTITUTE FOR SECURITY AND PRIVACY

Seongmin Lee and Marcel Böhme

STATISTICAL METHOD

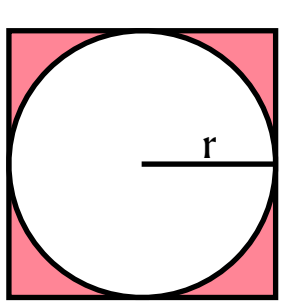


E.g., Monte Carlo method

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

The statistical method is useful
- when the target **model is inaccurate/unknown**,
- as it performs **complexity-agnostic** to the model,
- to **approximate** the quantity.

ANALYTICAL METHOD



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

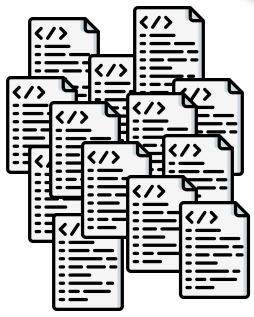
The analytic method is useful if
- the target model is **exactly known**,
- scale/complexity of the target **model is small/moderate**,
- and the **exact quantity is required**.

More feasible!

🎯 TARGET ANALYSIS: QUANTITATIVE REACHABILITY

Q. What is the probability of reaching a certain program state?

Many software verification task is answered by the reaching probability.
- The probability of reaching an erroneous state.



MODERN SOFTWARES

Analysis of the modern software faces
- an industrial scale huge code base
- heterogenous in-analyzable features, e.g., 3rd party library/binary
- cross-language/inter-procedural interface
- a nature of undecidability

RESEARCH GOAL

To investigate the performance of *statistical* method vs. *analytical* method for program analysis tasks.

STATISTICAL REACHABILITY ANALYSIS (SRA)

- $\Pr(s)$: A probability of an arbitrary program execution reaching the target state $s \in S$

Challenge of SRA

"How can it deal with the unobserved state s?"

If the target state is unobserved from the samples, the empirical probability is 0, which may lead to a *false positive* result.

Existing blackbox estimators

Given program execution samples O for analysis,

1. Laplace estimator

$$Lap(s,O) = \frac{|\{o \in O \mid REACH(o,s)\}| + \alpha}{|O| + \alpha},$$

where α is a smoothing parameter.

2. Good-Turing estimator

$$GoTu(s,n) = \begin{cases} \frac{|\{o \in O \mid REACH(o,s)\}|}{|O|}, & \text{if } c_s > 0, \\ \frac{f_1(O)}{|O|}, & \text{otherwise,} \end{cases}$$

where $f_1(O)$ is a number of singleton events (events that happen only once).

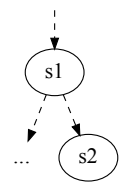
STRUCTURE-AWARE SRA

Limitation of existing estimators

Black-box estimators are unaware of the semantics of the program.

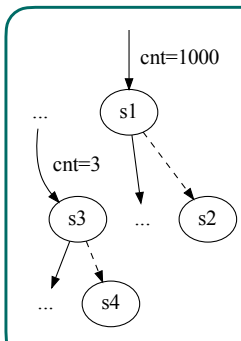
$\Pr(s_1) \geq \Pr(s_2)$; yet,
- $Lap(s_1, O) = Lap(s_2, O)$
- $GoTu(s_1, O) = GoTu(s_2, O))$

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



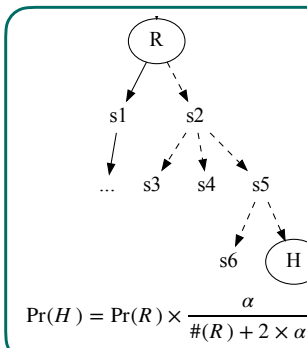
Structure-aware estimator

To address the limitation, we design a structure-aware estimator that reflects the *dependence relation* between the program states.



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

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



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

EMPIRICAL EVALUATION

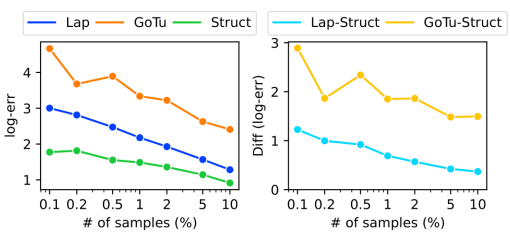
Evaluation 1. Statistical method vs. Analytic method
- How accurate is the estimated probability?
- How efficient is the estimation?
- Subject programs: 142 Java programs from Competition on Software Verification 2021
- Baseline: two analytic estimators PSE, Preach

Evaluation 2. Black-box vs. Structure-aware
- How fast can the estimator be closer to the ground-truth reaching probability?
- Subject programs: 5 middle-size (Siemens suite) + 5 large-size (open-source) C programs

Evaluation 1

Program	GT	Est₀(PSE)	T(PSE)	Est₀(PR)	T(PR)	Est₀(Lap)	T(Lap)
ExMIT-T	~0	4.7E-10 (O)	.866s	7.4E-06 (O)	14.9s	1.0E-06 (O)	0.044s
Exc1-F	0.49	NL (X)	-	0.500 (O)	13.5s	0.489 (O)	0.006s
Exc2-F	0.2	NL (X)	-	0.125 (X)	14.6s	0.199 (O)	0.005s
Exc4-F	0.25	NL (X)	-	0.125 (X)	14.7s	0.248 (O)	0.014s
Exc6-F	1.0	NL (X)	-	2.5E-10 (X)	14.8s	0.990 (O)	0.001s
Exc8-F	0.5	NL (X)	-	0.500 (X)	14.7s	0.500 (O)	0.005s
Exc10-F	0.25	NL (X)	-	0.250 (O)	14.5s	0.250 (O)	0.005s
Exc10-T	~0	NL (X)	-	1.2E-10 (O)	14.5s	1.0E-06 (O)	0.005s
Exc12-F	0.5	0.500 (O)	.934s	0.500 (O)	14.6s	0.501 (O)	0.004s
Exc12-T	0.375	0.250 (X)	.966s	0.375 (O)	14.6s	0.376 (O)	0.007s
Exc13-T	~0	0 (O)	.909s	9.4E-11 (O)	15.7s	1.0E-06 (O)	0.007s
Exc14-T	0.25	0.5 (X)	.868s	0.25 (O)	11.9s	0.251 (O)	0.010s
Exc15-T	0.25	0.125 (X)	.958s	0.25 (O)	13.1s	0.251 (O)	0.011s
Exc18-F	0.5	NL (X)	-	0.500 (O)	14.5s	0.502 (O)	0.011s
Exc19-T	0.25	0.375 (X)	.958s	0.245 (O)	14.5s	0.251 (O)	0.015s
Exc20-F	0.25	NL (X)	-	0.125 (X)	13.6s	0.249 (O)	0.008s
Exc20-T	0.5	0.500 (O)	.903s	0.5 (X)	14.5s	0.500 (O)	0.008s
Exc26-F	0.5	NL (X)	-	0.245 (X)	14.7s	0.500 (O)	0.004s
Exc27-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)	0.043s
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
Sunette-F	0.25	0.250 (O)	.928s	4.7E-10 (X)	13.8s	0.249 (O)	0.039s
Sunette-T	~0	2.4E-9 (O)	.928s	2.4E-09 (O)	14.4s	1.0E-06 (O)	0.039s
Assign-T	0	0 (O)	.841s	0.25 (X)	14.6s	1.0E-06 (O)	0.040s
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	2.5E-10 (O)	10.6s	1.0E-06 (O)	0.044s
if_jump1	0	0 (O)	.850s	9.4E-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)	-	2.2E-04 (O)	0.045s

Evaluation 2



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

Time
- PSE: < 1sec
- PReach: < 1min
- **SRA: ~ 0.01sec**

Error when 10% of samples needed to reach
- Lap: 1.28 orders of magnitude
- GoTu: 2.41 orders of magnitude
- **Struct: 0.91** orders of magnitude

TAKEAWAY

- The statistical methods can ***successfully estimate*** the reaching probability with high precision, generally in a short period of time. On the other hand, the analytical methods often fail to estimate the accurate reaching probability due to their ***scalability issues***.

- Considering the ***semantic information*** of the program, the structure-aware statistical estimation provides a ***more accurate*** estimate than the black-box statistical estimation.