DD2552 - Seminars on Theoretical Computer Science, Programming Languages and Formal Methods, Seminar 5

Karl Palmskog (palmskog@kth.se)

2021-09-13

Last Seminar and Today

Last seminar:

- CTMCs and CSL

Today:

- Effiency of algorithms
- Towards statistical verification using CSL

Continuous Stochastic Logic (CSL)

$$\phi ::= \top \mid a \mid \neg\phi \mid \phi \wedge \phi \mid P_{\geq\theta}(\psi)$$
$$\psi ::= \phi\, U^{\leq t}\, \phi$$
$$t \in R^{\geq 0}, \quad \theta \in [0,1]$$

Deciding CSL formulas on large CTMCs

- we want decide $\mathcal{M}, s \models P_{\geq\theta}(\psi)$
- numerical algorithms compute a measure p on ψ and compare to θ
- computation uses the CTMC rate matrix Q
- may take many iterations over Q to reach certainty
- in the literature, state spaces of size greater than 10^9 considered intractable

What can we do to manage tractability?

- abstract our problem to smaller models
- use sparse matrix representations
- computing probabilities is **too hard**
- symbolic representations and reasoning

Insight: simulations are cheap

- Monte Carlo experiments of stochastic systems has long history
- idea: randomly pick transitions/intervals up to some bound
- result is a **trace** for the system (path prefix)
- generally: each trace is *independent* of another

Insight: simulations abstract from model

- trace analysis and generation can be decoupled
- underlying system could be DTMC, CTMC, or more general process

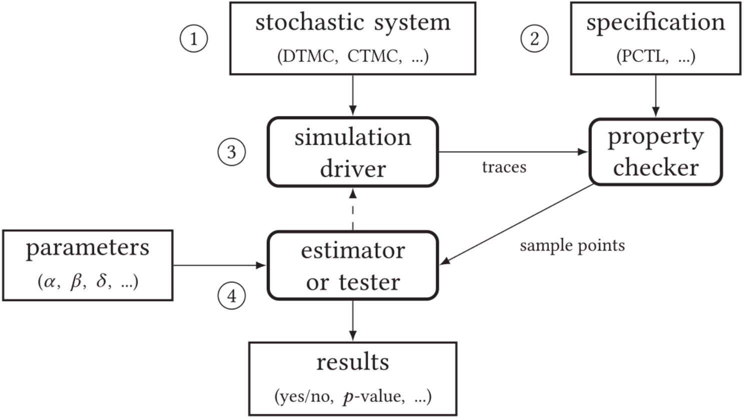
Basic statistical model checking idea

- want to know if $\mathcal{M}, s \models P_{\geq\theta}(\psi)$
- generate trace of $\mathcal{M}$ (prefix of the path π)
- check if ψ holds in the trace or not
- continue to generate traces and keep track of outcomes
- analyze all outcomes and decide if probability that ψ holds is $\geq \theta$ or not

Costs and Benefits

- sampling traces can (usually) never lead to absolute certainty like numerics
- but each trace (usually) cheap to maintain
- in many systems, we can continue to sample until time bound reached or acquired enough certainty

Overview of the statistical model checking process



Probability and error approach

- each trace has a "true" (1) or "false" (0) outcome
- each outcome can be viewed as observation of Bernoulli random variable X with parameter p (underlying probability)
- observations input to either **hypothesis tester** or **estimator**

Hypothesis testing

- we have mutually contradicting hypotheses:
  - $H_0 : p \geq \theta$
  - $H_1 : p < \theta$
- many hypothesis tests available
- tests can take parameters α and β which bound the probability of error
- example: single sampling plan (SSP)
- example: sequential testing

Errors in hypothesis testing

	Decision	
Truth	*accept* H_0, *reject* H_1	*reject* H_0, *accept* H_1
$p \geq \theta$: H_0 *true*, H_1 *false*	correct ($>1-\alpha$)	type I error ($\leq\alpha$)
$p < \theta$: H_0 *false*, H_1 *true*	type II error ($\leq\beta$)	correct ($>1-\beta$)

The conditions inside parentheses are on the probability for the given outcome.

A possible single sampling plan

- determine $n > 0$ consistent with given α and β bounds
- obtain observed outcomes x_i from traces, $1 \leq i \leq n$
- accept H_0 whenever we have

$$\sum_{i=1}^{n} x_i / n \geq \theta$$

- otherwise, accept H_1

Sequential testing

- in SSP, n must be determined beforehand
- sequential tests make “dynamic” decisions after each outcome
- canonical example: the sequential probability ratio test (SPRT)
- SPRT minimizes the sample size under certain assumptions