Verifying probabilistic procedural programs

Javier Esparza

Software Reliability and Security Group

University of Stuttgart

Joint work with Kousha Etessami.

Based on papers by Kousha Etessami and Mihalis Yannakakis,

by Javier Esparza, Antonín Kučera and Richard Mayr,

and by Tomáš Brázdil, Antonín Kučera, and Oldřich Stražovský

Motivation

Model checkers of the first generation (SPIN,SMV,Murphi, . . .) only work for finite-state systems

Programs with recursive procedures may be infinite-state,
even if all variables have a finite range (unbounded call stack)

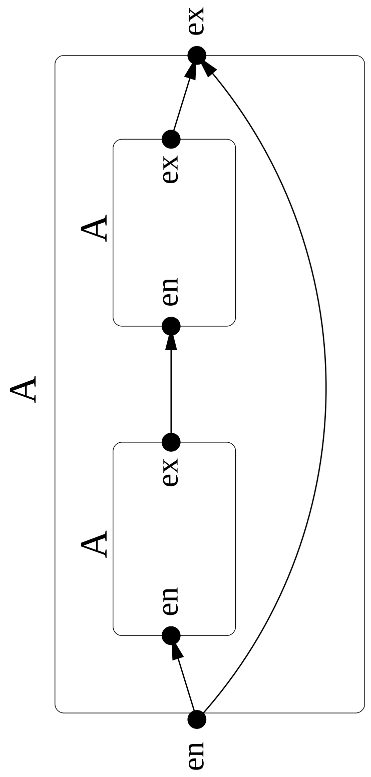
Non-recursive procedure calls can be eliminated using inlining,
but inlining may cause an exponential blow-up in the size of the program.
This is inefficient (and unnecessary)

Goal: Design model checkers that work directly on the procedural representation

Abstract model: recursive state machines (RSMs) and pushdown systems (PDSs)

RSMs and PDSs

Recursive state machines



Pushdown systems

$$pA \hookrightarrow p\epsilon$$
$$pA \hookrightarrow pAA$$

A quick comparison:

- Almost equivalent models (linear translations)
- RSMs make the procedure structure explicit. The structure can be exploited to obtain algorithms with slightly better complexity
  (Alur, Etessami, Yannakakis 01)
- Pushdown systems are more abstract, and have found applications outside the analysis of programs, like authorization problems
  (Schwoon, Jha, Reps, Stubblebine 03)

In this talk we'll use pushdown systems, but only because of the speaker!

Pushdown systems

A pushdown system is a triple (P, Γ, δ), where

- P is a finite set of control locations
- Γ is a finite stack alphabet
- $\delta \subseteq (P \times \Gamma) \times (P \times \Gamma^*)$ is a finite set of rules.

A configuration is a pair $p\alpha$, where $p \in P$, $\alpha \in \Gamma^*$
Semantics: A (possibly infinite) transition system with configurations as states and transitions given by

$$\text{If} \quad pX \hookrightarrow q\alpha \in \delta \quad \text{then} \quad pX\beta \longrightarrow q\alpha\beta \quad \text{for every } \beta \in \Gamma^*$$

Normalization: $|\alpha| \leq 2$, termination only by empty stack

From programs to pushdown systems

State of a procedural program: $(\, g, n, l, (n_1, l_1) \ldots (n_k, l_k)\,)$, where

- g is a valuation of the global variables,
- n is the value of the program pointer,
- l is a valuation of local variables of the current active procedure,
- n_i is a return address, and
- l_i is a saved valuation of the local variables of a calling procedure

Modelled as a configuration $pXY_1 \ldots Y_k$ where

$$p = g \qquad X = (n, l) \qquad Y_i = (n_i, l_i)$$

Correspondence between program statements and rules

procedure call	$pX \hookrightarrow qYX$
return	$pX \hookrightarrow q\varepsilon$
statement	$pX \hookrightarrow qY$

Current state

Efficient algorithms for reachability and model-checking problems

- Alur, Etessami, Yannakakis: Analysis of Recursive State Machines, CAV 2001
- Benedikt, Godefroid, Reps: Model Checking of Unrestricted Hierarchical State Machines, ICALP 2001
- Esparza, Hansel, Rossmanith, Schwoon: Efficient Algorithms for Model Checking Pushdown Systems, CAV 2000

MOPED: A model-checking tool for pushdown systems (Schwoon)

Weighted PDS library: (Schwoon, Reps, Jha)

Applications in different areas:

- Finding security bugs in C programs (Chen, Dean, Wagner)
- Analyzing Java programs and Java bytecode (NASA IS Project and JCAVE project)
- Interprocedural Dataflow Analysis (Reps, Schwoon, Jha)
- Authorization problems (Schwoon, Jha, Reps, Stubblebine)
- Finding bugs in Windows XP drivers (Ball, Rajamani,Schwoon)

Current work on

Counterexample-based abstraction refinement

Tailoring the algorithms for different applications

Extensions of the model in order to handle

- Concurrency
- Dynamic process creation
- Stochastic behaviour (randomized algorithms, stochastic models)

Stochastic verification

Finite Markov chains as model of probabilistic while-programs with finite datatypes

Decidability and complexity problems extensively studied
[Lehman and Shelah 82, Vardi 85, Courcoubetis and Yannakakis 95, ...]

Good tools for finite-state systems (e.g. PRISM)

We introduce probabilistic pushdown systems as a model of procedural programs with finite datatypes

Probabilistic pushdown systems

A probabilistic pushdown system (PPDS) is a tuple $\mathbf{P} = (P, \Gamma, \delta, Prob)$, where

- (P, Γ, δ) is a PDS, and
- $Prob \colon \delta \to (0..1]$ such that for every pair pX:

$$\sum_{pX \hookrightarrow q\alpha} Prob(pX \hookrightarrow q\alpha) = 1$$

Notation: We write $pX \overset{x}{\hookrightarrow} q\alpha$ for $Prob(pX \hookrightarrow q\alpha) = x$

Semantics: A (possibly infinite) Markov chain with configurations as states and transition probabilities given by

$$\text{If} \quad pX \overset{x}{\hookrightarrow} q\alpha \in \delta \quad \text{then} \quad pX\beta \xrightarrow{x} q\alpha\beta \quad \text{for every } \beta \in \Gamma^*$$

Probabilistic verification

Qualitative properties: does a program property hold with probability 1?

(Has the set of program runs satisfying the property measure 1 ?)

Quantitative properties: does a program property hold with probability at least ρ ?

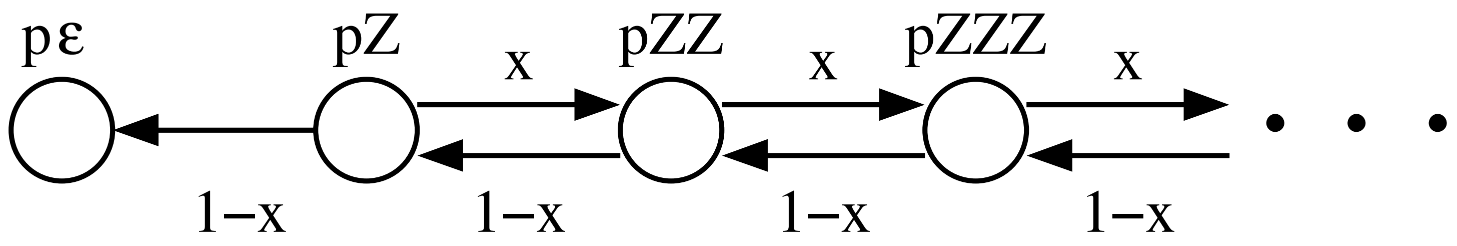
(Is the measure of the set of program runs satisfying the property at least ρ?)

In this talk:

- Reachability of control states
- Repeated reachability of control states
- Verification of Büchi specifications
- Verification of PCTL specifications

A one-state PPDS

$$pZ \overset{x}{\hookrightarrow} pZZ$$
$$pZ \overset{1-x}{\hookrightarrow} p\varepsilon$$



Even qualitative properties depend on the actual values of the probabilities

$\longrightarrow$ qualitative problems cannot be solved by graph-theoretical methods only

A basic result

Define $[pXq]$ as the probability of, starting at the configuration pX, eventually reaching the configuration $q\epsilon$.

Theorem: The $[pXq]$'s are the least solution of the following system of equations:

$$\langle pXq \rangle = \sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x + \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot \sum_{t \in P} \langle rYt \rangle \cdot \langle tZq \rangle$$

The system is of the form $\mathbf{x} = P(\mathbf{x})$, and the sequence $\mathbf{0}, P(\mathbf{0}), P^2(\mathbf{0}) \ldots$ converges to the least solution.

Some observations

No closed-form solution: The least solution of the equations can be a tuple of algebraic numbers of arbitrary degree

Slow convergence: We may need 2^n applications of P to gain n-bits of precision

Very small and large probabilities: The probability of $[pXq]$ in a PDS of size n may be as small as $1/2^{2^n}$ or as large as $1 - 1/2^{2^n}$

Checking reachability properties

Theorem: The problem $[pXq] \overset{?}{\leq} \rho$ can be solved in PSPACE for every $0 \leq \rho \leq 1$

Reduction to the decision problem for the existential theory of the reals

Theorem: The SQUARE-ROOT-SUM problem is polynomially reducible to the problem $[pXq] \overset{?}{\geq} 1$

Given: $(d_1, \ldots, d_n) \in \mathbb{N}^n$ and $k \in \mathbb{N}$

Decide whether: $\displaystyle\sum_{i=1}^{n} \sqrt{d_i} \leq k$

Theorem: The problem $[pXq] \overset{?}{\geq} 1$ can be solved in PTIME for one-state PPDS (or even for single-exit RSMs)

Numerical computation

Newton's method to solve $\mathbf{x} = F(\mathbf{x})$:

$$\mathbf{x}_{k+1} := \mathbf{x}_k - (F'(\mathbf{x}_k))^{-1} \cdot F(\mathbf{x}_k)$$

where

$$F'(\mathbf{x}) = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} \cdots \frac{\partial f_1}{\partial x_n} \\ \vdots \vdots \vdots \\ \frac{\partial f_n}{\partial x_1} \cdots \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

Theorem: Newton's method (multivariate) converges monotonically to the least fixed point of $\mathbf{x} = P(\mathbf{x})$

Newton's method converges fast for typical examples (but no bounds yet!)

Checking repeated reachability (qualitative)

- Given: an initial configuration p_0X_0, a control state p_r
- Decide whether: p_r is repeatedly reached w.p.1, i.e. whether the runs that visit infinitely many configurations of the form $p_r\alpha$ have measure 1

We construct a finite Markov chain M with initial state s_0 s.t.

$$p_r \text{ is repeatedly reached from } p_0X_0 \text{ w.p.1}$$
$$\Longleftrightarrow$$
$$\text{certain states of } M \text{ are repeatedly reached from } s_0 \text{ w.p.1}$$

that can be decided using grapth-theoretical methods

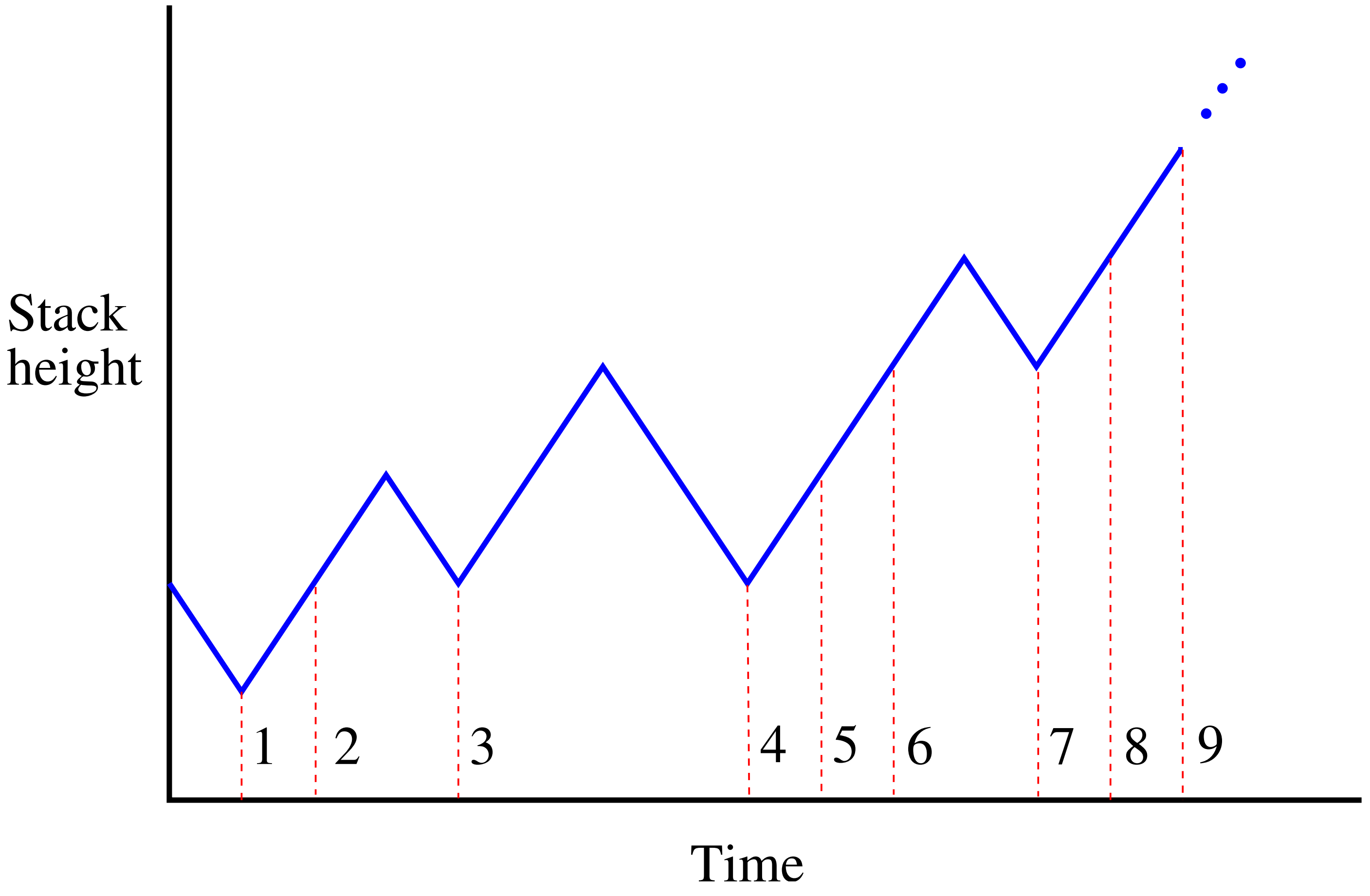
Minima of an infinite run

Let $w = p_0\alpha_0\, p_1\alpha_1\, p_2\alpha_2\, \cdots$ be an infinite run of a PPDS

$p_i\alpha_i$ is a minimum of w if $|\alpha_i| \geq |\alpha_j|$ for all $j \geq i$
(i.e., if α_i "stays forever in the stack")

Extract from w the subsequence $p_{m_1}\alpha_{m_1}\ p_{m_2}\alpha_{m_2}\ \ldots$ of minima

The i-th minimum of w is the i-th configuration of the subsequence of minima


Stack height
1
2
3
4
5
6
7
8
9
Time

The memoryless property

Given a configuration $c = pX\alpha$, let pX be the head and α the tail of c

Theorem (loosely formulated):
For every $i \geq 1$, the probability that the $i+1$-th minimum of a run has head pX depends only on the head of the i-th minimum (and is in particular independent of i).

We construct a Markov chain (not yet the one we want!) with

- the possible heads as states,
- transition probabilities given by:

  $pX \xrightarrow{x} qY$ if x is the probability of, starting from a minimum with head pX, reaching the next minimum at a configuration with head qY

Computing the transition probabilities

Let $[pX \Rightarrow qY]$ be the probability of $pX \longrightarrow qY$ in the new Markov chain

Define $[qY{\uparrow}] = 1 - \sum_{r \in P} [qYr]$

(i.e., $[qY{\uparrow}]$ is the probability that a run starting at qY does not terminate)

Theorem:

$$[pX \Rightarrow qY] \;=\; \sum_{pX \overset{x}{\hookrightarrow} rZY} x \cdot [rZq] \cdot [qY{\uparrow}] \;+\; \sum_{pX \overset{x}{\hookrightarrow} qYZ} x \cdot [qY{\uparrow}]$$

The finite Markov chain for the rep. reach. problem

States of the form $(pX, b) \xrightarrow{x} (qY, b')$, where b, b' booleans,

Transition probabilities given by:

- $(pX, b) \xrightarrow{x} (qY, 1)$ if x as above, but requiring that p_r is visited between the two minima, and

- $(pX, b) \xrightarrow{x} (qY, 0)$ if x as above, but requiring that p_r is not visited between the two minima

Theorem: The state p_r is repeatedly reachable w.p.1 iff the finite Markov chain above has a unique bottom s.c.c., and this s.c.c. contains a state of the form $(pX, 1)$ for some pX

Corollary: The (qualitative and quantitative) repeated reachability problem can be solved in PSPACE

Results on checking Büchi specifications

Theorem: The qualitative (quantitative) model-checking problem for deterministic Büchi specifications can be solved by an algorithm using polynomial space in the size of the PPDS and linear time in the size of the Büchi automaton automaton

Theorem: The qualitative (quantitative) model-checking problem for arbitrary Büchi specifications can be solved by an algorithm using polynomial space in the size of the PPDS and exponential time in the size of the Büchi automaton

Checking PCTL specifications

Syntax:

$$\varphi \; ::= \; \texttt{tt} \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathbf{X}^{\geq\varrho}\varphi \mid \varphi_1 \mathbf{U}^{\geq\varrho}\varphi_2$$

where *a* is an atomic proposition and ρ is a probability

Semantics: Let $[\![\varphi]\!]$ denote the set of states satisfying ϕ

$$\begin{aligned} [\![\mathbf{X}^{\geq\varrho}\varphi]\!] &= \{p\alpha \mid \quad \mathcal{P}(\, Run(p\alpha, \mathbf{X}[\![\varphi]\!])\,) \geq \varrho \quad \} \\ [\![\varphi_1 \mathbf{U}^{\geq\varrho}\varphi_2]\!] &= \{p\alpha \mid \quad \mathcal{P}(\, Run(p\alpha, [\![\varphi_1]\!]\mathbf{U}[\![\varphi_2]\!])\,) \geq \varrho \quad \} \end{aligned}$$

Qualitative fragment of PCTL: $\rho = 1$

Model checking the qualitative fragment

In the finite-state case:

- Compute the set of states satisfying the subformulas of a formula
- Derive from them the set of states satisfying the formula

Problem: The set of states satisfying the subformulas can be infinite

Theorem: Let φ be a qualitative PCTL formula, and let ν be a regular valuation. The set of configurations that satisfy φ under the valuation ν is effectively regular

Unfortunately (see counterexample in paper), the theorem no longer holds for general PCTL formulas

Theorem: The qualitative model-checking problem for PCTL and pushdown systems is EXPTIME-hard and solvable in EXPSPACE.

Theorem: The quantitative model-checking problem for PCTL is undecidable

Good approximation algorithms for one-state PPDSs (single-exit RSMs)

Conclusions

Probabilistic verification is feasible for models beyond while-programs

Very nice mathematics!

Key point: convergence rate of numerical algorithms

Very likely to have good applications for software models (polynomial procedures)

Applicability to ‘large’ randomized algorithms remains to be seen

The probability space

Run: maximal path of configurations (infinite or finite but ending at configuration with empty stack)

Sample space: runs starting at an initial configuration $p_0\alpha_0$

σ-algebra: generated by the basic cilinders $Run(w)$, the set of runs that start with the finite sequence w of configurations.

Probability function: the probability of $Run(w)$ is the product of the probabilities associated to the sequence of rules that ‘generate’ w