Verification of Infinite-State Probabilistic Systems

Richard Mayr

University of Edinburgh, UK

1st EATCS Young Researchers School. Telč. 31. July 2014

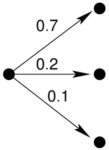
Outline

Modeling Uncertainty + Limited Control

- Markov Chain: Just probabilities
  Questions: What is the probability of property X ?
- Markov Decision Process: Probabilities + A Controller
  Questions: Find a controller which achieves that
  $Prob(\text{Property X}) \geq 0.99$.
- Stochastic game: Probabilities + 2 players.
  Questions: Is there a strategy for the player (controller) that achieves $Prob(\text{Property X}) \geq 0.99$ against every strategy of the opponent (i.e., the environment) ?
  Types of games: Simple (turn-based) games vs. concurrent games.

Different models for time: Discrete time (i.e. step-by-step) vs. Continuous time

Discrete time: Transition probabilities.



$\forall s \in S.\ \sum_{s' \in S} P(s, s') = 1$

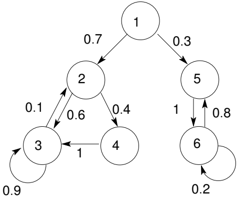
Continuous-time: Transition rates $R(s, s')$.

$$P(s, s', t) = \frac{R(s, s')}{E(s)} \left(1 - e^{-E(s)t}\right)$$

where $E(s) = \sum_{s'' \in S} R(s, s'')$.

In this tutorial: Just discrete time.

Finite-State DTMC

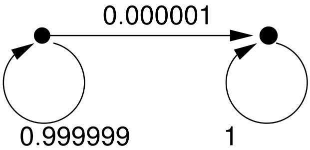


- State of the system can be expressed by a vector $\vec{x} = (x_0, \ldots, x_{n-1})$ where $x_i \geq 0$ is the probability to be in state s_i and $\sum_{0 \leq i \leq n-1} x_i = 1$.
- Transition described by matrix multiplication $\vec{x}' := \vec{x}M$
- If $\vec{x} = \vec{x}M$ then $\vec{x}$ is the steady-state (stationary distribution), i.e., the probability to be in states 'in the long run'. $\longrightarrow$ Solving linear equation system.
- Tools to analyze **finite** DTMC/CTMC: E.g., PRISM, MRMC.

Properties of Finite-State DTMC

- For finite-state DTMC the steady-state always exists, provided that the MC is irreducible (i.e., strongly connected).
  If not strongly connected then analyze each bottom strongly connected component.
- Property (C): If some state is always reachable, then it is eventually reached (and even ∞-often) with probability 1.

$$Prob(\Box\Diamond F \vee \Diamond\tilde{F}) = 1, \quad \tilde{F} = \overline{Pre^*(F)}$$

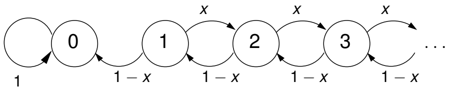


Infinite-State DTMC

Many results from finite Markov chains do not carry over:

- Steady-state need not exist.
- Property (C) need not hold.

Gambler's ruin:



Start at (1), probability of eventually visiting (0) is

1	if $x \leq 1/2$
$(1-x)/x$	if $x > 1/2$

For $x = 1/2$ the point (0), i.e., ruin, will eventually be reached with probability 1 (but the expected number of steps to get there is infinite).
For $x > 1/2$ the steady-state does not exist.

Program-induced Classes of Infinite DTMC

1. Probabilistic pushdown automata (PPDA).
   Finite DTMC calling each other recursively. Unbounded depth of the stack.
   Equivalent to: Recursive Markov chains (RMC), Tree-like quasi-birth-death processes (Tree-QBD).
   Subclasses:
   - Probabilistic One-Counter Automata. Quasi-birth-death processes.
   - Stateless PPDA. Stochastic context-free grammars. 1-exit RMC.
2. Probabilistic vector addition systems with states (PVASS).
   Probabilistic extension of vector addition systems (Petri nets).
   Concurrency, unbounded process creation, synchronization.
3. Probabilistic lossy channel systems (PLCS).
   Finite automata, communicating by asynchronous message-passing via unboundedly buffered channels with 'first-in-first-out' principle. Channels are unreliable and messages can be lost in transit.

Program-induced Classes of Infinite DTMC

1. Probabilistic pushdown automata (PPDA).
   Finite DTMC calling each other recursively. Unbounded depth of the stack.
   Equivalent to: Recursive Markov chains (RMC), Tree-like quasi-birth-death processes (Tree-QBD).
   Subclasses:
   - Probabilistic One-Counter Automata. Quasi-birth-death processes.
   - Stateless PPDA. Stochastic context-free grammars. 1-exit RMC.
2. Probabilistic vector addition systems with states (PVASS).
   Probabilistic extension of vector addition systems (Petri nets).
   Concurrency, unbounded process creation, synchronization.
3. Probabilistic lossy channel systems (PLCS).
   Finite automata, communicating by asynchronous message-passing via unboundedly buffered channels with 'first-in-first-out' principle. Channels are unreliable and messages can be lost in transit.

Program-induced Classes of Infinite DTMC

1. Probabilistic pushdown automata (PPDA).
   Finite DTMC calling each other recursively. Unbounded depth of the stack.
   Equivalent to: Recursive Markov chains (RMC), Tree-like quasi-birth-death processes (Tree-QBD).
   Subclasses:
   - Probabilistic One-Counter Automata. Quasi-birth-death processes.
   - Stateless PPDA. Stochastic context-free grammars. 1-exit RMC.
2. Probabilistic vector addition systems with states (PVASS).
   Probabilistic extension of vector addition systems (Petri nets).
   Concurrency, unbounded process creation, synchronization.
3. Probabilistic lossy channel systems (PLCS).
   Finite automata, communicating by asynchronous message-passing via unboundedly buffered channels with 'first-in-first-out' principle. Channels are unreliable and messages can be lost in transit.

Program-induced Classes of Infinite DTMC

1. Probabilistic pushdown automata (PPDA).
   Finite DTMC calling each other recursively. Unbounded depth of the stack.
   Equivalent to: Recursive Markov chains (RMC), Tree-like quasi-birth-death processes (Tree-QBD).
   Subclasses:
   - Probabilistic One-Counter Automata. Quasi-birth-death processes.
   - Stateless PPDA. Stochastic context-free grammars. 1-exit RMC.
2. Probabilistic vector addition systems with states (PVASS).
   Probabilistic extension of vector addition systems (Petri nets).
   Concurrency, unbounded process creation, synchronization.
3. Probabilistic lossy channel systems (PLCS).
   Finite automata, communicating by asynchronous message-passing via unboundedly buffered channels with 'first-in-first-out' principle. Channels are unreliable and messages can be lost in transit.

Probabilistic Pushdown Automata

$\Delta = (Q, \Gamma, \delta, Prob)$, where

- Q is a finite set of *control states*.
- Γ is a finite *stack alphabet*.
- $\delta \subseteq Q \times \Gamma \times Q \times \Gamma^*$ is a finite *transition relation*. Written $pX \to q\alpha$.
- *Prob* assigns a probability $Prob(pX \to q\alpha) \in (0, 1]$ to each transition s.t. $\sum_{q,\alpha} Prob(pX \to q\alpha) = 1$.

Example:
$$pX \stackrel{0.7}{\to} pXX$$
$$rX \stackrel{1}{\to} pX$$
$$pX \stackrel{0.3}{\to} r$$

RMC view:

- X is a procedure with start state p.
- It either terminates in state r (with prob. 0.3) or calls two instances of itself (with prob. 0.7).

Starting at pX, what is the probability of eventual termination?

General Verification Problems for PPDA

- Computing termination probabilities.
- Model-checking PPDA.
  - PCTL. A probabilistic extension of CTL. E.g.,

$$[\![\phi_1 \,\mathcal{U}^{\geq 0.7}\, \phi_2]\!] = \{s \mid P(s, [\![\phi_1]\!] \,\mathcal{U}\, [\![\phi_2]\!]) \geq 0.7\}$$

  - Model-checking with LTL. Given some LTL formula ϕ, what is the probability measure of those runs satisfying ϕ ?
- Reward models: What is the expectation/variance of the accumulated cost/reward of those runs which reach a given set of final states.
  E.g., expected time to termination; expected number of visits of a defined state.
- Adding controllable transitions: Recursive Markov decision processes (MDP) and recursive stochastic games.

Main Idea for PPDA

Behavior of (P)PDA can be decomposed sequentially.

$$pXY \to \ldots$$

Stack symbol Y does not play any role before symbol X is popped from the stack.
Decompose the computation into two parts:

$$pXY \to \ldots \to qY$$

and

$$qY \to \ldots$$

(If X is never popped then Y is irrelevant. Behaves like pX.)
The only connection between the two parts is the control-state q.
There are only finitely many cases which state q is.
Otherwise, the two parts are independent. Thus probabilities multiply.

Main Idea for PPDA (cont.)

Given control-states p and q and a stack symbol X, let

$$[pXq]$$

be the probability that, starting at configuration pX, one **eventually** pops the symbol X from the stack and is then at control-state q. These are called the selective termination probabilities.

$$[pXq] = Prob(pX \rightarrow \ldots \rightarrow q)$$

Once one knows the values $[pXq]$ for all combinations of p, X, q, one can compute (almost) all other verification questions.

Computing Basic Probabilities

The probabilities $[pXq]$ form the least solution of an effectively constructible system of polynomial (but generally non-linear) equations.
Let $\{\langle pXq\rangle \mid p, q \in Q, X \in \Gamma\}$ be a set of variables.

$$\begin{aligned}\langle pXq\rangle &= \sum_{pX \xrightarrow{y} rYZ} y \cdot \sum_{t \in Q} \langle rYt\rangle \cdot \langle tZq\rangle \\ &+ \sum_{pX \xrightarrow{y} rY} y \cdot \langle rYq\rangle \quad + \sum_{pX \xrightarrow{y} q\varepsilon} y\end{aligned}$$

Due to the monotonicity of the equations, there exists a least solution $\langle pXq\rangle^*$ for the variables and $\langle pXq\rangle^* = [pXq]$.

Fixpoints of Polynomial Equation Systems

Let $\vec{x} = \{\langle pXq \rangle \mid p, q \in Q, X \in \Gamma\}$.
Then we get a system of polynomial equations

$$\vec{x} = P(\vec{x})$$

$P : \mathbb{R}^n \mapsto \mathbb{R}^n$ defines a monotone operator on $\mathbb{R}^n_{\geq 0}$.
It has a least fixpoint $\vec{x}^* \in \mathbb{R}^n_{\geq 0}$.
I.e., $\vec{y}^* = P(\vec{y}^*) \ \rightarrow \ \vec{x}^* \leq \vec{y}^*$.

Theorem

- $\vec{x}^*$ *is the vector of termination probabilities.*
- $\vec{x}^* = \lim_{k \to \infty} P^k(\vec{0})$

Example

Example: $pX \overset{0.7}{\to} pXX$
$pX \overset{0.3}{\to} p$

$$[pXp] \quad = \quad 0.3 + 0.7 \cdot [pXp]^2$$

Complexity upper bound

Theorem

Questions about $\vec{x}^*$ can be effectively expressed in the existential fragment of the first-order theory of the reals $\exists(\mathbb{R}, +, *, \leq)$.
And thus solved in PSPACE.
Given i (in unary), $\vec{x}^*$ can be approximated to i bits of precision in PSPACE.

Proof.

Let $\vec{z} \in \mathbb{R}^n$. The question

$$\vec{x}^* \leq \vec{z}$$

is equivalent to

$$\exists \vec{x}.\ \vec{x} = P(\vec{x}) \wedge \vec{x} \leq \vec{z}$$

There are PSPACE decision procedures for $\exists(\mathbb{R}, +, *, \leq)$ [Canny'89, Renegar'92].
Approximation to within *i* bits can be done by binary search using *i* queries to $\exists(\mathbb{R}, +, *, \leq)$. $\square$

Complexity lower bounds

Some "hard" problems.

Problem (Sqrt-Sum)

Given $x_1, \ldots, x_n, k \in \mathbb{N}$, decide wether $\sum_{i=1}^{n} \sqrt{x_i} \leq k$.
This is solvable in PSPACE (by $\exists(\mathbb{R}, +, *, \leq)$), but it is open whether it is in NP or even the polynomial time hierarchy.

Problem (Pos-SLP)

Given an arithmetic circuit (Straight Line Program) over operations $\{+, -, *\}$ with integer inputs, decide whether the output is > 0.
PosSLP captures all one can do in polynomial time in the unit-cost arithmetic RAM model of computation.

Both Sqrt-Sum and Pos-SLP are in the counting hierarchy $P^{PP^{PP^{PP}}}$ [Allender et. al. 2006].

Complexity lower bounds (cont.)

Many PPDA problems are at least as hard as Sqrt-Sum, Pos-SLP.

Theorem (Etessami-Yannakakis. 2005, 2007)

Sqrt-Sum and Pos-SLP are polynomial time reducible to the following problems:

- *Given a PPDA, decide whether it terminates with probability one, i.e.,* $[pXq] = 1$.
- *Given a stateless PPDA (i.e., stochastic context-free grammar) and a rational p, decide whether it terminates with probability* $\geq p$.
- *Given a PPDA, compute any non-trivial approximation of* $[pXq]$. *For any fixed* $\epsilon > 0$*, given a PPDA such that either (a)* $[pXq] = 1$*, or (b)* $[pXq] \leq \epsilon$*. Decide which of (a) or (b) is the case.*

Approximating Solutions

Termination probabilities $\vec{x}^*$ are the least solution of a monotone system of polynomial equations

$$\vec{x} = P(\vec{x})$$

where $P : \mathbb{R}^n \mapsto \mathbb{R}^n$ defines a monotone operator on $\mathbb{R}^n_{\geq 0}$.
$\vec{x}^*$ is the least fixpoint of P.

$$\vec{x}^* = \lim_{n \to \infty} P^n(\vec{0})$$

Why not just do simple **value iteration** ?
Let $\vec{x}^0 := \vec{0}$ and $\vec{x}^{i+1} := P(\vec{x}^i) = P^{i+1}(\vec{0})$ for $i = 1, 2, 3, \ldots$.

Value iteration can require exponentially many iterations

Consider the PPDA

$$\begin{array}{rcl} pY & \stackrel{1/2}{\rightarrow} & p \\ pY & \stackrel{1/2}{\rightarrow} & pYY \end{array}$$

Equation in one variable x (i.e., $\langle pYp \rangle$).

$$x = (1/2)x^2 + 1/2$$

Fact: $x^* = 1$, but $|1 - P^m(0)| \geq 1/2^i$ for all $m \leq 2^i$.

I.e., one needs exponentially (2^i) many iterations to get within i bits of precision (additive error $1/2^i$) of the solution x^*.
There are other pathological cases with doubly exponentially large/small probabilities, i.e., ϵ or $1 - \epsilon$ where $\epsilon = \mathcal{O}(\frac{1}{2^{2^n}})$.

Faster Approximation Methods

Newton's method

For a general function $F : \mathbb{R}^n \to \mathbb{R}^n$ we seek the solution to $F(\vec{x}) = \vec{0}$.
Guess an initial vector $\vec{x}_0$ and compute the sequence $\vec{x}_k$, for $k \to \infty$, where:

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

$F'(\vec{x})$ is the **Jacobian matrix** of partial derivatives

$$F'(\vec{x}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \cdots & \frac{\partial F_1}{\partial x_n} \\ \vdots & \vdots & \vdots \\ \frac{\partial F_n}{\partial x_1} & \cdots & \frac{\partial F_n}{\partial x_n} \end{bmatrix}$$

- Method only defined if all matrices $F'(\vec{x}_k)$ are non-singular.
- Even when defined, it can diverge (even for univariate polynomials).
- But if it does converge, it is typically quite fast.

Newton's method applied to PDDA

Let $F(\vec{x}) = P(\vec{x}) - \vec{x}$.

In the equation system $\vec{x} = P(\vec{x})$, analyze dependencies of variables, and decompose the system into **strongly connected components** (SCCs). Eliminate variables that are certainly zero (or other constants).

Theorem (Decomposed Newton's method for RMC/PPDA [Etessami-Yannakakis'05])

Starting at $\vec{x}_0 := \vec{0}$ and working bottom-up on the SCCs of the decomposition DAG of $\vec{x} = P(\vec{x})$, Newton's method monotonically converges from below to the least fixpoint, i.e., $\vec{x}^*$.

- Implemented in PReMo tool (http://groups.inf.ed.ac.uk/premo) [Wojtczak-Etessami'07]. Large benchmarks from NLP (up-to 500000 variables) yield good results.
- Still no magic. [Esparza, Kiefer, Luttenberger,'07] gave examples requiring exponentially many iterations to converge to within additive error $< 1/2$.

Probabilistic One-Counter Automata, Quasi-birth-death processes (QBD)

Unlike for general PPDA, polynomially many iterations of Newton's method suffice.

Theorem (Etessami-Wojtczak-Yannakakis'08)

For discrete-time Quasi-Birth-Death Processes, polynomially many (in the encoding size of the system and parameter i) many iterations of Newton's method suffice to get termination probabilities within additive error $1/2^i$.

Proof.

(ideas). Some interesting decomposition properties in the QBD case. Shortest paths to a given control-state have polynomial length (unlike the possibly exponential length for general PPDA). Some results from [Esparza-Kiefer-Luttenberger]. □

Remarks on Probabilistic One-Counter Automata, Quasi-birth-death processes (QBD)

- Special Matrix Analytic numerical methods have been developed for many years for analyzing QBDs and related Structured Markov chains (e.g., M/G/1-Type). See, e.g., the books: [Neuts'81],[Latouche-Ramaswami'99],[Bini-Latouche-Meini'05].
- Key matrix analytic methods are **logarithmic reduction** and **cyclic reduction**. (Implemented in tools like SMCSolver [Bini-Meini-Steffe-Van Houdt'06].)
- These methods far outperform decomposed Newton's method on dense instances of QBDs, but decomposed Newton's method can outperform them on very sparse instances (see [Etessami-Wojtczak-Yannakakis'08-'10] for some comparisons).

Subclass: Stateless PPDA

- Equivalent to Stochastic context-free grammars, 1-exit RMC.
- Stochastic context-free grammars are a fundamental model in statistical natural language processes, and are also used extensively in biological sequence analysis (RNA secondary structure analysis).
- As far as termination probabilities are concerned, this is also equivalent to **multi-type branching processes (MT-BP)**. (Just ignore the order of elements on the stack.)
- MT-BPs are a classic and heavily studied class of stochastic processes ([Kolmogorov'1940s]), with many applications. Here termination probabilities are called extinction probabilities.

Subclass: Stateless PPDA (cont.)

Theorem (Etessami-Yannakakis'05)

For stochastic context-free grammars (and equivalent models), deciding whether the termination probability is $= 1$ is in polynomial time.

Proof.

Eigenvalue methods and graph-theoretic methods. Key problem can be reduced to deciding whether certain moment matrices (Jacobian of $P(x)$ evaluated at the all 1 vector) have spectral radius > 1. ([Kolmogorov-Sevastyanov,'47,Harris'63]) $\square$

Model Checking with PCTL

Theorem (Esparza, Kučera, Mayr'04)

Model checking with the qualitative fragment of PCTL (probability bound questions limited to $=0$ or $=1$) is decidable. The denotations of qualitative PCTL formulae are effectively regular sets of PPDA configurations.

Theorem (Brazdil, Kučera, Strazovsky'05)

General quantitative model checking of PPDA with PCTL is undecidable. (E.g., questions like prob $= 0.5$). In particular, denotations of formulae are not regular in general.

LTL Model Checking

Theorem (Esparza, Kučera, Mayr'04)

Given a PPDA A with initial configuration p_0X and an LTL formula ϕ, the probability $\mathcal{P}(runs(p_0X) \cap [\![\phi]\!])$ is effectively expressible in $(\mathbb{R}, +, *, \leq)$.

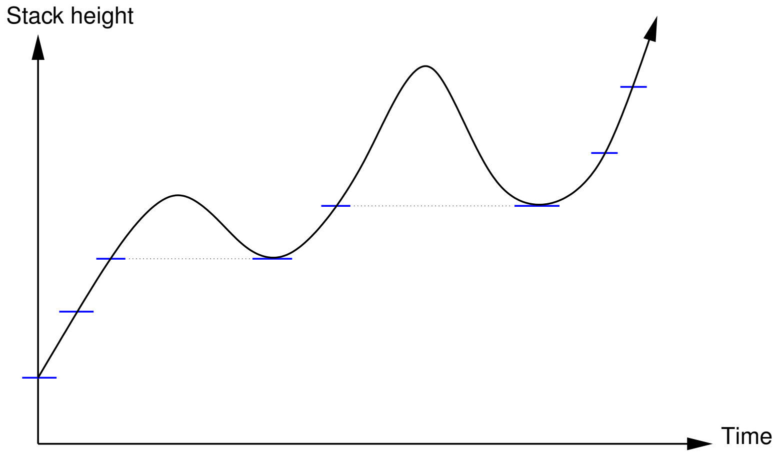
- Complexity of quantitative LTL model checking:
  PSPACE in $|A|$. EXPSPACE in $|\phi|$. [Etessami-Yannakakis'05'11].
- Even for non-probabilistic stateless PDA, LTL model checking is EXPTIME-complete [Bouajjani-Esparza-Maler'97, Mayr'98].

LTL Model Checking (cont.)

Proof.

Transform ϕ into a deterministic Muller automaton and sync. it with the PPDA, obtaining a deterministic Muller-PPDA.
Construct a finite-state Markov chain of the form $(pX, A) \stackrel{\alpha}{\to} (qY, B)$, where pX, qY are heads of the det. Muller-PPDA, α is the probability that qY is the head of the next minimum configuration in the run after the previous minimum head pX, and B is the set of states of the det. Muller aut. that were visited in between.
This finite MC represents the whole system up-to some runs with prob. zero. Analyze the bottom strongly connected components of this finite MC. (Similarly as done for finite systems by Courcoubetis, Yannakakis; JACM'95.) Everything is effectively constructible and expressible in $(\mathbb{R}, +, *, \leq)$. $\square$

LTL Model Checking: Illustration



Future Minimum: Stack height will never fall below this in the future
$\rightarrow$ Only control-state and top stack symbol matter

Extensions: Markov Reward Models

Simple reward functions $f(p\alpha) = f(p)$ (depending only on the control-state) assign a reward to configurations.

Let $[E(pXq)]$ be the expected accumulated reward of runs from pX to $q\epsilon$, provided that $q\epsilon$ is reached. (Conditional expectation).

Can be computed as minimal solution of a system of recursive equations [Esparza-Kučera-Mayr'05].

$\langle E(pXq)\rangle = 0$ for $[pXq] = 0$. Otherwise,

$$\langle E(pXq)\rangle = \frac{1}{[pXq]}\left(\sum_{pX \xrightarrow{x} q\varepsilon} x \cdot f(q) + \sum_{pX \xrightarrow{x} rYZ} x \cdot K_{pX,rYZ}\right)$$

where the term $K_{pX,rYZ}$ is given by

$$\sum_{s\in Q} [rYs][sZq](f(r) + \langle E(rYs)\rangle + \langle E(sZq)\rangle)$$

Variance

$[Q(pXq)]$, the conditional second moment of the distribution of accumulated rewards can be computed as the least solution of $\langle Q(pXq)\rangle = 0$ for $[pXq] = 0$, else

$$\begin{aligned}\langle Q(pXq)\rangle \quad = \quad & \frac{1}{[pXq]}\left(\sum_{pX \xrightarrow{x} q\varepsilon} x \cdot f(q)^2 + \right.\\ & \left. + \sum_{pX \xrightarrow{x} rYZ} x \cdot \sum_{s \in Q} [rYs][sZq] K_{pX,rYZ,s}\right)\end{aligned}$$

where the expression $K_{pX,rYZ,s}$ stands for

$$\langle Q(rYs)\rangle + \langle Q(sZq)\rangle + f(r)^2 +$$
$$2[E(rYs)][E(sZq)] + 2f(r)[E(rYs)] + 2f(r)[E(sZq)]$$

The conditional variance $V = [Q(pXq)] - [E(pXq)]^2$.

Recursive MDPs and Recursive Stochastic Games

Theorem (Etessami-Yannakakis'05)

For general recursive MDPs, even the qualitative termination value problem (is the value $= 1$), is undecidable. Even any non-trivial approximation of the optimal termination value is not computable.

By reduction from the emptiness problem for Probabilistic Finite Automata (PFA) [Rabin'63]. Let the player guess a word and store it on the stack. Then run the PFA on this word.

Theorem (Etessami-Yannakakis'05'07)

Quantitative termination value problems for 1-exit recursive MDPs and SSGs (stateless PPDA games) are in PSPACE using $\exists(\mathbb{R}, +, *, \leq)$.

Use systems of polynomial equations with additional min- and max-operators. Least fixed point gives precisely the game values.

Probabilistic Vector Addition Systems with States (PVASS)

- Configurations of the form $(q, \vec{x})$, where $q \in Q$ is a control-state and $\vec{x} \in \mathbb{N}^n$.
- Transition rules of the form $(q, \vec{y}) \stackrel{w}{\to} (q', \vec{y}')$.
  $w \in \mathbb{N}$ is a **transition weight**, not a probability.
- Induce transitions $(q, \vec{x}) \stackrel{p}{\to} (q', \vec{x} - \vec{y} + \vec{y}')$ if $\vec{x} \geq \vec{y}$.
- Probability p is given by

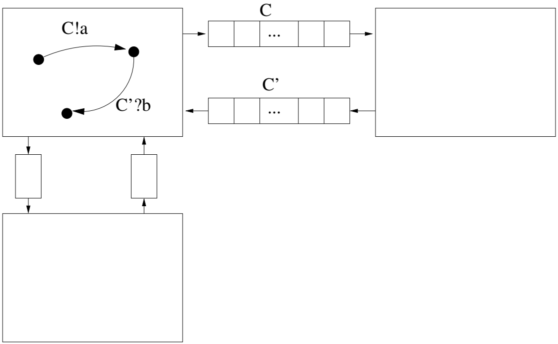
$$p = \frac{w}{w_1 + \cdots + w_k}$$

  where $w_1, \ldots, w_k$ are the weights of all transitions enabled at $(q, \vec{x})$.

FIFO Channel Systems

Finite automata which communicate with each other by

- asynchronous message passing
- communication channels with unbounded buffers
- FIFO: first-in first-out
- Channels can encode Turing-tape
- Simulate Turing machines [Brand & Zafiropoulo, 1983].
- Undecidable verification problems.



Probabilistic Lossy Channel Systems (PLCS)

Lossy channel systems.

- Messages in transit can be lost spontaneously, i.e., channel content word w can change to a subword w'.
- Subword order is a well-quasi-order on strings by Higman's Lemma.
- Transition relation is monotone w.r.t. subword order.
- Well quasi-ordered (aka well-structured) transition systems. Reachability and termination are decidable [Abdulla-Jonsson'96].
- Boundedness, universal termination, and LTL model-checking still undecidable [Mayr'2000].

Probabilistic Lossy Channel Systems.

- At every step, every message in transit is lost with probability $\lambda > 0$ (independently of each other).
- At bigger configurations more messages are in transit. On average more messages are lost per step. Strong downward drift.
- System has a finite attractor, i.e., a finite core region that is almost certainly (i.e., with probability 1) re-visited.

PVASS and PLCS vs. PPDA

Even without probabilities, VASS/LCS are a lot harder than PDA.

PDA: Reachability is in PTIME. Set of reachable configurations is effectively regular.

VASS: Control-state reachability is EXPSPACE-complete.
Reachability is decidable, but the exact complexity is open.
Set of reachable configurations is not semilinear.

LCS: (Control-state) reachability is decidable, but quite hard ($\mathcal{F}_{\omega^\omega}$ in the fast-growing hierarchy).
Set of reachable configurations is regular, but not effectively regular. (Undecidable space-boundedness problem.)

PVASS and PLCS vs. PPDA

Unlike PPDA, the PVASS/PLCS models do not have nice decomposition properties.

- For PVASS and PLCS, there is no characterization of selective termination probabilities by systems of polynomial equations.
- For PVASS and PLCS, it is not even known whether the selective termination probabilities are always algebraic numbers.

Theorem (Abdulla-Henda-Mayr'07)

For PVASS/PLCS it is not possible to effectively construct formulae containing transition-weights/loss-rates that express selective termination probabilities in $(\mathbb{R}, +, *, \leq)$.

Still, some qualitative and quantitative questions about PVASS/PLCS can be solved.

Conditions on infinite transition graphs

Coarse

Transition probability cannot get arbitrarily small.

$\exists \alpha > 0 \, \forall s, t.$

$P(s,t) > 0 \Rightarrow P(s,t) \geq \alpha$

+

Finitely Spanning

Bounded distance from target F

$\exists l. \forall s$

$(s \to^{*} F) \Rightarrow (s \xrightarrow{\leq l} F)$

Finite Attractor

$\exists$ finite $A \subseteq S \, \forall s$

$\mathcal{P}(s \models \Diamond A) = 1$

Globally Coarse

Prob. of reaching F cannot get arbitrarily small.

$\exists \alpha > 0 \, \forall s.$

$\mathcal{P}(s \models \Diamond F) > 0 \Rightarrow$
$\mathcal{P}(s \models \Diamond F) \geq \alpha$

Convergence Property (C)

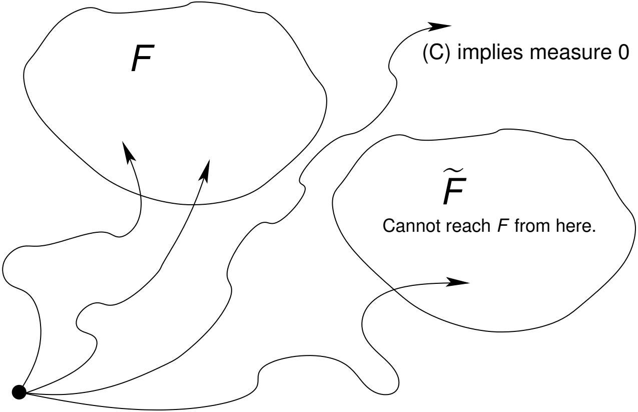
If F always reachable then
F reached infinitely often with prob. 1

$Prob(\Box\Diamond F \vee \Diamond \tilde{F}) = 1$

$\tilde{F} = \overline{Pre^{*}(F)}$

Methods for PVASS, PLCS

$\widetilde{F} = \overline{Pre^*(F)}$.
Path exploration to compute $\mathcal{P}(s_{init} \models \Diamond F)$ up-to $\delta > 0$.



Properties of PPDA, PVASS, PLCS

- Probabilistic vector addition systems with states (PVASS); concurrency.
- Probabilistic lossy channel systems (PLCS); unreliable async. communication.
- Probabilistic pushdown automata (PPDA); recursion.

	PVASS	PLCS	PPDA
Coarse	Yes	No	Yes
Finitely spanning	If F upward-closed	If F upward-closed	No
Globally coarse	If F upward-closed	No	No
Finite attractor	No	Yes	No
Property (C)	If F upward-closed	Yes	No

Almost sure reachability

Lemma

For any Markov chain with property (C),
$\mathcal{P}(s_{init} \models \Diamond F) = 1 \iff s_{init} \not\models \widetilde{F}\ \underline{Before}\ F \iff s_{init} \models \forall(\neg F\,\mathcal{U}\,\widetilde{F})$.

Theorem (Abdulla-Henda-Mayr'07)

For PLCS the question $\mathcal{P}(s_{init} \models \Diamond F) = 1$ is decidable for regular F.

Theorem (Abdulla-Henda-Mayr'07)

For PVASS the question $\mathcal{P}_{\mathcal{C}}(s_{init} \models \Diamond F) = 1$ is

- *decidable, if F specified only by condition on control-states.*
- *undecidable, if F is a general upward-closed set.*

Proof.

For decidability, backward reachability analysis in a modified system.
For undecidability, encoding of 2-CM. Unfaithful simulation leads to state in F. □

Büchi conditions

Lemma

For any Markov chain with property (C),
$\mathcal{P}(s_{init} \models \Box\Diamond F) = 1 \iff s_{init} \notin Pre^*(\widetilde{F})$.

Theorem (Abdulla-Henda-Mayr'07)

For PLCS (with general F) and PVASS (with upward-closed F), almost-sure Büchi, i.e., the question $\mathcal{P}(s_{init} \models \Box\Diamond F) = 1$, is decidable.

Theorem (Abdulla-Henda-Mayr'07)

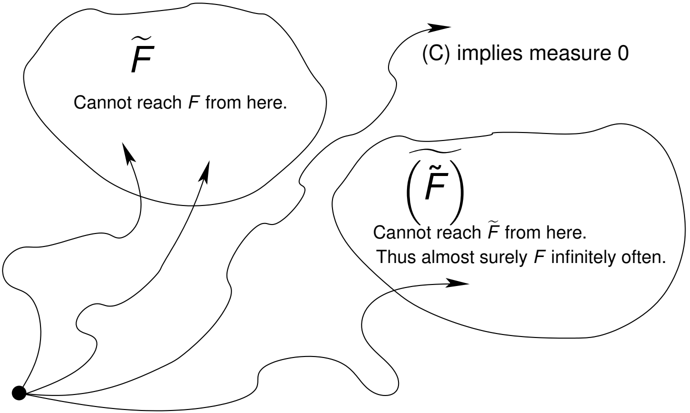
For any Markov chain with a finite attractor
$\mathcal{P}(s_{init} \models \Box\Diamond F) > 0 \iff s_{init} \in Pre^*(\overline{\widetilde{F}})$. *Decidable for PLCS.*

Proof idea: There is a minimal $\alpha > 0$ s.t. for any point in the finite attractor the probability of reaching F or $\widetilde{F}$ is either 0 or $\geq \alpha$.

This equivalence does not hold for PVASS. Decidability is open.

Path Exploration for Büchi Conditions

Path exploration to compute $\mathcal{P}(s_{init} \models \Box\Diamond F)$ up-to $\delta > 0$.



This only works if (C) holds for F and $\widetilde{F}$.
True for PLCS, but not generally for PVASS.

Games on PVASS

Theorem (Abdulla-Henda-Mayr'07)

Given an MDP on a PVASS and let $F = \{q\} \times \mathbb{N}$. It is undecidable whether the controller can achieve to reach F almost surely.

Proof.

Direct encoding of Minsky-machine. Unfaithful "zero"-step at nonzero counter value $c > 0$ is punished probabilistically in the following step. Go to a sink-state with probability $1/2$ iff $c > 0$. $\square$

Games on PLCS

Theorem (Abdulla-Henda-de Alfaro-Mayr-Sandberg'08)

2-player stochastic games on PLCS with almost-sure reachability or Büchi objectives are memoryless determined and decidable.

Theorem (Baier-Bertrand-Schnoebelen'06)

MDP on PLCS with almost-sure co-Büchi objectives generally require infinite memory to win, and are also undecidable.
No nontrivial approximation of the co-Büchi probability is computable.

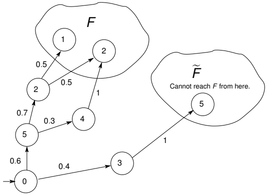
Theorem (Abdulla-Clemente-Mayr-Sandberg'13)

2-player stochastic parity games on PLCS are decidable, provided that the players are limited to finite-memory strategies.

Winning with finite memory is a different problem, not a subproblem.
In some cases, it is possible to win, but only with infinite memory.

Markov Reward Models

Assign a reward to paths in Markov chains, e.g., delay, average/peak memory used, average/peak bandwidth, etc.
Particular case: State/Transition rewards.



$\mathcal{P}_{\mathcal{C}}(s_{init} \models \Diamond F) = 0.6$.
Conditional expectation: E(cumulative reward until F | F is reached) $=$ $(0.6{*}0.7{*}0.5{*}8 + 0.6{*}0.7{*}0.5{*}9 + 0.6{*}0.3{*}1{*}11)/0.6 = 5.55/0.6 = 9.25$

Markov Reward Models (Cont.)

General case: Define reward-function f : Paths $\to \mathbb{R}$ that assigns rewards $f(\pi)$ to paths π.

Consider exponentially bounded reward functions $f(\pi) \leq k_1 \alpha_1^{|\pi|}$.

This subsumes all polynomially bounded reward functions.

Problem: Approximate conditional expected reward until F.

Random variable $X_f(\pi) := f(\pi)$, if π reaches F and 0 otherwise.

Approximate $\frac{E(X_f)}{\mathcal{P}_{\mathcal{C}}(s \models \Diamond F)}$. Path exploration up-to depth d.

Three types of paths:

1. Paths π that have reached F. Add $Prob(\pi) * f(\pi)$ to reward.
2. Paths that have reached $\widetilde{F}$. Contribute nothing. Stop.
3. Paths outside F and $\widetilde{F}$. How likely are they? How much reward can they contribute if they reach F in the future?

Markov Reward Models (Cont.)

General case: Define reward-function f : Paths $\to \mathbb{R}$ that assigns rewards $f(\pi)$ to paths π.

Consider exponentially bounded reward functions $f(\pi) \leq k_1 \alpha_1^{|\pi|}$.

This subsumes all polynomially bounded reward functions.

Problem: Approximate conditional expected reward until F.

Random variable $X_f(\pi) := f(\pi)$, if π reaches F and 0 otherwise.

Approximate $\frac{E(X_f)}{\mathcal{P}_{\mathcal{C}}(s \models \Diamond F)}$. Path exploration up-to depth d.

Three types of paths:

1. Paths π that have reached F. Add $Prob(\pi) * f(\pi)$ to reward.
2. Paths that have reached $\widetilde{F}$. Contribute nothing. Stop.
3. Paths outside F and $\widetilde{F}$. How likely are they? How much reward can they contribute if they reach F in the future?

Markov Reward Models (Cont.)

General case: Define reward-function f : Paths $\to \mathbb{R}$ that assigns rewards $f(\pi)$ to paths π.

Consider exponentially bounded reward functions $f(\pi) \le k_1 \alpha_1^{|\pi|}$.
This subsumes all polynomially bounded reward functions.

Problem: Approximate conditional expected reward until F.

Random variable $X_f(\pi) := f(\pi)$, if π reaches F and 0 otherwise.

Approximate $\frac{E(X_f)}{\mathcal{P}_\mathcal{C}(s \models \Diamond F)}$. Path exploration up-to depth d.

Three types of paths:

1. Paths π that have reached F. Add $Prob(\pi) * f(\pi)$ to reward.
2. Paths that have reached $\widetilde{F}$. Contribute nothing. Stop.
3. Paths outside F and $\widetilde{F}$. How likely are they? How much reward can they contribute if they reach F in the future?

Eager Markov Chains

Markov chain is eager w.r.t. F iff

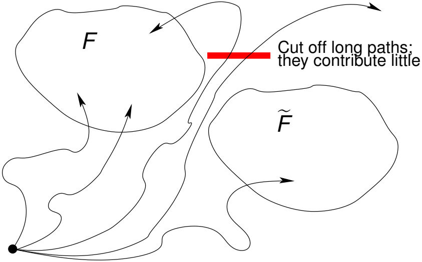
$$\mathcal{P}_{\mathcal{C}}(s \models \Diamond^{\geq n} F) \leq k_2 \, \alpha_2^n$$

for some $\alpha_2 < 1$.
Probability to reach F ‘late’ falls exponentially.
Trivially true for finite Markov chains, but also for infinite ones induced by

- PLCS, with every F.
- NTM, with every F.
- PVASS, with upward-closed F.

Eager Markov Chains: Reward Approximation

Approximate $E(X_f)$ by path exploration.
If $\alpha_1 \cdot \alpha_2 < 1$ then probability of long paths decreases faster than possible reward increases. $\longrightarrow$ Convergence.
Explore paths till sufficient depth; obtain lower bound.
Long paths contribute very little; obtain close upper bound.



Eager Markov Chains: Theorem

Theorem

Given a

- *Markov chain that is eager w.r.t. F, i.e.,* $\mathcal{P}_{\mathcal{C}}(s \models \Diamond^{\geq n} F) \leq k_2 \alpha_2^n$
- *Exponentially bounded reward function f.* $f(\pi) \leq k_1 \alpha_1^{|\pi|}$.
- $\alpha_1 \cdot \alpha_2 < 1$

Then the conditional expected reward until reaching F

$$\frac{E(X_f)}{\mathcal{P}_{\mathcal{C}}(s \models \Diamond F)}$$

can be effectively approximated up-to any error-margin $\delta > 0$.

Who is Eager and Why?

Which classes of infinite Markov chain are eager and how to compute parameters k_2 and α_2 ?

Probabilistic vector addition systems (PVASS) and
Noisy Turing Machines (NTM):
There is a fixed bound K s.t. for every state the minimal distance from F is bounded by K.
At any moment, the probability to reach F directly in $\leq K$ steps, is uniformly bounded from below.
$\Longrightarrow$ Probability to reach F 'late' falls exponentially.

Who is Eager and Why? (cont.)

Probabilistic lossy channel systems (PLCS):

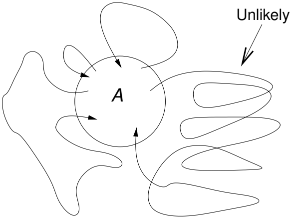
Different argument, based on properties of the finite attractor.

If many messages are in the channels then, with high probability, many messages will be lost in the next step.

$\Longrightarrow$ **Strong pull towards an attractor**.

Eager Finite Attractor

A subset A is an attractor iff $\forall s. \mathcal{P}_{\mathcal{C}}(s \models \Diamond A) = 1$.
Attractor is eager if the probability to stay outside it for $\geq n$ steps falls exponentially in n, i.e., $\leq b\beta^n$ for $\beta < 1$.

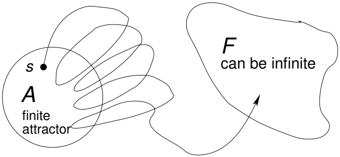


Sufficient condition: A function measuring the distance from A. In every step, prob. $> 1/2$ of getting closer, otherwise distance may increase by at most 1. (Proof by reduction to Gambler's ruin problem.) E.g., PLCS satisfy that.

Eagerness

Theorem

Markov chain with finite eager attractor is eager w.r.t. **every** *F.*



Long paths from s to F are exponentially unlikely. Why?
Long paths from s to F visit A either
Often: Every time A is visited: fixed chance to go to F directly.
Thus unlikely.
Rarely: Path stays outside A for long periods. Also unlikely.

Eagerness: How often is often?

Path-length n. 'Visiting A often' means $> \lfloor n/c \rfloor$ times (cont.c)
The probability of paths (of length n) visiting A '**often**' falls exponentially in $\lfloor n/c \rfloor$ and thus exponentially in n.
The probability of paths (of length n) visiting A '**rarely**', i.e., $\leq \lfloor n/c \rfloor$ times, is bounded by

$$\sum_{t=1}^{\lfloor n/c \rfloor} \binom{n-1}{t-1} b^t \beta^{n-t}$$

The paths are cut into t pieces between visits to A.
Each piece has some length x_i where $\sum_{i=1}^{t} x_i = n$.
$\binom{n-1}{t-1}$ is the number of ways to cut the path into t pieces.
Each piece of length x_i has upper probability bound $b\beta^{x_i - 1}$.
Altogether bounded by the product $\prod_{i=1}^{t} b\beta^{x_i - 1} = b^t \beta^{n-t}$.

Eagerness: How often is often?

Path-length n. 'Visiting A often' means $> \lfloor n/c \rfloor$ times (cont.c)
The probability of paths (of length n) visiting A '**often**' falls exponentially in $\lfloor n/c \rfloor$ and thus exponentially in n.
The probability of paths (of length n) visiting A '**rarely**', i.e., $\leq \lfloor n/c \rfloor$ times, is bounded by

$$\sum_{t=1}^{\lfloor n/c \rfloor} \binom{n-1}{t-1} b^t \beta^{n-t}$$

The paths are cut into t pieces between visits to A.
Each piece has some length x_i where $\sum_{i=1}^{t} x_i = n$.
$\binom{n-1}{t-1}$ is the number of ways to cut the path into t pieces.
Each piece of length x_i has upper probability bound $b\beta^{x_i-1}$.
Altogether bounded by the product $\prod_{i=1}^{t} b\beta^{x_i-1} = b^t \beta^{n-t}$.

Eagerness: How often is often? (Cont.)

Does this fall exponentially in n ?
Yes, for the right constant c.

The probability of paths (of length n) visiting A '**rarely**' is bounded by

$$\sum_{t=1}^{\lfloor n/c \rfloor} \binom{n-1}{t-1} b^t \beta^{n-t} \leq \left(\left(\frac{c}{c-1} \right) (2c)^{1/c} \left(\frac{1}{c} + \frac{b}{\beta} \right)^{1/c} \cdot \beta \right)^n$$

For sufficiently large c (depending on b and $\beta < 1$) the base is < 1.

Conclusion

- Many properties of finite DTMC do not carry over to infinite DTMC, but some weaker properties are often retained.
- Program-induced infinite DTMC have a particular structure which can be used in the analysis.
  - Sequential decomposition (PPDA).
  - Monotonicity; Finite distance to UC target sets (PVASS).
  - Finite Attractor (PLCS).

Open Questions

- Are $Prob(\Diamond F)$, $Prob(\Box\Diamond F)$ algebraic for PVASS, PLCS?
- Let $\vec{0}$ be the empty PVASS configuration.
  Is $Prob(\Diamond\vec{0}) = 1$ decidable?
  Can $Prob(\Diamond\vec{0})$ be approximated?
  Complex questions about multi-dimensional random walks.
- More efficient numerical approximation methods?
- Acceleration techniques?
- Infinite-state systems with continuous-time semantics?