Computing rewards in probabilistic pushdown systems

Javier Esparza

Software Reliability and Security Group

University of Stuttgart

Joint work with

Tomáš Brázdil Antonín Kučera Richard Mayr

Motivation for pushdown systems

Model checkers of the first generation (SPIN, SMV, Murphi, . . .) only work for systems (programs) with finitely many states.

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

Systems with non-recursive procedures can be “flattened” using inlining. However, inlining may cause an exponential blow-up in the size of the system. This is inefficient and unnecessary!

Goal: Design verifiers that work directly on the procedural representation.

Formalisms: pushdown systems and recursive Markov chains.

Pushdown systems

A pushdown system (PDS) 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 (notation: $pX \hookrightarrow q\alpha$).

A configuration is a pair $p\alpha$, where $p \in P$ and $\alpha \in \Gamma^*$.
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 procedural programs to pushdown systems

Full state of a procedural programs: $(\, g, n, l, (n_1, l_1) \dots (n_k, l_k)\,)$, 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 the caller procedures.

Modelled as a configuration $pXY_1 \dots Y_k$ where

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

pX (the configuration's head) contains the current program state.

$Y_1 \dots Y_k$ (the configuration's tail) contains activation records.

Correspondence between program instructions and PDS rules:

procedure call	$pX \hookrightarrow qYZ$
return	$pX \hookrightarrow q\varepsilon$
assignment	$pX \hookrightarrow qY$

In this talk: probabilistic pushdown systems as a model of probabilistic programs with procedures.

- Probabilities postulated, guessed, or estimated.

Probabilistic pushdown systems

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

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

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

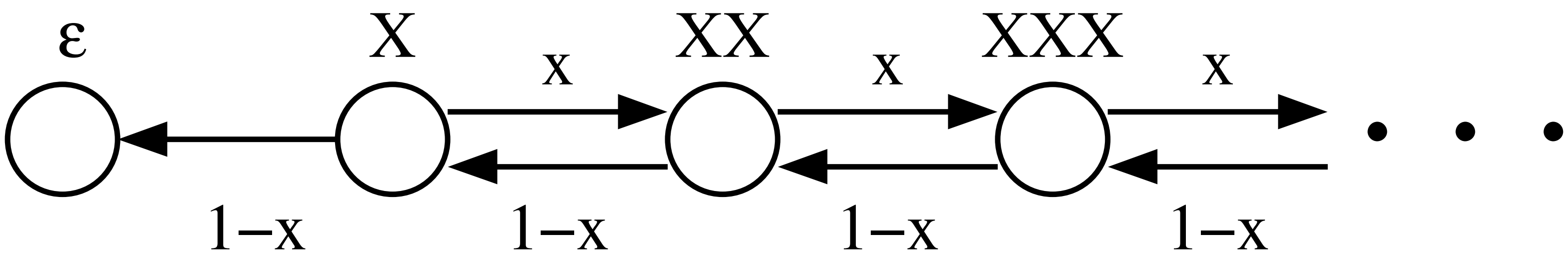
Notation: We write $pX \overset{x}{\hookrightarrow} q\alpha$ for $Prob(pX \hookrightarrow q\alpha) = x$.

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

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

A small example

$$pX \xhookrightarrow{x} pXX$$
$$pX \xhookrightarrow{1-x} p\varepsilon$$



Questions

Correctness properties:

- Will the program terminate?
  (Is the measure of the terminating runs at least ρ?)

  Will every request be granted?
  (Is the measure of the infinite runs in which requests are granted at least ρ?)

Performance properties:

- How long will it take the program to terminate?
  (Which is the expected running time)

  How long will it take in average to serve a request?
  (Which is the expected limit of the service time?)

Rewards

Let $(S, \longrightarrow, Prob)$ be a (finite or infinite) Markov chain.

A reward function is a function $f : S \to \mathbb{R}^+$ that assigns to each state a nonnegative real number.

- Average time spent in the state, costs or gains collected by visiting the state, 0 or 1 according to whether the state is "important" or not.

  Rewards could also be assigned to transitions.

The function F assigns to each finite path of the chain its accumulated reward (initial state not included).

The gain is the random variable $G(w)$ that assigns to an infinite run $w = s_0 s_1 s_2 \dots$ the value

$$G(w) = \begin{cases} \lim\limits_{n \to \infty} \dfrac{F(s_0 \dots s_n)}{n} & \text{if the limit exists} \\ \perp & \text{otherwise} \end{cases}$$

Rewards in probabilistic pushdown systems

In PPDS the states of the Markov chain are configurations $p\alpha$.

We restrict ourselves to

- Simple reward functions:

$$f(p\alpha) = g(p)$$

- Linear reward functions:

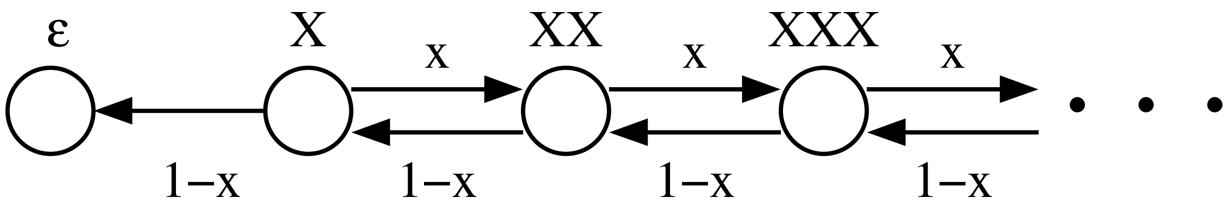
$$f(p\alpha) = g(p) + \sum_{\mathbf{X} \in \Gamma} h(X) \cdot \#_X(\alpha)$$

We consider the following problems:

- Compute the expected accumulated reward of the terminating runs.
- Compute the expected gain of the infinite runs.

Preliminaries: Termination probabilities

$$\begin{array}{rcl} pX & \xhookrightarrow{x} & pXX \\ pX & \xhookrightarrow{1-x} & p\varepsilon \end{array}$$



The PPDS terminates with probability 1 iff $x \leq 1/2$

Termination probabilities no loner determined by the chain's topology only, as in the finite state case.

A basic result

Let $[pXq]$ be the probability of, starting at the configuration pX, eventually reaching the configuration $q\epsilon$ (i.e., terminating in state q).

Theorem: The $[pXq]$ are the least solution of the following system of equations on the variables $\{\langle pXq\rangle \mid p, q \in P, X \in \Gamma\}$:

$$\begin{aligned}
\langle pXq\rangle \;=\;& \sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x \\
+& \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot \langle rYq\rangle \\
+& \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot \sum_{s\in P} \langle rYs\rangle \cdot \langle sZq\rangle
\end{aligned}$$

The system has the form $\overrightarrow{x} = F(\overrightarrow{x})$ for a quadratic form F.

Solving the system

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

Reduction to the decision problem for the existential theory of the reals, (first-order logic over the signature $(0, 1, +, *, <)$, interpreted on the reals)

The $[pXq]$s can be approximated using Newton's method [EY05].

Expected accumulated reward

Let $E^f(pXq)$ be the conditional expected accumulated reward of a path from pX to $q\varepsilon$ w.r.t. a simple reward function f.

Theorem [EKM05]:

$$
\begin{aligned}
E^f(pXq) = \frac{1}{[pXq]} \cdot \Bigg(& \sum_{pX \xhookrightarrow{x} q\varepsilon} x \cdot f(q) \\
& + \sum_{pX \xhookrightarrow{x} rY} x \cdot [rYq] \cdot (f(r) + E^f(rYq)) \\
& + \sum_{pX \xhookrightarrow{x} rYZ} x \cdot \sum_{s \in P} [rYs] \cdot [sZq] \cdot (f(r) + E^f(rYs) + E^f(sZq)) \Bigg)
\end{aligned}
$$

Expected gain: The finite case

Recall that for $\sigma = s_0 s_1 s_2 \ldots$:

$$G(\sigma) = \begin{cases} \lim\limits_{n\to\infty} \dfrac{F(s_0 \ldots s_n)}{n} & \text{if the limit exists} \\ \perp & \text{otherwise} \end{cases}$$

If the chain has a stationary distribution π then

$$E(G) = \sum_{s \in S} \pi(s) \cdot f(s)$$

If the chain is not strongly connected, then compute for each bottom strongly connected C

- C's stationary distribution,
- the gain of a trajectory that gets trapped in C, and
- the probability of getting trapped in C.

Expected gain: The PPDS case

PPDS chains may have infinitely many bottom strongly connected components.

Irreducible PPDS-chains may have no stationary distribution.

Even if the stationary distribution π exists, we are not yet done. We have

$$E(G) = \sum_{p\alpha \in P\Gamma^*} f(p\alpha) \cdot \pi(p\alpha) = \sum_{p \in P} f(p) \cdot \sum_{\alpha \in \Gamma^*} \pi(p\alpha)$$

and computing

$$\sum_{\alpha \in \Gamma^*} \pi(p\alpha)$$

may be complicated!

And yet . . .

Theorem [EKM05,BEK05] (loosely formulated):
$E(G)$ exists for every simple or linear reward function and every PPDS such that $E^{1}(pXq)$ are finite for every pXq. Moreover:

- $E(G)$ is expressible in the first-order theory of the reals, and
- $E(G)$ can be obtained by solving a system of linear equations whose coefficients are functions of the $[pXq]$.
  (Computing the $[pXq]$ remains the only difficult problem.)

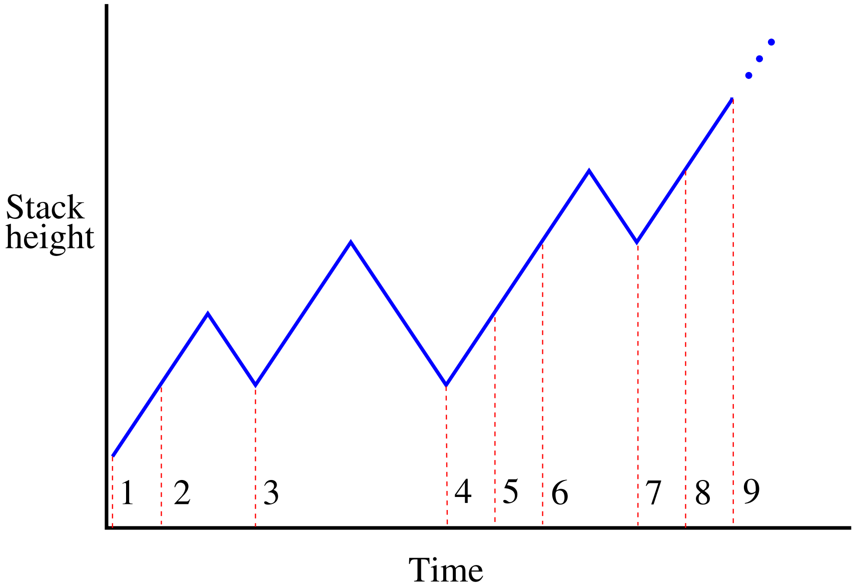
Sketch of the computational procedure:

- Compute a finite-state abstraction $\mathbf{A}$ of the infinite Markov chain.
- Compute $E(G)$ from the stationary distribution of the (bottom strongly connected components of the) chain $\mathbf{A}$.

The abstraction I: 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, where $|\alpha_0| = 1$.

$p_i\alpha_i$ is a minimum of w if $|\alpha_j| \geq |\alpha_i|$ for all $j \geq i$. (The tail 'stays in the stack').



The i-th minimum of w is the i-th configuration of the subsequence of minima.
Observe: the first minimum is $p_0\alpha_0$.

Let $p_i\alpha_i$ and $p_{i+1}\alpha_{i+1}$ be the i-th and $(i+1)$-th minima, respectively.
$p_{i+1}\alpha_{i+1}$ is a jump if $|\alpha_i| < |\alpha_{i+1}|$, and a bump if $|\alpha_i| = |\alpha_{i+1}|$.

The abstraction II: The memoryless property

Recall: pX is the head of $pX\alpha$, and α is its tail

Theorem [EKM04] (loosely formulated):
For every $i \geq 1$, the probability that the $(i+1)$-th minimum of a run has head pX and is a jump (or a bump) depends only on the head of the i-th minimum (and is in particular independent of i).

We define the finite-state Markov chain $\mathbf{A}$ as follows:

- the states are pairs $\langle pX, m\rangle$, where pX is a head and $m \in \{0, +\}$;
- a transition $\langle pX, m\rangle \Rightarrow \langle qY, +\rangle$ is assigned the probability of reaching a jump with head qY from a minimum with head pX;
- a transition $\langle pX, m\rangle \Rightarrow \langle qY, +\rangle$ is assigned the probability of reaching a bump with head qY from a minimum with head pX.

We also call a transition of this chain a macro-step.

The abstraction III: Computing transition probabilities

Let $[\langle pX, m\rangle \Rightarrow \langle qY, m'\rangle]$ be the probability of $\langle pX, m\rangle \Rightarrow \langle qY, m'\rangle$.

Define $[pX\!\uparrow] = 1 - \sum_{q \in P} [pXq]$ (This is the prob. of non-termination).

Theorem [EKM04]:

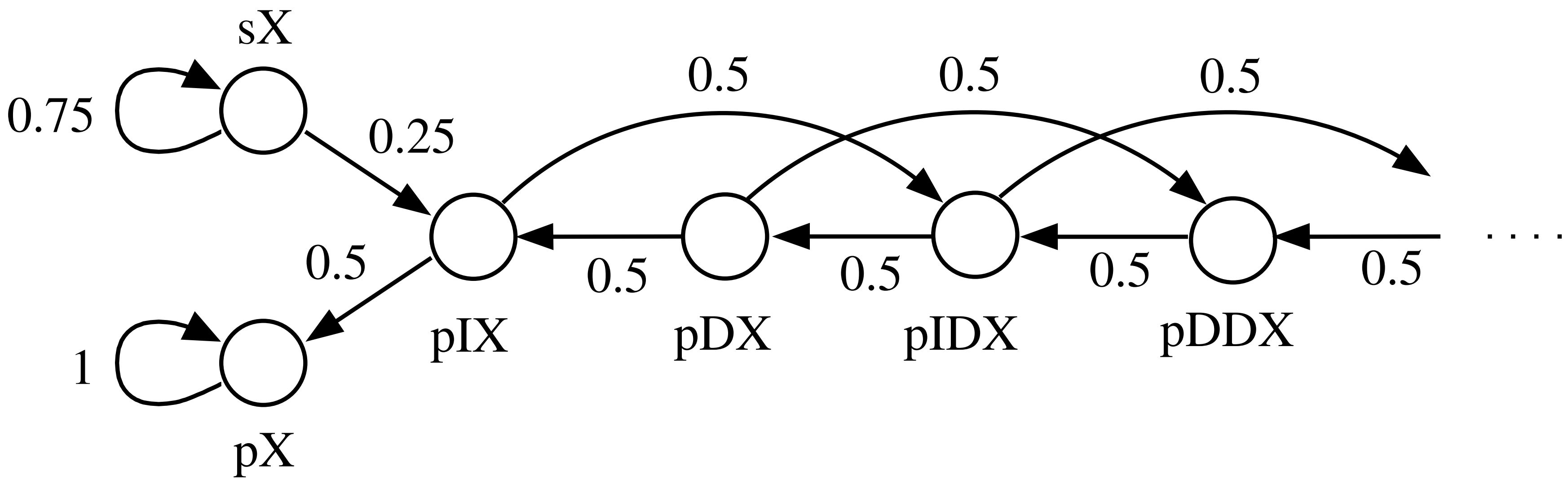
$$[\langle pX, m\rangle \Rightarrow \langle qY, +\rangle] = \frac{1}{[pX\!\uparrow]} \sum_{pX \overset{x}{\hookrightarrow} qYZ} x \cdot [qY\!\uparrow]$$

$$[\langle pX, m\rangle \Rightarrow \langle qY, 0\rangle] = \frac{1}{[pX\!\uparrow]} \left(\sum_{pX \overset{x}{\hookrightarrow} rZY} x \cdot [rZq] \cdot [qY\!\uparrow] + \sum_{pX \overset{x}{\hookrightarrow} qY} x \cdot [qY\!\uparrow] \right)$$

Again, computing the $[pXq]$ remains the only difficult problem!

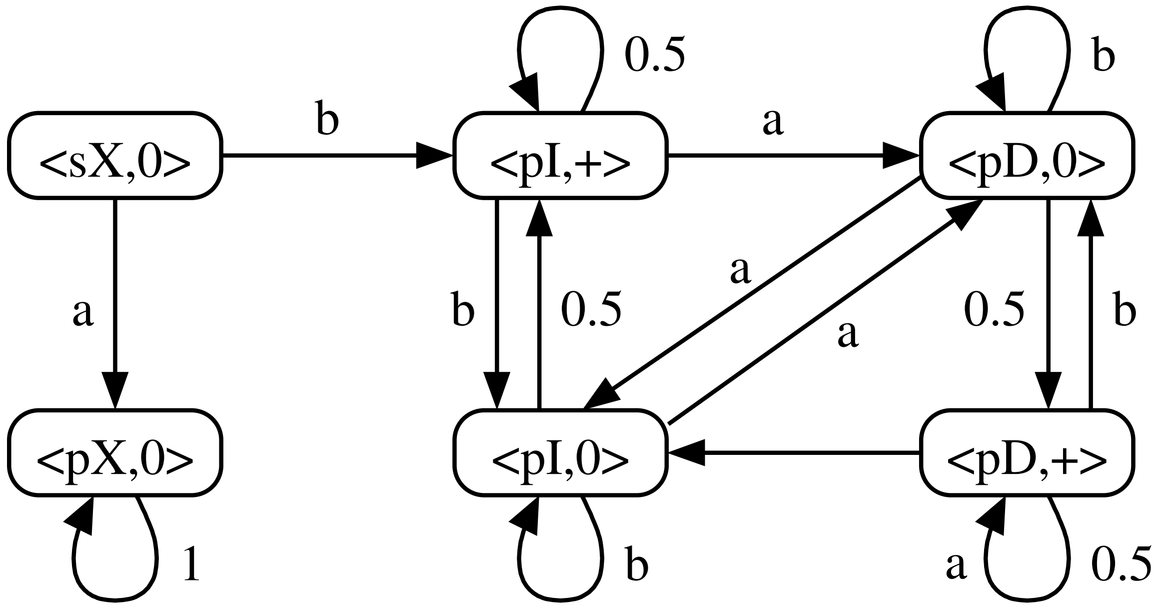
An example

sX	$\xrightarrow{0.75}$	sX	pI	$\xrightarrow{0.75}$	pID	pD	$\xrightarrow{0.5}$	pI	pX	$\xrightarrow{1}$	pX
sX	$\xrightarrow{0.25}$	pIX	pI	$\xrightarrow{0.50}$	pϵ	pD	$\xrightarrow{0.5}$	pDD			



The chain $\mathbf{A}$

$$
\begin{array}{llllllllllll}
\mathsf{sX} & \xrightarrow{0.75} & \mathsf{sX} & \mathsf{pI} & \xrightarrow{0.75} & \mathsf{pID} & \mathsf{pD} & \xrightarrow{0.5} & \mathsf{pI} & pX & \xrightarrow{1} & \mathsf{pX} \\
\mathsf{sX} & \xrightarrow{0.25} & \mathsf{pIX} & \mathsf{pI} & \xrightarrow{0.50} & \mathsf{p}\epsilon & \mathsf{pD} & \xrightarrow{0.5} & \mathsf{pDD} & & &
\end{array}
$$



$$a = \frac{\sqrt{5}-1}{2}$$

$$b = \frac{3-\sqrt{5}}{4}$$

Observe: there are runs not “represented” in $\mathbf{A}$, but they have measure 0.

Accumulated reward along a macro-step

Let $E^f(\langle pX, m\rangle \Rightarrow \langle qY, m'\rangle)$ denote the conditional expected accumulated reward along a macro-step $\langle pX, m\rangle \Rightarrow \langle qY, m'\rangle$.

$$E^f(\langle pX, m\rangle \Rightarrow \langle qY, +\rangle) \;=\; f(q)$$

$$\begin{aligned} E^f(\langle pX, m\rangle \Rightarrow \langle qY, 0\rangle) \;=&\\ \frac{1}{[\langle pX, m\rangle \Rightarrow \langle qY, 0\rangle]} \cdot &\Bigg(\sum_{pX \xhookrightarrow{x} rZY} x \cdot [rZq] \cdot [qY{\uparrow}] \cdot (f(q) + E^f(rZq)) \\ &+ \sum_{pX \xhookrightarrow{x} qY} x \cdot [qY{\uparrow}] \cdot f(q) \Bigg) \end{aligned}$$

We look at these as (conditional) “macro-rewards”.

Computing the gain for irreducible $\mathbf{A}$

Computing the gain for irreducible $\mathbf{A}$

$$E(G) \;=\; E\left(\lim_{n\to\infty} \frac{\text{acc. reward after } n \text{ steps}}{n}\right)$$

Computing the gain for irreducible $\mathbf{A}$

$$
\begin{aligned}
E(G) &= E\left(\lim_{n\to\infty} \frac{\text{acc. reward after } n \text{ steps}}{n}\right) \\
&= \lim_{n\to\infty} \left(\frac{E(\text{acc. reward after } n \text{ steps})}{n}\right)
\end{aligned}
$$

Computing the gain for irreducible $\mathbf{A}$

$$
\begin{aligned}
E(G) &= E\left(\lim_{n\to\infty} \frac{\text{acc. reward after } n \text{ steps}}{n}\right) \\
&= \lim_{n\to\infty} \left(\frac{E(\text{acc. reward after } n \text{ steps})}{n}\right) \\
&= \lim_{n\to\infty} \left(\frac{E(\text{acc. reward after } n \text{ macro-steps})}{E(\text{steps executed after } n \text{ macro-steps})}\right)
\end{aligned}
$$

Computing the gain for irreducible $\mathbf{A}$

$$
\begin{aligned}
E(G) &= E\left(\lim_{n\to\infty} \frac{\text{acc. reward after } n \text{ steps}}{n}\right) \\
&= \lim_{n\to\infty} \left(\frac{E(\text{acc. reward after } n \text{ steps})}{n}\right) \\
&= \lim_{n\to\infty} \left(\frac{E(\text{acc. reward after } n \text{ macro-steps})}{E(\text{steps executed after } n \text{ macro-steps})}\right) \\
&= \frac{E(\text{aver. macro-reward in steady st. })}{E(\text{aver. length of a macro-step in steady st. })}
\end{aligned}
$$

Computing the gain for irreducible $\mathbf{A}$

$$
\begin{aligned}
E(G) &= E\left(\lim_{n\to\infty} \frac{\text{acc. reward after } n \text{ steps}}{n}\right) \\
&= \lim_{n\to\infty} \left(\frac{E(\text{acc. reward after } n \text{ steps})}{n}\right) \\
&= \lim_{n\to\infty} \left(\frac{E(\text{acc. reward after } n \text{ macro-steps})}{E(\text{steps executed after } n \text{ macro-steps})}\right) \\
&= \frac{E(\text{aver. macro-reward in steady st. })}{E(\text{aver. length of a macro-step in steady st. })} \\
&= \frac{\displaystyle\sum_{\langle pX,m\rangle \stackrel{x}{\Rightarrow} \langle qY,m'\rangle} (\pi(\langle pX,m\rangle)\cdot x)\cdot E^{f}(\langle pX,m\rangle \Rightarrow \langle qY,m'\rangle)}{\displaystyle\sum_{\langle pX,m\rangle \stackrel{x}{\Rightarrow} \langle qY,m'\rangle} (\pi(\langle pX,m\rangle)\cdot x)\cdot E^{1}(pXq)}
\end{aligned}
$$

Linear reward functions

Recall: $f(p\alpha) = g(p) + \sum_{\mathbf{X}\in\Gamma} h(X) \cdot \#_X(\alpha)$

Key issue: compute $E^f(pXq)$.

$$E^f(pXq) = \frac{1}{[pXq]} \cdot \Bigg(\sum_{pX \overset{x}{\hookrightarrow} q\varepsilon} x \cdot f(q)$$

$$+ \sum_{pX \overset{x}{\hookrightarrow} rY} x \cdot [rYq] \cdot (f(rY) + E^f(rYq))$$

$$+ \sum_{pX \overset{x}{\hookrightarrow} rYZ} x \cdot \sum_{s\in P} [rYs] \cdot [sZq] \cdot (f(rYZ) + E^f(rYs) + E^f(sZq)) + h(Z) \cdot E^1(rYs) \Bigg)$$

Probability of performance

What is the probability that the average service time of a run is between 30 and 32 seconds?

What is the probability of those runs where the average service time is between 30 and 32 seconds, and the average deviation from 31 seconds is at most 5 seconds?

Theorem [BEK05] (very informally): Whether a run satisfies the property depends only on

- the BSCC hit by its corresponding “macro-run”, and
- whether the stack content at the hit point belongs to an effectively computable regular language.

Conclusions

PPDS are a "tractable" class of infinite-state Markov chains.

Classical performance evaluation problems solvable by means of a finite-state abstraction.

Price to pay: from linear to quadratic systems of equations.

Conclusions

PPDS are a “tractable” class of infinite-state Markov chains.

Classical performance evaluation problems solvable by means of a finite-state abstraction.

Price to pay: from linear to quadratic systems of equations.

Thank you for your attention!

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 cylinders $Run(w)$, 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