Analyzing probabilistic pushdown automata

**Tomáš Brázdil · Javier Esparza · Stefan Kiefer ·
Antonín Kučera**

Abstract The paper gives a summary of the existing results about algorithmic analysis of
probabilistic pushdown automata and their subclasses.

Keywords Markov chains · Pushdown automata · Model checking

1 Introduction

Discrete time Markov chains (Markov chains for short) have been used for decades to ana-
lyze stochastic systems in many disciplines, from Physics to Economics, from Engineering
to Linguistics, and from Biology to Computer Science.

Probabilistic model-checking introduces a new element in the practices of Markov chain
users. By extending simple programming languages and specification logics with stochastic
components, probabilistic model checkers allow scientists from all disciplines to describe,
modify, and explore Markov Chain models with far more flexibility and far less human
effort. At the same time, this more prominent rôle of modeling formalisms is drawing the
attention of computer science theorists to the classes of Markov chains generated by natural
programming languages. In particular, one such class has been extensively studied since the
mid 00s: Markov chains generated by programs with (possibly recursive) procedures. This

T. Brázdil · A. Kučera (✉)
Masaryk University, Brno, Czech Republic
e-mail: kucera@fi.muni.cz

T. Brázdil
e-mail: brazdil@fi.muni.cz

J. Esparza
TU München, Munich, Germany
e-mail: esparza@in.tum.de

S. Kiefer
University of Oxford, Oxford, UK
e-mail: stekie@cs.ox.ac.uk

class is captured by two equivalent formalisms: *probabilistic Pushdown Automata (pPDA)* [8, 14, 17, 25, 26], and *Recursive Markov Chains* [29–32]. Intuitively, the equivalence of the two models derives from the well-known fact that a recursive program can be compiled into a "plain" program that manipulates a stack, i.e., into a pushdown automaton.

Apart from being a natural model for probabilistic programs with procedures, pPDA are strongly related to several classes of stochastic processes that have been extensively studied within and outside computer science. Since recursion is a particular modality of reproduction, many questions about pPDA are related to similar questions about *branching processes*, the class of stochastic processes modeling populations whose individuals may reproduce and die. Branching processes have numerous applications in nuclear physics, genomics, ecology, and computer science [4, 35]. Probabilistic PDA are also a generalization of *stochastic context-free grammars*, very much used in natural language processing and molecular biology, and of many variants of *one-dimensional random walks*.

Markov chains generated by pPDA may have infinitely many states, which makes their analysis challenging. Properties that researchers working on finite-state Markov chains take for granted (e.g., that every state of an irreducible Markov chain is visited infinitely often almost surely) fail for pPDA chains. In particular, analysis questions that in the finite case only depend on the topology of the chain (e.g., whether a given state is recurrent), depend for pPDA on nontrivial mathematical properties of the actual values of the transition probabilities. At the same time, pPDA exhibit a lot of structure, which in the last years has been exploited to prove a wealth of surprising results. Polynomial algorithms have been found for many analysis problems, and surprising connections have been discovered to various areas of probability theory and mathematics in general (spectral theory, martingale theory, numerical methods, etc.).

This paper surveys the theory of pPDA and of two important subclasses. Loosely speaking, a PDA is a finite automaton whose transitions push and pop symbols into and from a stack. An important class of pPDA, equivalent to stochastic context-free grammars, are those whose automaton only has one state. For historical reasons, this class is called pBPA (probabilistic Basic Process Algebra). The second important subclass are pPDA whose stack alphabet contains only one symbol, apart from the special bottom-of-the-stack marker which cannot be removed. Since in this case the stack content is completely determined by the number of symbols in it, they are equivalent to *probabilistic one-counter automata* (pOC), i.e., to finite automata whose transitions increment and decrement a counter which can also be tested for zero. It turns out that several analysis questions for pBPA and pOC can be solved efficiently (in polynomial time) by developing specific methods that are not applicable to general pPDA.

The paper covers a large number of analysis problems, organized into several blocks. We start by recalling basic notions in Sect. 2, where we also set the scope of this paper by listing the problems of our interest. Section 3 contains results on basic analysis questions: probability of termination, and probability of reaching a given state or set of states. These quantities can be captured as the least solutions of effectively constructable systems of polynomial equations with positive coefficients [25, 31], and the corresponding decision problems are mostly solved by analyzing these systems. In particular, one can approximate the least solution by a decomposed variant of Newton's method [23, 37] which leads to efficient approximation algorithms for pBPA [27] and pOC [28].

In Sect. 4, we present a recent result of [17] which allows to transform every pPDA into an "equivalent" pBPA where every stack symbol terminates with probability 1 or 0. Such pBPA are easier to analyze, and since the transformation is in some sense effective, this results leads to substantial simplifications in constructions and proofs that have been previously formulated for general pPDA, which is explicitly documented in subsequent sections.

Section 5 studies the expected termination time or, more generally, the expected total reward accumulated along a terminating run with respect to a given reward function. The presented results for pPDA are based mainly on [26]. For pOC, we present more recent results of [15] that establish and utilize a link between pOC and martingale theory. Further, we study the distribution of termination time in pPDA. Here, the transformation of Sect. 4 plays an important role and allows to define suitable martingales which are then used to derive tight tail bounds for termination time. This part is based on [17].

Section 6 presents results on the analysis of long-run average properties such as the expected long-run average reward per visited configuration (i.e., mean payoff). In this section we again utilize the transformation of Sect. 4 and show how it can be used to simplify the original proofs of [14, 25].

Finally, Sect. 7 is devoted to the decidability and complexity results for model-checking pPDA and its subclasses against formulae of linear-time and branching-time logics. This section is based mainly on [12, 19, 25, 30].

Although the overview of existing results given in this paper is not completely exhaustive (for example, we have not included the material on analyzing various discounted properties of pPDA [8], or the results on checking probabilistic bisimilarity [18]), we believe that the presented proof sketches reflect most of the crucial ideas that have been invented in this area in the last decade.

2 Preliminaries

In the paper we use $\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, and $\mathbb{R}$ to denote the sets of positive integers, integers, rational numbers, and real numbers, respectively. When $\mathbb{A}$ is some of these sets, we use $\mathbb{A}^{\geq 0}$ to denote the subset of all non-negative elements of $\mathbb{A}$, and $\mathbb{A}^{\geq 0}_{\infty}$ to denote the set $\mathbb{A}^{\geq 0} \cup \{\infty\}$, where ∞ is treated according to the standard conventions, i.e., $c < \infty$ and $\infty + c = \infty - c = \infty \cdot d = \infty$ for all $c, d \in \mathbb{R}$ where $d > 0$, and we also put $\infty \cdot 0 = 0$. The cardinality of a given set M is denoted by $|M|$. If M is a problem instance, then $\|M\|$ denotes the length of the corresponding binary encoding of M. In particular, rational numbers are always encoded *as fractions of binary numbers*, unless we explicitly state otherwise.

For every finite or countably infinite set S, the symbols S^* and S^{ω} denote the sets of all finite words and all infinite words over S, respectively. The length of a given word $u \in S^* \cup S^{\omega}$ is denoted by $len(u)$, where $len(u) = \infty$ for every $u \in S^{\omega}$. The individual letters in u are denoted by $u(0), u(1), \ldots$. The empty word is denoted by ε, where $len(\varepsilon) = 0$. We also use S^+ to denote the set $S^* \smallsetminus \{\varepsilon\}$. A binary relation $\to \subseteq S \times S$ is *total* if for every $s \in S$ there is some $t \in S$ such that $s \to t$.

A *path* in $M = (S, \to)$, where S is a finite or countably infinite set and $\to \subseteq S \times S$ a total relation, is a word $w \in S^* \cup S^{\omega}$ such that $w(i-1) \to w(i)$ for every $1 \leq i < len(w)$. A given $t \in S$ is *reachable* from a given $s \in S$, written $s \to^* t$, if there is a finite path w such that $w(0) = s$ and $w(len(w) - 1) = t$. A *run* is an infinite path. For every run w and every $i \in \mathbb{Z}^{\geq 0}$, we use w_i to denote the run obtained from w by erasing the first i letters (note that $w = w(0) \ldots w(i-1) \, w_i$). The sets of all finite paths and all runs in M are denoted by $FPath(M)$ and $Run(M)$, respectively. For every $w \in FPath(M)$, the sets of all finite paths and runs that start with w are denoted $FPath(M, w)$ and $Run(M, w)$, respectively. In particular, $Run(M, s)$, where $s \in S$, is the set of all runs initiated in s. In the following we often write just $FPath$, $Run(w)$, etc., if the underlying structure M is clear from the context.

Let $\delta > 0$, $x \in \mathbb{Q}$, and $y \in \mathbb{R}$. We say that x approximates y up to the relative error δ, if either $y \neq 0$ and $|x - y|/|y| \leq \delta$, or $x = y = 0$. Further, we say that x approximates y up to the absolute error δ if $|x - y| \leq \delta$.

2.1 Basic notions of probability theory

Let A be a finite or countably infinite set. A *probability distribution* on A is a function $f : A \to \mathbb{R}^{\geq 0}$ such that $\sum_{a \in A} f(a) = 1$. A distribution f is *rational* if $f(a)$ is rational for every $a \in A$, *positive* if $f(a) > 0$ for every $a \in A$, and *Dirac* if $f(a) = 1$ for some $a \in A$. The set of all distributions on A is denoted by $\mathscr{D}(A)$.

A *σ-field* over a set Ω is a set $\mathscr{F} \subseteq 2^{\Omega}$ that includes Ω and is closed under complement and countable union. A *measurable space* is a pair $(\Omega, \mathscr{F})$, where Ω is a set called *sample space* and $\mathscr{F}$ is a σ-field over Ω. A *probability measure* over a measurable space $(\Omega, \mathscr{F})$ is a function $\mathscr{P} : \mathscr{F} \to \mathbb{R}^{\geq 0}$ such that, for each countable collection $\{\Omega_i\}_{i \in I}$ of pairwise disjoint elements of $\mathscr{F}$, we have that $\mathscr{P}(\bigcup_{i \in I} \Omega_i) = \sum_{i \in I} \mathscr{P}(\Omega_i)$, and moreover $\mathscr{P}(\Omega) = 1$. A *probability space* is a triple $(\Omega, \mathscr{F}, \mathscr{P})$ where $(\Omega, \mathscr{F})$ is a measurable space and $\mathscr{P}$ is a probability measure over $(\Omega, \mathscr{F})$. The elements of $\mathscr{F}$ are called *(measurable) events*. Given two events $A, B \in \mathscr{F}$ such that $\mathscr{P}(B) > 0$, the *conditional probability* of A under the condition B, written $\mathscr{P}(A \mid B)$, is defined as $\mathscr{P}(A \cap B)/\mathscr{P}(B)$.

Let $(\Omega, \mathscr{F}, \mathscr{P})$ be a probability space. A *random variable* is a function $X : \Omega \to \mathbb{R}_{\infty}$ such that $X^{-1}(I) \in \mathscr{F}$ for every open interval I in $\mathbb{R}$ (note that $X^{-1}(\{\infty\}) = \Omega \smallsetminus X^{-1}(\mathbb{R})$ and hence $X^{-1}(\{\infty\}) \in \mathscr{F}$). The expected value of X is denoted by $\mathbb{E}(X)$, and for every event $B \in \mathscr{F}$ such that $\mathscr{P}(B) > 0$ we use $\mathbb{E}(X \mid B)$ to denote the conditional expected value of X under the condition B.

In this paper we employ basic results and tools of *martingale theory* (see, e.g., [39, 43]) to analyze the distribution and tail bounds for certain random variables.

Definition 1 An infinite sequence of random variables $m^{(0)}, m^{(1)}, \ldots$ over the same probability space is a *martingale* if for all $i \in \mathbb{Z}^{\geq 0}$ we have the following:

- $\mathbb{E}(|m^{(i)}|) < \infty$;
- $\mathbb{E}(m^{(i+1)} \mid m^{(1)}, \ldots, m^{(i)}) = m^{(i)}$ almost surely.

Two generic results about martingales that are relevant for the purposes of this paper are *Azuma's inequality* and the *optional stopping theorem*. Let $m^{(0)}, m^{(1)}, \ldots$ be a martingale such that $|m^{(k)} - m^{(k-1)}| < d$ for all $k \in \mathbb{N}$, and let $\tau : \Omega \to \mathbb{Z}^{\geq 0}$ be a random variable over the underlying probability space of $m^{(0)}, m^{(1)}, \ldots$ such that $\mathbb{E}(\tau)$ is finite and τ is a stopping time, i.e., for all $k \in \mathbb{Z}^{\geq 0}$ the occurrence of the event $\tau = k$ depends only on the values $m^{(0)}, \ldots, m^{(k)}$. Then Azuma's inequality states that for every $b > 0$ we have that both $\mathscr{P}(m^{(n)} - m^{(0)} \geq b)$ and $\mathscr{P}(m^{(n)} - m^{(0)} \leq -b)$ are bounded by

$$\exp\left(\frac{-b^2}{2nd^2}\right),$$

and the optional stopping theorem guarantees that $\mathbb{E}(m^{(\tau)}) = \mathbb{E}(m^{(0)})$.

The semantics of probabilistic pushdown automata is defined in terms of discrete-time Markov chains, which are recalled in our next definition.

Definition 2 A *(discrete-time) Markov chain* is a triple $M = (S, \to, Prob)$ where S is a finite or countably infinite set of *vertices*, $\to \subseteq S \times S$ is a total *transition relation*, and *Prob* is a function which to each transition $s \to t$ of M assigns its probability $Prob(s \to t) \in (0, 1]$ so that for every $s \in S$ we have $\sum_{s \to t} Prob(s \to t) = 1$.

In the rest of this paper we also write $s \xrightarrow{x} t$ to indicate that $s \to t$ and $Prob(s \to t) = x$. To every $s \in S$ we associate the probability space $(Run(s), \mathscr{F}, \mathscr{P})$ where $\mathscr{F}$ is the σ-field generated by all *basic cylinders Run(w)* where $w \in FPath(s)$, and $\mathscr{P} : \mathscr{F} \to [0, 1]$ is the unique probability function such that $\mathscr{P}(Run(w)) = \Pi_{i=1}^{len(w)-1} x_i$ where $w(i-1) \xrightarrow{x_i} w(i)$ for every $1 \le i < len(w)$ (the empty product is equal to 1).

2.2 First-order theory of the reals

At many places in this paper, we rely on decision procedures for various fragments of $(\mathbb{R}, +, *, \le)$, i.e., first-order theory of the reals (also known as "Tarski algebra"). Given a closed first-order formula Φ over the signature $\{+, *, \le\}$, the problem whether Φ holds in the universe of all real numbers, with the standard interpretation of $+$ and $*$, is decidable [40] (note that "$-$" and "$/$" are easily definable from "$+$" and "$*$", and hence they can be freely used in formulae of Tarski algebra).

The existential fragment of $(\mathbb{R}, +, *, \le)$ is decidable in polynomial space [20] (the same upper bound of course holds also for the universal fragment), and the fragment where the alternation depth of the universal/existential quantifiers is bounded by a fixed constant is decidable in exponential time [33].

Some of the results presented in next sections are obtained by demonstrating that certain quantities can be effectively encoded by formulae of Tarski algebra in the following sense:

Definition 3 We say that a given tuple $(c_1, \dots, c_n)$ of reals is *expressible* in some fragment of Tarski algebra if there exists a formula Φ of the respective fragment with n free variables $x_1, \dots, x_n$ such that for every tuple $(d_1, \dots, d_n)$ of reals we have that $\Phi(x_1/d_1, \dots, x_n/d_n)$ holds iff $d_i = c_i$ for all $1 \le i \le n$. (Here $\Phi(x_1/d_1, \dots, x_n/d_n)$ is the closed formula obtained from Φ by substituting each x_i with d_i.)

2.3 Probabilistic pushdown automata

Pushdown automata (PDA) are a natural model for sequential systems with recursion. Important subclasses of PDA are stateless pushdown automata (BPA)[1] and one-counter automata (OC). Probabilistic variants of these models, denoted by pPDA, pBPA, and pOC, respectively, are obtained by associating probabilities to transition rules so that the total probability of all rules applicable to a given configuration is one. Thus, every pPDA, pBPA, and pOC generates an infinite-state Markov chain. Let us note that PDA are equivalent (in a well-defined sense) to recursive state machines (RSM), and the BPA subclass corresponds to 1-exit RSM [2, 3]. There are efficient (linear-time) translations between these models, and the same applies to their probabilistic variants.

Definition 4 A *probabilistic pushdown automaton (pPDA)* is a tuple $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ where Q is a finite set of *control states*, Γ is a finite *stack alphabet*, $\hookrightarrow \subseteq (Q \times \Gamma) \times (Q \times \Gamma^*)$ is a finite set of *rules* such that

– for every $pX \in Q \times \Gamma$ there is at least one rule of the form $pX \hookrightarrow q\alpha$,
– for every rule $pX \hookrightarrow q\alpha$ we have that $len(\alpha) \le 2$,

---

[1]The "BPA" acronym stands for Basic Process Algebra and is used mainly for historical reasons.

and *Prob* is a function which to every rule $pX \hookrightarrow q\alpha$ assigns its probability $Prob(pX \hookrightarrow q\alpha)$ $\in (0, 1]$ so that for all $pX \in Q \times \Gamma$ we have that $\sum_{pX \hookrightarrow q\alpha} Prob(pX \hookrightarrow q\alpha) = 1$. A *configuration* of Δ is an element of $Q \times \Gamma^*$.

If a pPDA Δ is used as an input to some algorithm, we implicitly assume that all transition probabilities are *rational*, unless we explicitly state otherwise. In particular, in Sect. 4 we also consider pBPA where the transitions probabilities are irrational but expressible in Tarski algebra, and in this case we use the corresponding formulae of $(\mathbb{R}, +, *, \leq)$ to represent transition probabilities (cf. Definition 3).

In the rest of this paper we write $pX \overset{x}{\hookrightarrow} q\alpha$ to indicate that $pX \hookrightarrow q\alpha$ and $Prob(pX \hookrightarrow q\alpha) = x$. The *head* of a configuration $pX\alpha$ is pX.

To Δ we associate the Markov chain M_Δ where $Q \times \Gamma^*$ is the set of vertices and the transitions are determined as follows:

– $p\varepsilon \overset{1}{\to} p\varepsilon$ for every $p \in Q$;
– for every $\beta \in \Gamma^*$, $pX\beta \overset{x}{\to} q\alpha\beta$ is a transition of M_Δ iff $pX \overset{x}{\hookrightarrow} q\alpha$ is a rule of Δ.

Since pPDA configurations are strings over a finite alphabet, we can interpret sets of configurations as languages. For our purposes, *regular* sets of configurations are particularly important.

Definition 5 Let $\mathcal{C} \subseteq Q \times \Gamma^*$ be a set of configurations. We say that $\mathcal{C}$ is *regular* if there is a deterministic finite-state automaton (DFA) $\mathcal{A}$ over the alphabet $Q \cup \Gamma$ such that $p\alpha \in \mathcal{C}$ iff the *reverse* of $p\alpha$ is accepted by $\mathcal{A}$. Further, we say that $\mathcal{C}$ is *simple* if there is a set $\mathcal{H} \subseteq Q \times \Gamma$ such that $p\alpha \in \mathcal{C}$ iff $\alpha \neq \varepsilon$ and the head of $p\alpha$ belongs to $\mathcal{H}$.

Remark 1 Since the DFA $\mathcal{A}$ of Definition 5 reads the stack in the bottom-up direction, one can easily simulate $\mathcal{A}$ on-the-fly in the stack alphabet of Δ. Formally, we construct another pPDA Δ' which has the same control states as Δ, the stack alphabet of Δ' is $\Gamma' = (\Gamma \times A) \cup Z_0$, where Z_0 is a fresh bottom-of-the-stack marker and A is the set of control states of $\mathcal{A}$, and the rules of Δ' simulate the execution of $\mathcal{A}$. For example, if $pX \overset{x}{\hookrightarrow} qYZ$ is a rule of Δ, then for every $a \in A$ we add the rule $p(X, a) \overset{x}{\hookrightarrow} q(Y, a')(Z, a)$ to Δ', where $a \overset{Z}{\to} a'$ is a transition in $\mathcal{A}$. Obviously, there is a bijective correspondence between $Run(M_\Delta, pX)$ and $Run(M_{\Delta'}, p(X, a_0)Z_0)$, where a_0 is the initial state of $\mathcal{A}$. Note that a configuration $qY\alpha$ of Δ is accepted by $\mathcal{A}$ iff the corresponding configuration $q(Y, a)\alpha'Z_0$ of Δ' satisfies $a \overset{Y}{\to} a' \overset{q}{\to} a_f$ where a_f is an accepting state of $\mathcal{A}$. Hence, the regular set of configurations of Δ encoded by $\mathcal{A}$ corresponds to a *simple* set of configurations in Δ' represented by an efficiently constructible set $\mathcal{H} \subseteq Q \times \Gamma'$. In particular, note that $q\varepsilon$ is recognized by $\mathcal{A}$ iff $qZ_0 \in \mathcal{H}$, which also explains the role of the symbol Z_0. Since the size of Δ' is *polynomial* in the size of Δ and $\mathcal{A}$, the above described construction is a generic technique for extending results about simple sets of configurations to regular sets of configurations without any complexity blowup. We use this simple principle at many places in this paper.

Important subclasses of pPDA are *stateless pPDA* (also known as *pBPA*) which do not have control states, and probabilistic one-counter automata (pOC) where the stack alphabet has just one symbol (apart from the bottom-of-the-stack marker) and can be interpreted as a counter.

Definition 6 A *pBPA* is a triple $\Delta = (\Gamma, \hookrightarrow, Prob)$ where Γ is a finite *stack alphabet*, $\hookrightarrow \subseteq \Gamma \times \Gamma^*$ is a finite set of *rules* such that
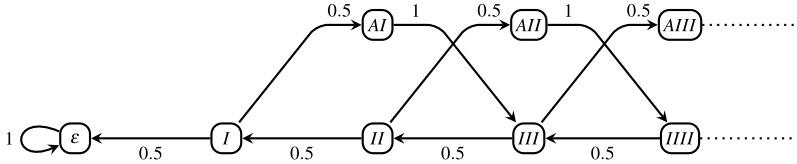
Fig. 1 A fragment of M_Δ

– for every $X \in \Gamma$ there is at least one rule of the form $X \hookrightarrow \alpha$,
– for every rule $X \hookrightarrow \alpha$ we have that $len(\alpha) \le 2$,

and *Prob* is a *probability assignment* which to each rule $X \hookrightarrow \alpha$ assigns its probability $Prob(X \hookrightarrow \alpha) \in (0, 1]$ so that for all $X \in \Gamma$ we have that $\sum_{X \hookrightarrow \alpha} Prob(X \hookrightarrow \alpha) = 1$. A *configuration* of Δ is an element of Γ^*.

Note that each pBPA Δ can be understood as a pPDA with just one control state p which is omitted in the rules and configurations of Δ. Thus, all notions introduced for pPDA can be adapted to pBPA. In particular, each pBPA Δ determines a unique Markov chain M_Δ where Γ^* is the set of vertices and the transitions are determined in the expected way.

Example 1 Consider a pBPA Δ with two stack symbols I, A and the rules

$$I \overset{0.5}{\hookrightarrow} \varepsilon, \quad I \overset{0.5}{\hookrightarrow} AI, \quad A \overset{1}{\hookrightarrow} II.$$

A fragment of M_Δ is shown in Fig. 1.

A formal definition of pOC adopted in this paper is consistent with the one used in recent works such as [10, 11, 15].

Definition 7 A *pOC* is a tuple $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$, where

– Q is a finite set of *control states*,
– $\delta^{>0} \subseteq Q \times \{-1, 0, 1\} \times Q$ and $\delta^{=0} \subseteq Q \times \{0, 1\} \times Q$ are the sets of *positive* and *zero rules* such that each $p \in Q$ has an outgoing positive rule and an outgoing zero rule;
– $P^{>0}$ and $P^{=0}$ are *probability assignments*; both assign to each $p \in Q$ a positive probability distribution over the outgoing transitions in $\delta^{>0}$ and $\delta^{=0}$, respectively, of p.

We say that Δ is *zero-trivial* if $\delta^{=0} = \{(p, 0, p) \mid p \in Q\}$. A *configuration* of Δ is a pair $p(i) \in Q \times \mathbb{Z}^{\ge 0}$.

The Markov chain M_Δ associated to Δ has $Q \times \mathbb{Z}^{\ge 0}$ as the set of vertices, and the transition are determined as expected. Note that the transitions enabled at $p(0)$ are not necessarily enabled at $p(i)$ where $i > 0$, and vice versa. If Δ is zero-trivial, then the only transition enabled at $p(0)$ is the loop $p(0) \overset{1}{\to} p(0)$. Also observe that $p(i) \overset{x}{\to} p(i+c)$ iff $p(j) \overset{x}{\to} p(j+c)$ for all $i, j > 0$ and $c \in \{-1, 0, 1\}$.

Each pOC can also be understood as a pPDA with just two stack symbols I and Z, where Z marks the bottom of the stack, and the number of pushed I's represents the counter value. The translations between the two models are linear as long as the counter changes are bounded (as in Definition 7) or encoded in unary.

2.4 The problems of interest

In this section we formally introduce the main concepts and notions used in performance and dependability analysis of probabilistic systems modeled by discrete-time Markov chains. In the next sections we show how to solve the associated algorithmic problems for pPDA and its subclasses.

For the rest of this section, we fix a Markov chain $M = (S, \rightarrow, Prob)$ where S is finite or countably infinite.

Reachability and termination Let $s \in S$ be an initial vertex and $T \subseteq S$ a set of *target* vertices. Let $Reach(s, T)$ be the set of all $w \in Run(s)$ such that $w(i) \in T$ for some $i \in \mathbb{Z}^{\geq 0}$ (sometimes we also write $Reach_M(s, T)$ to prevent confusions about the underlying Markov chain M). The probability of reaching T from s is defined as $\mathscr{P}(Reach(s, T))$.

For pPDA and its subclasses, we also distinguish a special form of reachability called *termination*. Intuitively, a recursive system terminates when its initial procedure terminates, i.e., the stack of activation records becomes empty. In pOC, termination corresponds to decreasing the counter to zero (this means that the set of zero rules is irrelevant, and therefore we restrict ourselves to *zero-trivial* pOC in the context of problems related to termination). Technically, we distinguish between two forms of termination.

- *Non-selective* termination, where the target set T consists of *all* configurations with empty stack (or zero counter).
- *Selective* termination, where T consists of *some* (selected) configurations with empty stack (or zero counter).

For pBPA, there is only one terminated configuration ε, and the two notions of termination coincide. For pPDA and pOC, selective termination intuitively corresponds to terminating with one of the distinguished output values.

The main algorithmic problem related to reachability is computing the probability $\mathscr{P}(Reach(s, T))$ for given s and T. For finite-state Markov chains with rational transition probabilities, $\mathscr{P}(Reach(s, T))$ is always rational and can be computed in polynomial time by solving a certain system of linear equations[2] (see, e.g., [5]). For infinite-state Markov chains generated by pPDA, we only consider *regular* sets T of target configurations encoded by the associated DFA (see Definition 5). Still, $\mathscr{P}(Reach(s, T))$ can be irrational and cannot be computed precisely in general. Hence, in this setting we refine the task of computing $\mathscr{P}(Reach(s, T))$ into the following problems:

- *Qualitative reachability/termination.* Given s and T, do we have that $\mathscr{P}(Reach(s, T)) = 1$?
- *Quantitative reachability/termination.* Given s, T, and a rational constant $\varrho \in (0, 1)$, do we have that $\mathscr{P}(Reach(s, T)) \leq \varrho$?
- *Approximating the probability of reachability/termination.* Given s, T, and a rational constant $\delta > 0$, compute $\mathscr{P}(Reach(s, T))$ up to the absolute/relative error δ.

Expected termination time and total accumulated reward Similarly as in the previous paragraph, let us fix an initial state $s \in S$ and a set of target vertices $T \subseteq S$. Further, we fix a *reward function* $f : S \rightarrow \mathbb{R}^{\geq 0}$.

---

[2]For every $t \in S$, we fix a fresh variable Y_t. If $t \in T$, we put $Y_t = 1$. If t cannot reach T at all, we put $Y_t = 0$. Otherwise, we put $Y_t = \sum_{t \xrightarrow{x} t'} x \cdot Y_{t'}$. The resulting system of linear equations has only one solution in $\mathbb{R}^{|S|}$ which is the tuple of all $\mathscr{P}(Reach(t, T))$.

We are interested in the expected total reward accumulated along a path from s to T. Formally, for every run w, let $Hit(w)$ be the least $j \in \mathbb{Z}_{\infty}^{\geq 0}$ such that $w(j) \in T$. We define a random variable $Acc : Run(s) \to \mathbb{R}_{\infty}^{\geq 0}$ where

$$Acc(w) = \sum_{i=0}^{Hit(w)-1} f\big(w(i)\big).$$

Note that if $Hit(w) = 0$, then the above sum is empty and hence $Acc(w) = 0$.

The task is to compute the expected value $\mathbb{E}(Acc)$. If $0 < \mathscr{P}(Reach(s, T)) < 1$, we are also interested in the conditional expectation $\mathbb{E}(Acc \mid Reach(s, T))$. Note that if $f(r) = 1$ for all $r \in S$, then $\mathbb{E}(Acc)$ corresponds to the expected number of transitions (or "time") needed to visit T from s.

For finite-state Markov chains, both $\mathbb{E}(Acc)$ and $\mathbb{E}(Acc \mid Reach(s, T))$ are easily computable in polynomial time. For pPDA and its subclasses, we face the same difficulties as in the case of reachability/termination, and also some new ones. In particular, $\mathbb{E}(Acc)$ can be infinite even if $\mathscr{P}(Reach(s, T)) = 1$ and f is bounded, which cannot happen for finite-state Markov chains. We also need to restrict the reward functions to some workable subclass. In particular, we consider reward functions that are

– *constant*, which suffices for modelling the discrete time;
– *simple*, i.e., for every configuration $pX\alpha$ with non-empty stack, $f(pX\alpha)$ depends just on the head pX (there are no restrictions on $f(p\varepsilon)$). Simple reward functions are sufficient for modelling the costs and payoffs determined just by the currently running procedure;
– *linear*, i.e., for every configuration $p\alpha$ we have that

$$f(p\alpha) = c(p) \cdot \left(d(p) + \sum_{X \in \Gamma} \#_X(\alpha) \cdot h(X) \right).$$

Here c, d are functions that assign a fixed non-negative value to every control state, and h does the same for stack symbols. Further, $\#_X(\alpha)$ denotes the number of occurrences of X in α. Note that by putting $c(p) = 0$, we can assign zero reward to all configurations of the form $p\alpha$.

Linear reward functions are useful for, e.g., analyzing the stack height or the total amount of memory allocated by all procedures currently stored in the stack of activation records.

Similarly to reachability/termination, we refine the problem of computing $\mathbb{E}(Acc)$ and $\mathbb{E}(Acc \mid Reach(s, T))$ for pPDA and its subclasses into several questions.

– *Finiteness*. Given s, T, and f, do we have $\mathbb{E}(Acc) < \infty$? Do we have $\mathbb{E}(Acc \mid Reach(s, T)) < \infty$?
– *Boundedness*. Given s, T, f, and $c \in \mathbb{R}^+$, do we have $\mathbb{E}(Acc) \leq c$? Do we have $\mathbb{E}(Acc \mid Reach(s, T)) \leq c$?
– *Approximating the expected accumulated reward*. Given s, T, f, and $\delta > 0$, compute $\mathbb{E}(Acc)$ and $\mathbb{E}(Acc \mid Reach(s, T))$ up to the absolute/relative error δ.

Apart from these algorithmic problems, we are also interested in the distribution of Acc, particularly in the special case when Acc models the termination time. Then $\mathscr{P}(Acc \leq c)$ is the probability that the initial configuration terminates after at most c transitions, and we ask how quickly this probability approaches the probability of termination as c increases. Thus, we obtain rather generic results about the asymptotic behavior of recursive probabilistic systems (see Theorem 11).

Mean payoff and other long-run average properties   The mean payoff is the long-run average of reward per visited vertex. Formally, we fix some reward function $f : S \to \mathbb{R}^{\geq 0}$, and define random variables *MPsup*, *MPinf* : $Run(s) \to \mathbb{R}_{\infty}^{\geq 0}$ as follows:

$$MPsup(w) = \limsup_{n \to \infty} \frac{\sum_{i=0}^{n} f(w(i))}{n+1},$$

$$MPinf(w) = \liminf_{n \to \infty} \frac{\sum_{i=0}^{n} f(w(i))}{n+1}.$$

For finite-state Markov chains, we have that $\mathscr{P}(MPsup \neq MPinf) = 0$, although there may be uncountably many $w \in Run(s)$ such that $MPsup(w) \neq MPinf(w)$. Further, both *MPsup* and *MPinf* assume only finitely many values $v_1, \ldots, v_n$ with positive probabilities $p_1, \ldots, p_n$, and these values and probabilities are computable in polynomial time by analyzing the bottom strongly connected components of the finite-state Markov chain. Hence, the expected values of *MPsup* and *MPinf* are the same and efficiently computable.

For pPDA and its subclasses, we consider the same classes of reward functions as in the case of total accumulated reward. First, we need to answer the question whether the mean payoff is well-defined for almost all runs, i.e., whether $\mathscr{P}(MPsup \neq MPinf) = 0$, and what is the distribution of *MPsup* and *MPinf*. The algorithmic problems concern mainly approximating the expected value of *MPsup* and *MPinf*, and approximating the distribution of these variables.

Model-checking linear-time and branching-time logics   Let s be a vertex of M. A *linear-time* specification is a property φ which is either true or false for every run, and we are interested in computing the probability $\mathscr{P}(\{w \in Run(s) \mid w \models \varphi\})$. The property φ can be encoded in linear-time logics such as LTL, or by finite-state automata with some ω-acceptance condition (Büchi, Rabin, Street, etc.). For finite-state Markov chains, the probability $\mathscr{P}(\{w \in Run(s) \mid w \models \varphi\})$ can be computed in time polynomial in the size of M. For pPDA and its subclasses, this probability can be irrational, and we refine the problem in the same way as for reachability. That is, we consider the *qualitative* and *quantitative* model-checking problems for linear-time specifications, and the associated *approximation* problem.

A *branching-time* specification is a state formula of a branching-time probabilistic logic such as PCTL or PCTL* [34]. Intuitively, these logics are obtained by replacing the existential and universal path quantifiers in CTL and CTL* (see, e.g., [22]) with the *probabilistic operator* $\mathscr{P}^{\sim \varrho}(\cdot)$, which says that the probability of satisfying a given path formula is $\sim$-related to ϱ. More precisely, the syntax of PCTL* *state* and *path* formulae Φ and φ, resp., is given by the following abstract syntax equations.

$$\Phi ::= a \mid \neg \Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathscr{P}^{\sim \varrho} \varphi,$$

$$\varphi ::= \Phi \mid \neg \varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathcal{X} \varphi \mid \varphi_1 \mathcal{U} \varphi_2.$$

Here a ranges over a countably infinite set Ap of atomic propositions, $\varrho \in [0, 1]$ is a rational constant, and $\sim \in \{\leq, <, \geq, >, =\}$.

The logic PCTL is a fragment of PCTL* where path formulae are given by the equation $\varphi ::= \mathcal{X} \Phi \mid \Phi_1 \mathcal{U} \Phi_2$. The *qualitative fragments* of PCTL and PCTL*, denoted by qPCTL and qPCTL*, resp., are obtained by restricting the allowed operator/number combinations in $\mathscr{P}^{\sim \varrho} \varphi$ subformulae to '$= 0$' and '$= 1$' (we do not include '< 1', '> 0' because these are

definable from '$=0$', '$=1$', and negation). Finally, a path formula φ is an *LTL formula* if all of its state subformulae are atomic propositions.

Now we define the semantics of PCTL*. Let us fix a valuation $\nu : Ap \rightarrow 2^S$. State formulae are interpreted over S, and path formulae are interpreted over *Run*. Hence, for given $s \in S$ and $w \in Run$ we define

$$
\begin{array}{llll}
s \models^\nu a & \text{iff} & s \in \nu(a), \\
s \models^\nu \neg\Phi & \text{iff} & s \not\models^\nu \Phi, \\
s \models^\nu \Phi_1 \wedge \Phi_2 & \text{iff} & s \models^\nu \Phi_1 \text{ and } s \models^\nu \Phi_2, \\
s \models^\nu \mathcal{P}^{\sim\varrho}\varphi & \text{iff} & \mathcal{P}(\{w \in Run(s) \mid w \models^\nu \varphi\}) \sim \varrho, \\
\\
w \models^\nu \Phi & \text{iff} & w(0) \models^\nu \Phi, \\
w \models^\nu \neg\varphi & \text{iff} & w \not\models^\nu \varphi, \\
w \models^\nu \varphi_1 \wedge \varphi_2 & \text{iff} & w \models^\nu \varphi_1 \text{ and } w \models^\nu \varphi_2, \\
w \models^\nu \mathcal{X}\varphi & \text{iff} & w_1 \models^\nu \varphi, \\
w \models^\nu \varphi_1\mathcal{U}\varphi_2 & \text{iff} & \text{there is } j \geq 0 \text{ s.t. } w_j \models^\nu \varphi_2 \text{ and } w_i \models^\nu \varphi_1 \text{ for all } 0 \leq i < j.
\end{array}
$$

The model-checking problem for PCTL, PCTL*, and their qualitative fragments is the question whether $s \models^\nu \Phi$ for a given vertex s and a state formula Φ of the respective logic.

For pPDA and its subclasses, the model-checking problems for linear/branching time logics are usually considered only for *regular valuations*, which assign to every atomic proposition a regular set of configurations (see Definition 5).

3 Reachability and termination

In this section we examine the problems of quantitative/qualitative reachability and the problems of approximating the probability $\mathcal{P}(Reach(s, T))$ up to the given absolute/relative error $\delta > 0$. We start with general pPDA and then show that some of the considered questions are solvable more efficiently for the pBPA and pOC subclasses.

3.1 Quantitative and qualitative reachability

Recall that the quantitative/qualitative reachability problems are the questions whether $\mathcal{P}(Reach(s, T))$ is bounded by a given rational $\varrho \in (0, 1)$ or equal to one, respectively.

Results for pPDA   First, we show how to solve the reachability problems for a simple set of target configurations. A generalization to arbitrary regular sets is then obtained by employing the generic construction recalled in Remark 1.

Let $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ be a pPDA and $\mathcal{C} \subseteq Q \times \Gamma^*$ a simple set of configurations where $\mathcal{H}$ is the associated set of heads (see Definition 5). For all $p, q \in Q$ and $X \in \Gamma$, let

- $[pX\bullet]$ denote the probability $\mathcal{P}(Reach(pX, \mathcal{C}))$;
- $[pXq]$ denote the probability of all $w \in Reach(pX, \{q\varepsilon\})$ such that w does not visit a configuration of $\mathcal{C}$.

One can easily verify that all such $[pX\bullet]$ and $[pXq]$ must satisfy the following:

- For all $pX \in \mathcal{H}$ and $q \in Q$, we have that $[pX\bullet] = 1$ and $[pXq] = 0$.

– For all $pX \notin \mathcal{H}$ and $q \in Q$, we have that

$$[pX\bullet] = \sum_{pX \xrightarrow{x} rY} x \cdot [rY\bullet] + \sum_{pX \xrightarrow{x} rYZ} x \cdot [rY\bullet] + \sum_{pX \xrightarrow{x} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZ\bullet],$$

$$[pXq] = \sum_{pX \xrightarrow{x} q\varepsilon} x + \sum_{pX \xrightarrow{x} rY} x \cdot [rYq] + \sum_{pX \xrightarrow{x} rYZ} \sum_{t \in Q} x \cdot [rYt] \cdot [tZq].$$

Now consider a system $Reach_\Delta$ of recursive equations obtained by constructing the above equalities for all $[pX\bullet]$ and $[pXq]$ where $p, q \in Q$ and $X \in \Gamma$, and replacing each occurrence of all $[pX\bullet]$ and $[pXq]$ with the corresponding fresh variables $\langle pX\bullet \rangle$ and $\langle pXq \rangle$. In general, $Reach_\Delta$ may have several solutions. It has been observed in [25, 31] that the tuple of all $[pX\bullet]$ and $[pXq]$ is exactly the *least* solution of $Reach_\Delta$ in $([0, 1]^k, \sqsubseteq)$, where $k = |\Gamma| \cdot (|Q|^2 + |Q|)$ and $\sqsubseteq$ is the component-wise ordering. Observe that if $\mathcal{C} = \emptyset$, then $[pX\bullet] = 0$ and $[pXq] = \mathscr{P}(Reach(pX, \{q\varepsilon\}))$ for all $p, q \in Q$ and $X \in \Gamma$. Hence, in the case of termination, it suffices to put $\mathcal{C} = \emptyset$ and consider a simpler system of equations $Term_\Delta$ which is the same as $Reach_\Delta$ but all $\langle pX\bullet \rangle$ variables and the corresponding equations are eliminated.

Example 2 Consider again the pBPA Δ of Example 1, and let $\mathcal{C} = \emptyset$. The system $Term_\Delta$ looks as follows:

$$\langle I \rangle = \frac{1}{2} + \frac{1}{2} \cdot \langle A \rangle \cdot \langle I \rangle,$$

$$\langle A \rangle = 1 \cdot \langle I \rangle \cdot \langle I \rangle.$$

Hence, $[I]$ is the least solution of $\langle I \rangle = \frac{1}{2} + \frac{1}{2}\langle I \rangle^3$, which is $\frac{\sqrt{5}-1}{2}$ (the golden ratio).

In general, the least solution of $Reach_\Delta$ cannot be given as a tuple of closed-form expressions. Still, we can decide if $[pX\bullet] \leq \varrho$ for a given rational $\varrho \in (0, 1)$ by encoding this question in Tarski algebra (see Sect. 2.2). More precisely, we construct a formula Φ which says

"there is $\mathbf{V} \in [0, 1]^k$ such that $\mathbf{V}$ is a solution of $Reach_\Delta$ and $\mathbf{V}_{\langle pX\bullet \rangle} \leq \varrho$".

Here $\mathbf{V}_{\langle pX\bullet \rangle}$ is the component of $\mathbf{V}$ corresponding to $\langle pX\bullet \rangle$. Note that if *some* solution $\mathbf{V}$ of $Reach_\Delta$ satisfies $\mathbf{V}_{\langle pX\bullet \rangle} \leq \varrho$, then also the *least* solution does. Since the formula Φ is existential and its size is polynomial in $\|\Delta\|$, the validity of Φ can be decided in space polynomial in $\|\Delta\|$ (see Sect. 2.2). Similarly, one can decide whether $[pX\bullet] \geq \varrho$ in polynomial space by constructing a universal formula

"for every $\mathbf{V} \in [0, 1]^k$ such that $\mathbf{V}$ is a solution of $Reach_\Delta$ we have $\mathbf{V}_{\langle pX\bullet \rangle} \geq \varrho$".

The same observations are valid also for the probability $[pXq]$, and be can be further extended to *regular* sets of target configurations (see Remark 1). Thus, we obtain the following:

Theorem 1 (see [25, 31]) *Let* $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ *be a pPDA,* $p\alpha \in Q \times \Gamma^*$ *a configuration of* Δ, *and* $\mathcal{C}$ *a regular set of configurations of* Δ *represented by a DFA* $\mathcal{A}$. *Further, let* $\varrho \in (0, 1)$ *be a rational constant. Then the problems whether* $\mathscr{P}(Reach(p\alpha, \mathcal{C})) = 1$ *and* $\mathscr{P}(Reach(p\alpha, \mathcal{C})) \leq \varrho$ *are in* **PSPACE**.

Interestingly, there are no lower complexity bounds known for the problems considered in Theorem 1. On the other hand, there is some indication that improving the presented **PSPACE** upper bound to some natural Boolean complexity class (e.g., the polynomial hierarchy) might be difficult. As observed in [31], even the problem whether $\mathscr{P}(Reach(pX, \{q\varepsilon\})) = 1$ is at least as hard as SQUARE-ROOT-SUM, whose exact complexity is a long-standing open problem in computational geometry[3].

As we shall see in Sects. 4, 5, and 6, the tuple of *termination probabilities*, i.e., the least solution of $Term_\Delta$, is useful for computing and approximating other interesting quantities. Since $Term_\Delta$ can have several solutions, it is not immediately clear that the tuple of all termination probabilities $[pXq]$ is efficiently expressible in the *existential fragment* of Tarski algebra in the sense of Definition 3. However, we can effectively extend the system $Term_\Delta$ by the following constraints:

(1) $0 \leq \langle pXq \rangle \leq 1$ for every $\langle pXq \rangle$;
(2) $\langle pXq \rangle = 0$ for every $\langle pXq \rangle$ such that $[pXq] = 0$;
(3) $\sum_{q \in Q} \langle pXq \rangle = 1$ for every $pX \in Q \times \Gamma$ such that $\sum_{q \in Q} [pXq] = 1$;
(4) $\sum_{q \in Q} \langle pXq \rangle < 1$ for every $pX \in Q \times \Gamma$ such that $\sum_{q \in Q} [pXq] < 1$.

Note that the constraints (1) and (2) can be computed in time polynomial in $\|\Delta\|$, and the constraints (3) and (4) can be computed in space polynomial in $\|\Delta\|$ by Theorem 1. It has been shown in [29, 30] that the system $Term_\Delta$ extended with the above constraints has a *unique* solution in $\mathbb{R}^k$, where $k = |Q|^2 \cdot |\Gamma|$. Thus, we obtain the following:

Theorem 2 *Let $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ be a pPDA. The tuple of all termination probabilities $[pXq]$ is expressible in the existential fragment of Tarski algebra. Moreover, the corresponding formula Φ is constructible in space polynomial in $\|\Delta\|$, and the length of Φ is polynomial in $\|\Delta\|$.*

Results for pBPA For pBPA, the *quantitative* termination, i.e., the question whether $\mathscr{P}(Reach(X, \{\varepsilon\})) \leq \varrho$ for a given rational $\varrho \in (0, 1)$, is still as hard as SQUARE-ROOT-SUM [31]. However, it has been also shown in [31] that the *qualitative* termination, i.e., the question whether $\mathscr{P}(Reach(X, \{\varepsilon\})) = 1$, is solvable in polynomial time. This is achieved by employing some results and tools of spectral theory. First, let us consider a pBPA $\Delta = (\Gamma, \hookrightarrow, Prob)$ such that

(1) for every $X \in \Gamma$ we have that $\mathscr{P}(Reach(X, \{\varepsilon\})) > 0$;
(2) for all $X, Y \in \Gamma$ there is a configuration of the form $Y\alpha$ reachable from X.

As a running example, we use a pBPA Θ with three stack symbols X, Y, Z, where

$$X \xrightarrow{0.25} YZ, \qquad Y \xrightarrow{0.5} XX, \qquad Z \xrightarrow{0.25} YX,$$
$$X \xrightarrow{0.75} \varepsilon, \qquad Y \xrightarrow{0.5} Z, \qquad Z \xrightarrow{0.25} XZ,$$
$$Z \xrightarrow{0.5} \varepsilon.$$

---

[3]An instance of SQUARE-ROOT-SUM is a tuple of positive integers $a_1, \ldots, a_n, b$, and the question is whether $\sum_{i=1}^n \sqrt{a_i} \leq b$. The problem is obviously in **PSPACE**, because it can be encoded in the existential fragment of Tarski algebra (see Sect. 2.2), and the best upper bound currently known is **CH** (counting hierarchy; see Corollary 1.4 in [1]). It is not known whether this bound can be further lowered to some natural Boolean subclass of **PSPACE**, and a progress in answering this question might lead to breakthrough results in complexity theory.

Assumptions (1) and (2) imply that either all stack symbols of Δ terminate with probability one, or all of them terminate with a positive probability strictly less than one. Hence, we need to decide whether the least solution of $Term_\Delta$ is equal to $(1, \ldots, 1)$ or strictly less than 1 in every component. To get some intuition, let us first realize that Δ can also be interpreted as a *multi-type branching process (MTBP)*. Intuitively, the symbols of Γ then correspond to different "species" which can evolve into finite collections of other species or die. For example, $X \xrightarrow{0.25} YZ$ means "one copy of X can evolve into one copy Y and one copy of Z with probability 0.25", while $X \xrightarrow{0.75} \varepsilon$ means "X dies with probability 0.75". The states of the MTBP determined by Δ are finite collections of species (stack symbols) where the ordering of symbols is irrelevant. Each occurrence of every stack symbol in the current state chooses a rule independently of the others, and the chosen rules are then executed simultaneously. The probability of this transition is obtained by multiplying the probabilities of the chosen rules. Thus, we obtain an infinite-state Markov chain B_Δ whose states are tuples of the form $(X_1^{k_1}, \ldots, X_n^{k_n})$, where $\{X_1, \ldots, X_n\} = \Gamma$ and $k_i \in \mathbb{Z}^{\geq 0}$ for every $1 \leq i \leq n$, and the transitions are defined in the way described above. For example, the set of transitions in B_Θ includes the following (for simplicity, in each tuple we omit all symbols with zero occurrence index):

$$(X, Y) \xrightarrow{0.375} (Z), \qquad (X, Y) \xrightarrow{0.125} (X^2, Y, Z), \qquad (X^2) \xrightarrow{0.375} (Y, Z).$$

The first transition is determined by the rules $X \xrightarrow{0.75} \varepsilon$ and $Y \xrightarrow{0.5} Z$, the second transition by the rules $X \xrightarrow{0.25} YZ$ and $Y \xrightarrow{0.5} XX$, and the third transition by the rules $X \xrightarrow{0.25} YZ$ and $X \xrightarrow{0.75} \varepsilon$. Note that the probability of the last transition is $2 \cdot 0.25 \cdot 0.75 = 0.375$ because both copies of X select their rules independently.

Intuitively, the only difference between M_Δ and B_Δ is that in M_Δ, the stack symbols are processed from left to right, while in B_Δ, all stack symbols are processed simultaneously. However, if the goal is to empty the stack, it does not really matter in what order we process the symbols because all of them have to disappear anyway. Formally, one can prove that for every $\alpha \in \Gamma^*$, the probability of reaching ε from α in M_Δ is the *same* as the probability of reaching the empty family from the family of symbols listed in α in B_Δ. In particular, we have that every symbol $X \in \Gamma$ terminates with probability 1 in M_Δ iff the family $(X_1, \ldots, X_n)$ reaches the empty family with probability 1 in B_Δ. Now consider an $n \times n$ matrix C where $C(i, j)$ is the expected number of X_j's obtained by performing a rule of X_i. For example, for the pBPA Θ we thus obtain the matrix

$$\begin{pmatrix} 0 & \frac{1}{4} & \frac{1}{4} \\ 1 & 0 & \frac{1}{2} \\ \frac{1}{2} & \frac{1}{4} & \frac{1}{4} \end{pmatrix}.$$

Here the symbols X, Y, Z are formally treated as X_1, X_2, X_3, respectively. Note that, e.g., $C(3, 1) = \frac{1}{4} \cdot 1 + \frac{1}{4} \cdot 1 + \frac{1}{2} \cdot 0 = \frac{1}{2}$. The three summands correspond to the three outgoing rules of Z, where for each rule we count the number of X's on its right-hand side.

It follows immediately that the i-th component of the vector $(1, \ldots, 1) \cdot C$ is the expected number of occurrences of X_i in a one-step successor of $(X_1, \ldots, X_n)$ in B_Δ. In general, one can easily verify that the i-th component of $(1, \ldots, 1) \cdot C^k$ is the expected number of occurrences of X_i in a k-step successor of $(X_1, \ldots, X_n)$. Intuitively, it is not surprising that $(X_1, \ldots, X_n)$ reaches the empty family with probability one iff the expected size of the family reached in k steps stays bounded as k increases. Indeed, denoting by S_k the size of the family in the k-th step, the Markov inequality implies that $\liminf_{k \to \infty} S_k$ is almost surely

finite. However, this means that almost every run must visit a particular family infinitely many times. As every symbol terminates with positive probability, almost all runs terminate.

Checking whether the expected size of the family stays bounded translates to checking whether the sum of all elements in C^k is bounded for all $k \geq 1$. Note that due to Perron-Frobenius theorem, the matrix C possesses a positive real dominant eigenvalue, say r (i.e., all other eigenvalues λ satisfy $|\lambda| < r$). The elements of C^k are bounded iff $r \leq 1$, and the latter condition can be checked in polynomial time [32].

For the pBPA Θ, the largest eigenvalue of C is strictly less than 0.9, hence (X, Y, Z) reaches the empty family with probability one, and hence all of the symbols X, Y, Z terminate with probability one in M_Θ.

If a given pBPA Δ does *not* satisfy the assumptions (1) and (2), we first determine and eliminate all stack symbols that cannot reach ε (this is easily achievable in polynomial time). Then, the variables of $Term_\Delta$ are ordered according to the associated dependency relation, and the resulting dependency graph is split into strongly connected components that are processed in the bottom-up direction, using the above described method as a procedure for resolving the most complicated subcase in the underlying case analysis. Thus, every stack symbol is eventually classified as terminating with probability 0, 1, or strictly in between. We refer to [32] for details.

The result about termination has been extended to general qualitative reachability in [13], even for a more general model of Markov decision processes generated by BPA. Thus, we obtain the following:

Theorem 3 Let $\Delta = (\Gamma, \hookrightarrow, Prob)$ be a pBPA, $\alpha \in \Gamma^*$ a configuration of Δ, and $\mathcal{C}$ a regular set of configurations of Δ represented by a DFA $\mathcal{A}$. The problem whether $\mathscr{P}(Reach(\alpha, \mathcal{C})) = 1$ is in **P**.

Results for pOC   Currently known results about the *quantitative* reachability/termination for pOC are essentially the same as for pPDA. However, the *qualitative* termination, i.e., the question whether $\mathscr{P}(Reach(p(k), T)) = 1$, where T is a subset of configurations with zero counter, is solvable in polynomial time. The underlying method is based on analyzing the *trend*, i.e., the long-run average change in counter value per transition. This is an important proof concept which turned out to be useful also in the more general setting of MDPs and stochastic games over one-counter automata (see, e.g., [9–11]). Therefore, we explain the main idea in greater detail.

Let $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$ be a zero-trivial pOC. We may safely assume that for all $p, q \in Q$ there is at most one rule $(p, c, q) \in \delta^{>0}$. The behavior of Δ for positive counter values is then fully captured by the associated *finite-state* Markov chain F_Δ, where the set of vertices is Q and each transition is assigned, in addition to its probability, a *weight*, which encodes the corresponding change in the counter value. More precisely, $p \xrightarrow{x,c} q$ is a transition of F_Δ with probability x and weight c iff $(p, c, q) \in \delta^{>0}$ and $P^{>0}(p, c, q) = x$. An example of F_Δ is shown in Fig. 2 (note that the underlying pOC Δ has ten control states).

Now consider an initial configuration $p(k)$ of Δ, and let T be the set of *all* configurations with zero counter (the selective case, when $T \subseteq Q \times \{0\}$, is discussed later). Our aim is to decide whether $\mathscr{P}(Reach(p(k), T)) = 1$. Obviously, almost every run $w \in Run(p(k))$ which does *not* visit T must visit a bottom strongly connected component (BSCC) C of F_Δ. For each such C we can easily compute the *trend* t_C which corresponds to the long-run average change in the counter value per transition (in other words, t_C is the mean payoff
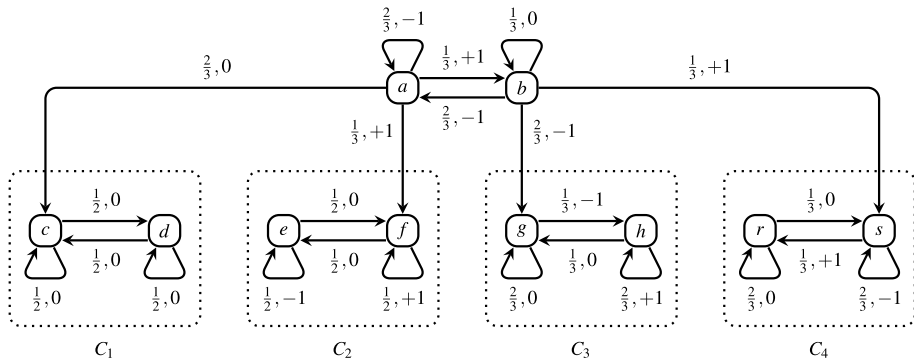
Fig. 2 A chain F_Δ and its bottom strongly connected components

determined by transition weights). More precisely, for every $q \in C$ we first compute

$$change_q = \sum_{q \xrightarrow{x,c} q'} x \cdot c$$

which is the expected change of counter value caused by an outgoing transition of q. Then, we take the weighted sum of all $change_q$ according to the unique *invariant distribution* π_C for C (intuitively, $\pi_C(q)$ gives the "frequency" of visits to q along a run initiated in (some) state of C; see, e.g., [21]). Hence,

$$t_C = \sum_{q \in C} \pi_C(q) \cdot change_q.$$

For the BSCCs C_1, C_2, C_3, and C_4 of Fig. 2 we obtain the trends 0, 0, $\frac{1}{6}$, and $-\frac{1}{6}$, respectively (note that the invariant distribution is uniform for each of these BSCCs). Now we distinguish three possibilities:

– If $t_C < 0$, then for every configuration $q(\ell)$ where $q \in C$ we have that $q(\ell)$ terminates with probability 1. Intuitively, this is because the counter tends to decrease on average, and hence it is eventually decreased to zero with probability 1.

– If $t_C > 0$, one might be tempted to conclude that $\mathscr{P}(Reach(q(\ell), T)) < 1$ for every $q \in C$ and $\ell \geq 1$. The intuition is basically correct, but for some small ℓ, the configuration $q(\ell)$ may still terminate with probability one, because the initial transitions of $q(\ell)$ may only decrease the counter even if the overall trend is positive. For example, consider the configuration $g(1)$ in the underlying pOC of Fig. 2. Although $g \in C_3$ and the trend of C_3 is positive, $g(1)$ terminates with probability one. In general, one can show that if $q(\ell)$ can reach a configuration with an *arbitrarily high* counter value without a prior visit to a configuration with zero counter, then $q(\ell)$ terminates with probability strictly less than one; otherwise, $q(\ell)$ terminates with probability one. This condition can be checked in polynomial time by the standard reachability analysis (see, e.g., [24]), and for every $q \in C$ we can easily compute a bound $k \in \mathbb{N}$ such that $\mathscr{P}(Reach(q(\ell), T)) < 1$ iff $\ell \geq k$. For example, for the state g of Fig. 2 we have that $k = 2$.

– If $t_C = 0$, then for every $q(\ell)$ where $q \in C$ we have that $q(\ell)$ terminates with probability one iff $q(\ell)$ can reach a configuration with zero counter. Again, this condition is easy to check in polynomial time, and for each q we can easily compute a bound $k \in \mathbb{Z}_\infty^{\geq 0}$ such

that $\mathscr{P}(Reach(q(\ell), T)) = 1$ iff $\ell < k$. For example, for the states c and e of Fig. 2 we have that the k is equal to 1 and ∞, respectively. Intuitively, the above condition captures the difference between two possible "types" of BSCCs with zero trend, which can be informally described as follows:

– In Type I case, the counter can be changed by an unbounded amount along a run in C (a concrete example is the component C_2 of Fig. 2). Then, given $q \in C$, the expected accumulated counter change between two visits of q in F_Δ is zero. At the same time, the accumulated change is negative with some positive probability. Thus, by standard results of theory of random walks (see, e.g., [21]), for every run w of F_Δ we have that the counter change accumulated along w fluctuates among arbitrarily large positive and negative values. However, then the corresponding run of M_Δ initiated in a configuration $q(\ell)$ eventually terminates.
– In Type II case, the counter change along every run in C is bounded. A concrete example is the component C_1 of Fig. 2, where the counter is not changed at all. Then, a run initiated in $q(\ell)$ terminates either with probability one or zero, depending on whether ℓ is small enough or not, respectively.

Using the above observations, we can determine if a configuration $q(\ell)$, where q belongs to some BSCC of F_Δ, terminates with probability one or not. If q does *not* belong to a BSCC of F_Δ, we simply check whether $q(\ell)$ can reach a configuration $q'(\ell')$, such that q' belongs to some BSCC and $q'(\ell')$ terminates with probability less than one. If so, then $q(\ell)$ terminates with probability less than one, otherwise it terminates with probability one.

For the *selective* termination, when $T \subseteq Q \times \{0\}$, we have that $\mathscr{P}(Reach(q(\ell), T)) = 1$ iff $\mathscr{P}(Reach(q(\ell), Q \times \{0\})) = 1$ and $q(\ell)$ *cannot* reach a configuration $r(0) \notin T$. Thus, we obtain the following:

Theorem 4 *Let $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$ be a zero-trivial pOC, $q(\ell)$ a configuration of Δ, and $T \subseteq Q \times \{0\}$ a set of target configurations. The problem whether $\mathscr{P}(Reach(q(\ell), T)) = 1$ is in P, assuming that ℓ is encoded in unary.*

3.2 Approximation results for reachability and termination

Now we consider the problem of approximating $\mathscr{P}(Reach(s, T))$ up to the given absolute/relative error $\delta > 0$.

Note that the results of Theorem 1 can be used to compute $\mathscr{P}(Reach(p\alpha, C))$ up to the given *absolute* error $\delta > 0$ in polynomial space by a simple binary search. However, since this algorithm uses a decision procedure for the existential fragment of Tarski algebra, it is not really practical. Observe that we can view the system $Reach_\Delta$ introduced in Sect. 3.1 more abstractly as a system of polynomial equations of the form $y_i = Pol_i(y_1, \ldots, y_n)$, where $n \in \mathbb{N}$, $1 \leq i \leq n$, and $Pol(y_1, \ldots, y_n)$ is a multivariate polynomial in the variables $y_1, \ldots, y_n$ with positive coefficients. Such systems are always monotone, and hence they have the least non-negative solution in $(\mathbb{R}_\infty^n, \sqsubseteq)$ by Knaster-Tarski theorem [41]. Also observe that if Δ is a pBPA, then $Reach_\Delta$ is always *probabilistic* in the sense that the sum of coefficients in every $Pol_i(y_1, \ldots, y_n)$ is bounded by 1, which does *not* hold for general pPDA.

Let us consider some (unspecified) system $\mathbf{y} = \mathbf{P}(\mathbf{y})$ of polynomial equations with positive coefficients. A naive approach to approximating the least non-negative solution of $\mathbf{y} = \mathbf{P}(\mathbf{y})$ is value iteration. We start with the vector of zeros $\mathbf{0}$ and successively compute $\mathbf{P}(\mathbf{0})$, $\mathbf{P}(\mathbf{P}(\mathbf{0}))$, $\mathbf{P}(\mathbf{P}(\mathbf{P}(\mathbf{0})))$, etc. This sequence of vectors is guaranteed to converge to the

least solution of the system, but the speed of this convergence can be very slow. In general, exponentially many iterations may be needed to produce another bit of precision. However, one can also apply more efficient methods. In [37], it has been shown that (a decomposed variant of) Newton's method, when applied to $\mathbf{y} = \mathbf{P}(\mathbf{y})$, converges *linearly* in the sense that after some initial number of iterations, it produces one bit of precision per iteration. In general, no bound is given for the initial number of iterations. A special variant of Newton's method applicable to *probabilistic* systems $\mathbf{y} = \mathbf{P}(\mathbf{y})$ has been designed and investigated in [23]. Although the method does not improve the worst-case upper bounds, it seems to be more robust and delivers better performance in practical examples. A recent work [27] shows that for probabilistic systems, the initial phase actually requires only *linearly* many iterations when all variables with value 0 and 1 are eliminated in a preprocessing phase (which is achievable in polynomial time). This already gives a polynomial-time approximation algorithm on the unit-cost rational arithmetic RAM. However, in [27] it is also shown that one can actually obtain a polynomial-time approximation algorithm on the standard Turing machine model by rounding down the intermediate results carefully. Interestingly, in the special case of $Term_\Delta$, when Δ is a pOC, Newton's method requires only *polynomially* many iterations in the initial phase after eliminating all variables which are equal to 0 (which is again achievable in polynomial time) [28].

To sum up, Newton's method can be used to approximate $\mathscr{P}(Reach(p\alpha, \mathcal{C}))$ for general pPDAs, but it does not allow for improving the **PSPACE** worst-case complexity bound obtained by employing the decision procedure for Tarski algebra. Still, there are at least two tractable subcases of pBPA and pOC.

Theorem 5 *Let $\Delta = (\Gamma, \hookrightarrow, Prob)$ be a pBPA, $\alpha \in \Gamma^*$ a configuration of Δ, $\mathcal{C}$ a regular set of configurations of Δ represented by a DFA $\mathcal{A}$, and $\delta > 0$ a rational constant represented as a fraction of binary numbers. Then there is $r \in \mathbb{Q}$ computable in polynomial time such that $|\mathscr{P}(Reach(\alpha, \mathcal{C})) - r| \leq \delta$.*

For pOC, it is not known whether the termination probabilities can be approximated in polynomial time on the standard Turing machine model. So, we can only state a somewhat weaker result.

Theorem 6 *Let $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$ be a pOC, $p(k)$ a configuration of Δ (where k is encoded in unary), $q \in Q$, and $\delta > 0$ a rational constant. Then there is $r \in \mathbb{Q}$ computable in polynomial time on the unit-cost rational arithmetic RAM such that $|\mathscr{P}(Reach(p(k), \{q(0)\})) - r| \leq \delta$.*

Approximating $\mathscr{P}(Reach(p\alpha, \mathcal{C}))$ up to a given *relative* error $\delta > 0$ is more problematic. It requires exponential time even for pBPA and termination, because the probability $\mathscr{P}(Reach(X, \{\varepsilon\}))$ can be *doubly-exponentially* small in the size of the underlying pBPA Δ. To see this, realize that pBPA can simulate repeated squaring; let $\Delta = (\{X_1, \ldots, X_{n+1}, Z\}, \hookrightarrow, Prob)$ where, for all $1 \leq i \leq n$,

$$X_i \overset{1}{\hookrightarrow} X_{i+1}X_{i+1}, \quad X_{n+1} \overset{0.5}{\hookrightarrow} Z, \quad X_{n+1} \overset{0.5}{\hookrightarrow} \varepsilon, \quad Z \overset{1}{\hookrightarrow} Z.$$

Then $\mathscr{P}(Reach(X_1, \{\varepsilon\})) = 1/2^{2^n}$. This argument does not work for pOC, where a positive probability of the form $\mathscr{P}(Reach(p(k), \{q(0)\}))$ can be only singly exponentially small in the size of the underlying pOC and the initial counter value k. Hence, it follows directly from Theorem 6 that $\mathscr{P}(Reach(p(k), \{q(0)\}))$ can be approximated up to the *relative* error $\delta > 0$ in polynomial time on the unit-cost rational arithmetic RAM.

4 Translating pPDA into pBPA

In this section we present the construction of [17] which transforms every pPDA into an equivalent pBPA where all stack symbols *terminate either with probability* 0 *or* 1. This transformation preserves virtually all interesting quantitative properties of the original pPDA (except, of course, termination probabilities) and it is in some sense effective. Thus, the study of general pPDA can be reduced to the study of a special type of pBPA, and the reduction step does not lead to any substantial increase in complexity (at least, for the problems considered in this survey).

For the rest of this section, we fix a pPDA $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$. For all $p, q \in Q$ and $X \in \Gamma$, we use

– $Run(pXq)$ to denote the set of all runs in M_Δ initiated in pX that visit $q\varepsilon$;
– $Run(pX\uparrow)$ to denote the set of all runs in M_Δ initiated in pX that do *not* visit a configuration with empty stack.

The probability of $Run(pXq)$ and $Run(pX\uparrow)$ is denoted by $[pXq]$ and $[pX\uparrow]$, respectively.

The idea behind the transformation of Δ into an equivalent pBPA $\Delta_\bullet$ is relatively simple and closely resembles the standard method for transforming a PDA into an equivalent context-free grammar (see, e.g., [36]). Formally, the stack alphabet $\Gamma_\bullet$ of $\Delta_\bullet$ is defined as follows:

– For all $p \in Q$ and $X \in \Gamma$ such that $[pX\uparrow] > 0$ we add a stack symbol $\langle pX\uparrow\rangle$ to $\Gamma_\bullet$.
– For all $p, q \in Q$ and $X \in \Gamma$ such that $[pXq] > 0$ we add a stack symbol $\langle pXq\rangle$ to $\Gamma_\bullet$.

Note that $\Gamma_\bullet$ is effectively constructible in polynomial space by applying the results of Sect. 3. Now we construct the rules $\hookrightarrow_\bullet$ of $\Delta_\bullet$ together with their probabilities. For all $\langle pXq\rangle \in \Gamma_\bullet$ we do the following:

– if $pX \overset{x}{\hookrightarrow} rYZ$, then for all $s \in Q$ such that $y = x \cdot [rYs] \cdot [sZq] > 0$ we put $\langle pXq\rangle \xrightarrow{y/[pXq]}_\bullet \langle rYs\rangle\langle sZq\rangle$;
– if $pX \overset{x}{\hookrightarrow} rY$ where $y = x \cdot [rYq] > 0$, we put $\langle pXq\rangle \xrightarrow{y/[pXq]}_\bullet \langle rYq\rangle$;
– if $pX \overset{x}{\hookrightarrow} q\varepsilon$, we put $\langle pXq\rangle \xrightarrow{x/[pXq]}_\bullet \varepsilon$.

For all $\langle pX\uparrow\rangle \in \Gamma_\bullet$ we do the following:

– if $pX \overset{x}{\hookrightarrow} rYZ$, then for every $s \in Q$ such that $y = x \cdot [rYs] \cdot [sZ\uparrow] > 0$ we put $\langle pX\uparrow\rangle \xrightarrow{y/[pX\uparrow]}_\bullet \langle rYs\rangle\langle sZ\uparrow\rangle$;
– for all $q \in Q$ and $Y \in \Gamma$ such that $y = [qY\uparrow] \cdot \sum_{pX \overset{z}{\hookrightarrow} qY\beta} z > 0$ we put $\langle pX\uparrow\rangle \xrightarrow{y/[pX\uparrow]}_\bullet$ $\langle qY\uparrow\rangle$.

Note that the transition probabilities of $\Delta_\bullet$ may take irrational values, but are effectively expressible in the existential fragment of Tarski algebra (see Theorem 2). Obviously, all symbols of the form $\langle pX\uparrow\rangle$ terminate with probability 0, and we show that all symbols of the form $\langle pXq\rangle$ terminate with probability 1 (see Theorem 7).

Remark 2 The translation from Δ to $\Delta_\bullet$ makes also good sense when Δ is a pBPA. Since qualitative termination for pBPA is decidable in polynomial time (see Theorem 3), one can also efficiently compute the set of all stack symbols Y of Δ such that $[Y\uparrow] > 0$, and hence the set of rules of $\Delta_\bullet$ is constructible in polynomial time (the rule probabilities may still take irrational values). Consequently, some interesting *qualitative* properties of pBPA are decidable in polynomial time, because they do not depend on the exact values of rule probabilities in $\Delta_\bullet$ (see Sect. 7.1).

Example 3 Consider a pPDA Δ with two control states p, q, one stack symbol X, and the following transition rules, where $a > 1/2$:

$$pX \xrightarrow{a} qXX, \quad pX \xrightarrow{1-a} q\varepsilon, \quad qX \xrightarrow{a} pXX, \quad qX \xrightarrow{1-a} p\varepsilon.$$

Clearly, $[pXp] = [qXq] = 0$. Using the results of Sect. 3, one can easily verify that $[pXq] = [qXp] = (1 - a)/a$. Hence, $[pX\uparrow] = [qX\uparrow] = (2a - 1)/a$. Consequently, the stack symbols of $\Delta_\bullet$ are $\langle pXq \rangle$, $\langle qXp \rangle$, $\langle pX\uparrow \rangle$, and $\langle qX\uparrow \rangle$, and the transition rules of $\Delta_\bullet$ are the following:

$$\langle pXq \rangle \xrightarrow{1-a}_\bullet \langle qXp \rangle \langle pXq \rangle, \qquad \langle qXp \rangle \xrightarrow{1-a}_\bullet \langle pXq \rangle \langle qXp \rangle,$$
$$\langle pXq \rangle \xrightarrow{a}_\bullet \varepsilon, \qquad \langle qXp \rangle \xrightarrow{a}_\bullet \varepsilon,$$
$$\langle pX\uparrow \rangle \xrightarrow{1-a}_\bullet \langle qXp \rangle \langle pX\uparrow \rangle, \qquad \langle qX\uparrow \rangle \xrightarrow{1-a}_\bullet \langle pXq \rangle \langle qX\uparrow \rangle,$$
$$\langle pX\uparrow \rangle \xrightarrow{a}_\bullet \langle qX\uparrow \rangle, \qquad \langle qX\uparrow \rangle \xrightarrow{a}_\bullet \langle pX\uparrow \rangle.$$

As $a > 1/2$, the resulting pBPA has a tendency to decrease the stack height. Hence, both $\langle pXq \rangle$ and $\langle qXp \rangle$ terminate with probability 1.

Every run of M_Δ initiated in pX that reaches $q\varepsilon$ can be "mimicked" by the associated run of $M_{\Delta_\bullet}$ initiated in $\langle pXq \rangle$. Similarly, *almost every*[4] run of M_Δ initiated in pX that does *not* visit a configuration with empty stack corresponds to some run of $M_{\Delta_\bullet}$ initiated in $\langle pX\uparrow \rangle$.

Example 4 Let Δ be a pPDA with two control states p, q, one stack symbol X, and the following transition rules:

$$pX \xrightarrow{0.5} pXX, \quad pX \xrightarrow{0.5} q\varepsilon, \quad qX \xrightarrow{1} q\varepsilon.$$

Then $[pXq] = 1$ and $[qXq] = 1$, which means that $\Delta_\bullet$ has just two stack symbols $\langle pXq \rangle$ and $\langle qXq \rangle$ and the rules

$$\langle pXq \rangle \xrightarrow{0.5}_\bullet \langle pXq \rangle \langle qXq \rangle, \quad \langle pXq \rangle \xrightarrow{0.5}_\bullet \varepsilon, \quad \langle qXq \rangle \xrightarrow{1}_\bullet \varepsilon.$$

The infinite run $pX, pXX, pXXX, \ldots$ cannot be mimicked in $M_{\Delta_\bullet}$, but since the total probability of all infinite runs initiated in pX is zero, almost all (but not *all*) of them can be mimicked in $M_{\Delta_\bullet}$.

The correspondence between the runs of M_Δ and $M_{\Delta_\bullet}$ is formally captured by a finite family of functions $(\cdot)_\odot$ where $\odot \in Q \cup \{\uparrow\}$. For every run $w \in Run(pX)$ in M_Δ, the function $(\cdot)_\odot$ returns an infinite sequence $w_\odot$ such that $w_\odot(i) \in \Gamma_\bullet^* \cup \{\times\}$ for every $i \in \mathbb{Z}^{\geq 0}$. The sequence $w_\odot$ is either a run of $M_{\Delta_\bullet}$ initiated in $\langle pX\odot \rangle$, or an *invalid sequence*. As we shall see, all invalid sequences have an infinite suffix of "$\times$" symbols and correspond to those runs of $Run(pX)$ that cannot be mimicked by a run of $Run(\langle pX\odot \rangle)$.

So, let $\odot \in Q \cup \{\uparrow\}$, and let w be a run of M_Δ initiated in pX. We define an infinite sequence $w_\odot$ over $\Gamma_\bullet^* \cup \{\times\}$ inductively as follows:

---

[4]Here "almost every" is meant in the usual probabilistic sense, i.e., the probability of the remaining runs is zero.

- $w_\odot(0)$ is either $\langle pX\odot\rangle$ or $\times$, depending on whether $\langle pX\odot\rangle \in \Gamma_\bullet$ or not, respectively.
- If $w_\odot(i) = \times$ or $w_\odot(i) = \varepsilon$, then $w_\odot(i+1) = w_\odot(i)$. Otherwise, we have that $w_\odot(i) = \langle pX\dagger\rangle\alpha$, where $\dagger \in Q \cup \{\uparrow\}$, and $w(i) = pX\gamma$ for some $\gamma \in \Gamma^*$. Let $pX \hookrightarrow r\beta$ be the rule of Δ used to derive the transition $w(i) \to w(i+1)$. We put

$$w_\odot(i+1) = \begin{cases} \alpha & \text{if } \beta = \varepsilon \text{ and } \dagger = r; \\ \langle rY\dagger\rangle\alpha & \text{if } \beta = Y \text{ and } [rY\dagger] > 0; \\ \langle rYs\rangle\langle sZ\dagger\rangle\alpha & \text{if } \beta = YZ, [sZ\dagger] > 0, \text{ and there is } k > i \text{ such that} \\ & \quad w(k) = sZ\gamma \text{ and } |w(j)| > |w(i)| \text{ for all } i < j < k; \\ \langle rY\uparrow\rangle\alpha & \text{if } \beta = YZ, \dagger = \uparrow, [rY\uparrow] > 0, \text{ and } |w(j)| > |w(i)| \\ & \quad \text{for all } j > i; \\ \times & \text{otherwise.} \end{cases}$$

We say that $w \in Run(pX)$ is *invalid* if $w_\odot(i) = \times$ for some $i \in \mathbb{Z}^{\geq 0}$. Otherwise, w is *valid*. It is easy to check that if w is valid, then $w_\odot \in Run(\langle pX\odot\rangle)$. Hence, $(\cdot)_\odot$ can be seen as a partial function from $Run(pX)$ to $Run(\langle pX\odot\rangle)$ which is defined only for valid runs. Further, for every valid $w \in Run(pX)$ and every $i \in \mathbb{Z}^{\geq 0}$ we have that

- $w(i) = rY\beta$ iff $w_\odot(i) = \langle rY\dagger\rangle\gamma$ for some $\dagger \in Q \cup \{\uparrow\}$ and $\gamma \in \Gamma_\bullet^*$,
- $w(i) = r\varepsilon$ iff $w_\odot(i) = \varepsilon$ and $\odot = r$.

Hence, $(\cdot)_\odot$ preserves all properties of runs that depend just on the heads of visited configurations. Further, $(\cdot)_\odot$ preserves the probability of measurable subsets of $Run(pX)$ with respect to the associated probability measure $\mathscr{P}_\odot$. More precisely, we define the probability space $(Run(pX), \mathscr{F}, \mathscr{P}_\odot)$, where $\mathscr{F}$ is the standard Borel σ-field generated by all basic cylinders (see Sect. 1) and $\mathscr{P}_\odot$ is the unique probability function such that for every $w \in FPath(pX)$ we have that

$$\mathscr{P}_\odot = \frac{\mathscr{P}(Run(w) \cap Run(pX\odot))}{[pX\odot]}$$

where $\mathscr{P}$ is the standard probability function. Now we can state the main theorem, which says that $(\cdot)_\odot$ is a probability preserving measurable function.

Theorem 7 (see [17]) *Let* $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ *be a pPDA,* $p \in Q$, $X \in \Gamma$, *and* $\odot \in \Gamma \cup \{\uparrow\}$ *such that* $[pX\odot] > 0$. *Then for every measurable subset* $R \subseteq Run(\langle pX\odot\rangle)$ *we have that* $(\cdot)_\odot^{-1}(R) \subseteq Run(pX)$ *is measurable and* $\mathscr{P}(R) = \mathscr{P}_\odot((\cdot)_\odot^{-1}(R))$.

In particular, Theorem 7 implies that all symbols of the form $\langle pXq\rangle$ which belong to $\Gamma_\bullet$ terminate with probability one, because

$$\mathscr{P}\big(Reach(\langle pXq\rangle, \{\varepsilon\})\big) = \mathscr{P}_q\big((\cdot)_q^{-1}\big(Reach(\langle pXq\rangle, \{\varepsilon\})\big)\big) = \mathscr{P}_q\big(Run(pXq)\big) = 1.$$

5 Termination time and total accumulated reward

Now we show how to compute the (conditional) expected total reward accumulated along a run initiated in a given configuration before visiting a target configuration from a given regular set. We also formulate generic tail bounds on the termination time in pPDA. This section is based mainly on the results presented in [15, 17, 26].

5.1 Computing and approximating the expected total accumulated reward

Results for pPDA and pBPA Let us fix a pPDA $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ and a *simple* reward function $f : Q \times \Gamma^* \to \mathbb{R}^{\geq 0}$ (recall that a reward function is simple if $f(pX\alpha) = f(pX\beta)$ for all $pX \in Q \times \Gamma$ and all $\alpha, \beta \in \Gamma^*$). As in the previous sections, for all $p, q \in Q$ and $X \in \Gamma$ we use $Run(pXq)$ to denote the set $Reach(pX, \{q\varepsilon\})$, and $[pXq]$ to denote the probability of $Run(pXq)$. If $[pXq] > 0$, then we also use E_{pXq} to denote the conditional expectation

$$\mathbb{E}\big(Acc \mid Run(pXq)\big)$$

where Acc is the random variable introduced in Sect. 2.4 and the set of target configurations contains just $q\varepsilon$. That is, E_{pXq} is the conditional expected total reward accumulated along a run initiated in pX before visiting $q\varepsilon$ under the condition that $q\varepsilon$ is eventually visited, where f is the underlying reward function.

We show that the tuple of all E_{pXq}, where $[pXq] > 0$, is the least solution of an effectively constructible system of *linear* equations $Expect_\Delta$, where the (fractions of) termination probabilities are used as coefficients. Since E_{pXq} can also be infinite, we need to consider the least solution of $Expect_\Delta$ in $((\mathbb{R}_\infty^{\geq 0})^k, \sqsubseteq)$, where k is the number of all triples (p, X, q) such that $[pXq] > 0$, and $\sqsubseteq$ is the component-wise ordering.

The system $Expect_\Delta$ is obtained as follows. First, we compute the set of all triples (p, X, q) such that $[pXq] > 0$ (this can be done in polynomial time). For each such (p, X, q) we fix a fresh variable $\langle E_{pXq} \rangle$ and construct the equation given below, where all summands with zero coefficients are immediately removed.

$$\langle E_{pXq} \rangle = f(pX) + \sum_{pX \overset{x}{\hookrightarrow} rY} \frac{x \cdot [rYq]}{[pXq]} \cdot \langle E_{rYq} \rangle$$

$$+ \sum_{pX \overset{x}{\hookrightarrow} rYZ} \sum_{t \in Q} \frac{x \cdot [rYt] \cdot [tZq]}{[pXq]} \cdot \big(\langle E_{rYt} \rangle + \langle E_{tZq} \rangle\big). \tag{1}$$

Example 5 Let us consider a pBPA Δ with just one stack symbol I and the rules

$$I \overset{x}{\hookrightarrow} II, \quad I \overset{1-x}{\hookrightarrow} \varepsilon.$$

Further, let $f(I) = 1$. The system $Expect_\Delta$ then contains just the following equation:

$$\langle E_I \rangle = 1 + x \cdot [I] \cdot \big(\langle E_I \rangle + \langle E_I \rangle\big).$$

If $x = \frac{2}{3}$, then $[I] = \frac{1}{2}$ and hence $E_I = 3$. If $x = \frac{1}{2}$, then $[I] = 1$ and the only solution to the above equation is ∞.

In general, $Expect_\Delta$ may have several solutions in $((\mathbb{R}_\infty^{\geq 0})^k, \sqsubseteq)$. However, if we identify and remove all variables which are equal to 0 or ∞ in the least solution, then the resulting system $Expect_\Delta'$ has exactly *one* solution in $((\mathbb{R}^{\geq 0})^k, \sqsubseteq)$. To see this, let us assume that $Expect_\Delta'$ has another solution ν apart from the least solution μ (note that $\nu - \mu \geq \mathbf{0}$). Since we eliminated all zero variables, μ is strictly positive in all components and hence there is $c > 0$ such that $(c, \ldots, c) \sqsubseteq \mu$. Let d be the maximal component of $\nu - \mu$. Then $d > 0$, and $\mu - \frac{c}{d}(\nu - \mu)$ is also a non-negative solution of $Expect_\Delta'$ which is strictly smaller than μ, and we have a contradiction.

Note that one can easily identify all $\langle E_{pXq} \rangle$ such that $E_{pXq} = 0$. This is because $E_{pXq} = 0$ iff every $w \in Run(pXq)$ visits only configurations with zero reward (except for the last configuration $q\varepsilon$), and this is a simple reachability question which can be solved in polynomial time by standards methods [24]. However, identifying the variables $\langle E_{pXq} \rangle$ such that $E_{pXq} = \infty$ is not so trivial because the coefficients of $Expect_\Delta$ are given only symbolically and their actual values are not at our disposal. Still, one can easily express the question whether $E_{pXq} = \infty$ in the existential fragment of Tarski algebra, and hence the system $Expect'_\Delta$ is constructible in polynomial space. This implies that the tuple of all E_{pXq} is effectively expressible in the existential fragment of Tarski algebra. The corresponding formula (cf. Definition 3) can be constructed in polynomial space and its size is polynomial in the size of Δ.

Remark 3 It is worth mentioning that in the special case when Δ is a pBPA such that *all* stack symbols terminate with probability one, the system $Expect'_\Delta$ and its only solution are computable in polynomial time. Here, the only problem is to identify all stack symbols X such that $E_X = \infty$, which can be done by constructing the dependency graph among the stack symbols (we say that X *depends* on Y if X can reach a configuration of the form $Y\alpha$), identifying strongly connected components (SCCs) in this graph, and processing them in the bottom-up direction. Note that if $E_Y = \infty$, then $E_X = \infty$ for all X that depend on Y. For a bottom SCC C, we simply consider a pBPA Δ_C obtained from Δ by restricting the stack alphabet to C, and check whether $Expect_{\Delta_C}$ has a solution in non-negative reals. If so, $E_X < \infty$ for all $X \in C$, otherwise $E_X = \infty$ for all $X \in C$. A similar procedure is used when processing the intermediate SCCs. So, we eventually decide whether $E_X = \infty$ for every $X \in \Gamma$.

The above observations can be immediately extended to the conditional expectation of the form

$$\mathbb{E}\big(Acc \mid Reach(pX, \mathcal{C})\big)$$

where $\mathcal{C}$ is a *simple* set of target configurations (see Definition 5). This is because we can modify a given pPDA Δ into another pPDA Δ' by

- adding a fresh control state t where $tX \xrightarrow{1} t\varepsilon$ for every stack symbol X of Δ;
- modifying the rules of Δ so that the only successor of every $qY\beta \in \mathcal{C}$ is the configuration $tY\beta$;
- extending the reward function f by stipulating $f(tX) = 0$ for every stack symbol X of Δ.

It follows immediately that $\mathbb{E}(Acc \mid Reach(pX, \mathcal{C}))$ computed for Δ is equal to $\mathbb{E}(Acc \mid Reach(pX, \{t\varepsilon\}))$ computed for Δ', and thus we can apply the results mentioned above. Further, we can generalize this observation to *regular* sets of target configurations by using the generic method of Remark 1. Thus, we obtain the following theorem:

Theorem 8 (see [26]) *Let $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ be a pPDA, $p\alpha \in Q \times \Gamma^*$ a configuration of Δ, $\mathcal{C}$ a regular set of configurations of Δ represented by a DFA $\mathcal{A}$ such that $\mathscr{P}(Reach(p\alpha, \mathcal{C})) > 0$, and f a simple reward function. Further, let $\varrho > 0$ and $\delta > 0$ be rational constants. Then the problems whether $\mathbb{E}(Acc \mid Reach(p\alpha, \mathcal{C})) < \infty$ and $\mathbb{E}(Acc \mid Reach(p\alpha, \mathcal{C})) \leq \varrho$ are in **PSPACE**. Further, there is $r \in \mathbb{Q}$ computable in polynomial space such that $|\mathbb{E}(Acc \mid Reach(p\alpha, \mathcal{C})) - r| \leq \delta$.*

In the special case when Δ is a pBPA where all stack symbols terminate with probability one, the conditional expectation $\mathbb{E}(Acc \mid Reach(\alpha, \{\varepsilon\}))$ is computable in polynomial time.

Theorem 8 can be easily extended to a more general class of *linear* reward functions. Recall that a reward function f is linear if there are functions $c, d : Q \to \mathbb{R}^{\geq 0}$ and $h : \Gamma \to \mathbb{R}^{\geq 0}$ such that for all $p\alpha \in Q \times \Gamma^*$ we have that

$$f(p\alpha) = c(p) \cdot \left(d(p) + \sum_{X \in \Gamma} \#_X(\alpha) \cdot h(X) \right)$$

where $\#_X(\alpha)$ denotes the number of occurrences of X in α. Again, we start by considering the conditional expectation

$$\mathbb{E}\big(Acc \mid Run(pXq)\big).$$

Intuitively, the main difference from the case when f was simple is that after executing a rule of the form $pX \xrightarrow{x} qYZ$, the symbol Z contributes to the total accumulated reward even if it is hidden in the middle of the stack. Using the results about simple reward functions, we can express the expected number of visits to a configuration with a given control state r along a path from pX to $q\varepsilon$. Thus, we can also express the "expected contribution" of Z to the total reward accumulated along such a path. These considerations lead to a system of equations similar to $Expect_\Delta$ (we refer to [26] for details). Hence, Theorem 8 holds also for linear reward functions without any change.

Let us note that Theorem 8 can be generalized even further; it holds for an arbitrary (fixed) conditional moment

$$\mathbb{E}\big(Acc^i \mid Reach(p\alpha, \mathcal{C})\big).$$

In particular, one can approximate the conditional variance of Acc up to an arbitrarily small absolute error $\varepsilon > 0$ in polynomial space [26].

Results for pOC   In this paragraph we present the results of [15] about the conditional expected termination time in zero-trivial pOC.

Let us fix a zero-trivial pOC $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$. Similarly as in Sect. 3.1, we assume that for all $p, q \in Q$ there is at most one rule $(p, c, q) \in \delta^{>0}$. Since we are primarily interested in the (conditional) expected termination time, we fix a constant reward function f which returns 1 for every configuration. Consistently with the notation previously adopted for pPDA, we use $Run(p{\downarrow}q)$ to denote the set $Reach(p(1), \{q(0)\})$, and $[p{\downarrow}q]$ to denote the probability of $Run(p{\downarrow}q)$. For every $i \in \mathbb{N}$, we also use $Run(p{\downarrow}q, i)$ to denote the set of all runs initiated in $p(1)$ that visit $q(0)$ for the first time in exactly i transitions, and $[p{\downarrow}q, i]$ to denote the probability of $Run(p{\downarrow}q, i)$. If $[p{\downarrow}q] > 0$, then $E_{p{\downarrow}q}$ denotes the conditional expectation

$$\mathbb{E}\big(Acc \mid Run(p{\downarrow}q)\big),$$

where $q(0)$ is the only target configuration. Also recall the finite-state Markov chain F_Δ which captures the behavior of Δ for positive counter values, and the definitions of the *expected counter change* at $q \in Q$, denoted by $change_q$, and the *trend* of a given bottom strongly connected component (BSCC) C of F_Δ, denoted by t_C (see Sect. 3.1). For every configuration $p(k)$ of Δ, we use

– $pre^*(p(k))$ to denote the set of all configurations that can reach $p(k)$ in M_Δ;
– $post^*(p(k))$ to denote the set of all configurations reachable from $p(k)$ in M_Δ.

Our aim is to show that the problem whether $E_{p{\downarrow}q} < \infty$ is decidable in polynomial time, and that the value of $E_{p{\downarrow}q}$ (if it is finite) can be efficiently approximated. A crucial step towards these results is the following theorem:

Theorem 9 (see [15]) *Let* $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$ *be a zero-trivial pOC, and let* $p, q \in Q$ *such that* $[p{\downarrow}q] > 0$. *Further, let* x_{min} *denote the smallest* (*positive*) *probability in* F_Δ.

(A) *If* q *is not in a BSCC of* F_Δ, *then* $E_{p{\downarrow}q} \leq 5|Q| / x_{min}^{|Q|+|Q|^3}$.
(B) *Let* $q \in C$, *where* C *is a BSCC of* F_Δ. *Further, let* $Conf = pre^*(q(0)) \cap post^*(p(1)) \cap C \times \mathbb{N}$. *Then*
   (a) *if* $Conf$ *is a finite set, then* $E_{p{\downarrow}q} \leq 20|Q|^3/x_{min}^{4|Q|^3}$;
   (b) *if* $Conf$ *is an infinite set*, *then*
      (1) *if* C *has trend* $t \neq 0$, *then* $E_{p{\downarrow}q} \leq 85000|Q|^6/(x_{min}^{5|Q|+|Q|^3} \cdot t^4)$;
      (2) *if* C *has trend* $t = 0$, *then* $E_{p{\downarrow}q}$ *is infinite.*

According to Theorem 9, the value of $E_{p{\downarrow}q}$ is either infinite or at most exponential in the size of Δ. Note that this does not hold for pBPA, where the value of E_X can be doubly exponential in the size of the underlying pBPA (an example is easy to construct by simulating repeated squaring similarly as in Sect. 3.2).

It follows from the results of [24] that the sets $pre^*(q(0))$ and $post^*(p(1))$ are regular and the associated DFA are computable in polynomial time. Hence, the finiteness of $Conf$ can be decided in polynomial time, and thus we obtain the following corollary to Theorem 9:

Corollary 1 *Let* $p, q \in Q$ *such that* $[p{\downarrow}q] > 0$. *The problem whether* $E_{p{\downarrow}q} < \infty$ *is in* **P**.

The bounds given in Theorem 9 also provide the missing piece of knowledge needed for efficient approximation of the expected termination time in pOC. Recall that the tuple of all $E_{p{\downarrow}q}$, where $[p{\downarrow}q] > 0$, is the least solution of the system of linear equations $Expect_\Delta$. Due to Corollary 1, we can eliminate all variables $\langle p{\downarrow}q \rangle$ such that $E_{p{\downarrow}q} = 0$ or $E_{p{\downarrow}q} = \infty$ in polynomial time, and thus construct the system $Expect'_\Delta$. We have already shown that $Expect'_\Delta$ has only one solution. Also recall that the coefficients of $Expect'_\Delta$ are fractions of termination probabilities, which can be computed up to an arbitrarily small positive error in polynomial time, assuming the unit-cost rational arithmetic RAM model of computation (see Theorem 6). Using the bounds of Theorem 9, for each $\delta > 0$ we can give $\varepsilon > 0$ (as a function of δ) such that solving a *perturbed* system $Expect'_\Delta$, where the coefficients are just approximated up to the absolute error ε, produces a solution whose absolute error is bounded by δ. Further, the size of ε (i.e., the length of the corresponding binary encoding) is *polynomial* in the size δ. Thus, we obtain the following:

Theorem 10 (see [15]) *Let* $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$ *be a zero-trivial pOC, and let* $p, q \in Q$ *such that* $[p{\downarrow}q] > 0$ *and* $E_{p{\downarrow}q} < \infty$. *Further, let* $\delta > 0$ *be a rational constant. Then the value of* $E_{p{\downarrow}q}$ *can be approximated up to the absolute error* δ *in polynomial time, assuming the unit-cost rational arithmetic RAM model of computation.*

In the rest of this subsection we sketch the main ideas behind the proof of Theorem 9. In particular, we indicate why $E_{p{\downarrow}q}$ is infinite only in case (B.b.2). First assume case (A), i.e., q is not in a BSCC of F_Δ. Then for all $s(\ell) \in post^*(p(1))$, where $\ell \geq |Q|$, we have that $s(\ell)$ can reach a configuration outside $pre^*(q(0))$ in at most $|Q|$ transitions. It follows that the probability of performing a path from $p(1)$ to $q(0)$ of length i decays exponentially in i, and hence

$$E_{p{\downarrow}q} = \sum_{i=1}^{\infty} i \cdot \frac{[p{\downarrow}q, i]}{[p{\downarrow}q]}$$

is finite. As an example, consider the states a and b of Fig. 2. A "long" path from $a(1)$ to $b(0)$ inevitably loops between the control states a and b. Since there is always a chance to enter some BSCC of F_Δ, the probability of executing a path of length i which loops between a and b decays exponentially in i.

Next assume case (B.a), i.e., C is a BSCC and *Conf* is a finite set. It is easy to show that the expected time for a run in $Run(p{\downarrow}q)$ to reach C is finite. Once the run has reached C, it basically moves within a Markov chain on *Conf*. By assumption, *Conf* is finite (which implies, by a pumping argument, that $|Conf| \leq 3|Q|^3$). Consequently, after the run has reached C, it reaches $q(0)$ in finite expected time.

Case (B.b) requires new non-trivial techniques. For the sake of simplicity, *from now on we assume that $Q = C$* (the general case requires only slight modifications of the arguments presented below). We employ a generic observation which connects the study of pOC to *martingale theory* (recall the definitions and results of Sect. 2.1). Let us fix an initial configuration $r(c) \in Q \times \mathbb{N}$. Our aim is to construct a suitable martingale over $Run(r(c))$. Let $p^{(i)}$ and $c^{(i)}$ be random variables which to every run $w \in Run(r(c))$ assign the control state and the counter value of the configuration $w(i)$, respectively. Note that if the expected change of counter value *change$_s$* was the *same* for every $s \in C$, we would have *change$_s$* $= t$ where t is the trend of C, and we could define a martingale $m^{(0)}, m^{(1)}, \ldots$ simply by

$$m^{(i)} = \begin{cases} c^{(i)} - i \cdot t & \text{if } c^{(j)} \geq 1 \text{ for all } 0 \leq j < i; \\ m^{(i-1)} & \text{otherwise.} \end{cases}$$

Since *change$_s$* is generally not constant, we might try to compensate the difference among the individual control states by adding a constant "weight" $\boldsymbol{v}_s$ to each $s \in C$. That is, we aim at designing a martingale of the form

$$m^{(i)} = \begin{cases} c^{(i)} + \boldsymbol{v}_{p^{(i)}} - i \cdot t & \text{if } c^{(j)} \geq 1 \text{ for all } 0 \leq j < i; \\ m^{(i-1)} & \text{otherwise.} \end{cases}$$

In [15], it is shown that there indeed exists a vector of suitable $\boldsymbol{v}_s \in \mathbb{R}$ such that the above stochastic process becomes a martingale. Further, the difference between the maximal and the minimal weight assigned to some state, denoted by *diff*, is bounded by $2|C|/x_{\min}^{|C|}$. Due to this result, powerful tools of martingale theory, such as Azuma's inequality and the optional stopping theorem (see Sect. 2.1) become applicable to pOC. In particular, we can resolve both case (B.b.1) and case (B.b.2) of Theorem 10.

Let us first consider case (B.b.1) when $t \neq 0$. By applying Azuma's inequality to the above martingale $m^{(0)}, m^{(1)}, \ldots$ over $Run(p(1))$, we show that there are $0 < a < 1$ and $h \in \mathbb{N}$ such that for all $i \geq h$ we have that $[p{\downarrow}q, i] \leq a^i$. This immediately implies that $E_{p{\downarrow}q}$ is finite, and the bound given in case (B.b.1) is obtained by analyzing the size of a and h.

Realize that for every $w \in Run(p{\downarrow}q, i)$ we have that

$$\big(m^{(i)} - m^{(0)}\big)(w) = \boldsymbol{v}_q - \boldsymbol{v}_p - i \cdot t.$$

Hence, $[p{\downarrow}q, i] \leq \mathscr{P}(m^{(i)} - m^{(0)} = \boldsymbol{v}_q - \boldsymbol{v}_p - i \cdot t)$. A simple computation reveals that for a sufficiently large $h \in \mathbb{N}$ and all $i \geq h$ we have the following:

– If $t < 0$, then

$$[p{\downarrow}q, i] \leq \mathscr{P}\big(m^{(i)} - m^{(0)} \geq (i/2) \cdot (-t)\big).$$

– If $t > 0$, then

$$[p\downarrow q, i] \leq \mathscr{P}\big(m^{(i)} - m^{(0)} \leq (i/2)\cdot(-t)\big).$$

In each step, the martingale value changes by at most $\textit{diff} + t + 1$, where $\textit{diff}$ is defined above. Hence, by applying Azuma's inequality, we obtain the following for all $t \neq 0$ and $i \geq h$:

$$[p\downarrow q, i] \leq \exp\left(-\frac{(i/2)^2 t^2}{2i\,(\textit{diff} + t + 1)^2}\right) = a^i.$$

Here $a = \exp(-t^2 8(\textit{diff} + t + 1)^2) < 1$. Hence, $E_{p\downarrow q} < \infty$, and the bound given in Theorem 9, case (B.b.2), is computed by means of the bounds on $\textit{diff}$ and h.

Finally, consider case (B.b.2), i.e., $t = 0$. We need to show that $E_{p\downarrow q} = \infty$. Since $\textit{pre}^*(q(0)) \cap \textit{post}^*(p(1)) \cap C \times \mathbb{N}$ is infinite, for an arbitrarily large $k \in \mathbb{N}$ there is a configuration $r(k) \in \textit{pre}^*(q(0)) \cap \textit{post}^*(p(1))$. We show that if k is sufficiently large, then the expected number of transitions needed to decrease the counter by some fixed constant b is infinite. This is achieved by analyzing the martingale $m^{(0)}, m^{(1)}, \ldots$ for $r(k)$, but this time we use the optional stopping theorem (see Sect. 2.1) to show that the probability of performing a finite path of length i which decreases the counter by b decays sufficiently slowly to make the expected length of this path infinite. It follows that $E_{p\downarrow q}$ is also infinite. We refer to [15] for details.

5.2 Distribution of termination time

In this section we present the results of [17] about the distribution of termination time in pPDA. These results do not have immediate algorithmic consequences, but bring a general insight into the behavior of probabilistic recursive programs. In particular, we show that stochastic computations defined by pPDA are "well-behaved" in the sense that if their expected termination time is finite, then the actual termination time is exponentially unlikely to deviate from this expectation (i.e., the probability of performing a run of length n decays exponentially in n).

Let us fix a pPDA $\Delta = (Q, \Gamma, \hookrightarrow, \textit{Prob})$ and a constant reward function $f : Q \times \Gamma \to \{0, 1\}$ which assigns 1 to all configurations. Similarly as in Sect. 5.1, for all $p, q \in Q$ and $X \in \Gamma$ we use $\textit{Run}(pXq)$ to denote $\textit{Reach}(pX, \{q\varepsilon\})$, and $[pXq]$ to denote the probability of $\textit{Run}(pXq)$. If $[pXq] > 0$, then E_{pXq} denotes $\mathbb{E}(\textit{Acc} \mid \textit{Run}(pXq))$ where $q\varepsilon$ is the only target configuration and f is the underlying reward function.

Our aim is to show that for all $p, q \in Q$ and $X \in \Gamma$ such that $[pXq] > 0$, there are essentially three possibilities:

– There is a "small" $k \in \mathbb{N}$ such that $\mathscr{P}(\textit{Acc} \geq n \mid \textit{Run}(pXq)) = 0$ for all $n \geq k$.
– E_{pXq} is finite and $\mathscr{P}(\textit{Acc} \geq n \mid \textit{Run}(pXq))$ decreases exponentially in n.
– E_{pXq} is infinite and $\mathscr{P}(\textit{Acc} \geq n \mid \textit{Run}(pXq))$ decreases "polynomially" in n.

A precise formulation of this result is given below.

Due to the translation presented in Sect. 4, we can equivalently consider pBPA where each stack symbol terminates with probability 1 or 0. Let X, Y be stack symbols. We say that X *depends* on Y if X can reach a configuration with head Y. Since the symbols which terminate with probability 0 are not interesting from the current point of view, they can be safely removed (obviously, the symbols which terminate with probability 1 do not depend on symbols which terminate with probability 0). So, for the rest of this section we assume that $\Delta = (\Gamma, \hookrightarrow, \textit{Prob})$ is a pBPA where *every* $X \in \Gamma$ terminates with probability 1. For every

$\alpha \in \Gamma^*$, we use E_α to denote the expected value of *Acc* over *Run*(α) where the only target configuration is ε (note that α terminates with probability 1). Observe that the dependence relation partitions Γ into "strongly connected components" formed by symbols that depend on each other, and one can also order these components into a directed acyclic graph (DAG) which has some finite height $h \geq 1$ (here the height of a DAG consisting of a single strongly connected component is defined as 1).

Theorem 11 (see [16, 17]) *Let $\Delta = (\Gamma, \hookrightarrow, Prob)$ be a pBPA where every $X \in \Gamma$ terminates with probability 1. Let x_{min} be the minimal probability assigned to a rule of Δ. Then for every $X_0 \in \Gamma$, one of the following is true (where Acc is interpreted as a random variable over Run(X_0)):*

(1) $\mathscr{P}(Acc \geq 2^{|\Gamma|}) = 0$.

(2) E_{X_0} *is finite and for all $n \in \mathbb{N}$ with $n \geq 2E_{X_0}$ we have that*

$$x_{min}^n \leq \mathscr{P}(Acc \geq n) \leq \exp\left(1 - \frac{n}{8E_{max}^2}\right).$$

Here, $E_{max} = \max\{E_X \mid X \text{ depends on } X_0\} < \infty$.

(3) E_{X_0} *is infinite and there is $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$ we have that*

$$c/\sqrt{n} \leq \mathscr{P}(Acc \geq n) \leq d_1/n^{d_2}$$

where $d_1 = 18h|\Gamma|/x_{min}^{3|\Gamma|}$, and $d_2 = 1/(2^{h+1} - 2)$. Here, h is the height of the DAG of strongly connected components of the dependence relation, and c is a suitable positive constant depending on Δ.

One can effectively distinguish between the three cases set out in Theorem 11. More precisely, case (1) can be recognized in polynomial time by looking only at the structure of the pBPA, i.e., disregarding the probabilities. Determining whether E_{X_0} is finite or infinite can be done in polynomial space by Theorem 8. This holds even if the transition probabilities of Δ are represented just symbolically by formulae of the existential fragment of Tarski algebra.

The proof of Theorem 11 is based on designing suitable martingales that are used to analyze the concentration of the termination probability. Here we only sketch the proof for the upper bound of Theorem 11 (2), which is perhaps the most interesting part. Observe that for every $\alpha \in \Gamma^*$ such that $\alpha \neq \varepsilon$, $E_\alpha < \infty$, and α terminates with probability one, performing one transition from α decreases the expected termination time by one on average. For every $w \in Run(X_0)$, we denote by $I(w)$ the maximal number $j \geq 0$ such that $w(j-1) \neq \varepsilon$. For every $i \geq 0$, we put

$$m^{(i)}(w) = E_{w(i)} + \min\{i, I(w)\}.$$

It is easy to see that $\mathbb{E}(m^{(i+1)} \mid m^{(i)}) = m^{(i)}$, i.e., $m^{(0)}, m^{(1)}, \ldots$ is a martingale. Let $E_{max} = \max\{E_X \mid X \text{ depends on } X_0\}$, and let $n \geq 2E_{X_0}$. By applying Azuma's inequality (see Sect. 2.1) we obtain

$$\mathscr{P}\left(m^{(n)} - E_{X_0} \geq n - E_{X_0}\right) \leq \exp\left(\frac{-(n - E_{X_0})^2}{2n(2E_{max})^2}\right) \leq \exp\left(\frac{2E_{X_0} - n}{8E_{max}^2}\right).$$

For every $w \in Run(X_0)$ we have that $w(n) \neq \varepsilon$ implies $m^{(n)} \geq n$. It follows:

$$\mathscr{P}(Acc \geq n) \leq \mathscr{P}\big(m^{(n)} \geq n\big) \leq \exp\left(\frac{2E_{X_0} - n}{8E_{\max}^2}\right) \leq \exp\left(1 - \frac{n}{8E_{\max}^2}\right).$$

The proof of Theorem 11 (3) is also based on designing and analyzing a suitable martingale, but the argument is more technical. We refer to [17] for details.

6 Mean payoff and other long-run properties

In this section we indicate how to analyze long-run average properties, such as the *expected mean payoff*, for pPDA. The key idea, originally presented in [14, 25], is to abstract the behavior of a given pPDA Δ into a *finite-state* Markov chain X_Δ, where possible evolutions of individual stack symbols are "summarized" and explicitly quantified. Since X_Δ preserves important behavioral aspects of Δ, the questions about long-run average properties of Δ can be reformulated as questions about X_Δ and solved by standard methods for finite-state Markov chains.

The original definition of X_Δ and the underlying analysis given in [14, 25] are somewhat technical. In this section, we follow a simpler approach enabled by the translation presented in Sect. 4. That is, we introduce the chain X_Δ only for the special type of pBPA obtained by this translation, which is remarkably simple. Then, we show how X_Δ can be used to compute/approximate the expected mean payoff for this special type of pBPA. For completeness, we also show how to lift the obtained results to general pPDA. Let us note that the chain X_Δ is also used in Sect. 7.1 to solve the model-checking problem for pPDA and linear-time logics.

For the rest of this section, we fix a pBPA $\Delta = (\Gamma, \hookrightarrow, Prob)$, where Γ is partitioned into two disjoint subsets $\Gamma_\downarrow$ and $\Gamma_\uparrow$ of all symbols that terminate with probability 1 and 0, respectively (cf. Sect. 4). Obviously, if $X \hookrightarrow \alpha$ and $X \in \Gamma_\downarrow$, then $\alpha \in \Gamma_\downarrow^*$. Similarly, if $X \hookrightarrow \alpha$ and $X \in \Gamma_\uparrow$, then α contains at least one symbol of $\Gamma_\uparrow$, and we can safely assume that $\alpha \in \Gamma_\downarrow^* \Gamma_\uparrow$.

Example 6 Let Δ be a pBPA with stack symbols A, B, X, Y, Z, where

$$
\begin{array}{lllll}
X \xrightarrow{0.2} AY, & Y \xrightarrow{0.2} BX, & Z \xrightarrow{0.1} AX, & A \xrightarrow{0.6} \varepsilon, & B \xrightarrow{0.8} \varepsilon, \\
X \xrightarrow{0.1} BY, & Y \xrightarrow{0.4} AY, & Z \xrightarrow{0.2} BY, & A \xrightarrow{0.2} B, & B \xrightarrow{0.1} BB, \\
X \xrightarrow{0.3} Y, & Y \xrightarrow{0.3} Z, & Z \xrightarrow{0.1} AZ, & A \xrightarrow{0.2} AA, & B \xrightarrow{0.1} AA, \\
X \xrightarrow{0.4} AZ, & Y \xrightarrow{0.1} AZ, & Z \xrightarrow{0.6} BZ.
\end{array}
$$

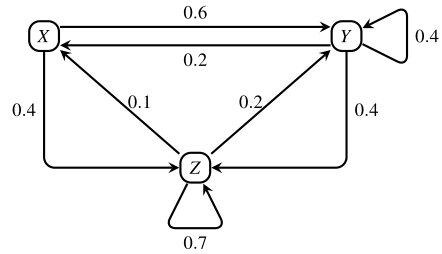Then $\Gamma_\downarrow = \{A, B\}$ and $\Gamma_\uparrow = \{X, Y, Z\}$.

Now we can define the promised Markov chain X_Δ.

Definition 8 Let X_Δ be a finite-state Markov chain where $\Gamma_\uparrow$ is the set of vertices, and $X \xrightarrow{x} Y$ is a transition of X_Δ iff $x = \sum_{X \hookrightarrow \alpha Y} y > 0$.

For the pBPA Δ of Example 6, the chain X_Δ is shown in Fig. 3.

Fig. 3 The chain X_Δ for the pBPA of Example 6

Let $Y \in \Gamma_\uparrow$. Obviously, for almost every $w \in Run(M_\Delta, Y)$ there is an infinite increasing sequence of indexes $i_0, i_1, \ldots$ such that $w(i) \in \Gamma_\uparrow$ iff $i = i_j$ for some $j \in \mathbb{Z}^{\geq 0}$. The sequence $w(i_0), w(i_1), \ldots$ is a run in X_Δ initiated in Y, which we call the *footprint* of w. Further, for every $j \in \mathbb{Z}^{\geq 0}$, the *j-th jump* of w is the finite path $w(i_j) \ldots w(i_{j+1})$ in M_Δ. Hence, the transitions of X_Δ represent the jumps in M_Δ. One can easily verify that the mapping Υ which to every run $w \in Run(M_\Delta, Y)$ assigns its footprint $\hat{w} \in Run(X_\Delta, Y)$ is defined almost surely and preserves probability.

Let $C_1, \ldots, C_n$ be the bottom strongly connected components (BSCCs) of X_Δ. For every $1 \leq i \leq n$, let $Reach_{X_\Delta}(Y, C_i)$ be the set of all $\hat{w} \in Run(X_\Delta, Y)$ that visit C_i, and let

$$p_i = \mathscr{P}\big(Reach_{X_\Delta}(Y, C_i)\big).$$

Note that $\sum_{i=1}^{n} p_i = 1$. Further, let π_i be the unique invariant distribution for C_i (see, e.g., [21]). Then for almost all $\hat{w} \in Reach_{X_\Delta}(Y, C_i)$ and every $Z \in C_i$ we have that

$$\lim_{i \to \infty} \frac{\#_Z^i(\hat{w})}{i + 1} = \pi_i(Z)$$

where $\#_Z^i(\hat{w})$ denotes the number of all indexes j such that $0 \leq j \leq i$ and $\hat{w}(j) = Z$. Since the above defined mapping Υ preserves probability, we obtain that almost every run $w \in Run(M_\Delta, Y)$ whose footprint $\hat{w}$ belongs to $Reach_{X_\Delta}(Y, C_i)$ satisfies

$$\lim_{i \to \infty} \frac{\#_Z^i(w)}{\#_{\Gamma_\uparrow}^i(w)} = \lim_{i \to \infty} \frac{\#_Z^i(\hat{w})}{i + 1} = \pi_i(Z)$$

where $\#_Z^i(w)$ and $\#_{\Gamma_\uparrow}^i(w)$ denote the number all indexes $0 \leq j \leq i$ such that $w(j) = Z$ and $w(j) \in \Gamma_\uparrow$, respectively.

Now let $f : \Gamma^* \to \mathbb{R}^{\geq 0}$ be a simple reward function (see Sect. 2.4). Recall that for every $A \in \Gamma_\downarrow$, we use E_A to denote $\mathbb{E}(Acc \mid Reach_{M_\Delta}(A, \varepsilon))$, i.e., the expected total reward accumulated along a terminating path initiated in A before visiting ε. Let $Z \xrightarrow{y} U$ be a transition of X_Δ, and let $E(Z \to U)$ be the expected total reward accumulated along a jump[5] represented by $Z \xrightarrow{y} U$. Observe that

$$E(Z \to U) = f(Z) + \sum_{Z \xrightarrow{x} AU} \frac{x}{y} \cdot E_A. \qquad (2)$$

---

[5]The last configuration of a jump does *not* contribute to the total accumulated reward.

Let C_i be a BSCC of X_Δ with the invariant distribution π_i. Then

$$\sum_{Z \in C_i} \pi_i(Z) \cdot \sum_{Z \xrightarrow{y} U} y \cdot E(Z \to U) = \sum_{Z \in C_i} \pi_i(Z) \cdot \left(f(Z) + \sum_{Z \xrightarrow{x} AU} x \cdot E_A \right) \qquad (3)$$

is the *long-run average reward per jump* in a run of $Run(M_\Delta, Z)$ whose footprint visits C_i. More precisely, for almost all $w \in Run(M_\Delta, Y)$ whose footprint visits C_i we have that the long-run average reward per jump is defined (i.e., the limit of partial averages computed for more and more jumps in w exists) and is equal to (3).

Similarly, we obtain that the *long-run average length* of a jump exists for almost all $w \in Run(M_\Delta, Y)$ whose footprint visits C_i and is equal to the same value (observe that the long-run average length of a jump is actually the long-run average reward per jump with respect to the constant reward function **1** which assigns 1 to every configuration; hence, we can just re-use (3) for another reward function).

Since the long-run average reward per jump as well as the long-run average length of a jump are the same for almost all $w \in Run(M_\Delta, Y)$ whose footprint visits C_i, the *long-run average reward per configuration* (i.e., the value of the random variables *MPsup* and *MPinf* introduced in Sect. 2.4) should be equal to

$$\frac{\sum_{Z \in C_i} \pi_i(Z) \cdot (f(Z) + \sum_{Z \xrightarrow{x} AU} x \cdot E_A^f)}{\sum_{Z \in C_i} \pi_i(Z) \cdot (1 + \sum_{Z \xrightarrow{x} AU} x \cdot E_A^1)} \qquad (4)$$

for almost all $w \in Run(M_\Delta, Y)$ whose footprint visits C_i. This is correct if (4) is defined, i.e., the numerator or the denominator is finite. If they are both infinite (which may happen), then the problem of computing the expected mean payoff becomes more problematic. We may even have that $MPsup(w) \neq MPinf(w)$ for almost all runs $w \in Run(M_\Delta, Y)$ whose footprint visits C_i. A more detailed analysis of this problematic case is still missing in the literature.

Observe that the denominator of (4) is finite if E_A^1 is finite for every $A \in \Gamma_\downarrow$. This motivates the following definition (later we also discuss the problem of computing the expected mean payoff for general pPDA, and therefore we formulate our definition for general pPDA).

Definition 9 Let $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ be a pPDA. We say that Δ is *well-terminating* if for all $p, q \in Q$ and $X \in \Gamma$ such that $[pXq] > 0$ we have that $E_{pXq} < \infty$, where the underlying reward function assigns 1 to every configuration (cf. Sect. 5.1).

From now on, we assume that the pBPA Δ fixed above is well-terminating, which means that E_A^1 is finite for all $A \in \Gamma_\downarrow$. Since f is simple, we actually have that E_A^f is also finite for all $A \in \Gamma_\downarrow$. For every BSCC C_i of X_Δ, we use v_i to denote the value defined by (4), and p_i to denote the probability of reaching C_i from Y in X_Δ. Now observe the following:

- The invariant distribution π_i is the unique solution of the system of linear equations $\mathbf{y} = \mathbf{y} \cdot \mathbf{M}$, where $\mathbf{M}$ is the transition matrix of C_i (see, e.g., [21]).
- The tuple of all E_A^f, where $A \in \Gamma_\downarrow$ and $E_A^f > 0$, is the unique solution of the system of linear equations $Expect'_\Delta$ in $((\mathbb{R}^{\geq 0})^k, \sqsubseteq)$ (see Sect. 5.1). Note that the system $Expect'_\Delta$ is constructible in polynomial time even if the transition probabilities of Δ are encoded just symbolically in Tarski algebra (since all E_A^f are finite, we only need to recognize those E_A^f that are equal to zero, i.e., we do *not* rely on the procedure outlined in Remark 3 which requires explicit values of transition probabilities).

Analogous observations hold for the tuple of all E_A^1. Here the situation is even simpler, because all of E_A^1 are positive.

– The probability p_i is a component of the unique solution of an efficiently constructible system of linear equations, where the transition probabilities of X_Δ are used as coefficients (see, e.g., [5]).

The above observations imply that if the transition probabilities of Δ are given *explicitly as rational numbers*, then both v_i and p_i are rational and computable in polynomial time. If Δ is obtained by running the translation algorithm of Sect. 4 with some well-terminating pPDA on input, then the transition probabilities of Δ are encoded just *symbolically* in the existential fragment of Tarski algebra (but they are already guaranteed to be positive). In this case, we cannot *compute* v_i and p_i directly, but we can still express them in the existential fragment of Tarski algebra by a formula constructible in polynomial time.

Note that $\mathscr{P}(MPsup = MPinf = v_i)$ is not necessarily equal to p_i, because there can be another BSCC C_j of X_Δ such that $v_j = v_i$, and then $\mathscr{P}(MPsup = MPinf = v_i)$ is at least $p_i + p_j$. Since all v_i's are expressible in the existential fragment of Tarski algebra, we can decide whether $v_i = v_j$ in space polynomial in $\|\Delta\|$, and thus identify all BSCCs with the same value.

Now we show how to lift the above results to general pPDA. Let us fix a well-terminating pPDA Δ, a simple reward function f over the configurations of Δ, and a configuration $p_0 X_0 \in Q \times \Gamma$ that cannot reach a configuration with empty stack (note that the problem whether a given pPDA Δ is well-terminating is decidable in polynomial space). If we interpret *MPsup* and *MPinf* as random variables over $Run(p_0 X_0)$, we obtain the following:

– Let $v_1, \ldots, v_m$ be the eligible values obtained for the symbol $\langle p_0 X_0 \uparrow \rangle$ of the pBPA $\Delta_\bullet$ by the method described above, and let $p_1, \ldots, p_m$ be the associated probabilities. Then almost all runs of $Run(p_0 X_0)$ can be split into m disjoint classes $\mathcal{R}_1, \ldots, \mathcal{R}_m$ such that
  – $\mathscr{P}(\mathcal{R}_i) = p_i$ for all $1 \le i \le m$;
  – for almost all $w \in \mathcal{R}_i$ we have that $MPsup(w) = MPinf(w) = v_i$.
  This follows from the correspondence between the runs in Δ and $\Delta_\bullet$ established in Sect. 4.

Thus, we obtain the following theorem:

Theorem 12 *Let $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ be a well-terminating pPDA, $f : Q \times \Gamma^* \to \mathbb{R}^{\ge 0}$ a simple reward function, and $p_0 X_0 \in Q \times \Gamma$ a configuration that cannot reach a configuration with empty stack. Then for almost all $w \in Run(p_0 X_0)$ we have that $MPsup(w) = MPinf(w)$, and there are at most $|Q| \cdot |\Gamma|$ distinct values that MPsup and MPinf may assume over $Run(p_0 X_0)$ with positive probability. Moreover, these values and the associated probabilities can by approximated up to a given absolute error $\delta > 0$ in polynomial space.*

Let us note that Theorem 12 holds also for linear reward functions.

7 Model-checking

Finally, we present the existing results about the model-checking problem for pPDA and formulae of linear-time and branching-time logics.

	pBPA	pPDA
quantitative DRA	in **PSPACE**	in **PSPACE**
qualitative DRA	in **P**	in **PSPACE**
quantitative LTL	**EXPTIME**-complete	**EXPTIME**-complete
qualitative LTL	**EXPTIME**-complete, in **P** for a fixed formula	**EXPTIME**-complete

Fig. 4 Summary of results for pPDA/pBPA and linear-time model-checking

7.1 Linear-time logics

Results for pPDA and pBPA   Model-checking pPDA against deterministic Büchi specification was already studied in [25], where it was shown that the *quantitative* model-checking problem, i.e., the question whether the probability of all runs accepted by a given deterministic Büchi automaton is bounded by a given $\varrho \in [0, 1]$, is solvable in exponential time. This result was extended to deterministic Muller automata (and hence all ω-regular properties) in [19]. The complexity of quantitative and qualitative model-checking problem for pPDA, pBPA and LTL was studied in greater detail in [30], where it was shown that for a given pPDA Δ and a given LTL formula φ, the quantitative and qualitative model-checking problem is **EXPTIME**-hard and solvable in space polynomial in $\|\Delta\|$ and time exponential in $\|\varphi\|$. Moreover, it was shown that the *qualitative* LTL model-checking problem is solvable in *polynomial time* for pBPA and every fixed LTL formula. The upper bounds for LTL require different techniques than the ones for automata specifications, and the lower bounds are inherited from the non-probabilistic case [6, 38].

The current knowledge about quantitative/qualitative linear-time model-checking is summarized in Fig. 4. Here DRA stands for *deterministic Rabin automata* specifications (see Definition 10 below).

Using the techniques presented in previous sections, the results about model-checking DRA specifications are actually easy to prove.

Definition 10 A *deterministic Rabin automaton (DRA)* is a tuple $\mathscr{D} = (D, \Sigma, \rightarrow, d_{init}, R)$, where D is a finite set of control states, Σ is a finite input alphabet, $\rightarrow \subseteq D \times \Sigma \times D$ is a deterministic and total transition relation, $d_0 \in D$ is an initial state, and $R = (E_1, F_1), \ldots, (E_n, F_n)$ is a Rabin acceptance condition, where $E_i, F_i \subseteq D$.

Let u be an infinite word over the alphabet Σ. A *computation* of $\mathscr{D}$ over u is an infinite sequence of states $c(u) = d_0, d_1, d_2, \ldots$ such that $d_0 = d_{init}$ and $d_i \xrightarrow{u(i)} d_{i+1}$ for all $i \in \mathbb{Z}^{\geq 0}$. We say that u is *accepted* by $\mathscr{D}$ if there is $1 \leq i \leq n$ such that all states of E_i appear in $c(u)$ finitely many times, and at least one state of F_i appears in $c(u)$ infinitely many times.

Let $\Delta = (Q, \Gamma, \hookrightarrow, Prob)$ be a pPDA, and let $p_0 X_0 \in Q \times \Gamma$ be a configuration of Δ that cannot reach a configuration with empty stack. Further, let $\mathscr{D} = (D, \Sigma, \rightarrow, d_{init}, R)$ be a DRA where $\Sigma = Q \times \Gamma$ is the input alphabet. We say that $w \in Run(p_0 X_0)$ is *recognized* by $\mathscr{D}$ if the corresponding word $p_0 X_0 \, p_1 X_1 \, p_2 X_2 \ldots$, where $p_i X_i$ is the head of $w(i)$, is accepted by $\mathscr{D}$. The set of all $w \in Run(p_0 X_0)$ that are recognized by $\mathscr{D}$ is denoted by $Run(p_0 X_0, \mathscr{D})$.

Our aim is to compute/approximate the probability of $Run(p_0 X_0, \mathscr{D})$. This is achieved in three steps.

Step I. We compute the synchronized product of Δ and $\mathscr{D}$, which is a pPDA $\Delta \times \mathscr{D}$ that behaves like Δ but also simulates the execution of $\mathscr{D}$ in its finite control. Hence, the set of

control states of $\Delta \times \mathscr{D}$ is $Q \times D$, and if $pX \xrightarrow{x} q\alpha$ is a rule of Δ and $d \xrightarrow{pX} d'$ a transition of $\mathscr{D}$, then $(p, d)X \xrightarrow{x} (q, d')\alpha$ is a rule of $\Delta \times \mathscr{D}$.

Let $w \in Run(M_{\Delta \times \mathscr{D}}, (p_0, d_{init})X_0)$, and let $\inf(w)$ be the set of all $d \in D$ such that w visits infinitely many configurations with head of the form $(q, d)Y$. We say that w is *accepting* if there is $1 \le i \le n$ such that $\inf(w) \cap E_i = \emptyset$ and $\inf(w) \cap F_i \ne \emptyset$. Let $Run(\Delta \times \mathscr{D}, (p_0, d_{init})X_0, Accept)$ be the set of all accepting runs of $Run(M_{\Delta \times \mathscr{D}}, (p_0, d_{init})X_0)$. Since $\mathscr{D}$ just "observes" the computation of Δ in $\Delta \times \mathscr{D}$ without any real influence, we have that

$$\mathscr{P}\big(Run(p_0X_0, \mathscr{D})\big) = \mathscr{P}\big(Run(\Delta \times \mathscr{D}, (p_0, d_{init})X_0, Accept)\big).$$

Step II. We translate $\Delta \times \mathscr{D}$ into the corresponding pBPA $\Delta \times \mathscr{D}_\bullet$ by the construction given in Sect. 4. Let $w \in Run(M_{\Delta \times \mathscr{D}_\bullet}, \langle (p_0, d_{init})X_0\uparrow\rangle)$, and let $\inf(w)$ be the set of all $d \in D$ such that w visits infinitely many configurations with head of the form $\langle (q, d)Y\odot\rangle$. Similarly as above, we say that w is *accepting* if there is $1 \le i \le n$ such that $\inf(w) \cap E_i = \emptyset$ and $\inf(w) \cap F_i \ne \emptyset$, and we use $Run(\Delta \times \mathscr{D}_\bullet, \langle (p_0, d_{init})X_0\uparrow\rangle, Accept)$ to denote the set of all accepting runs. Due to the correspondence between the runs in $M_{\Delta \times \mathscr{D}}$ and $M_{\Delta \times \mathscr{D}_\bullet}$ established in Sect. 4, we can conclude that

$$\mathscr{P}\big(Run(\Delta \times \mathscr{D}, (p_0, d_{init})X_0, Accept)\big) = \mathscr{P}\big(Run(\Delta \times \mathscr{D}_\bullet, \langle (p_0, d_{init})X_0\uparrow\rangle, Accept)\big).$$

Another important observation is that if Δ is a pBPA, then the set of rules of $\Delta \times \mathscr{D}_\bullet$ is computable in time *polynomial* in $\|\Delta\|$ and $\|\mathscr{D}\|$. This does not follow *immediately* from Remark 2, because $\Delta \times \mathscr{D}$ is *not* a pBPA. However, if Δ is a pBPA, then the control unit of $\Delta \times \mathscr{D}$ just stores the current state of $\mathscr{D}$, and hence for every configuration dY of $\Delta \times \mathscr{D}$ we have that it terminates with the same probability as the configuration Y of Δ. Thus, all information need for computing the set of rules of $\Delta \times \mathscr{D}_\bullet$ can be obtained by analyzing the pBPA Δ.

Step III. We construct the Markov chain $X_{\Delta \times \mathscr{D}_\bullet}$ of Sect. 6, and classify each BSCC of $X_{\Delta \times \mathscr{D}_\bullet}$ as either "good" or "bad" with respect to the Rabin condition R. It turns out that almost all runs $w \in Run(\Delta \times \mathscr{D}_\bullet, \langle (p_0, d_{init})X_0\uparrow\rangle)$ whose footprint visits a good BSCC of $X_{\Delta \times \mathscr{D}_\bullet}$ are accepting, and almost all runs $w \in Run(\Delta \times \mathscr{D}_\bullet, \langle (p_0, d_{init})X_0\uparrow\rangle)$ whose footprint visits a bad BSCC of $X_{\Delta \times \mathscr{D}_\bullet}$ are *not* accepting. Hence, $\mathscr{P}(Run(p_0X_0, \mathscr{D}))$ is equal to the probability of visiting a good BSCC in $X_{\Delta \times \mathscr{D}_\bullet}$, and from this we obtain the desired results.

More concretely, let C be a BSCC of $X_{\Delta \times \mathscr{D}_\bullet}$. Note that the elements of C are symbols of the form $\langle (q, d)Y\uparrow\rangle$. We compute the set $C_{\inf}$ of all $d' \in D$ such that there exists a configuration with head of the form $\langle (r, d')Z\odot\rangle$ reachable from some $\langle (q, d)Y\uparrow\rangle \in C$ in the Markov chain $M_{\Delta \times \mathscr{D}_\bullet}$. Note that $C_{\inf}$ can be computed in *polynomial time* if the set of rules of $\Delta \times \mathscr{D}_\bullet$ is given (the precise probabilities of these rules are irrelevant). In particular, if Δ is a pBPA, then $C_{\inf}$ can be computed in time *polynomial* in $\|\Delta\|$ and $\|\mathscr{D}\|$ (see above). We declare C as *good* if there is $1 \le i \le n$ such that $C_{\inf} \cap E_i = \emptyset$ and $C_{\inf} \cap F_i \ne \emptyset$. Now it suffices to realize that for almost every run $w \in Run(\langle (p_0X_0)X\uparrow\rangle)$ in $M_{\Delta \times \mathscr{D}_\bullet}$ whose footprint in $X_{\Delta \times \mathscr{D}_\bullet}$ visits C we have that $\inf(w) = C_{\inf}$. So, the probability

$$\mathscr{P}\big(Run(\Delta \times \mathscr{D}_\bullet, \langle (p_0, d_{init})X_0\uparrow\rangle, Accept)\big)$$

is equal to the probability of visiting a good BSCC of $X_{\Delta \times \mathscr{D}_\bullet}$ from $\langle (p_0, d_{init})X_0\uparrow\rangle$ in $X_{\Delta \times \mathscr{D}_\bullet}$. Since $X_{\Delta \times \mathscr{D}_\bullet}$ has finitely many states, this probability is a component in the unique solution of an effectively constructible system of linear equations whose coefficients

are the transition probabilities of $X_{\Delta \times \mathscr{D}_\bullet}$ (see, e.g., [5]). Since the transition probabilities of $X_{\Delta \times \mathscr{D}_\bullet}$ are effectively expressible in the existential fragment of Tarski algebra (cf. Theorem 2), the same can be said about the probability of our interest. Thus, we obtain the **PSPACE** upper bounds given in Fig. 4.

If Δ is a pBPA, then the rules of $\Delta \times \mathscr{D}_\bullet$ are computable in time polynomial in $\|\Delta\|$ and $\|\mathscr{D}\|$ (see above). This means that the transitions of $X_{\Delta \times \mathscr{D}_\bullet}$ are also computable in polynomial time (see Definition 8). Hence, we can decide in polynomial time whether *all* BSCC of $X_{\Delta \times \mathscr{D}_\bullet}$ reachable from $\langle (p_0, d_{init}) X_0 \uparrow \rangle$ are good. Thus, we obtain the **P** upper bound for qualitative DRA properties and pBPA of Fig. 4.

Results for pOC   The qualitative and quantitative model-checking problems for pOC and linear-time properties encoded by DRA was analyzed in [15]. To some extent, the method is similar to the one for pPDA described in the previous paragraph, but the underlying analytical techniques are different. Again, it is shown that the probability of all runs that are recognized[6] by a given DRA is equal to the probability of visiting a "good" BSCC in a suitable finite-state Markov chain $Y_{\Delta \times \mathscr{D}}$ (the chain $Y_{\Delta \times \mathscr{D}}$ is somewhat different from the chain $X_{\Delta \times \mathscr{D}_\bullet}$ used for pPDA). The set of transitions of $Y_{\Delta \times \mathscr{D}}$ is computable in polynomial time, and the "good" BSCCs of $Y_{\Delta \times \mathscr{D}}$ are also computable in polynomial time. Thus, we obtain the following:

Theorem 13 *Let $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$ be a pOC and $\mathscr{D} = (D, \Sigma, \rightarrow, d_{init}, R)$ a DRA where $\Sigma = Q \times \{0, 1\}$. For every configuration $p(k)$, where k is encoded in unary, the problem whether almost all runs initiated in $p(k)$ are recognized by $\mathscr{D}$ is in **P**.*

Further, one can efficiently *approximate* the probability of reaching a good BSCC in $Y_{\Delta \times \mathscr{D}}$ by approximating the values of its transition probabilities and solving the corresponding system of linear equations. The underlying analysis is not completely simple and relies on the *divergence gap theorem*, which bounds a positive non-termination probability in pOC away from zero (this theorem has been established in [15] by analyzing the martingale constructed in Sect. 5.1). This leads to the following result:

Theorem 14 *Let $\Delta = (Q, \delta^{=0}, \delta^{>0}, P^{=0}, P^{>0})$ be a pOC, $\mathscr{D} = (D, \Sigma, \rightarrow, d_{init}, R)$ a DRA where $\Sigma = Q \times \{0, 1\}$, and $\delta > 0$ a rational constant. For every configuration $p(k)$, where k is encoded in unary, the probability of all runs initiated in $p(k)$ that are recognized by $\mathscr{D}$ can be approximated up to the relative error δ in polynomial time on the unit-cost rational arithmetic RAM.*

7.2 Branching-time logics

The currently known results about model-checking pPDA and pBPA against branching-time formulae of PCTL and PCTL* established in [12, 19] are summarized in Fig. 5. The abbreviation "p.c." stands for *program complexity*, i.e., the respective upper bounds hold for an *arbitrary* fixed formula, and the lower bounds hold for *some* fixed formula.

The undecidability of the model-checking problem for pPDA and PCTL is proven by reduction from the Post's correspondence problem (see below). The fundamental idea of

---

[6]Formally, the "head" of a given pOC configuration $p(k)$ is either $(p, 0)$ or $(p, 1)$, depending on whether $k = 0$ or $k > 0$, respectively. The input alphabet of the corresponding DRA is then $Q \times \{0, 1\}$.

	pBPA	pPDA
PCTL	?	undecidable (fixed formula)
PCTL*	undecidable (fixed formula)	undecidable (fixed formula)
qPCTL	**EXPTIME**-complete	**EXPTIME**-complete
qPCTL*	**2-EXPTIME**-complete	**2-EXPTIME**-complete
qPCTL (p.c.)	**P**	**EXPTIME**-complete
qPCTL* (p.c.)	**P**	**EXPTIME**-complete

Fig. 5 Summary of results for pPDA/pBPA and branching-time model-checking

encoding words into probabilities which lies behind this proof appeared for the first time in [14]. The undecidability proof for pBPA and PCTL* is obtained as a slight modification of the construction used for pPDA and PCTL. However, the question whether this undecidability result can be extended to pBPA and PCTL is still open.

The **2-EXPTIME** (and **EXPTIME**) upper bounds on model-checking pPDA against qPCTL* (and qPCTL) rely on the results of [30] about LTL model-checking for pPDA. Using these results, one can show that the set of all configurations that almost surely satisfy a given path formula with regular valuations of atomic propositions is also *effectively regular*, i.e., the associated DFA is effectively constructible. From this one obtains a model checking algorithm for qPCTL*.

The **2-EXPTIME** (and **EXPTIME**) lower bounds on model-checking pPDA against qPCTL* (and qPCTL) are based on standard techniques from non-probabilistic model-checking algorithms [7, 42].

We sketch the main idea of the undecidability proof for model-checking pPDA against PCTL formulae. The result is obtained by reduction from (a slightly modified version of) the Post's correspondence problem (PCP). An instance of PCP consists of two sequences $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ of words over the alphabet $\Sigma = \{A, B, \bullet\}$ such that all x_i and y_j have the same length m. The question is whether there is a finite sequence $i_1, \cdots, i_k$ of indexes such that $x_{i_1} \cdots x_{i_k}$ and $y_{i_1} \cdots y_{i_k}$ are the same words after erasing all occurrences of "$\bullet$". Given such an instance, we construct a pPDA Δ where

– the number of control states is $\mathcal{O}(m \cdot n)$, and there are always three distinguished control states g, c, and t;
– the stack alphabet of Δ is fixed and contains the symbols Z, Y, and nine symbols of the form (z, z') where $z, z' \in \Sigma$.

Further, we define a PCTL formula

$$\Phi \equiv \Diamond^{>0}\big(cZ \wedge \Diamond^{=1/2} tY\big)$$

where the atomic propositions cZ and tY are valid in exactly all configurations with head cZ and tY, respectively. The formula $\Diamond^{\sim\varrho}(\Psi)$ is an abbreviation for $\mathscr{P}^{\sim\varrho}(true\,\mathcal{U}\,\Psi)$.

Our construction ensures that the given PCP instance has a solution iff $gZ \models^\nu \Phi$, where ν is the simple valuation described above (from now on, we omit the ν superscript in $\models^\nu$). From the initial configuration gZ, the pPDA Δ tries to "guess" a solution to our PCP instance by pushing pairs of words (x_i, y_i) successively to the stack. Since x_i and y_i have the same length m, this is implemented by pushing pairs of letters from Σ. For example, if $x_i = AAB$ and $y_i = BA\bullet$, then the pair (x_i, y_i) is stored as a sequence of three stack symbols $(A, B), (A, A), (B, \bullet)$, where $(B, \bullet)$ is on top of the stack. After storing a chosen pair of words, the automaton can either go on with guessing another pair of words, or enter a *checking* configuration by changing the control state from g to c and pushing the symbol
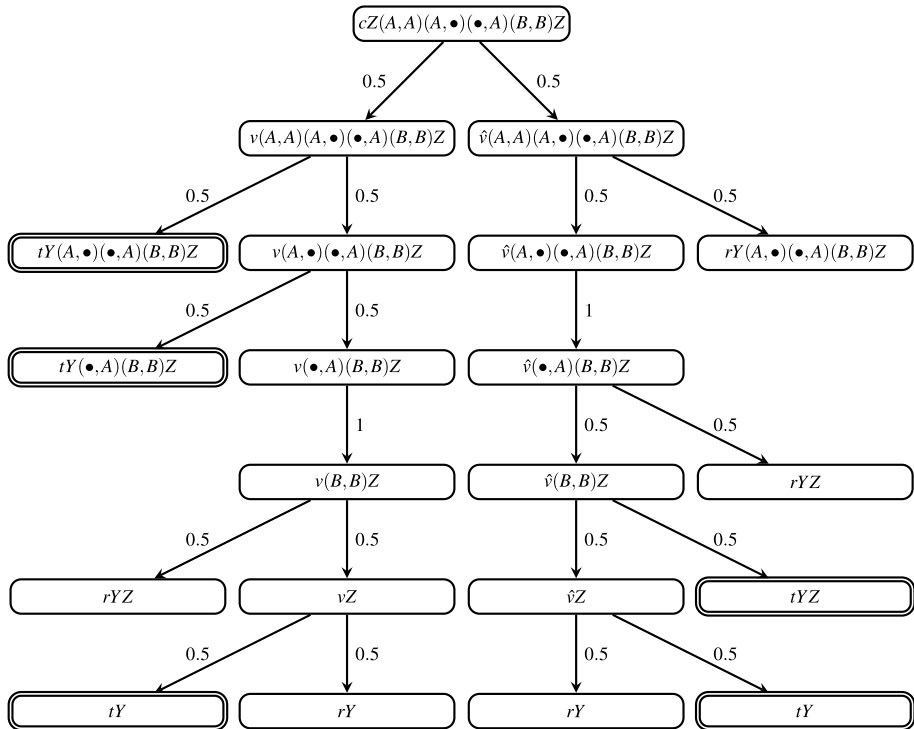
Fig. 6 Configurations reachable from $cZ(A, A)(A, \bullet)(\bullet, A)(B, B)Z$

Z on top of the stack. The transition probabilities do not matter here, and hence we assume they are distributed uniformly. This "guessing phase" is formalized below. Note that a pair (x_i, y_i) needs to be pushed "from right to left" because the top of the stack is on the left-hand side (we use $\hat{x}_i$ and $\hat{y}_i$ to denote the reverse of x_i and y_i, respectively). Since transition probabilities are distributed uniformly, we do not write them explicitly. The symbol "|" separates alternatives.

$$gX \to g_1^1 X \mid \cdots \mid g_n^1 X,$$

$$g_i^j X \to g_i^{j+1}\big(\hat{x}_i(j), \hat{y}_i(j)\big)X,$$

$$g_i^{m+1} X \to cZX \mid gX.$$

Here $1 \le i \le n$, $1 \le j \le m$, $\hat{x}_i(j)$ and $\hat{y}_i(j)$ denote the jth letters in $\hat{x}_i$ and $\hat{y}_i$, respectively, and X ranges over the stack alphabet.

Obviously, a configuration of the form $cZ\alpha$ is reachable from gZ iff $\alpha \equiv (a_1, b_1) \cdots (a_\ell, b_\ell)Z$ and there is a sequence $i_1, \ldots, i_k$ such that $a_\ell \cdots a_1 = x_{i_1} \cdots x_{i_k}$ and $b_\ell \cdots b_1 = y_{i_1} \cdots y_{i_k}$. The crucial part of the construction is the next phase which verifies that the guess was correct, i.e., that the words stored in the first and the second component of stack symbols are the same when "$\bullet$" is disregarded. First, we erase Z from the top of the stack and change the control state to either v or $\hat{v}$, which intuitively means that we are going to read the letters stored either in the first or in the second component of stack symbols, respectively. Then, we start to erase the symbols from the stack one by one according to the

following transition rules (again, the transition probabilities are distributed uniformly):

$$cZ \to v\varepsilon \mid \hat{v}\varepsilon, \quad v(A,z) \to tY \mid v\varepsilon, \quad \hat{v}(z,A) \to rY \mid \hat{v}\varepsilon, \quad tY \to tY,$$
$$v(B,z) \to rY \mid v\varepsilon, \quad \hat{v}(z,B) \to tY \mid \hat{v}\varepsilon, \quad rY \to rY.$$
$$v(\bullet,z) \to v\varepsilon, \quad \hat{v}(z,\bullet) \to \hat{v}\varepsilon,$$
$$vZ \to tY \mid rY, \quad \hat{v}Z \to tY \mid rY,$$

Here z ranges over Σ. We claim that a checking configuration satisfies the formula $\Diamond^{=1/2}\, tY$ iff the previous guess was correct. To get some intuition, let us evaluate the probability of reaching a configuration with the head tY for, e.g., a configuration $cZ(A,A)(A,\bullet)(\bullet,A)(B,B)Z$. By inspecting the part of M_Δ reachable from $cZ(A,A)(A,\bullet)$ $(\bullet,A)(B,B)Z$ (see Fig. 6), one can easily confirm that this probability is equal to

$$\frac{1}{2}\left(\left(1 \cdot \frac{1}{2} + 1 \cdot \frac{1}{2^2} + 0 \cdot \frac{1}{2^3} + 1 \cdot \frac{1}{2^4}\right) + \left(0 \cdot \frac{1}{2} + 0 \cdot \frac{1}{2^2} + 1 \cdot \frac{1}{2^3} + 1 \cdot \frac{1}{2^4}\right)\right)$$

which can be written in binary as follows: $\frac{1}{2}(0.1101 + 0.0011)$. The first three digits after the binary point in 0.1101 and 0.0011 reflect the structure of the words $AA\bullet B$ and $A\bullet AB$ stored in the stack, and the last 1 is due to the Z at the very bottom. Note that the role of A in the two words is dual—in the first case, it generates 1's, and in the second case it generates 0's (similarly for B). The symbol $\bullet$ is just popped from the stack with probability one, which does not influence the probability of reaching a configuration satisfying tY. Note that the probabilities 0.1101 and 0.0011 are "complementary" and their sum is equal to 1. This "complementarity" breaks down iff the words stored in the first and the second component of stack symbols are *not* the same, in which case the sum is different from 1.

The above construction does not directly work for pBPA, because here we cannot carry any information down the stack when popping the symbols. It is possible to overcome this problem, but the constructed formula becomes more complicated and it is expressible only in PCTL*.

References

1. Allender E, Bürgisser P, Kjeldgaard-Pedersen J, Miltersen P (2008) On the complexity of numerical analysis. SIAM J Comput 38:1987–2006
2. Alur R, Chaudhuri S, Etessami K, Madhusudan P (2005) On-the-fly reachability and cycle detection for recursive state machines. In: Proceedings of TACAS 2005. Lecture notes in computer science, vol 3440. Springer, Berlin, pp 61–76
3. Alur R, Etessami K, Yannakakis M (2001) Analysis of recursive state machines. In: Proceedings of CAV 2001. Lecture notes in computer science, vol 2102. Springer, Berlin, pp 207–220
4. Athreya K, Ney P (1972) Branching processes. Springer, Berlin
5. Baier C, Katoen JP (2008) Principles of model checking. MIT Press, Cambridge
6. Bouajjani A, Esparza J, Maler O (1997) Reachability analysis of pushdown automata: application to model checking. In: Proceedings of CONCUR'97. Lecture notes in computer science, vol 1243. Springer, Berlin, pp 135–150
7. Bozzelli L (2006) Complexity results on branching-time pushdown model checking. In: Proceedings of VMCAI 2006. Lecture notes in computer science, vol 3855. Springer, Berlin, pp 65–79

8. Brázdil T, Brožek V, Holeček J, Kučera A (2008) Discounted properties of probabilistic pushdown automata. In: Proceedings of LPAR 2008. Lecture notes in computer science, vol 5330. Springer, Berlin, pp 230–242

9. Brázdil T, Brožek V, Etessami K (2010) One-counter stochastic games. In: Proceedings of FST&TCS 2010. Leibniz international proceedings in informatics, vol 8. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp 108–119

10. Brázdil T, Brožek V, Etessami K, Kučera A (2011) Approximating the termination value of one-counter MDPs and stochastic games. In: Proceedings of ICALP 2011, Part II. Lecture notes in computer science, vol 6756. Springer, Berlin, pp 332–343

11. Brázdil T, Brožek V, Etessami K, Kučera A, Wojtczak D (2010) One-counter Markov decision processes. In: Proceedings of SODA 2010. SIAM, Philadelphia, pp 863–874

12. Brázdil T, Brožek V, Forejt V (2009) Branching-time model-checking of probabilistic pushdown automata. Electron Notes Theor Comput Sci 239:73–83

13. Brázdil T, Brožek V, Forejt V, Kučera A (2008) Reachability in recursive Markov decision processes. Inf Comput 206(5):520–537

14. Brázdil T, Esparza J, Kučera A (2005) Analysis and prediction of the long-run behavior of probabilistic sequential programs with recursion. In: Proceedings of FOCS 2005. IEEE Computer Society, Los Alamitos, pp 521–530

15. Brázdil T, Kiefer S, Kučera A (2011) Efficient analysis of probabilistic programs with an unbounded counter. In: Proceedings of CAV 2011. Lecture notes in computer science, vol 6806. Springer, Berlin, pp 208–224

16. Brázdil T, Kiefer S, Kučera A, Hutařová Vařeková I (2010) Runtime analysis of probabilistic programs with unbounded recursion. CoRR abs/1007.1710

17. Brázdil T, Kiefer S, Kučera A, Hutařová Vařeková I (2011) Runtime analysis of probabilistic programs with unbounded recursion. In: Proceedings of ICALP 2011, Part II. Lecture notes in computer science, vol 6756. Springer, Berlin, pp 319–331

18. Brázdil T, Kučera A, Stražovský O (2004) Deciding probabilistic bisimilarity over infinite-state probabilistic systems. In: Proceedings of CONCUR 2004. Lecture notes in computer science, vol 3170. Springer, Berlin, pp 193–208

19. Brázdil T, Kučera A, Stražovský O (2005) On the decidability of temporal properties of probabilistic pushdown automata. In: Proceedings of STACS 2005. Lecture notes in computer science, vol 3404. Springer, Berlin, pp 145–157

20. Canny J (1988) Some algebraic and geometric computations in PSPACE. In: Proceedings of STOC'88. ACM Press, New York, pp 460–467

21. Chung K (1967) Markov chains with stationary transition probabilities. Springer, Berlin

22. Emerson E (1991) Temporal and modal logic. In: Handbook of theoretical computer science B, pp 995–1072

23. Esparza J, Gaiser A, Kiefer S (2010) Computing least fixed points of probabilistic systems of polynomials. In: Proceedings of STACS 2010. Leibniz international proceedings in informatics, vol 5. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, pp 359–370

24. Esparza J, Hansel D, Rossmanith P, Schwoon S (2000) Efficient algorithms for model checking pushdown systems. In: Proceedings of CAV 2000. Lecture notes in computer science, vol 1855. Springer, Berlin, pp 232–247

25. Esparza J, Kučera A, Mayr R (2004) Model-checking probabilistic pushdown automata. In: Proceedings of LICS 2004. IEEE Computer Society, Los Alamitos, pp 12–21

26. Esparza J, Kučera A, Mayr R (2005) Quantitative analysis of probabilistic pushdown automata: Expectations and variances. In: Proceedings of LICS 2005. IEEE Computer Society, Los Alamitos, pp 117–126

27. Etessami K, Stewart A, Yannakakis M (2012) Polynomial time algorithms for multi-type branching processes and stochastic context-free grammars. In: Proceedings of STOC 2012. ACM Press, New York, pp 579–588

28. Etessami K, Wojtczak D, Yannakakis M (2008) Quasi-birth-death processes, tree-like QBDs, probabilistic 1-counter automata, and pushdown systems. In: Proceedings of 5th Int. Conf. on Quantitative Evaluation of Systems (QEST'08). IEEE Computer Society, Los Alamitos

29. Etessami K, Yannakakis M (2005) Algorithmic verification of recursive probabilistic systems. In: Proceedings of TACAS 2005. Lecture notes in computer science, vol 3440. Springer, Berlin, pp 253–270

30. Etessami K, Yannakakis M (2005) Checking LTL properties of recursive Markov chains. In: Proceedings of 2nd int conf on quantitative evaluation of systems (QEST'05). IEEE Computer Society, Los Alamitos, pp 155–165

31. Etessami K, Yannakakis M (2005) Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. In: Proceedings of STACS 2005. Lecture notes in computer science, vol 3404. Springer, Berlin, pp 340–352

32. Etessami K, Yannakakis M (2009) Recursive Markov chains, stochastic grammars, and monotone systems of nonlinear equations. J Assoc Comput Mach 56:1–66
33. Grigoriev D (1988) Complexity of deciding Tarski algebra. J Symb Comput 5(1–2):65–108
34. Hansson H, Jonsson B (1994) A logic for reasoning about time and reliability. Form Asp Comput 6:512–535
35. Harris T (1963) The theory of branching processes. Springer, Berlin
36. Hopcroft J, Ullman J (1979) Introduction to automata theory, languages, and computation. Addison-Wesley, Reading
37. Kiefer S, Luttenberger M, Esparza J (2007) On the convergence of Newton's method for monotone systems of polynomial equations. In: Proceedings of STOC 2007. ACM Press, New York, pp 217–226
38. Mayr R (1998) Strict lower bounds for model checking BPA. In: Electronic notes in theoretical computer science, vol 18
39. Rosenthal J (2006) A first look at rigorous probability theory. World Scientific, Singapore
40. Tarski A (1951) A decision method for elementary algebra and geometry. Univ. of California Press, Berkeley
41. Tarski A (1955) A lattice-theoretical fixpoint theorem and its applications. Pac J Math 5(2):285–309
42. Walukiewicz I (2000) Model checking CTL properties of pushdown systems. In: Proceedings of FST&TCS'2000. Lecture notes in computer science, vol 1974. Springer, Berlin, pp 127–138
43. Williams D (1991) Probability with Martingales. Cambridge University Press, Cambridge