.

MODEL CHECKING PROBABILISTIC PUSHDOWN AUTOMATA

JAVIER ESPARZA, ANTONÍN KUČERA, AND RICHARD MAYR

INSTITUTE FOR FORMAL METHODS IN COMPUTER SCIENCE, UNIVERSITY OF STUTTGART, UNIVERSITÄT STR. 38, 70569 STUTTGART, GERMANY.
 E-mail address: esparza@informatik.uni-stuttgart.de

FACULTY OF INFORMATICS, MASARYK UNIVERSITY, BOTANICKÁ 68A, CZ-60200 BRNO, CZECH REPUBLIC
 E-mail address: tony@fi.muni.cz

DEPARTMENT OF COMPUTER SCIENCE, NORTH CAROLINA STATE UNIVERSITY, 900 MAIN CAMPUS DRIVE, CAMPUS BOX 8207, RALEIGH NC 27695, USA.
 E-mail address: mayr@csc.ncsu.edu

ABSTRACT. We consider the model checking problem for probabilistic pushdown automata (pPDA) and properties expressible in various probabilistic logics. We start with properties that can be formulated as instances of a generalized random walk problem. We prove that both qualitative and quantitative model checking for this class of properties and pPDA is decidable. Then we show that model checking for the qualitative fragment of the logic PCTL and pPDA is also decidable. Moreover, we develop an error-tolerant model checking algorithm for PCTL and the subclass of stateless pPDA. Finally, we consider the class of ω-regular properties and show that both qualitative and quantitative model checking for pPDA is decidable.

1. INTRODUCTION

Probabilistic systems can be used for modeling systems that exhibit uncertainty, such as communication protocols over unreliable channels, randomized distributed systems, or fault-tolerant systems. Finite-state models of such systems often use variants of probabilistic automata whose underlying semantics is defined in terms of homogeneous Markov chains, which are also called "fully probabilistic transition systems" in this context. For fully probabilistic finite-state systems, algorithms for various (probabilistic) temporal logics like LTL, PCTL, PCTL*, probabilistic μ-calculus, etc., have been presented in [LS82, HS84, Var85, CY88, HJ94, ASB⁺95, CY95, HK97, CSS03]. As for infinite-state systems, most works so far considered probabilistic lossy channel systems [IN97] which model asynchronous

communication through unreliable channels [BE99, ABIJ05, AR03, BS03]. A notable recent result is the decidability of quantitative model checking of liveness properties specified by Büchi-automata for probabilistic lossy channel systems [Rab03]. In fact, this algorithm is *error tolerant* in the sense that the quantitative model checking is solved only up to an arbitrarily small (but non-zero) given error.

In this paper we consider *probabilistic pushdown automata (pPDA)*, which are a natural model for probabilistic sequential programs with possibly recursive procedure calls. There is a large number of results about model checking of non-probabilistic PDA or similar models (see for instance [AEY01, BS97, EHRS00, Wal01]), but the probabilistic extension has so far not been considered. As a related work we can mention [MO98], where it is shown that a restricted subclass of pPDA (where essentially all probabilities for outgoing arcs are either 1 or 1/2) generates a richer class of languages than non-deterministic PDA. Another work [AMP99] shows the equivalence of pPDA and probabilistic context-free grammars. There are also recent results of [BKS05, EY05, EY] which are directly related to the results presented in this paper. A detailed discussion is postponed to Section 6.

Here we consider model checking problems for pPDA and its natural subclass of *stateless pPDA* denoted pBPA[1] and various probabilistic logics. We start with a class of properties
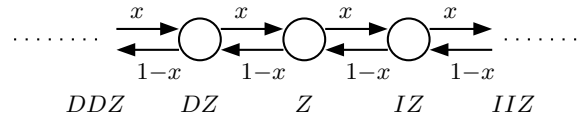

Figure 1: Bernoulli random walk as a pBPA

that can be specified as a generalized *random walk problem*. To get a better intuition about this class of problems, realize that some random walks can easily be specified by pBPA systems. For example, consider a pBPA with just three *stack symbols* Z, I, D and *transitions* $Z \xrightarrow{x} IZ$, $Z \xrightarrow{1-x} DZ$, $I \xrightarrow{x} II$, $I \xrightarrow{1-x} \varepsilon$, $D \xrightarrow{1-x} DD$, and $D \xrightarrow{x} \varepsilon$, where $x \in [0, 1]$ and ε denotes the empty string. A transition $X \xrightarrow{x} w$ means that if the current top stack symbol is X, then it can be replaced by w with probability x. The transition graph of this pBPA with Z as initial stack content (see Fig. 1) is the well-known *Bernoulli walk*. A typical question examined in the theory of random walks is "Do we eventually revisit a given state (with probability one)?", or more generally "What is the probability of reaching a given state from another given state?" For example, it is a standard result that the state Z of Fig. 1 is revisited with probability 1 iff $x = 1/2$. This simple example indicates that answers to qualitative questions about pPDA (i.e., whether something holds with probability 1 or 0) depend on the exact probabilities of individual transitions. This is different from finite-state systems where qualitative properties depend only on the topology of a given finite-state Markov chain [HJ94].

The generalized random walk problem is formulated as follows: Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be subsets of the set of states of a given Markov chain, and let s be a state of $\mathcal{C}_1$. What is the probability that a run initiated in s hits a state of $\mathcal{C}_2$ via a path leading only through the states of $\mathcal{C}_1$? Let us denote this probability by $\mathcal{P}(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$. The problem of computing $\mathcal{P}(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ has previously been considered (and solved) for finite-state systems, where this probability can be computed precisely [HJ94, CY95]. In Section 3, we propose a solution for pPDA

---

[1]This is a standard notation adopted in concurrency theory. The subclass of stateless PDA corresponds to a natural subclass of ACP known as Basic Process Algebra [BW90].

applicable to those sets $\mathcal{C}_1, \mathcal{C}_2$ which are *regular*, i.e., recognizable by finite-state automata (realize that pPDA configurations can be written as words of the form $p\alpha$, where p is a control state and α a sequence of stack symbols). More precisely, we show that the problem whether $\mathcal{P}(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \sim \varrho$, where $\sim \in \{\leq, <, \geq, >, =\}$ and $\varrho \in [0, 1]$, is decidable. Interestingly, this is achieved without explicitly computing the probability $\mathcal{P}(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$. Moreover, for an arbitrary precision $0 < \lambda < 1$ we can compute rational lower and upper approximations $\mathcal{P}^\ell, \mathcal{P}^u \in [0, 1]$ such that $\mathcal{P}^\ell \leq \mathcal{P}(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \leq \mathcal{P}^u$ and $\mathcal{P}^u - \mathcal{P}^\ell \leq \lambda$.

In Section 4, we consider the model checking problem for pPDA and the logic PCTL. This is a more general problem than the one about random walks (the class of properties expressible in PCTL is strictly larger). In Section 4.1, we give a model checking algorithm for the *qualitative fragment* of PCTL and pPDA processes. For general PCTL formulas and pBPA processes, an *error tolerant* model checking algorithm is developed in Section 4.2. The question whether this result can be extended to pPDA is left open.

Finally, in Section 5 we prove that both qualitative and quantitative model checking for the class of ω-regular properties is decidable for pPDA. In [EKM04], it was shown that the qualitative and quantitative model-checking problem is decidable for pPDA and a subclass of ω-regular properties that are definable by deterministic Büchi automata. Later, it has been observed in [BKS05] that the technique can easily be generalized to Muller automata, and thus the decidability result was extended to all ω-regular properties (in [BKS05], some complexity results were also presented). The construction presented in this paper is a slightly generalized and polished version of the algorithms given in [EKM04, BKS05], which can now be seen as instances of a more abstract result.

In Section 6 we conclude by remarks on open problems and recent related work of [BKS05, EY05, EY].

2. Preliminary Definitions

Definition 2.1. A *probabilistic transition system* is a triple $\mathcal{T} = (S, \rightarrow, Prob)$ where S is a finite or countably infinite set of *states*, $\rightarrow \subseteq S \times S$ is a *transition relation*, and $Prob$ is a function which to each transition $s \rightarrow t$ of $\mathcal{T}$ assigns its probability $Prob(s \rightarrow t) \in (0, 1]$ so that for every $s \in S$ we have

$$\sum_{s \rightarrow t} Prob(s \rightarrow t) \in \{0, 1\}$$

The sum above is 0 iff s does not have any outgoing transitions.

In the rest of this paper we also write $s \xrightarrow{x} t$ instead of $Prob(s \rightarrow t) = x$. A *path* in $\mathcal{T}$ is a finite or infinite sequence $w = s_0; s_1; \cdots$ of states such that $s_i \rightarrow s_{i+1}$ for every i. We also use $w(i)$ to denote the state s_i of w (by writing $w(i) = s$ we implicitly impose the condition that the length of w is at least $i + 1$). A *run* is a maximal path, i.e., a path which cannot be prolonged. The sets of all finite paths, all runs, and all infinite runs of $\mathcal{T}$ are denoted *FPath*, *Run*, and *IRun*, respectively[2]. Similarly, the sets of all finite paths, runs, and infinite runs that start in a given $s \in S$ are denoted *FPath(s)*, *Run(s)*, and *IRun(s)*, respectively.

Each $w \in FPath$ determines a *basic cylinder Run(w)* which consists of all runs that start with w. To every $s \in S$ we associate the probabilistic space $(Run(s), \mathcal{F}, \mathcal{P})$ where $\mathcal{F}$ is the σ-field generated by all basic cylinders $Run(w)$ such that w starts with s, and

---

[2]In this paper, $\mathcal{T}$ is always clear from the context.

$\mathcal{P} : \mathcal{F} \to [0,1]$ is the unique probability function such that $\mathcal{P}(Run(w)) = \Pi_{i=0}^{m-1} x_i$ where $w = s_0; \cdots ; s_m$ and $s_i \xrightarrow{x_i} s_{i+1}$ for every $0 \leq i < m$ (if $m = 0$, we put $\mathcal{P}(Run(w)) = 1$).

2.1. **The Logic PCTL.** PCTL, the probabilistic extension of CTL, was defined in [HJ94]. Let $Ap = \{a, b, c, \dots\}$ be a countably infinite set of *atomic propositions*. The syntax of PCTL[3] is given by the following abstract syntax equation:

$$\varphi \quad ::= \quad \mathtt{tt} \mid a \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathcal{X}^{\sim\varrho}\varphi \mid \varphi_1 \mathcal{U}^{\sim\varrho}\varphi_2$$

Here a ranges over Ap, $\varrho \in [0,1]$, and $\sim \in \{\leq, <, \geq, >\}$. Let $\mathcal{T} = (S, \to, Prob)$ be a probabilistic transition system. For all $s \in S$, all $\mathcal{C}, \mathcal{C}_1, \mathcal{C}_2 \subseteq S$, and all $k \in \mathbb{N}_0$, let

- $Run(s, \mathcal{X}\mathcal{C}) = \{w \in Run(s) \mid w(1) \in \mathcal{C}\}$
- $Run(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) = \{w \in Run(s) \mid \exists i \geq 0 : w(i) \in \mathcal{C}_2 \text{ and } w(j) \in \mathcal{C}_1 \text{ for all } 0 \leq j < i\}$
- $FPath^k(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) = \{s_0; \cdots; s_\ell \in FPath(s) \mid 0 \leq \ell \leq k, s_\ell \in \mathcal{C}_2 \text{ and } s_j \in \mathcal{C}_1 \setminus \mathcal{C}_2 \text{ for all } 0 \leq j < \ell\}$
- $FPath(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) = \bigcup_{k=0}^{\infty} FPath^k(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$

The set $Run(s, \mathcal{X}\mathcal{C})$ is clearly $\mathcal{P}$-measurable, and the same holds for $Run(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ because

$$\mathcal{P}(Run(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)) = \sum_{w \in FPath(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)} \mathcal{P}(Run(w)).$$

In the rest of this paper, we usually write $\mathcal{P}(s, \mathcal{X}\mathcal{C})$ and $\mathcal{P}(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ instead of $\mathcal{P}(Run(s, \mathcal{X}\mathcal{C}))$ and $\mathcal{P}(Run(s, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2))$, respectively.

Let $\nu : Ap \to 2^S$ be a *valuation*. The denotation of a PCTL formula φ over $\mathcal{T}$ w.r.t. ν, denoted $[\![\varphi]\!]^\nu$, is defined inductively as follows:

$$\begin{aligned}
[\![\mathtt{tt}]\!]^\nu &= S \\
[\![a]\!]^\nu &= \nu(a) \\
[\![\neg\varphi]\!]^\nu &= S \setminus [\![\varphi]\!]^\nu \\
[\![\varphi_1 \wedge \varphi_2]\!]^\nu &= [\![\varphi_1]\!]^\nu \cap [\![\varphi_2]\!]^\nu \\
[\![\mathcal{X}^{\sim\varrho}\varphi]\!]^\nu &= \{s \in S \mid \mathcal{P}(s, \mathcal{X}[\![\varphi]\!]^\nu) \sim \varrho\} \\
[\![\varphi_1 \mathcal{U}^{\sim\varrho}\varphi_2]\!]^\nu &= \{s \in S \mid \mathcal{P}(s, [\![\varphi_1]\!]^\nu \mathcal{U} [\![\varphi_2]\!]^\nu) \sim \varrho\}
\end{aligned}$$

As usual, we write $s \models^\nu \varphi$ instead of $s \in [\![\varphi]\!]^\nu$.

The *qualitative fragment* of PCTL is obtained by restricting the allowed operator/number combinations to '≤ 0' and '≥ 1', which will be also written as '$= 0$' and '$= 1$', resp. (Observe that '< 1', '> 0' are definable from '≤ 0', '≥ 1', and negation; for example, $a\mathcal{U}^{<1}b \equiv \neg(a\mathcal{U}^{\geq 1}b)$.)

2.2. **Probabilistic PDA.**

Definition 2.2. A *probabilistic pushdown automaton (pPDA)* is a tuple $\Delta = (Q, \Gamma, \delta, Prob)$ where Q is a finite set of *control states*, Γ is a finite *stack alphabet*, $\delta \subseteq Q \times \Gamma \times Q \times \Gamma^*$ is a finite *transition relation* (we write $pX \to q\alpha$ instead of $(p, X, q, \alpha) \in \delta$), and $Prob$ is a function which to each transition $pX \to q\alpha$ assigns its probability $Prob(pX \to q\alpha) \in (0, 1]$ and satisfies $\sum_{pX \to q\alpha} Prob(pX \to q\alpha) \in \{0, 1\}$ for all $p \in Q$ and $X \in \Gamma$.

A *pBPA* is a pPDA with just one control state. Formally, a pBPA is understood as a triple $\Delta = (\Gamma, \delta, Prob)$ where $\delta \subseteq \Gamma \times \Gamma^*$.

---

[3]For simplicity we omit the bounded 'until' operator of [HJ94].

In the rest of this paper we adopt a more intuitive notation, writing $pX \xrightarrow{x} q\alpha$ instead of $Prob(pX \to q\alpha) = x$. A *configuration* of Δ is an element of $Q \times \Gamma^*$. The set of all configurations of Δ is denoted by $\mathcal{C}(\Delta)$. We also assume (w.l.o.g.) that if $pX \to q\alpha \in \delta$, then $|\alpha| \leq 2$. It is easy to transform an arbitrary pair (Δ, F), where Δ is a pPDA and F is a a PCTL formula or ω-property, into another pair (Δ', F') such that Δ' satisfies the assumption above and Δ satisfies F if and only if Δ' satisfies F'. Moreover, the transformation takes linear time. For instance, a transition rule $pX \xrightarrow{x} qYZW$ of Δ is transformed into two transitions $pX \xrightarrow{x} p'Y'W$ and $p'Y' \xrightarrow{1} qYZ$ in Δ', where p', Y' are a fresh control state and a fresh stack symbol, respectively.

To Δ we associate the probabilistic transition system $\mathcal{T}_\Delta$ where $\mathcal{C}(\Delta)$ is the set of states and the probabilistic transition relation is determined as follows: $pX\beta \xrightarrow{x} q\alpha\beta$ is a transition of $\mathcal{T}_\Delta$ iff $pX \xrightarrow{x} q\alpha$ is a transition of Δ and $\beta \in \Gamma^*$.

The model checking problem for pPDA configurations and PCTL formulate (i.e., the question whether $p\alpha \models^\nu \varphi$ for given $p\alpha$, φ, and ν) is clearly undecidable for general valuations. Therefore, we restrict ourselves to *regular* valuations which to every $a \in Ap$ assign a *regular set of configurations*:

Definition 2.3. A Δ-*automaton* is a triple $\mathcal{A} = (St, \gamma, Acc)$ where St is a finite set of *states* s.t. $Q \subseteq St$, $\gamma : St \times \Gamma \to St$ is a (total) *transition function*, and $Acc \subseteq St$ a set of *accepting states*.

The function γ is extended to the elements of Γ^* in the standard way. Each Δ-automaton $\mathcal{A}$ determines a set $\mathcal{C}(\mathcal{A}) \subseteq \mathcal{C}(\Delta)$ given by $p\alpha \in \mathcal{C}(\mathcal{A})$ iff $\gamma(p, \alpha^R) \in Acc$. Here α^R is the reverse of α, i.e., the word obtained by reading α from right to left.

We say that a set $\mathcal{C} \subseteq \mathcal{C}(\Delta)$ is *regular* iff there is a Δ-automaton $\mathcal{A}$ such that $\mathcal{C} = \mathcal{C}(\mathcal{A})$.

In other words, regular sets of configurations are recognizable by finite-state automata which read the stack bottom-up (the bottom-up direction was chosen just for technical convenience).

An important technical step is that one can reduce the model-checking problem for regular valuations to the problem for *simple* valuations that assign to each atomic proposition a *simple* set of configurations. Loosely speaking, a set of configurations is simple if we can decide whether a configuration belongs to the set by inspecting only its control state and its top stack symbol.

Definition 2.4. A set of configurations $\mathcal{C} \subseteq \mathcal{C}(\Delta)$ is *simple* if there is a set $G \subseteq Q \times (\Gamma \cup \{\varepsilon\})$ such that for each $p\alpha \in \mathcal{C}(\Delta)$ we have that $p\alpha \in \mathcal{C}$ iff either $\alpha = \varepsilon$ and $p\varepsilon \in G$, or $\alpha = X\beta$ and $pX \in G$.

The reason why we only need to consider simple valuations is a bisimilarity property. Let $\mathcal{C}_1, \cdots, \mathcal{C}_k \subseteq \mathcal{C}(\Delta)$ be regular sets of configurations, and assume that all we can observe from a configuration is whether it belongs to $\mathcal{C}_i$ for every $1 \leq i \leq k$. Loosely speaking, Lemma 2.5 below states that we can effectively construct another pPDA Δ' and *simple* sets of configurations $\mathcal{C}'_1, \cdots, \mathcal{C}'_k \subseteq \mathcal{C}(\Delta)$ such that Δ and Δ' are bisimilar with respect to these observables (in the usual definition of bisimilarity one observes transitions between configurations, while here we observe the configurations themselves, but otherwise the notion is the same). The idea of the construction is to take Δ-automata $\mathcal{A}_1, \cdots, \mathcal{A}_k$ accepting the sets $\mathcal{C}_1, \cdots, \mathcal{C}_k$, and construct Δ' such that the following holds: If the current configuration of Δ is $p\alpha$, then in the simulating configuration of Δ' the topmost stack symbol stores the states reached by the Δ-automata after reading α^R from the initial state p. Although this

construction is standard (see, e.g., [EKS03]), we include an explicit proof for the sake of completeness.

Lemma 2.5. *For each pPDA $\Delta = (Q, \Gamma, \delta, Prob)$ and regular sets $\mathcal{C}_1, \cdots, \mathcal{C}_k \subseteq \mathcal{C}(\Delta)$ there effectively exists a pPDA $\Delta' = (Q, \Gamma', \delta', Prob')$, simple sets $\mathcal{C}'_1, \cdots, \mathcal{C}'_k \subseteq \mathcal{C}(\Delta')$, and an injective mapping $\mathcal{G} : \mathcal{C}(\Delta) \to \mathcal{C}(\Delta')$ such that for each $p\alpha \in \mathcal{C}(\Delta)$ the following conditions are satisfied:*

- *for each $1 \leq j \leq k$ we have $p\alpha \in \mathcal{C}_j$ iff $\mathcal{G}(p\alpha) \in \mathcal{C}'_j$;*
- *if $p\alpha \xrightarrow{x} q\beta$, then $\mathcal{G}(p\alpha) \xrightarrow{x} \mathcal{G}(q\beta)$;*
- *if $\mathcal{G}(p\alpha) \xrightarrow{x} s$ for some $s \in \mathcal{C}(\Delta')$, then there is $p\alpha \xrightarrow{x} q\beta$ such that $\mathcal{G}(q\beta) = s$.*

Moreover, if $\mathcal{C} \subseteq \mathcal{C}(\Delta')$ is regular, then $\mathcal{G}^{-1}(\mathcal{C})$ is also regular.

Proof. For each $1 \leq i \leq k$, let $\mathcal{A}_i = (St_i, \gamma_i, Acc_i)$ be a Δ-automaton such that $\mathcal{C}(\mathcal{A}_i) = \mathcal{C}_i$. Let $States = \prod_{i=1}^{k} \prod_{p \in Q} St_i$. For given $\vec{s} \in States$, $1 \leq i \leq k$, and $p \in Q$, we denote by $\vec{s}(i, p)$ the component of $\vec{s}$ which corresponds to i and p.

We put $\Gamma' = \Gamma \times States$. The transition function δ' and probabilities $Prob'$ are defined as follows:

- if $pX \xrightarrow{x} q\varepsilon \in \delta$, then $p(X, \vec{s}) \xrightarrow{x} q\varepsilon$ for each $\vec{s} \in States$;
- if $pX \xrightarrow{x} qY \in \delta$, then $p(X, \vec{s}) \xrightarrow{x} q(Y, \vec{s})$ for each $\vec{s} \in States$;
- if $pX \xrightarrow{x} qYZ \in \delta$, then $p(X, \vec{s}) \xrightarrow{x} q(Y, \vec{t})(Z, \vec{s})$ for all $\vec{s}, \vec{t} \in States$ such that $\gamma_i(\vec{s}(i, r), Z) = \vec{t}(i, r)$ for all $1 \leq i \leq k$ and $r \in Q$.

So, the Δ-automata $\mathcal{A}_1, \cdots, \mathcal{A}_k$ are simulated "on-the-fly" by storing the vector of current states directly in the stack. Hence, the information whether a given $\mathcal{A}_i$ accepts the current configuration is available in the topmost stack symbol. For every $1 \leq i \leq k$, the underlying set G_i of $\mathcal{C}'_i$ (see Definition 2.4) is defined by

$$G_i = \{p(X, \vec{s}) \mid \gamma_i(\vec{s}(i, p), X) \in Acc_i\} \cup \{p\varepsilon \mid p\varepsilon \in \mathcal{C}_i\}$$

The function $\mathcal{G}$ is defined by $\mathcal{G}(p\varepsilon) = p\varepsilon$, and $\mathcal{G}(pX_1 \cdots X_k) = p(X_1, \vec{s}_1) \cdots (X_k, \vec{s}_k)$, where $\vec{s}_k(i, q) = q$, and $\vec{s}_j(i, q) = \gamma_i(\vec{s}_{j+1}(i, q), X_{j+1})$ for all $1 \leq j < k$. It follows immediately from the definition of δ' and $Prob'$ that the parts of $\mathcal{T}_\Delta$ and $\mathcal{T}_{\Delta'}$ which are reachable from $p\alpha$ and $\mathcal{G}(p\alpha)$ are isomorphic (for every $p\alpha \in \mathcal{C}(\Delta)$).

Let $\mathcal{C} \subseteq \mathcal{C}(\Delta')$ be a regular set of configurations. Since some configurations of $\mathcal{C}$ can be "inconsistent" in the sense that the vectors of states that are stored together with the original stack symbols do not correspond to a valid computation of the $\mathcal{A}_i$ automata, the set $\mathcal{G}^{-1}(\mathcal{C})$ is not a simple projection of $\mathcal{C}$ "forgetting" the vectors of states from the stack symbols. Fortunately, $\mathcal{G}(\mathcal{C}(\Delta))$ is (obviously) a regular set, so we can construct a Δ'-automaton recognizing the set $\mathcal{C} \cap \mathcal{G}(\mathcal{C}(\Delta))$ and apply the mentioned projection. $\square$

3. Random Walks on pPDA Graphs

In this section we address the following problem. Let Δ be a pPDA, let $p_1\alpha_1$ be an initial configuration, let $\mathcal{C}_1, \mathcal{C}_2$ be two simple sets of configurations, and let ρ be a threshold probability. Is the probability of executing a run $p_1\alpha_1; p_2\alpha_2; p_3\alpha_3 \cdots$ that satisfies $\mathcal{C}_1 \mathcal{U} \mathcal{C}_2$, denoted by $\mathcal{P}(p_1\alpha_1, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$, at least ρ? We show that the problem is decidable.

The plan of the section is as follows. First, we show in Lemma 3.4 that $\mathcal{P}(p_1\alpha_1, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ is equal to a polynomial expression in the following probabilities:

- Let pX be an initial configuration (notice that there is only one symbol on the stack), and let q be a control state q. The probability of reaching $q\varepsilon$ visiting only configurations of $\mathcal{C}_1 \smallsetminus \mathcal{C}_2$ along the way is denoted by $[pXq]$
- Let pX be an initial configuration and let τ be a threshold probability. The probability of reaching some configuration of $\mathcal{C}_2$ with nonempty stack, visiting only configurations of $\mathcal{C}_1$ along the way, is denoted by $[pX\bullet]$.

Second, in Theorem 3.5, we show that the probabilities $[pXq]$ and $[pX\bullet]$ are the least solution of a system of quadratic equations. So our original problem reduces to determining whether a polynomial expression on this least solution has at least the value ρ. Finally, we observe in Theorem 3.7 that this question can be reduced to deciding the truth of a formula in the first-order arithmetic of the reals (i.e., in the theory $(\mathbb{R}, +, *, \leq)$). Since this theory is known to be decidable [Tar51], our original question is decidable.

For the rest of this section, let us fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$ and two simple sets $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}(\Delta)$. Let $G_1, G_2 \subseteq Q \times (\Gamma \cup \{\varepsilon\})$ be the sets associated to $\mathcal{C}_1, \mathcal{C}_2$ in the sense of Definition 2.4.

Definition 3.1. To simplify our notation, we adopt the following conventions:

- For each $\mathcal{C} \subseteq \mathcal{C}(\Delta)$, let $\mathcal{C}^\bullet = \mathcal{C} \smallsetminus (Q \times \{\varepsilon\})$. Observe that if $\mathcal{C}$ is simple, then so is $\mathcal{C}^\bullet$.
- For every $\mathcal{C} \subseteq \mathcal{C}(\Delta)$ and every $\beta \in \Gamma^*$, the symbol $\mathcal{C}\beta$ denotes the set $\{p\alpha\beta \mid p\alpha \in \mathcal{C}\}$.
- For all $p, q \in Q$ and $X \in \Gamma$, we use $[pXq]$ to abbreviate $\mathcal{P}(pX, \mathcal{C}_1 \smallsetminus \mathcal{C}_2 \,\mathcal{U}\, \{q\varepsilon\})$, and $[pX\bullet]$ to abbreviate $\mathcal{P}(pX, \mathcal{C}_1 \,\mathcal{U}\, \mathcal{C}_2^\bullet)$.
- Let A be a set of finite paths which end in the same state t, and let B a set of finite or infinite paths that start in t. Then the symbol $A \odot B$ denotes the set of paths $\{v; w \mid v \in A, t; w \in B\}$.

The proof of Lemma 3.4. our first milestone, requires the following two auxiliary results:

Lemma 3.2. *Let $\mathcal{T} = (S, \rightarrow, Prob)$ be a probabilistic transition system. Let $s, t \in S$ and $\mathcal{C}_1, \mathcal{C}_2 \subseteq S$. Further, let $A = FPath(s, (\mathcal{C}_1 \smallsetminus \mathcal{C}_2) \,\mathcal{U}\, \{t\})$ and $B = FPath(t, \mathcal{C}_1 \,\mathcal{U}\, \mathcal{C}_2)$. Then*

$$\sum_{w \in A \odot B} \mathcal{P}(Run(w)) = \sum_{w \in A} \mathcal{P}(Run(w)) \cdot \sum_{w \in B} \mathcal{P}(Run(w)).$$

Proof. Immediate. $\qquad\square$

Lemma 3.3. *For all $p\alpha \in \mathcal{C}(\Delta)$ and $\beta \in \Gamma^*$ we have that $\mathcal{P}(p\alpha, \mathcal{C}_1 \,\mathcal{U}\, \mathcal{C}_2)$ is equal to $\mathcal{P}(p\alpha\beta, \mathcal{C}_1^\bullet \beta \,\mathcal{U}\, \mathcal{C}_2\beta)$.*

Proof. For every finite path $w = p_1\alpha_1; \cdots ; p_n\alpha_n$ of $FPath(p\alpha)$, let $w^{+\beta}$ denote the finite path $p_1\alpha_1\beta; \cdots ; p_n\alpha_n\beta$ of $FPath(p\alpha\beta)$. Realize that $\mathcal{P}(Run(w)) = \mathcal{P}(Run(w^{+\beta}))$, because w and $w^{+\beta}$ execute the same transitions. One can easily verify that $w \in FPath(p\alpha, \mathcal{C}_1 \,\mathcal{U}\, \mathcal{C}_2)$ iff $w^{+\beta} \in FPath(p\alpha\beta, \mathcal{C}_1^\bullet \beta \,\mathcal{U}\, \mathcal{C}_2\beta)$. From this we get

$$
\begin{aligned}
\mathcal{P}(p\alpha, \mathcal{C}_1 \,\mathcal{U}\, \mathcal{C}_2) &= \sum_{w \in FPath(p\alpha, \mathcal{C}_1 \,\mathcal{U}\, \mathcal{C}_2)} \mathcal{P}(Run(w)) \\
&= \sum_{w \in FPath(p\alpha\beta, \mathcal{C}_1^\bullet \beta \,\mathcal{U}\, \mathcal{C}_2\beta)} \mathcal{P}(Run(w)) \\
&= \mathcal{P}(p\alpha\beta, \mathcal{C}_1^\bullet \beta \,\mathcal{U}\, \mathcal{C}_2\beta)
\end{aligned}
$$

$\square$

Now we show how to compute $\mathcal{P}(pX_1 \cdots X_n, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ from the finite family of all $[pXq]$, $[pX\bullet]$ probabilities. First, realize that

$$\mathcal{P}(pX_1 \cdots X_n, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \;\; = \;\; [pX_1\bullet] + \sum_{q \in Q} [pX_1 q] \cdot \mathcal{P}(qX_2 \cdots X_n, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$$

The meaning of this equation is intuitively clear. If we repeatedly expand the probabilities of the form $\mathcal{P}(qX_j \cdots X_n, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ in the above equation (until j becomes n), we obtain the equation presented in the following lemma:

Lemma 3.4. *For each $pX_1 \cdots X_n \in \mathcal{C}(\Delta)$ where $n \geq 0$ we have that $\mathcal{P}(pX_1 \cdots X_n, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ is equal to*

$$\sum_{i=1}^{n} \sum_{\substack{(q_1, \cdots, q_i) \in Q^i \\ where\ p=q_1}} [q_i X_i \bullet] \cdot \prod_{j=1}^{i-1} [q_j X_j q_{j+1}] \; + \; \sum_{\substack{(q_1, \cdots, q_{n+1}) \in Q^{n+1} \\ where\ p=q_1\ and\ q_{n+1}\varepsilon \in \mathcal{C}_2}} \prod_{j=1}^{n} [q_j X_j q_{j+1}]$$

with the convention that empty sum is equal to 0 and empty product is equal to 1.

Proof. By induction on n. For $n = 0$ we have that $\mathcal{P}(p\varepsilon, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ is equal either to 1 or 0, depending on whether $p\varepsilon$ belongs to $\mathcal{C}_2$ or not, resp. Now let $n \geq 1$, and let β denote the sequence $X_2 \cdots X_n$. The set $Run(pX_1\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ is equal to

$$\biguplus_{w \in FPath(pX_1\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)} Run(w)$$

Let $\mathcal{C}' = \{q\alpha\beta \mid q \in Q, \alpha \in \Gamma^+\}$. We have that

$$FPath(pX_1\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \; = \; FPath(pX_1\beta, \mathcal{C}_1 \cap \mathcal{C}' \mathcal{U} \mathcal{C}_2 \cap \mathcal{C}') \; \uplus$$

$$\biguplus_{q \in Q} FPath(pX_1\beta, (\mathcal{C}_1 \smallsetminus \mathcal{C}_2) \cap \mathcal{C}' \mathcal{U} \{q\beta\}) \odot FPath(q\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$$

Now observe that for every *simple* set $\mathcal{C} \subseteq \mathcal{C}(\Delta)$ we have that $\mathcal{C} \cap \mathcal{C}' = \mathcal{C}^\bullet \beta$. Hence, the above equation can be rewritten as follows:

$$FPath(pX_1\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \; = \; FPath(pX_1\beta, \mathcal{C}_1^\bullet \beta \mathcal{U} \mathcal{C}_2^\bullet \beta) \; \uplus$$

$$\biguplus_{q \in Q} FPath(pX_1\beta, (\mathcal{C}_1 \smallsetminus \mathcal{C}_2)^\bullet \beta \mathcal{U} \{q\beta\}) \odot FPath(q\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$$

Using Lemma 3.3 and Lemma 3.2, we obtain that

$$\mathcal{P}(pX_1\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \; = \mathcal{P}(pX_1, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2^\bullet) \; +$$

$$\sum_{q \in Q} \mathcal{P}(pX_1\beta, (\mathcal{C}_1 \smallsetminus \mathcal{C}_2) \mathcal{U} \{q\beta\}) \cdot \mathcal{P}(q\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$$

This can also be written as

$$\mathcal{P}(pX_1\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \; = [pX_1\bullet] + \sum_{q \in Q} [pX_1 q] \cdot \mathcal{P}(q\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$$

Now it suffices to apply induction hypothesis to $\mathcal{P}(q\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ and restructure the resulting expression. $\qquad \square$

Now we show that the probabilities $[pXq]$, $[pX\bullet]$ form the least solution of an effectively constructible system of quadratic equations. This can be seen as a generalization of a similar result for finite-state systems [HJ94, CY95]. In the finite-state case, the equations are linear and can be further modified so that they have a *unique* solution (which is then computable, e.g., by Gauss elimination). In the case of pPDA, the equations are not linear and cannot be generally solved by analytical methods. The question whether the equations can be further modified so that they have a unique solution is left open; we just note that the method used for finite-state systems is insufficient (this is demonstrated by Example 3.6).

Let $\mathcal{V} = \{\langle pXq \rangle, \langle pX\bullet \rangle \mid p, q \in Q, X \in \Gamma\}$ be a set of "variables". Let us consider the system of recursive equations constructed as follows:

- if $pX \notin G_1 \smallsetminus G_2$, then $\langle pXq \rangle = 0$ for each $q \in Q$; otherwise, we put

$$\langle pXq \rangle = \sum_{pX \xrightarrow{x} rYZ} x \cdot \sum_{t \in Q} \langle rYt \rangle \cdot \langle tZq \rangle \; + \sum_{pX \xrightarrow{x} rY} x \cdot \langle rYq \rangle \; + \sum_{pX \xrightarrow{x} q\varepsilon} x$$

- if $pX \in G_2$, then $\langle pX\bullet \rangle = 1$; if $pX \notin G_1 \cup G_2$, then $\langle pX\bullet \rangle = 0$; otherwise we put

$$\langle pX\bullet \rangle = \sum_{pX \xrightarrow{x} rYZ} x \cdot (\langle rY\bullet \rangle \; + \sum_{t \in Q} \langle rYt \rangle \cdot \langle tZ\bullet \rangle) \; + \sum_{pX \xrightarrow{x} rY} x \cdot \langle rY\bullet \rangle$$

The intuition behind these equations is easy to understand. For the sake of simplicity, assume $G_1 = Q \times \Gamma$ and $G_2 = \emptyset$ (this corresponds to $\mathcal{C}_1 = \mathcal{C}(\Delta)$ and $\mathcal{C}_2 = \emptyset$). In this case, we only have the two "long" equations. Consider the first one, the intuition for the second one being similar. In order to reach $q\varepsilon$ from pX, the pPDA must make at least one move. Since we assume than the transitions $pX \xrightarrow{x} q\alpha$ of a pPDA satisfy $|\alpha| \leq 2$, here are three possible kinds of moves: moves that increase the stack length by one, moves that do not change the stack length, and moves that decrease the stack length. The three summands in the equations correspond to these three kinds of moves. Since no transition can be executed when the stack is empty, the only way to reach $q\varepsilon$ by means of a length-decreasing move is to apply a transition $pX \xrightarrow{x} q\varepsilon$, if it exists (third summand). If the first transition is length-keeping, i.e., of the form $pX \xrightarrow{x} rY$, then, after the transition, we must reach $q\varepsilon$ from rY (second summand). Finally, if the first transition is of the form $pX \rightarrow rYZ$, then the pPDA must first go from rYZ to some configuration tZ along a path of configurations having with Z as bottom stack symbol, and then from tZ to $q\varepsilon$. Intuitively (see the next theorem for the formal proof), the probability of reaching tZ from rYZ along such a path is equal to the probability of reaching $t\varepsilon$ from rY, and so we get the first summand.

For given $t \in [0,1]^{|\mathcal{V}|}$, $p, q \in Q$, and $X \in \Gamma$ we use $\langle pXq \rangle_t$ and $\langle pX\bullet \rangle_t$ to denote the component of t which corresponds to the variable $\langle pXq \rangle$ and $\langle pX\bullet \rangle$, respectively. The above defined system of equations determines a unique operator $\mathcal{F} : [0,1]^{|\mathcal{V}|} \rightarrow [0,1]^{|\mathcal{V}|}$ where $\mathcal{F}(t)$ is the tuple of values obtained by evaluating the right-hand sides of the equations where all $\langle pXq \rangle$ and $\langle pX\bullet \rangle$ are substituted with $\langle pXq \rangle_t$ and $\langle pX\bullet \rangle_t$, respectively.

Theorem 3.5. *The operator $\mathcal{F}$ has the least fixed-point μ. Moreover, for all $p, q \in Q$ and $X \in \Gamma$ we have that $\langle pXq \rangle_\mu = [pXq]$ and $\langle pX\bullet \rangle_\mu = [pX\bullet]$.*

Proof. Since $\mathcal{F}$ is monotonic and continuous, it has the least fixed point $\mu = \bigvee_{k=0}^{\infty} \mathcal{F}^k(\vec{0})$, where $\vec{0}$ is the tuple of zeros. One can readily check that the tuple π of all $[pXq]$ and $[pX\bullet]$ probabilities forms a solution of the above system; this is done just by partitioning the associated sets of runs into appropriate disjoint subsets similarly as in the proof of

Lemma 3.4. Hence, $\mu \leq \pi$. To prove that also $\pi \leq \mu$, we approximate the $[pXq]$ and $[pX\bullet]$ probabilities in the following way: For each $k \in \mathbb{N}_0$ we define

- $[pXq]^k = \sum\limits_{w \in FPath^k(pX, \mathcal{C}_1 \smallsetminus \mathcal{C}_2 \,\mathcal{U}\, \{q\varepsilon\})} \mathcal{P}(Run(w))$

- $[pX\bullet]^k = \sum\limits_{w \in FPath^k(pX, \mathcal{C}_1 \,\mathcal{U}\, \mathcal{C}_2^\bullet)} \mathcal{P}(Run(w))$

Let π^k be the tuple of all $[pXq]^k$ and $[pX\bullet]^k$ probabilities. Clearly $\pi = \lim_{k\to\infty} \pi^k$. By induction on k we prove that $\pi^k \leq \mu$ for each $k \in \mathbb{N}_0$, hence also $\pi \leq \mu$ as needed.

The base case ($k = 0$) follows immediately. We show that if $[pXq]^k \leq \langle pXq\rangle_\mu$ and $[pX\bullet]^k \leq \langle pXq\rangle_\mu$, then also $[pXq]^{k+1} \leq \langle pXq\rangle_\mu$ and $[pX\bullet]^{k+1} \leq \langle pXq\rangle_\mu$. If $pX \notin G_1 \smallsetminus G_2$, then $[pXq]^{k+1} = \langle pXq\rangle_\mu = 0$. Otherwise, by applying the definitions we obtain

$$
\begin{aligned}
[pXq]^{k+1} \;=\; & \sum_{pX \xrightarrow{x} rYZ} x \cdot \sum_{w \in FPath^k(rYZ, \mathcal{C}_1 \smallsetminus \mathcal{C}_2 \,\mathcal{U}\, \{q\varepsilon\})} \mathcal{P}(Run(w)) \\
& + \sum_{pX \xrightarrow{x} rY} x \cdot \sum_{w \in FPath^k(rY, \mathcal{C}_1 \smallsetminus \mathcal{C}_2 \,\mathcal{U}\, \{q\varepsilon\})} \mathcal{P}(Run(w)) \\
& + \sum_{pX \xrightarrow{x} q\varepsilon} x
\end{aligned}
$$

and

$$
\langle pXq\rangle_\mu \;=\; \sum_{pX \xrightarrow{x} rYZ} x \cdot \sum_{t \in Q} \langle rYt\rangle_\mu \cdot \langle tZq\rangle_\mu \;+\; \sum_{pX \xrightarrow{x} rY} x \cdot \langle rYq\rangle_\mu \;+\; \sum_{pX \xrightarrow{x} q\varepsilon} x
$$

Since

$$
\sum_{w \in FPath^k(rY, \mathcal{C}_1 \smallsetminus \mathcal{C}_2 \,\mathcal{U}\, \{q\varepsilon\})} \mathcal{P}(Run(w)) \;=\; [rYq]^k,
$$

we have

$$
\sum_{w \in FPath^k(rY, \mathcal{C}_1 \smallsetminus \mathcal{C}_2 \,\mathcal{U}\, \{q\varepsilon\})} \mathcal{P}(Run(w)) \;\leq\; \langle rYq\rangle_\mu
$$

by induction hypothesis. Further,

$$
\sum_{pX \xrightarrow{x} rYZ} x \cdot \sum_{w \in FPath^k(rYZ, \mathcal{C}_1 \smallsetminus \mathcal{C}_2 \,\mathcal{U}\, \{q\varepsilon\})} \mathcal{P}(Run(w))
$$

is surely bounded by

$$
\sum_{pX \xrightarrow{x} rYZ} x \cdot \sum_{t \in Q} [rYt]^k \cdot [tZq]^k,
$$

which is bounded by

$$
\sum_{pX \xrightarrow{x} rYZ} x \cdot \sum_{t \in Q} \langle rYt\rangle_\mu \cdot \langle tZq\rangle_\mu
$$

by induction hypothesis. To sum up, we have that $[pXq]^{k+1} \leq \langle pXq\rangle_\mu$. The inequality $[pX\bullet]^{k+1} \leq \langle pX\bullet\rangle_\mu$ is proved similarly. $\qquad \square$

Example 3.6. Let us consider the pBPA system Δ of Fig. 1, and let $\mathcal{C}_1 = \Gamma^*$, $\mathcal{C}_2 = \{Z\}$. Then we obtain the following system of equations (since Δ has only one control state p, we write $\langle X, \bullet \rangle$ and $\langle X, \varepsilon \rangle$ instead of $\langle pX\bullet \rangle$ and $\langle pXp \rangle$, resp.):

$$
\begin{aligned}
\langle Z, \bullet \rangle &= 1 \\
\langle Z, \varepsilon \rangle &= x\langle I, \varepsilon \rangle\langle Z, \varepsilon \rangle + (1-x)\langle D, \varepsilon \rangle\langle Z, \varepsilon \rangle \\
\langle I, \bullet \rangle &= x(\langle I, \bullet \rangle + \langle I, \varepsilon \rangle\langle I, \bullet \rangle) \\
\langle I, \varepsilon \rangle &= x\langle I, \varepsilon \rangle\langle I, \varepsilon \rangle + 1-x \\
\langle D, \bullet \rangle &= (1-x)(\langle D, \bullet \rangle + \langle D, \varepsilon \rangle\langle D, \bullet \rangle) \\
\langle D, \varepsilon \rangle &= (1-x)\langle D, \varepsilon \rangle\langle D, \varepsilon \rangle + x
\end{aligned}
$$

As the least solution we obtain the probabilities $[Z, \bullet] = 1$, $[Z, \varepsilon] = 0$, $[I, \bullet] = 0$, $[I, \varepsilon] = \min\{1, (1-x)/x\}$, $[D, \bullet] = 0$, $[D, \varepsilon] = \min\{1, x/(1-x)\}$. By applying Lemma 3.4 we further obtain that, e.g., $\mathcal{P}(IIZ, \mathcal{C}_1\, \mathcal{U}\, \mathcal{C}_2) = [I, \bullet] + [I, \varepsilon] \cdot ([I, \bullet] + [I, \varepsilon] \cdot [Z, \bullet]) = \min\{1, (1-x)^2/x^2\}$. $\square$

In Example 3.6 it is possible to compute a closed form for the least solution of the system of equations, but in general this is not true. However, many important properties of the least solution are decidable, because the decision problem can be reduced to the problem of deciding the truth of a formula in the first-order theory of the reals. For our purposes, it suffices to consider the class of properties defined in the next theorem.

Theorem 3.7. *Let* $Const = \mathbb{Q} \cup \{[pXq], [pX\bullet] \mid p, q \in Q \text{ and } X \in \Gamma\}$, *where* $\mathbb{Q}$ *is the set of all rational constants. Let* E_1, E_2 *be expressions built over* $Const$ *using '·' and '+', and let* $\sim \in \{<, =\}$. *It is decidable whether* $E_1 \sim E_2$.

Proof. We show that, due to Theorem 3.5, $E_1 \sim E_2$ is effectively expressible as a closed formula of $(\mathbb{R}, +, *, \leq)$. Hence, the theorem follows from the decidability of first-order arithmetic of reals [Tar51].

For all $p, q \in Q$ and $X \in \Gamma$, let $x(pXq)$, $x(pX\bullet)$, $y(pXq)$, and $y(pX\bullet)$ be first order variables, and let $\vec{X}$ and $\vec{Y}$ be the vectors of all $x(pXq)$, $x(pX\bullet)$, and $y(pXq)$, $y(pX\bullet)$ variables, respectively. Let us consider the formula Φ constructed as follows:

$\exists \vec{X} \;:\; \vec{0} \leq \vec{X} \leq \vec{1} \quad \wedge \quad \vec{X} = \mathcal{F}(\vec{X})$

$\quad \wedge \; (\forall \vec{Y} \;:\; (\vec{0} \leq \vec{Y} \leq \vec{1} \;\wedge\; \vec{Y} = \mathcal{F}(\vec{Y})) \;\Rightarrow\; \vec{X} \leq \vec{Y}))$

$\quad \wedge \; E_1[\vec{X}/\pi] \sim E_2[\vec{X}/\pi]$

Observe that the conditions $\vec{X} = \mathcal{F}(\vec{X})$ and $\vec{Y} = \mathcal{F}(\vec{Y})$ are expressible only using multiplication, summation, and equality. The expressions $E_1[\vec{X}/\pi]$ and $E_2[\vec{X}/\pi]$ are obtained from E_1 and E_2 by substituting all $[pXq]$ and $[pX\bullet]$ with $x(pXq)$ and $x(pX\bullet)$, respectively. It follows immediately that $E_1 \sim E_2$ iff Φ holds. $\square$

An immediate consequence of Theorem 3.7 is the following:

Theorem 3.8. *Let* $p\alpha \in \mathcal{C}(\Delta)$, $\varrho \in \mathbb{Q} \cap [0,1]$, $\sim \in \{\leq, <, \geq, >\}$ *and* $0 < \lambda < 1$. *It is decidable whether* $\mathcal{P}(p\alpha, \mathcal{C}_1\, \mathcal{U}\, \mathcal{C}_2) \sim \varrho$. *Moreover, there effectively exist rational numbers* $\mathcal{P}^\ell, \mathcal{P}^u$ *such that* $\mathcal{P}^\ell \leq \mathcal{P}(p\alpha, \mathcal{C}_1\, \mathcal{U}\, \mathcal{C}_2) \leq \mathcal{P}^u$ *and* $\mathcal{P}^u - \mathcal{P}^\ell \leq \lambda$.

Proof. We can assume w.l.o.g. that $\alpha = X$ for some $X \in \Gamma$. Note that $\mathcal{P}(pX, \mathcal{C}_1\, \mathcal{U}\, \mathcal{C}_2) \sim \varrho$ iff $[pX\bullet] + \sum_{q\varepsilon \in \mathcal{C}_2}[pXq] \sim \varrho$ by Lemma 3.4. Hence, we can apply Theorem 3.7. The numbers $\mathcal{P}^\ell, \mathcal{P}^u$ are computable, e.g., by the algorithm of Fig. 2. $\square$

Input: $pX \in \mathcal{C}(\Delta)$, $0 < \lambda < 1$
Output: $\mathcal{P}^\ell$, $\mathcal{P}^u$

```
1:   Pℓ := 0; Pu := 1;
2:   for  i = 1  to  ⌈− log₂ λ⌉
3:       if  [pX•] + ∑_{q∈C₂} [pXq] ≥ (Pu − Pℓ)/2
4:           then  Pℓ := (Pu − Pℓ)/2
5:           else  Pu := (Pu − Pℓ)/2
6:       fi
```

Figure 2: Computing $\mathcal{P}^\ell, \mathcal{P}^u$

4. Model Checking PCTL for pPDAs

In this section we study the model-checking problem for PCTL formulas with regular valuations and pPDA.

4.1. **Qualitative Fragment of PCTL.** We give a model checking algorithm for the qualitative fragment of PCTL, i.e., for the fragment in which only 0 and 1 are allowed as probability thresholds.

Recall that in order to check if a CTL formula φ holds of a finite state system we first recursively compute the sets of states that satisfy the subformulas of φ lying right below φ in the syntax tree, and then we apply a semantic operator that gets these sets of states as inputs and produces the set of states satisfying φ as output. In the case of a PDA (no probabilities), these sets of states (they are now sets of configurations) can be infinite. Therefore, in order to apply a similar algorithm it is necessary to prove that the sets have a finite representation. This was done in [BEM97]: It was shown that in the case of regular valuations the sets are always regular, and so can be finitely represented by, say, finite automata. In this section we prove that the same property also holds for pPDA and for the qualitative fragment of PCTL, and that the constructions showing the regularity of the sets are effective.

By Lemma 2.5, we only need to show that if the sets of configurations satisfying the subformulas of φ are simple, then the set of configurations satisfying φ is regular. We need to consider four cases, corresponding to formulas of the form $\mathcal{X}^{=0}\varphi$, $\mathcal{X}^{=1}\varphi$, $\varphi_1\,\mathcal{U}^{=0}\varphi_2$, and $\varphi_1\,\mathcal{U}^{=1}\varphi_2$. they are dealt with in Lemma 4.1, Lemma 4.2, and Lemma 4.3.

For the rest of this section we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$.

Lemma 4.1. Let $\mathcal{C} \subseteq \mathcal{C}(\Delta)$ be a simple set. The sets $\{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{X}\mathcal{C}) = 1\}$ and $\{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{X}\mathcal{C}) = 0\}$ are effectively regular.

Proof. Follows immediate from the fact that $p\alpha$ has only finitely many successors in the probabilistic transition system associated to Δ.. □

Lemma 4.2. Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}(\Delta)$ be simple sets. The set $\{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) = 1\}$ is effectively regular.

Proof. Let $R(pX) = \{q \in Q \mid [pXq] > 0\}$ for all $p \in Q$, $X \in \Gamma$. For each $i \in \mathbb{N}_0$ we define the set $S_i \subseteq \mathcal{C}(\Delta)$ inductively as follows:

- $S_0 = \{q\varepsilon \mid q\varepsilon \in \mathcal{C}_2\} \cup \{qX\alpha \mid [qX•] = 1, \alpha \in \Gamma^*\}$
- $S_{i+1} = \{pX\beta \mid [pX•] + \sum_{q\in R(pX)}[pXq] = 1 \text{ and } \forall q \in R(pX) : q\beta \in S_i\}$

Using Lemma 3.4, we can easily check that $\bigcup_{i=0}^{\infty} S_i = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) = 1\}$. To see that the set $\bigcup_{i=0}^{\infty} S_i$ is effectively regular, for each $p \in Q$ we construct a finite automaton $\mathcal{M}_p$ such that $L(\mathcal{M}_p) = \{\alpha \in \Gamma^* \mid p\alpha \in \bigcup_{i=0}^{\infty} S_i\}$. A Δ-automaton $\mathcal{A}$ recognizing the set $\bigcup_{i=0}^{\infty} S_i$ can then be constructed using standard algorithms of automata theory (in particular, note that regular languages are effectively closed under reverse). The states of $\mathcal{M}_p$ are all subsets of Q, $\{p\}$ is the initial state, Γ is the input alphabet, the final states are those $T \subseteq Q$ where for every $q \in T$ we have that $q\varepsilon \in \mathcal{C}_2$ (in particular, note that $\emptyset$ is a final state), and the transition function is given by $T \xrightarrow{X} U$ iff for every $q \in T$ we have that $[qX\bullet] + \sum_{r \in R(qX)}[qXr] = 1$ and $U = \bigcup_{q\in T} R(qX)$. Note that $\emptyset \xrightarrow{X} \emptyset$ for each $X \in \Gamma$. The definition of $\mathcal{M}_p$ is effective due to Theorem 3.7. It is straightforward to check that $L(\mathcal{M}_p) = \{\alpha \in \Gamma^* \mid p\alpha \in \bigcup_{i=0}^{\infty} S_i\}$. $\qquad\square$

Lemma 4.3. *Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}(\Delta)$ be simple sets. The set $\{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) = 0\}$ is effectively regular.*

Proof. Let $R(pX) = \{q \in Q \mid [pXq] > 0\}$ for all $p \in Q$, $X \in \Gamma$. For each $i \in \mathbb{N}_0$ we define the set $S_i \subseteq \mathcal{C}(\Delta)$ inductively as follows:

- $S_0 = \{q\varepsilon \mid q\varepsilon \notin \mathcal{C}_2\}$
- $S_{i+1} = \{pX\beta \mid [pX\bullet] = 0 \text{ and } \forall q \in R(pX) : q\beta \in S_i\}$

The fact $\bigcup_{i=0}^{\infty} S_i = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) = 0\}$ follows immediately from Lemma 3.4. The set $\bigcup_{i=0}^{\infty} S_i$ is effectively regular, which can be shown by constructing a finite automaton $\mathcal{M}_p$ recognizing the set $\{\alpha \in \Gamma^* \mid p\alpha \in \bigcup_{i=0}^{\infty} S_i\}$. This construction and the rest of the argument are very similar to the ones of the proof of Lemma 4.2. Therefore, they are not given explicitly. $\qquad\square$

Theorem 4.4. *Let φ be a qualitative PCTL formula and ν a regular valuation. The set $\{p\alpha \in \mathcal{C}(\Delta) \mid p\alpha \models^{\nu} \varphi\}$ is effectively regular.*

Proof. By induction on the structure of φ. The cases when $\varphi \equiv \mathtt{tt}$ and $\varphi \equiv a$ follow immediately. For Boolean connectives we use the fact that regular sets are closed under complement and intersection. The other cases are covered by Lemma 4.1, 4.2, and 4.3. Here we also need Lemma 2.5, because the regular sets of configurations must effectively be replaced with simple ones before applying Lemma 4.1, 4.2, and 4.3. $\qquad\square$

4.2. **Model Checking PCTL for pBPA Processes.** In this section we consider arbitrary PCTL properties with regular valuations, but restrict ourselves to pBPA processes. We provide an error-tolerant model-checking algorithm. Since it is not so obvious what is meant by error tolerance in the context of PCTL model checking, this notion is defined formally. More precisely, we first show that for every formula there is an equivalent negation-free formula, and then we provide a definition for negation-free formulas.

Let $\mathcal{T} = (S, \rightarrow, Prob)$ be a probabilistic transition system and $0 < \lambda < 1$, let φ be a PCTL formula, and let ν be a regular valuation (i.e., for every atomic proposition a the set $\nu(a)$ of configurations is regular). We observe that there is a negation-free formula φ' and a regular valuation ν' such that $[\![\varphi]\!]^{\nu} = [\![\varphi']\!]^{\nu'}$. First, negations can be "pushed inside" to atomic propositions using dual connectives (note that, e.g., $\neg(\varphi \mathcal{U}^{\geq \varrho} \psi)$ is equivalent to $\varphi \mathcal{U}^{< \varrho} \psi$). Moreover, since regular sets are closed under complement, $[\![\neg a]\!]^{\nu}$ is also regular for every a. We construct φ' by replacing each negation $\neg a$ by a fresh atomic proposition b, and we extend ν to ν' by defining $\nu(b) = [\![\neg a]\!]^{\nu}$.

For every negation-free PCTL formula φ and valuation ν we define the denotation of φ over $\mathcal{T}$ w.r.t. ν with *error tolerance* λ, denoted $[\![\varphi]\!]^\nu_\lambda$, in the same way as $[\![\varphi]\!]^\nu$. The only exception is $\varphi_1\,\mathcal{U}^{\sim\varrho}\varphi_2$ where

- if $\sim\,\in\{<,\leq\}$, then $[\![\varphi_1\,\mathcal{U}^{\sim\varrho}\varphi_2]\!]^\nu_\lambda = \{s\in S \mid \mathcal{P}(s,[\![\varphi_1]\!]^\nu_\lambda\,\mathcal{U}\,[\![\varphi_2]\!]^\nu_\lambda) \sim \varrho + \lambda\}$
- if $\sim\,\in\{>,\geq\}$, then $[\![\varphi_1\,\mathcal{U}^{\sim\varrho}\varphi_2]\!]^\nu_\lambda = \{s\in S \mid \mathcal{P}(s,[\![\varphi_1]\!]^\nu_\lambda\,\mathcal{U}\,[\![\varphi_2]\!]^\nu_\lambda) \sim \varrho - \lambda\}$

Notice that every negation-free formula φ satisfies $[\![\varphi]\!]^\nu \subseteq [\![\varphi]\!]^\nu_\lambda$.

An *error tolerant PCTL model checking algorithm* is an algorithm which, for each PCTL formula φ, valuation ν, $s\in S$, and $0<\lambda<1$, outputs YES/NO so that

- if $s\in[\![\varphi]\!]^\nu$, then the answer is YES;
- if the answer is YES, then $s\in[\![\varphi]\!]^\nu_\lambda$.

For the rest of this section, let us fix a pBPA $\Delta=(\Gamma,\delta,Prob)$. Since Δ has just one (or "none") control state p, we write $[X,\bullet]$ and $[X,\varepsilon]$ instead of $[pX\bullet]$ and $[pXp]$, respectively.

We need the following obvious generalization of Lemma 4.1 (use the same proof):

Lemma 4.5. *Let $\mathcal{C}\subseteq\mathcal{C}(\Delta)$ be a simple set, $\varrho\in[0,1]$, and $\sim\,\in\{\leq,<,\geq,>\}$. The set $\{\alpha\in\mathcal{C}(\Delta)\mid\mathcal{P}(\alpha,\mathcal{XC})\sim\varrho\}$ is effectively regular.*

Proof. Immediate. $\qquad\square$

Input: pBPA Δ, $0<\lambda<1$
Output: $n,\kappa,\nu,[X,\bullet]^\ell,[X,\varepsilon]^\ell,[X,\bullet]^u,[X,\varepsilon]^u$

```
1:   S := {X ∈ Γ | [X, ε] ≠ 1};
2:   ν := 1; n := ∞;
3:   for each  X ∈ S  do
4:       [X, ε]ℓ := 0; [X, •]ℓ := 0; [X, ε]u := 1; [X, •]u := 1;
5:   done
6:   repeat
7:       for each  X ∈ Γ  do
8:           avgε := ([X, ε]u − [X, ε]ℓ)/2;
9:           avg• := ([X, •]u − [X, •]ℓ)/2;
10:          if  [X, ε] ≥ avgε  then  [X, ε]ℓ := avgε;
11:              else  [X, ε]u := avgε;
12:          if  [X, •] ≥ avg•  then  [X, •]ℓ := avg•;
13:              else  [X, •]u := avg•;
14:      done
15:      ν := ν/2;
16:      κ := max{[X, ε]u | X ∈ S};
17:      if  κ < 1  then  n := ⌈(log(λ/3)/ log κ⌉
18: until  κ < 1  and  n(ν + ν(n + 1)(1 + ν)ⁿ) ≤ λ/3
```

Figure 3: A part of the algorithm for pBPA

The following lemma presents the crucial part of the algorithm. This is the place where we need the assumption that Δ is a pBPA.

Lemma 4.6. *Let $\mathcal{C}_1,\mathcal{C}_2\subseteq\mathcal{C}(\Delta)$ be simple sets. For all $\varrho\in[0,1]$ and $0<\lambda<1$ there effectively exist Δ-automata $\mathcal{A}^\geq$ and $\mathcal{A}^\leq$ such that for all $\alpha\in\mathcal{C}(\Delta)$ we have that*

- *if $\mathcal{P}(\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \geq \varrho$ (or $\mathcal{P}(\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \leq \varrho$), then $\alpha \in \mathcal{C}(\mathcal{A}^{\geq})$ (or $\alpha \in \mathcal{C}(\mathcal{A}^{\leq})$, respectively.)*
- *if $\alpha \in \mathcal{C}(\mathcal{A}^{\geq})$ (or $\alpha \in \mathcal{C}(\mathcal{A}^{\leq})$), then $\mathcal{P}(\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \geq \varrho - \lambda$ (or $\mathcal{P}(\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \leq \varrho + \lambda$, respectively.)*

Proof. We describe just the construction of $\mathcal{A}^{\geq}$ (the Δ-automaton $\mathcal{A}^{\leq}$ is constructed similarly). Let $S = \{X \in \Gamma \mid [X, \varepsilon] \neq 1\}$. For each $\beta \in S^*$ we define the set $Cl(\beta) = \{\alpha \in \Gamma^* \mid \alpha|_S = \beta\}$, where $\alpha|_S$ is the word obtained by deleting in α all occurrences of symbols in $\Gamma \smallsetminus S$. It follows directly from Lemma 3.4 that for all $\beta \in S^*$ and $\alpha \in Cl(\beta)$ we have that $\mathcal{P}(\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) = \mathcal{P}(\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$. Further, for all $n \in \mathbb{N}_0$ and $\beta \in \bigcup_{i=0}^{n} S^i$ we define the set

$$Gen_n(\beta) = \begin{cases} Cl(\beta) & \text{if } \alpha \in S^i \wedge i < n \\ \{\alpha\alpha' \mid \alpha \in Cl(\beta), \alpha' \in \Gamma^*\} & \text{if } \alpha \in S^n \end{cases}$$

We prove that for every $0 < \lambda < 1$ there effectively exist $n \in \mathbb{N}_0$ and $\mathcal{G} \subseteq \bigcup_{i=0}^{n} S^i$ such that for every $\alpha \in \Gamma^*$ we have that

- if $\mathcal{P}(\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \geq \varrho$, then $\alpha \in \bigcup_{\beta \in \mathcal{G}} Gen_n(\beta)$;
- if $\alpha \in \bigcup_{\beta \in \mathcal{G}} Gen_n(\beta)$, then $\mathcal{P}(\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \geq \varrho - \lambda$.

This suffices for our purposes, because the set $\bigcup_{\beta \in \mathcal{G}} Gen_n(\beta)$ is clearly recognizable by an effectively constructible Δ-automaton $\mathcal{A}^{\geq}$.

The crucial part of the algorithm for computing the set $\mathcal{G}$ is shown in Fig. 3. The algorithm starts by computing the set S (note that S is effectively computable due to Theorem 3.7). For each $X \in S$, there are four rational variables $[X, \varepsilon]^{\ell}$, $[X, \varepsilon]^u$, $[X, \bullet]^{\ell}$, and $[X, \bullet]^u$ whose values are lower and upper approximations of the probabilities $[X, \varepsilon]$ and $[X, \bullet]$, resp. These variables are initialized in lines 3–5 and successively refined in lines 7–14. Note that the conditions of the **if** statements in lines 10 and 12 are effective due to Theorem 3.7. The current "precision", i.e., the difference between the upper and the lower approximation is stored in the rational variable ν. The subtle point is the termination condition. First, one necessary condition for termination is that $\kappa = \max\{[X, \varepsilon]^u \mid X \in S\}$ becomes less than one. This must happen eventually, because $[X, \varepsilon] < 1$ for every $X \in S$. An important observation is that κ can only *decrease* by performing the assignment in line 16. This means that $n = \lceil \log(\lambda/3)/\log \kappa \rceil$ also only *decreases* (since both λ and κ are less than 1, we have $\log(\lambda/3)/\log \kappa = |\log(\lambda/3)|/|\log \kappa|$; and if $0 < \kappa' < \kappa < 1$, then $|\log \kappa'| > |\log \kappa|$). Therefore, we eventually find a sufficiently small ν such that $n(\nu + \nu(n+1)(1+\nu)^n) \leq \lambda/3$.

The output of the algorithm of Fig. 3 are the (values of the) variables n, ν, κ, $[X, \varepsilon]^{\ell}$, $[X, \varepsilon]^u$, $[X, \bullet]^{\ell}$, and $[X, \bullet]^u$ where X ranges over S. For each $\beta \in S^*$, let $\mathcal{P}^{\ell}(\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ and $\mathcal{P}^u(\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ be the lower and upper approximations of $\mathcal{P}(\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)$ obtained by using the formula of Lemma 3.4 where $[X, \varepsilon]^{\ell}$, $[X, \bullet]^{\ell}$, and $[X, \varepsilon]^u$, $[X, \bullet]^u$ are used instead of $[X, \varepsilon]$, $[X, \bullet]$, respectively. The set $\mathcal{G}$ is constructed as follows:

$$\begin{aligned} \mathcal{G} &= \{\beta \in S^i \mid 0 \leq i < n, \mathcal{P}^u(\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \geq \varrho\} \\ &\cup \{\beta \in S^n \mid \mathcal{P}^u(\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) \geq \varrho - \lambda/3\} \end{aligned}$$

To verify that the set $\mathcal{G}$ has the properties mentioned above, we need to formulate two auxiliary observations.

(a) for all $\beta \in S^n$ and $\alpha \in \Gamma^*$ we have that

$$|\mathcal{P}(\beta, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2) - \mathcal{P}(\beta\alpha, \mathcal{C}_1 \mathcal{U} \mathcal{C}_2)| \leq \lambda/3$$

This follows immediately from the following (in)equalities:

$$\mathcal{P}(\beta\alpha, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \;=\; \mathcal{P}(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2^{\bullet}) \;+\; \mathcal{P}(\beta, \mathcal{C}_1\smallsetminus\mathcal{C}_2\,\mathcal{U}\,\{\varepsilon\}) \cdot \mathcal{P}(\alpha, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2)$$

$$\mathcal{P}(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \;\leq\; \mathcal{P}(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2^{\bullet}) \;+\; \mathcal{P}(\beta, \mathcal{C}_1\smallsetminus\mathcal{C}_2\,\mathcal{U}\,\{\varepsilon\})$$

$$\mathcal{P}(\beta, \mathcal{C}_1\smallsetminus\mathcal{C}_2\,\mathcal{U}\,\{\varepsilon\}) \;\leq\; \lambda/3$$

The first two (in)equalities are obtained just by applying Lemma 3.4. The last one is derived as follows: $\mathcal{P}(\beta, \mathcal{C}_1\smallsetminus\mathcal{C}_2\,\mathcal{U}\,\{\varepsilon\})$ is surely bounded by κ^n (by Lemma 3.4 and the definition of κ). Since $n = \lceil \log(\lambda/3)/\log\kappa \rceil$, we have $n \cdot \log\kappa \leq \log(\lambda/3)$. Hence, $\log\kappa^n \leq \log(\lambda/3)$, thus $\kappa^n \leq \lambda/3$.

(b) for each $\beta \in \bigcup_{i=0}^{n} S^i$ we have that

$$\mathcal{P}^u(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) - \mathcal{P}(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \leq \lambda/3$$

Let $k = length(\beta)$. A straightforward induction on k reveals that $\mathcal{P}^u(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \leq (k+1) \cdot (1+\nu)^k$. Now we prove (again by induction on k) that

$$\mathcal{P}^u(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) - \mathcal{P}(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \leq k(\nu + \nu(k+1)(1+\nu)^k)$$

The base case (when $k = 0$) is immediate, because $\mathcal{P}^u(\varepsilon, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) = \mathcal{P}(\varepsilon, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2)$. Now let $\beta = X\beta'$. By definition, $\mathcal{P}^u(X\beta', \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) - \mathcal{P}(X\beta', \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2)$ is equal to

$$[X, \bullet]^u + [X, \varepsilon]^u \cdot \mathcal{P}^u(\beta', \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \;-\; ([X, \bullet] + [X, \varepsilon] \cdot \mathcal{P}(\beta', \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2)) \qquad (4.1)$$

Since $[X, \bullet]^u \leq [X, \bullet] + \nu$ and $[X, \varepsilon]^u \leq [X, \varepsilon] + \nu$, the expression (4.1) is bounded by

$$\nu \;+\; [X, \varepsilon] \cdot (\mathcal{P}^u(\beta', \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) - \mathcal{P}(\beta', \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2)) \;+\; \nu \cdot \mathcal{P}^u(\beta', \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \qquad (4.2)$$

By applying induction hypothesis and the facts that $[X, \varepsilon] \leq 1$ and $\mathcal{P}^u(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \leq (k+1) \cdot (1+\nu)^k$ (see above), we obtain that the expression (4.2) is bounded by

$$\nu + k(\nu + \nu(k+1)(1+\nu)^k) + \nu(k+1)(1+\nu)^k$$

which is bounded by $(k+1)(\nu + \nu(k+2)(1+\nu)^{k+1})$ as required. This finishes the inductive step.

Since $n(\nu + \nu(n+1)(1+\nu)^n) \leq \lambda/3$ and $k \leq n$, we have $\mathcal{P}^u(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) - \mathcal{P}(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \leq k(\nu + \nu(k+1)(1+\nu)^k) \leq \lambda/3$.

Now we are ready to prove that the set $\mathcal{G}$ has the required properties. Let $\alpha \in \Gamma^*$ such that $\mathcal{P}(\alpha, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho$, and let $\beta = \alpha|_S$. There are two possibilities:

- $length(\beta) < n$. Then $\mathcal{P}^u(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho$, hence $\beta \in \mathcal{G}$ and $\alpha \in \bigcup_{\beta \in \mathcal{G}} Gen_n(\beta)$.
- $length(\beta) \geq n$. Let $\beta = \gamma\gamma'$ where $length(\gamma) = n$. Due to the observation (a) above we have that $\mathcal{P}(\gamma, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho - \lambda/3$, hence also $\mathcal{P}^u(\gamma, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho - \lambda/3$, which means that $\gamma \in \mathcal{G}$ and thus $\alpha \in \bigcup_{\beta \in \mathcal{G}} Gen_n(\beta)$.

Now let $\alpha \in Gen_n(\beta)$ for some $\beta \in \mathcal{G}$. Again, we distinguish two possibilities:

- $length(\beta) < n$. Then $\mathcal{P}^u(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho$, which means that $\mathcal{P}(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho - \lambda/3$ by the observation (b) above. Hence, $\mathcal{P}(\alpha, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho - \lambda/3$.
- $length(\beta) = n$. Then $\mathcal{P}^u(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho - \lambda/3$, which means that $\mathcal{P}(\beta, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho - 2\lambda/3$ due to the observation (b). Further, for every $\alpha' \in \Gamma$ we have that $\mathcal{P}(\beta\alpha', \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho - \lambda$ due to the observation (a) above. Hence, $\mathcal{P}(\alpha, \mathcal{C}_1\,\mathcal{U}\,\mathcal{C}_2) \geq \varrho - \lambda$ as required.

The automaton $\mathcal{A}^{\leq}$ is constructed similarly. Here, the set $\mathcal{G}$ is computed using the lower approximations $[X, \bullet]^{\ell}$ and $[X, \varepsilon]^{\ell}$. Since this construction is analogous to the one just presented, it is not given explicitly. $\square$

Theorem 4.7. *There is an error-tolerant PCTL model checking algorithm for pBPA processes.*

Proof. The proof is similar to the one of Theorem 4.4, using Lemma 4.5 and 4.6 instead of Lemma 4.1, 4.2, and 4.3. Note that Lemma 2.5 is applicable also to pBPA (the system Δ' constructed in Lemma 2.5 has the same set of control states as the original system Δ). $\square$

5. MODEL CHECKING ω-REGULAR SPECIFICATIONS

In this section we show that the qualitative and quantitative model-checking problem for pPDA and ω-regular properties are decidable. At the very core of our result are observations leading to the definition of a finite Markov chain M_Δ. Intuitively, each transition of M_Δ corresponds to a sequence of transitions of the probabilistic transition system $\mathcal{T}_\Delta$ associated to Δ. This allows to reduce the model-checking problem to a problem about M_Δ, which, since M_Δ is finite, can be solved using well-known techniques. In [EKM04], the Markov chain M_Δ was used to show that the qualitative and quantitative model-checking problem for properties expressible by deterministic Büchi automata is decidable. Later, it was observed in [BKS05] that the technique can easily be generalized to deterministic Muller automata. Thus, the decidability result was extended to all ω-regular properties. In this paper we go a bit further, and prove the decidability of a slightly larger class. The previous result about the ω-regular case follows as a corollary.

The section is structured as follows. Given a pPDA Δ, we first introduce the notion of *minima of a run* and Δ-*observing automaton*. We use observing automata as specifications: an infinite run satisfies the specification iff it is accepted by the automaton (section 5.1). Using the notion of minima, we define the finite Markov chain M_Δ (section 5.2), and show that the probability that a run is accepted by a Δ-observing automaton is effectively expressible in $(\mathbb{R}, +, *, \leq)$ (section 5.3). Finally, we show that the model-checking problem for ω-regular properties is a special case of the problem of deciding if a run is accepted by a Δ-observing automaton with at least a given probability (section 5.4).

For the rest of this section, we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$.

5.1. **Minima of a run.** Loosely speaking, a configuration of a run is a minimum if all configurations placed after it in the run have the same or larger stack length.

Definition 5.1. Let $w = p_1\alpha_1; p_2\alpha_2, \cdots$ be an infinite run in $\mathcal{T}_\Delta$. A configuration $p_i\alpha_i$ is a *minimum* of w if $|\alpha_i| \leq |\alpha_j|$ for every $j \geq i$. We say that $p_i\alpha_i$ is the kth *minimum* of w if $p_i\alpha_i$ is a minimum and there are exactly $k-1$ indices $j < i$ such that $p_j\alpha_j$ is a minimum. We denote the kth minimum of w by $\min_k(w)$.

Sometimes we abuse language and use $\min_i(w)$ to denote not only a configuration, but the particular *occurrence* of the configuration that corresponds to the ith minimum.

Example 5.2. In the run $w_1 = (Z; DZ)^\omega$ of the pBPA shown in the introduction we have $\min_i(w_1) = Z$ for every $i \geq 1$. In the run $w_2 = Z; DZ; DDZ; \ldots$ we have $\min_1(w_2) = Z$ and $\min_i(w_2) = D$ for every $i \geq 2$. Every odd configuration of w_1 is a minimum, and every configuration of w_2 is a minimum. $\square$
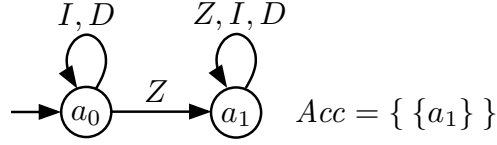
Figure 4: An observing automaton

Since stack lengths are bounded from below, every infinite run has infinitely many minima, and so it can be divided into an infinite sequence of fragments, or "jumps", each of them leading from one minimum to the next.

We are interested in those properties of a run that can be decided by extracting a finite amount of information from each jump, independently of its length. Consider for instance the property "the control state p is visited infinitely often along the run". It can be reformulated as "there are infinitely many jumps along which the state p is visited". In order to decide the property all we need is a bit of information for each jump, telling whether it is "visiting" or "non-visiting". We consider properties in which this finite amount of information can be extracted by letting a finite automaton go over the jump reading the *heads* of the configurations:

Definition 5.3. Given a configuration $pX\alpha$ of Δ, we call pX the *head* and α the *tail* of $pX\alpha$. The set $Q \times \Gamma$ of all heads of Δ is also denoted by $\mathcal{H}(\Delta)$.

More precisely, we consider automata with the set of heads as alphabet. An oracle tells the automaton to start reading heads immediately after the run leaves a minimum (i.e., the first head read is the one of the configuration immediately following the minimum), stop after reading the head of the next minimum, report its state, and reset itself to an initial state that depends on the head of the minimum.

Definition 5.4. A Δ-*observing automaton* is a tuple $\mathcal{A} = (A, \xi, a_o, Acc)$ where A is finite set of *observing states*, $\xi : A \times \mathcal{H}(\Delta) \to A$ is a (total) *transition function*, $a_0 \in A$ is an *initial state*, and Acc is a set of subsets of A, also called an *acceptance set*.

Let w be an infinite run in $\mathcal{T}_\Delta$ and let $i \in \mathbb{N}$. The ith *observation* of $\mathcal{A}$ over w, denoted $Obs_i(w)$, is the state reached by $\mathcal{A}$ after reading the heads of all configurations between $\min_i(w)$ and $\min_{i+1}(w)$, including $\min_{i+1}(w)$ but not including $\min_i(w)$. [4] The *observation* of $\mathcal{A}$ on w, denoted by $Obs(w)$, is the sequence $Obs_1(w)Obs_2(w)\dots$.

We say that an infinite run $w \in Run(pX)$ is *accepting* if the set of states of A that occur infinitely often in $Obs(w)$ belongs to Acc; otherwise, w is *rejecting*.

Example 5.5. Figure 4 shows a Δ-observing automaton for the pBPA of the introduction (see also Figure 1). For every infinite run w and every $i \geq 0$, we have $Obs_i(w) = b$ if some configuration of the ith jump has Z as topmost stack symbol. So a run is accepting iff it visits configurations with head Z infinitely often. □

For the rest of the section we fix a Δ-observing automaton $\mathcal{A} = (A, \xi, a_0, Acc)$. Let $Run(pX, Acc)$ be the set of all accepting runs initiated in pX. Our aim is to show that $\mathcal{P}(Run(pX, Acc))$ is effectively definable in $(\mathbb{R}, +, *, \leq)$.

---

[4]Notice that the automaton starts observing after the first minimum of the run.

5.2. **The Markov chain M_Δ.** For all $pX \in \mathcal{H}(\Delta)$ and all $i \in \mathbb{N}$ we define a random variable $V_{pX}^{(i)}$ over $Run(pX)$. Loosely speaking, $V_{pX}^{(i)}$ assigns to a run starting at the configuration pX the head of its ith minimum, and the ith observation of the Δ-observing automaton $\mathcal{A}$. Formally, the possible values of $V_{pX}^{(i)}$ are pairs of the form (qY, a), where $qY \in \mathcal{H}(\Delta)$ and $a \in A$. There is also a special value $\bot$, where $\bot \notin \mathcal{H}(\Delta) \times A$. For a given $w \in Run(pX)$, the value $V_{pX}^{(i)}(w)$ is determined as follows: If w is finite, then $V_{pX}^{(i)}(w) = \bot$; otherwise, $V_{pX}^{(i)}(w) = (qY, Obs_i(w))$, where qY is the head of $\min_i(w)$. Notice that the random variables are well defined, because they assign to each run exactly one value.

Given possible values $v_1, \ldots, v_n$ for the variables $V_{pX}^{(1)}, \ldots, V_{pX}^{(n)}$, we are going to prove the following two results:

- the probability that a run satisfies $V_{pX}^{(1)} = v_1, \ldots, V_{pX}^{(n)} = v_n$ is expressible in $(\mathbb{R}, +, *, \leq)$ (Lemma 5.10); and
- the probability that $V_{pX}^{(i+1)} = v_{i+1}$ depends only on the value of $V_{pX}^{(i)}$, but neither on i nor on the value of $V_{pX}^{(k)}$ for $k < i$ (Lemma 5.11 and 5.12).

The second result will allow us to define the finite Markov chain M_Δ, while the first one will show that its transition probabilities are expressible in $(\mathbb{R}, +, *, \leq)$.

The proof of Lemma 5.10 is rather technical (as we shall, see, Lemma 5.11 and 5.12 are easy corollaries of Lemma 5.10). We need three auxiliary lemmas. Intuitively, the first one states that the probability of executing an infinite run from a configuration pX is equal to the probability of executing an infinite run from $pX\beta$ such that the stack content never goes *"below"* β. For every finite or infinite path $w = p_1\alpha_1; p_2\alpha_2; \cdots$ in $\mathcal{T}_\Delta$ and every $\beta \in \Gamma^*$, the symbol $w^{+\beta}$ denotes the path $p_1\alpha_1\beta; p_2\alpha_2\beta; \cdots$ obtained from w by concatenating β to the stack content in every configuration. Similarly, if R is a set of paths in $\mathcal{T}_\Delta$ and $\beta \in \Gamma^*$, then $[R]^{+\beta}$ denotes the set $\{w^{+\beta} \mid w \in R\}$.

Lemma 5.6. *Let $pX \in Q \times \Gamma$ and $\beta \in \Gamma^*$. Then $\mathcal{P}([IRun(pX)]^{+\beta}) = \mathcal{P}(IRun(pX))$.*

Proof. Let $Dead = Q \times \{\varepsilon\} \cup \{qY\alpha \mid qY \text{ has no transitions in } \delta, \alpha \in \Gamma^*\}$. We have that

$$
\begin{aligned}
\mathcal{P}([IRun(pX)]^{+\beta}) &= 1 - \mathcal{P}(pX\beta, \mathcal{C}(\Delta)^\bullet \beta \, \mathcal{U} \, Dead\,\beta) \\
&= 1 - \mathcal{P}(pX, \mathcal{C}(\Delta) \, \mathcal{U} \, Dead) \quad \text{(by Lemma 3.3)} \\
&= \mathcal{P}(IRun(pX)).
\end{aligned}
$$

$\square$

The second lemma states that prefixing a measurable set of runs with a finite path yields a measurable set of runs, and relates the probabilities of both sets.

Lemma 5.7. *Let $s_0; \cdots; s_n$ be a path in a probabilistic transition system, and let R be a measurable subset of $Run(s_n)$. Then $\{s_0; \cdots; s_n\} \odot R$ is a measurable subset of $Run(s_0)$, and moreover $\mathcal{P}(\{s_0; \cdots; s_n\} \odot R) = \Pi_{i=1}^{n} x_i \cdot \mathcal{P}(R)$, where $s_i \xrightarrow{x_{i+1}} s_{i+1}$ for every $0 \leq i < n$. (The '$\odot$' operator has been introduced in Definition 3.1.)*

Proof. Standard. $\square$

The third lemma shows that the probability of starting from the configuration qY reaching the configuration $q\varepsilon$ with the observing automaton in state a is expressible in $(\mathbb{R}, +, *, \leq)$.

Definition 5.8. Let R be a $\mathcal{P}$-measurable set of runs of $\mathcal{T}_\Delta$ starting at the same initial configuration. We say that $\mathcal{P}(R)$ is *well-definable* if there effectively exist a pPDA Δ' and a finite family of probabilities of the form $\mathcal{P}(Run(qY, \mathcal{C}_1 \; \mathcal{U} \; \mathcal{C}_2))$, where $qY \in \mathcal{H}(\Delta')$ and $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{C}(\Delta')$ are simple sets, such that $\mathcal{P}(R)$ is effectively definable from this family of probabilities using only summation, multiplication, and rational constants.

Note that if $\mathcal{P}(R)$ is well-definable, it can be expressed in $(\mathbb{R}, +, *, \leq)$ using the results of Section 3.

For all $qY \in \mathcal{H}(\Delta)$, $r \in Q$, $Z \in \Gamma$, and $a \in A$, let $Run(qY, r, Z, a) \subseteq Run(qY)$ be the set of all runs $w = s_0; \cdots; s_n$ such that $s_0 = qY$, $s_n = r\varepsilon$, and the automaton $\mathcal{A}$ reaches the state a after reading the heads of configurations $s_0, \cdots, s_{n-1}, rZ$.

Lemma 5.9. $\mathcal{P}(Run(qY, r, Z, a))$ *is well-definable.*

Proof. We put $\Delta' = (Q \times A, \Gamma, \delta', Prob')$ to be the synchronized product of Δ and $\mathcal{A}$, i.e., $(p, \bar{a})X \xrightarrow{x} (t, \hat{a})\alpha$ is a rule of Δ' iff $pX \xrightarrow{x} t\alpha$ is a rule of Δ and $\xi(\bar{a}, pX) = \hat{a}$. Let $\bar{A} = \{\bar{a} \in A \mid \xi(\bar{a}, rZ) = a\}$. Now we can easily check that $\mathcal{P}(Run(qY, r, Z, a))$ is equal to

$$\mathcal{P}(\, (q, a_0)Y, \mathcal{C}(\Delta') \; \mathcal{U} \; \{(r, \bar{a})\varepsilon\})$$

$\square$

We can now prove our main technical result:

Lemma 5.10. *For all $pX \in \mathcal{H}(\Delta)$, $n \in \mathbb{N}$, and $v_1, \cdots, v_n \in (\mathcal{H}(\Delta) \times A) \cup \{\bot\}$, the probability of $V_{pX}^{(1)} = v_1 \wedge \cdots \wedge V_{pX}^{(n)} = v_n$ is well-definable. In particular, for every rational constant y there is an effectively constructible formula of $(\mathbb{R}, +, *, \leq)$ which holds if and only if $\mathcal{P}(V_{pX}^{(1)} = v_1 \wedge \cdots \wedge V_{pX}^{(n)} = v_n) = y$.*

Proof. By induction on n we prove that $\mathcal{P}(V_{pX}^{(1)} = v_1 \wedge \cdots \wedge V_{pX}^{(n)} = v_n)$ is well-definable. The base case when $n = 1$ follows immediately, because $\mathcal{P}(V_{pX}^{(1)} = v_1)$ equals either $\mathcal{P}(IRun(pX))$, $1 - \mathcal{P}(IRun(pX))$, or 0, depending on whether $v_1 = (pX, a_0)$, $v_1 = \bot$, or $(pX, a_0) \neq v_1 \neq \bot$, respectively. Observe that $\mathcal{P}(IRun(pX)) = 1 - \mathcal{P}(pX, \mathcal{C}(\Delta) \; \mathcal{U} \; Dead)$, where $Dead = Q \times \{\varepsilon\} \cup \{qY\alpha \mid qY \text{ has no transitions in } \delta, \alpha \in \Gamma^*\}$.

Now let $n \geq 2$. For each $1 \leq i \leq n$, let Sat_i be the set of all runs that satisfy $V_{pX}^{(1)} = v_1 \wedge \cdots \wedge V_{pX}^{(i)} = v_i$. If $\mathcal{P}(Sat_{n-1}) = 0$, which is decidable by induction hypothesis, then $\mathcal{P}(Sat_n) = 0$ as well. If $\mathcal{P}(Sat_{n-1}) \neq 0$ and there is an $i \leq n-1$ such that $v_i = \bot$, then for all $j \leq n-1$ we have that $v_j = \bot$, and $\mathcal{P}(Sat_n)$ is equal either to $\mathcal{P}(Sat_{n-1})$ or 0, depending on whether $v_n = \bot$ or not, respectively. If $\mathcal{P}(Sat_{n-1}) \neq 0$, $v_i \neq \bot$ for all $i \leq n-1$, and $v_n = \bot$, then $\mathcal{P}(Sat_n) = 0$. So, the only interesting case is when $\mathcal{P}(Sat_{n-1}) \neq 0$ and $v_i \neq \bot$ for all $i \leq n$. Since

$$\mathcal{P}(Sat_n) = \frac{\mathcal{P}(V_{pX}^{(n)} = v_n \mid Sat_{n-1})}{\mathcal{P}(Sat_{n-1})}$$

and $\mathcal{P}(Sat_{n-1})$ is well-definable by induction hypothesis, it suffices to show that the conditional probability $\mathcal{P}(V_{pX}^{(n)} = v_n \mid Sat_{n-1})$ is also well-definable. For this we use a general

result of basic probability theory saying that if A, B are events and $B = \uplus_{i \in I} B_i$, where I is a finite or countably infinite index set, then

$$\mathcal{P}(A \mid B) = \frac{\sum_{i \in I} \mathcal{P}(A \mid B_i) \cdot \mathcal{P}(B_i)}{\mathcal{P}(B)}$$

An immediate consequence of this equation is that if the probability $\mathcal{P}(A|B_i)$ is independent of i, then $\mathcal{P}(A|B) = \mathcal{P}(A|B_i)$. In our case, A is the event $V_{pX}^{(n)} = v_n$, and B is Sat_{n-1}. Let

$$Chop = \{w(0); \cdots ; w(\min_{n-1}(w)) \mid w \in Sat_{n-1}\}.$$

Observe that if $y \in Chop$, then the last configuration of y is of the form $p_{n-1}X_{n-1}\alpha$. We denote the α by $Stack(y)$. For every $y \in Chop$, let

$$Sat_{n-1}(y) = \{y\} \odot [IRun(p_{n-1}X_{n-1})]^{+Stack(y)} \tag{5.1}$$

Now we can easily check that

$$Sat_{n-1} = \biguplus_{y \in Chop} Sat_{n-1}(y)$$

Hence, *Chop* plays the role of I, and $Sat_{n-1}(y)$ plays the role of B_i. We show that $\mathcal{P}(V_{pX}^{(n)} = v_n \mid Sat_{n-1}(y))$ is independent of y, which means that

$$\mathcal{P}(V_{pX}^{(n)} = v_n \mid Sat_{n-1}(y)) = \mathcal{P}(V_{pX}^{(n)} = v_n \mid Sat_{n-1}).$$

By definition of conditional probability,

$$\mathcal{P}(V_{pX}^{(n)} = v_n \mid Sat_{n-1}(y)) = \frac{\mathcal{P}(V_{pX}^{(n)} = v_n \wedge Sat_{n-1}(y))}{\mathcal{P}(Sat_{n-1}(y))} \tag{5.2}$$

The denominator of the fraction in equation (5.2) is well-definable, because

$$\mathcal{P}(Sat_{n-1}(y)) = \mathcal{P}(Run(y)) \cdot \mathcal{P}(IRun(p_{n-1}X_{n-1}))$$

Here we used Lemma 5.6, Lemma 5.7, and equation (5.1). Now we show that $\mathcal{P}(V_{pX}^{(n)} = v_n \wedge Sat_{n-1}(y))$ is also well-definable. Let R be the set of all runs satisfying $V_{pX}^{(n)} = v_n \wedge Sat_{n-1}(y)$, and let $v_n = (p_n X_n, a_n)$ and $\alpha = Stack(y)$. Obviously, each $w \in R$ starts with y. Now let us consider what transitions can be performed from the final state $p_{n-1}X_{n-1}\alpha$ of y.

- Obviously, transitions which decrease the stack cannot be performed, because $p_{n-1}X_{n-1}\alpha$ would not be a minimum then (i.e., w would not belong to R).
- If a transition of the form $p_{n-1}X_{n-1}\alpha \xrightarrow{x} rZ\alpha$ is performed, then $rZ\alpha$ must be the n-th minimum, because the stack cannot be decreased below Z (otherwise, $p_{n-1}X_{n-1}\alpha$ would not be a minimum). So, if $w \in R$, we must have that $rZ = p_n X_n$ and $\xi(a_0, rZ) = a_n$.
- If a transition of the form $p_{n-1}X_{n-1}\alpha \xrightarrow{x} rPQ\alpha$ is performed, then the stack cannot be decreased below Q. Now there are two possibilities:
  - If the stack is never decreased below P, then the configuration $rPQ\alpha$ is the n-th minimum. Hence, if $w \in R$, we must have that $rP = p_n X_n$ and $\xi(a_0, rP) = a_n$.
  - If the stack is decreased below P, i.e., if a sequence of transitions of the form $rPQ\alpha \rightarrow^* tQ\alpha$ is performed (where the stack is never decreased to $Q\alpha$ except in the last configuration), then $tQ\alpha$ is the n-th minimum. Hence, if $w \in R$, we must have that $tQ = p_n X_n$ and the automaton $\mathcal{A}$ reaches a_n by reading the word consisting of heads of configurations in the sequence $rPQ\alpha \rightarrow^* tQ\alpha$.

From the above discussion, it follows that R can be partitioned as follows:

$$R \;=\; \biguplus_{p_{n-1}X_{n-1}\xrightarrow{x}p_nX_n} \{y\}\odot\{p_{n-1}X_{n-1}\alpha;,p_nX_n\alpha\}\odot[IRun(p_nX_n)]^{+\alpha}$$

$$\biguplus_{\substack{p_{n-1}X_{n-1}\xrightarrow{x}p_nX_nY \\ Y\in\Gamma}} \{y\}\odot\{p_{n-1}X_{n-1}\alpha;p_nX_nY\alpha\}\odot[IRun(p_nX_n)]^{+Y\alpha}$$

$$\biguplus_{\substack{p_{n-1}X_{n-1}\xrightarrow{x}qYX_n \\ q\in Q,Y\in\Gamma}} \{y\}\odot[Run(qY,p_n,X_n,a_n)]^{+\alpha}\odot[IRun(p_nX_n)]^{+\alpha}$$

Using Lemma 5.6, Lemma 5.7, Lemma 5.9 and the above equation, we obtain that

$$\mathcal{P}(V_{pX}^{(n)}{=}v_n\wedge Sat_{n-1}(y))=\mathcal{P}(Run(y))\cdot\mathcal{P}(IRun(p_nX_n))\cdot S$$

where

$$S \;=\; \sum_{p_{n-1}X_{n-1}\xrightarrow{x}p_nX_n} x \;+\; \sum_{\substack{p_{n-1}X_{n-1}\xrightarrow{x}p_nX_nY \\ Y\in\Gamma}} x \;+\;$$

$$\sum_{\substack{p_{n-1}X_{n-1}\xrightarrow{x}qYX_n \\ q\in Q,Y\in\Gamma}} x\cdot\mathcal{P}(Run(qY,p_n,X_n,a_n)) \tag{5.3}$$

Equation (5.2) can now be rewritten to

$$\mathcal{P}(V_{pX}^{(n)}{=}v_n\mid Sat_{n-1}(y))=\frac{\mathcal{P}(IRun(p_nX_n))}{\mathcal{P}(IRun(p_{n-1}X_{n-1}))}\cdot S \tag{5.4}$$

where the meaning of S is given by equation (5.3). So, $\mathcal{P}(V_{pX}^{(n)}{=}v_n\mid Sat_{n-1}(y))$ is indeed independent of y, and hence equation (5.4) also defines the probability $\mathcal{P}(V_{pX}^{(n)}{=}v_n\mid Sat_{n-1})$. $\qquad\square$

Loosely speaking, the following lemma proves the memoryless property required to define a Markov chain: The probability of $V_{pX}^{(n)}=v_n$ depends only on the value of $V_{pX}^{(n-1)}$, not on the values of $V_{pX}^{(n-2)},\dots,V_{pX}^{(1)}$.

Lemma 5.11. *The conditional probability of $V_{pX}^{(n)}=v_n$ on the hypothesis $V_{pX}^{(1)}=v_1\wedge\cdots\wedge V_{pX}^{(n-1)}=v_{n-1}$ is equal to the probability of $V_{pX}^{(n)}=v_n$ conditioned on $V_{pX}^{(n-1)}=v_{n-1}$, assuming that the probability of $V_{pX}^{(1)}=v_1\wedge\cdots\wedge V_{pX}^{(n-1)}=v_{n-1}$ is non-zero.*

Proof. The result follows immediately from Equation (5.4) in the proof of Lemma 5.10: The right side on the equation does not depend on the values of $V_{pX}^{(n-2)},\dots,V_{pX}^{(1)}$. $\qquad\square$

Finally, as another consequence of Lemma 5.10 we obtain that the probability of $V_{pX}^{(n)}=v_n$ does not depend on n:

Lemma 5.12. *The conditional probability of $V_{pX}^{(n)}{=}(q'Y',a')$ on the hypothesis $V_{pX}^{(n-1)}{=}(qY,a)$ is equal to the conditional probability of $V_{qY}^{(2)}{=}(q'Y',a')$ on the hypothesis*

$V_{qY}^{(1)}{=}(qY, a_0)$, *assuming that* $\mathcal{P}(V_{pX}^{(n-1)}{=}(qY, a)) \neq 0$. *Moreover, the hypothesis that a run* w *satisfies* $V_{qY}^{(1)}(w){=}(qY, a_0)$ *is the same as the hypothesis that* $w \in IRun(qY)$.

Proof. The first part follows immediately from the fact that n appears only as an index in Equation (5.4). For the second, observe that, by definition, a run w starting at qY satisfies $V_{qY}^{(1)}(w){=}qY$ if (1) it is infinite and (2) its first minimum has head qY. But (1) and the fact that all configurations of an infinite run have length 1 or greater imply that the first configuration of the run is also its first minimum, and so, since w starts at qY, they imply (2). So a run w starting at qY satisfies $V_{qY}^{(1)}{=}qY$ iff it is infinite, i.e., iff $w \in IRun(qY)$. $\square$

Example 5.13. In order to give some intuition for these results, and in particular for the proof of Lemma 5.10, consider the special case in which the initial configuration is pX for some $p \in P, X \in \Gamma$, and the observing automaton $\mathcal{A}$ has one single state. In this case, the automaton always makes the same observation, and so we can write $V_{pX}^{(n)} = qY$ instead of $V_{pX}^{(n)} = (qY, a)$. We wish to obtain an expression for $\mathcal{P}(V_{pX}^{(2)}{=}qY)$. By the second part of Lemma 5.12 we have

$$\mathcal{P}(V_{pX}^{(1)}{=}pX) = \mathcal{P}(IRun(pX))$$

and therefore

$$\mathcal{P}(V_{pX}^{(2)}{=}qY) = \mathcal{P}(V_{pX}^{(2)}{=}qY \mid V_{pX}^{(1)}{=}pX) \cdot \mathcal{P}(IRun(pX))$$

Now we can apply equation 5.3 in the proof of Lemma 5.10 and obtain

$$\mathcal{P}(V_{pX}^{(2)}{=}qY) = \mathcal{P}(IRun(qY)) \cdot S$$

and, by Equation 5.2

$$\mathcal{P}(V_{pX}^{(2)}{=}qY) \;\; = \sum_{pX \xrightarrow{x} qY} x \cdot \mathcal{P}(IRun(qY)) \quad +$$
$$\sum_{\substack{pX \xrightarrow{x} qYZ \\ Z \in \Gamma}} x \cdot \mathcal{P}(IRun(qY)) \quad +$$
$$\sum_{\substack{pX \xrightarrow{x} rZY \\ r \in Q, Z \in \Gamma}} x \cdot \mathcal{P}(rZ, (Q \times \Gamma^*)\,\mathcal{U}\,\{q\varepsilon\}) \cdot \mathcal{P}(IRun(qY)) \qquad (5.5)$$

Let us interpret this equation. In order to reach the second minimum at qY there are only three possibilities for the first move. The first possibility is to move directly from pX to qY; in this case we must continue with any run that never terminates, since every infinite run of the form $pX; qY; \cdots$ necessarily has qY as second minimum. The probability of this case is captured by the first summand of Equation 5.5. The second possibility is to move from pX to qYZ for some $Z \in \Gamma$; in this case we must continue with an infinite run in which the stack content always has at least length 2, i.e., with a run of the form

$$pX; \; qYZ; \; q_1\alpha_1 Z; \; \dots; \; q_i\alpha_i Z; \; \dots$$

where all the α's are nonempty. This gives the second summand. Finally, the third possibility is to move from pX to rZY for some $r \in P, Z \in \Gamma$; we must then continue with a run that eventually "pops the Z" while entering state q, i.e., with a run of the form

$$pX; \; rZY; \; r_1\alpha_1 Y; \; \dots; \; r_n\alpha_n Y; \; qY; \; q_1\beta_1; \; \dots; \; q_i\beta_i; \; \dots$$

where all the α's and β's are nonempty. This gives the third summand.   □

Lemma 5.11 and 5.12 allow us to define the finite Markov chain M_Δ.

Definition 5.14. The finite-state Markov chain M_Δ has the following set of states

$$\{(qY,a) \mid qY \in \mathcal{H}(\Delta), a \in A, \mathcal{P}(V_{qY}^{(1)}=(qY,a_0)) > 0\} \cup \mathcal{H}(\Delta) \cup \{\bot\}$$

and the following transition probabilities:

- $Prob(\bot \rightarrow \bot) = 1$,
- $Prob(pX \rightarrow (qY,a_0)) = \mathcal{P}(V_{pX}^{(1)}=(qY,a_0))$,
- $Prob(pX \rightarrow \bot) = \mathcal{P}(V_{pX}^{(1)}=\bot)$,
- $Prob((qY,a) \rightarrow (q'Y',a')) = \mathcal{P}(V_{qY}^{(2)}=(q'Y',a') \mid V_{qY}^{(1)}=(qY,a_0))$.

One can readily check that M_Δ is indeed a Markov chain, i.e., for every state s of M_Δ we have that the sum of probabilities of all outgoing transitions of s is equal to one. Observe also that if both (qY,a) and (qY,a') are states of M_Δ, then they have the "same" outgoing arcs (i.e., $(qY,a) \xrightarrow{x} (rZ,\bar{a})$ iff $(qY,a') \xrightarrow{x} (rZ,\bar{a})$, where $x > 0$).

Example 5.15. We construct the Markov Chain M_Δ for the pBPA Δ of Figure 1 and the observing automaton $\mathcal{A}$ of Figure 4. In fact, as we shall see, the states and transition probabilities of the chain depend on the value of the parameter x.

Since the pBPA has one single control state, we omit it. The set of heads is then $\mathcal{H}(\Delta) = \{Z,I,D\}$ and the set of states of the observing automaton is $A = \{a_0, a_1\}$. In order to determine the states of the Markov chain we have to compute the pairs (Y,a) such that $\mathcal{P}(V_Y^{(1)} = (Y,a)) \geq 0$. Recall the definition of $\mathcal{P}(V_Y^{(1)} = (Y,a))$. This is the probability of, starting at the configuration Y, executing an infinite run such that (i) the head of the first minimum is Y, and (ii) the first observation of $\mathcal{A}$ is the state a. Since the initial configuration Y has the shortest possible length in an infinite run, (i) always holds. So $\mathcal{P}(V_Y^{(1)} = (Y,a))$ is the probability of executing an infinite run such that (ii) holds. Recall that the first observation of an observing automaton is the state it reaches after reading the sequence of heads between the first and the second minimum, excluding the first, but including the second. In the case of the automaton $\mathcal{A}$ of Figure 4, the first observation is a_0 if the sequence of heads does not contain the head Z, and a_1 otherwise.

The values of $\mathcal{P}(V_X^{(1)} = (X,a))$ for $X \in \{Z,I,D\}$ and $a \in \{a_0,a_1\}$ are as follows:

$$\mathcal{P}(V_X^{(1)} = (X,a)) = \begin{cases} \min\{2x, 2-2x\} & \text{if} \quad X = Z \quad \text{and} \quad a = a_1 \\ \max\{0, (2x-1)/x\} & \text{if} \quad X = I \quad \text{and} \quad a = a_0 \\ \max\{0, (1-2x)/(1-x)\} & \text{if} \quad X = D \quad \text{and} \quad a = a_0 \\ 0 & \text{otherwise} \end{cases}$$

These values can be obtained using the definitions, but in this simple case we can also use more direct methods. Consider for instance $\mathcal{P}(V_Z^{(1)} = (Z,a_1))$. This is the probability of, starting at Z, executing an infinite run and visiting again a configuration with head Z before reaching the second minimum. Observe that all runs that start at Z are infinite, that the only configuration they visit with head Z is Z itself, and that Z is always a minimum. So $\mathcal{P}(V_Z^{(1)} = (Z,a_1))$ is the probability of, starting at the configuration Z, eventually reaching Z again. This probability is equal to $x \cdot [I,\varepsilon] + (1-x) \cdot [D,\varepsilon]$, where $[I,\varepsilon]$ and $[D,\varepsilon]$ are defined in Example 3.6. We get

$$\begin{aligned}
\mathcal{P}(V_Z^{(1)} = (Z, a_1)) &= x \cdot [I, \varepsilon] + (1 - x) \cdot [D, \varepsilon] \\
&= x \cdot \min\{1, (1 - x)/x\} + (1 - x) \cdot \min\{1, x/(1 - x)\} \\
&= \min\{2x, 2 - 2x\}
\end{aligned}$$

Observe that the states of M_Δ depend on x. The states are $\bot, Z, I, D$ and

$$\begin{array}{ll}
(D, a_0) & \text{if } x = 0, \\
(Z, a_1), (D, a_0) & \text{if } 0 < x < 1/2, \\
(Z, a_1) & \text{if } x = 1/2, \\
(Z, a_1), (I, a_0) & \text{if } 1/2 < x < 1, \\
(I, a_0) & \text{if } x = 1.
\end{array}$$

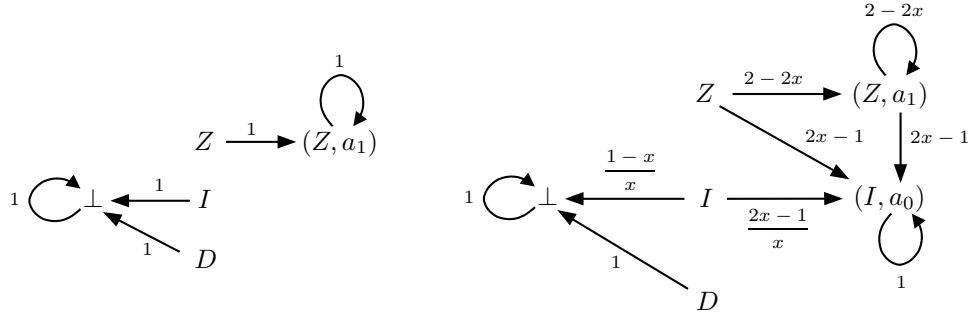The Markov chain for the cases $x = 1/2$ and $1/2 < x < 1$ are shown in Figure 5.

Figure 5: The Markov chain M_Δ for $x = 1/2$ (left) and for $1/2 < x < 1$ (right)

Let us obtain the transition probability from (I, a_0) to itself in the case $1/2 < x < 1$. According to Definition 5.14, the probability is equal to $\mathcal{P}(V_I^{(2)} = (I, a_0) \mid V_I^{(1)} = (I, a_0))$, i.e., to the probability of, assuming the first minimum has head I, reaching the second minimum at head I again, visiting no configuration with head Z in-between. Let us see that this probability is 1. If the first minimum is $I\alpha$ for some $\alpha \in \{Z, I, D\}^*$, then all subsequent configurations of the run are of the form $\beta\alpha$ for a nonempty β (notice that we assume that the run is infinite, because finite runs have no minima). So β must have head I and so, in particular, the next minimum will also have head I. □

Not every run of Δ is "represented" in the Markov chain M_Δ. Consider for instance the case $x = 1/2$ and its corresponding chain M_Δ on the left of Figure 5. Every configuration of the run $Z; IZ; IIZ; IIIZ; \ldots$ is a minimum, but its sequence of heads, i.e., ZI^ω, does not correspond to any path of M_Δ. We show, however, that the "not represented" runs have probability 0.

A *trajectory* in M_Δ is an infinite sequence $\sigma(0)\sigma(1)\cdots$ of states of M_Δ where $Prob(\sigma(i) \to \sigma(i+1)) > 0$ for every $i \in \mathbb{N}_0$. To every run $w \in Run(pX)$ of Δ we associate its *footprint*, denoted σ_w, which is an infinite sequence of states of M_Δ defined as follows:

- $\sigma_w(0) = pX$
- if w is finite, then for every $i \in \mathbb{N}$ we have $\sigma_w(i) = \bot$;
- if w is infinite, then for every $i \in \mathbb{N}$ we have $\sigma_w(i) = (p_i X_i, Obs_i(w))$, where $p_i X_i$ is the head of $\min_i(w)$.

We say that a given $w \in Run(pX)$ is *good* if σ_w is a trajectory in M_Δ. Our next lemma reveals that almost all runs are good.

Lemma 5.16. *Let $pX \in \mathcal{H}(\Delta)$, and let Good be the subset of all good runs of $Run(pX)$. Then $\mathcal{P}(Good) = 1$.*

Proof. Let $Bad = Run(pX) \setminus Good$. Let $Fail$ be the set of all finite sequences $v_0 \cdots v_{i+1}$ of states of M_Δ such that $i \in \mathbb{N}_0$, $v_0 = pX$, $v_0 \cdots v_i$ is a trajectory in M_Δ, and $Prob(v_i \to v_{i+1}) = 0$, where $Prob$ is the probability assignment of M_Δ. Each $y \in Fail$ determines a set $Bad_y = \{w \in Bad \mid \sigma_w \text{ starts with } y\}$. Obviously, $Bad = \biguplus_{y \in Fail} Bad_y$. We prove that $\mathcal{P}(Bad_y) = 0$ for each $y \in Fail$. Let $y = v_0 \cdots v_{i+1}$. By applying definitions, we obtain

$$
\begin{aligned}
\mathcal{P}(Bad_y) &= \mathcal{P}(V_{pX}^{(1)}{=}v_1 \wedge \cdots \wedge V_{pX}^{(i+1)}{=}v_{i+1}) \\[2mm]
&= \frac{\mathcal{P}(V_{pX}^{(i+1)}{=}v_{i+1} \mid V_{pX}^{(i)}{=}v_i \wedge \cdots \wedge V_{pX}^{(1)}{=}v_1)}{\mathcal{P}(V_{pX}^{(i)}{=}v_i \wedge \cdots \wedge V_{pX}^{(1)}{=}v_1)}
\end{aligned}
$$

Since $\mathcal{P}(V_{pX}^{(i)}{=}v_i \wedge \cdots \wedge V_{pX}^{(1)}{=}v_1) \neq 0$, the last fraction makes sense and it is equal to

$$
\frac{Prob(v_i \to v_{i+1})}{\mathcal{P}(V_{pX}^{(i)}{=}v_i \wedge \cdots \wedge V_{pX}^{(1)}{=}v_1)}
$$

which equals zero. $\qquad\square$

5.3. $\mathcal{P}(Run(pX, Acc))$ is effectively definable in $(\mathbb{R}, +, *, \leq)$. Recall that our aim is to show that $\mathcal{P}(Run(pX, Acc))$ is effectively definable in $(\mathbb{R}, +, *, \leq)$. We will achieve this in Theorem 5.22 as an easy corollary of Lemma 5.20. This lemma states that $\mathcal{P}(pX, Acc)$ is the probability of, starting at pX, hitting so-called *accepting* bottom strongly connected component of M_Δ. As usual, a strongly connected component of M_Δ is a maximal set of mutually reachable states, and bottom strongly connected components are those from which no other strongly connected components can be reached.

Definition 5.17. Let C be a bottom strongly connected component of M_Δ. We say that C is *accepting* if $C \neq \{\perp\}$ and the set $\{a \in A \mid (qY, a) \in C \text{ for some } qY \in \mathcal{H}(\Delta)\}$ is an element of Acc (remember that Acc is the acceptance set introduced after Definition 5.4). Otherwise, C is *rejecting*.

We say that a given pair (qY, a), where $qY \in \mathcal{H}(\Delta)$ and $a \in A$, is *recurrent*, if it belongs to some bottom strongly connected component of M_Δ.

We say that a run $w \in Run(pX)$ *hits* a pair $(qY, a) \in \mathcal{H}(\Delta) \times A$ if there is some $i \in \mathbb{N}$ such that the head of $\min_i(w)$ is qY and $Obs_i(w) = a$. The next lemma says that an infinite run eventually hits a recurrent pair. In this lemma and the next we use the following well-known results for finite Markov chains (see e.g. [Fel66]):

- A run visits some bottom strongly connected component of the chain with probability 1.
- If a run visits some state of a bottom strongly connected component C, then it visits all states of C infinitely often with probability 1.

Lemma 5.18. *Let us assume that $\mathcal{P}(IRun(pX)) > 0$. Then the conditional probability that $w \in Run(pX)$ hits a recurrent pair on the hypothesis that w is infinite is equal to one.*

Proof. Let *Rec* denote the event that a run of $Run(pX)$ hits a recurrent pair. Due to Lemma 5.16, we have that

$$\mathcal{P}(Rec \mid IRun(pX)) = \mathcal{P}(Rec \mid IRun(pX) \cap Good) \qquad (5.6)$$

A run belongs to $IRun(pX) \cap Good$ iff its footprint is a trajectory in M_Δ that does not hit the state $\perp$. A run $w \in IRun(pX) \cap Good$ satisfies *Rec* iff its footprint hits (some) recurrent pair (qY, a). It follows directly from the definition of M_Δ that the right-hand side of equation (5.6) is equal to the probability that a trajectory from pX in M_Δ hits a bottom strongly connected component on the hypothesis that the state $\perp$ is not visited. Since M_Δ is finite, this happens with probability one. $\qquad\square$

So, an infinite run eventually hits a recurrent pair. Now we prove that if this pair belongs to an accepting/rejecting bottom strongly connected component of M_Δ, then the run will be accepting/rejecting with probability one.

Lemma 5.19. *The conditional probability that $w \in Run(pX)$ is accepting/rejecting on the hypothesis that the first recurrent pair hit by w belongs to an accepting/rejecting bottom strongly connected component of M_Δ is equal to one.*

Proof. The argument is similar as in the proof of Lemma 5.18. Let C be a bottom strongly connected component of M_Δ. By ergodicity, the conditional probability that an infinite trajectory in M_Δ hits each state of C infinitely often on the hypothesis that the trajectory hits C is equal to one. $\qquad\square$

A simple consequence of Lemma 5.19 is:

Lemma 5.20. *(cf. Proposition 4.1.5 of* [CY95]*) Let $pX \in \mathcal{H}(\Delta)$. $\mathcal{P}(pX, Acc)$ is equal to the probability that a trajectory from pX in M_Δ hits an accepting bottom strongly connected component of M_Δ.*

Example 5.21. Consider the pBPA of Figure 1 and the observing automaton of Figure 4. $\mathcal{P}(Z, Acc)$ is the probability of, starting at Z, executing a run that visits configurations with head Z infinitely often. In the case $x = 1/2$, the bottom strongly connected components of M_Δ are $\{\perp\}$ and $\{(Z, a_1)\}$, which are rejecting and accepting, respectively. Starting at the state Z of M_Δ, the probability of hitting $\{(Z, a_1)\}$ is 1, and so $\mathcal{P}(Z, Acc) = 1$. In the case $1/2 < x < 1$, the bottom strongly connected components of M_Δ are $\{\perp\}$ and $\{(I, a_0)\}$, which are both rejecting, and so $\mathcal{P}(Z, Acc) = 0$.

Since the probability of hitting a given bottom strongly connected component of a given finite-state Markov chain is effectively definable in $(\mathbb{R}, +, *, \leq)$ by the results of Section 3, and the transition probabilities in M_Δ are well-definable too, we can conclude the following:

Theorem 5.22. $\mathcal{P}(Run(pX, Acc))$ *is effectively expressible in $(\mathbb{R}, +, *, \leq)$. In particular, for every rational constant y and every $\sim \in \{\leq, <, \geq, >, =\}$ there effectively exists a formula of $(\mathbb{R}, +, *, \leq)$ which holds iff $\mathcal{P}(Run(pX, Acc)) \sim y$.*

5.4. **Decidability of ω-regular properties.** As a simple corollary of Theorem 5.22, we obtain the decidability of the qualitative/quantitative model-checking problem for pPDA and ω-regular properties. Recall that a language of infinite words over a finite alphabet is *ω-regular* iff it can be accepted by a (deterministic) Muller automaton.

Definition 5.23. A deterministic *Muller automaton* is a tuple $\mathcal{B} = (\Sigma, B, \varrho, b_I, \mathcal{F})$, where Σ is a finite *alphabet*, B is a finite set of *states*, $\varrho \colon B \times \Sigma \to B$ is a (total) *transition function* (we write $b \xrightarrow{a} b'$ instead of $\varrho(b, a) = b'$), b_I is the *initial state*, and $\mathcal{F} \subseteq 2^B$ is a set of *accepting sets*.

An infinite word w over the alphabet Σ is *accepted* by $\mathcal{B}$ if $Inf(w) \in \mathcal{F}$, where $Inf(w)$ is the set of all $b \in B$ that appear infinitely often in the unique run of $\mathcal{B}$ over the word w.

We consider specifications given by Muller automata having $\mathcal{H}(\Delta)$ as their alphabet. It is well known that every LTL formula whose atomic propositions are interpreted over simple sets can be encoded into a deterministic Muller automaton having $\mathcal{H}(\Delta)$ as alphabet. Our results can be extended to atomic propositions interpreted over arbitrary regular sets of configurations using the same technique as in [EKS03].

Let us fix a deterministic Muller automaton $\mathcal{B} = (\mathcal{H}(\Delta), B, \varrho, b_I, \mathcal{F})$. An infinite run w of $\mathcal{T}_\Delta$ is *accepted by* $\mathcal{B}$ if the associated sequence of heads of configurations in w is accepted by $\mathcal{B}$. Let $Run(pX, \mathcal{B})$ be the set of all $w \in Run(pX)$ that are accepted by $\mathcal{B}$. We show that $Run(pX, \mathcal{B})$ is effectively expressible in $(\mathbb{R}, +, *, \leq)$, and so we can decide if it is larger than, smaller than, or equal to some threshold ρ.

Loosely speaking, we proceed as follows. We compute the synchronized product Δ' of Δ and $\mathcal{B}$. Then, we define a Δ'-observing automaton $\mathcal{A}$ whose states are sets of states of $\mathcal{B}$. The automaton observes heads of Δ', which are of the form $(p, b)X$, where pX is a head of Δ and b is a state of b. At the end of a "jump", $\mathcal{A}$ returns the set of states of B that were visited during the jump. Hence, the observation $Obs(w)$ of the automaton on a run w is a sequence $B_1 B_2 \ldots$ of sets of states of B containing full information about which states were visited in which jump. Now it is just a matter of setting the acceptance set of $\mathcal{A}$ adequately: The acceptance sets of $\mathcal{A}$ are the sets $\{b_1, \ldots, b_n\}$ of states of $\mathcal{A}$ such that the union $b_1 \cup \ldots \cup b_n$ is an element of $\mathcal{F}$.

Theorem 5.24. $\mathcal{P}(Run(pX, \mathcal{B}))$ *is effectively expressible in* $(\mathbb{R}, +, *, \leq)$. *In particular, for every rational constant y and every $\sim \in \{\leq, <, \geq, >, =\}$ there effectively exists a formula of $(\mathbb{R}, +, *, \leq)$ which holds iff $\mathcal{P}(Run(pX, \mathcal{B})) \sim y$. (Hence, for each $0 < \lambda < 1$ we can compute rationals $\mathcal{P}^\ell, \mathcal{P}^u$ such that $\mathcal{P}^\ell \leq \mathcal{P}(pX, Acc) \leq \mathcal{P}^u$ and $\mathcal{P}^u - \mathcal{P}^\ell \leq \lambda$.)*

Proof. Let $\Delta' = (Q \times B, \Gamma, \delta', Prob')$ be the synchronized product of Δ and $\mathcal{B}$, i.e., $(p, b)X \xrightarrow{x} (t, b')\alpha$ is a rule of Δ' iff $pX \xrightarrow{x} t\alpha$ is a rule of Δ and $\varrho(b, pX) = b'$. Consider the Δ'-observing automaton $\mathcal{A} = (A, \xi, \mathcal{I}, Acc)$ where $A = 2^B$, $a_0 = \emptyset$, $\xi(M, (p, b)Y) = M \cup \{b\}$ for all $M \subseteq B$ and $(p, b)Y \in \mathcal{H}(\Delta')$, and Acc is defined as follows: for every $a_1, \ldots, a_n \in 2^B$, $\{a_1, \ldots, a_n\} \in Acc$ iff $a_1 \cup \ldots \cup a_n \in \mathcal{F}$.

It is easy to check that

$$\mathcal{P}(Run(pX, \mathcal{B})) = \mathcal{P}(Run((p, b_I)X, Acc))$$

Now it suffices to apply Theorem 5.22. $\qquad\square$

6. Conclusions

We have provided model checking algorithms for probabilistic pushdown automata against PCTL specifications, and against ω-regular specifications represented by Muller automata. Contrary to the case of probabilistic finite automata, qualitative properties (i.e., whether a property holds with probability 0 or 1), depend on the exact values of transition probabilities.

There are many possibilities for future work. An obvious question is what is the complexity of the obtained algorithms. Of course, this depends on the complexity of the corresponding fragments of first order arithmetic of reals. It is known that the fragment obtained by fixing the alternation depth of quantifiers is decidable in exponential time [Gri88], and that the existential fragment (and hence also the universal fragment) is decidable even in polynomial space [Can88]. The formulas constructed in Section 3 have a fixed alternation depth, and so we can conclude that the qualitative/quantitative random walk problem is decidable in exponential time. Actually, we can do even better—if we are interested whether $\mathcal{P}(pX, \mathcal{C}_1 \, \mathcal{U} \, \mathcal{C}_2) \leq \varrho$, we can simply ask if there is *some* solution of the corresponding system of quadratic equations (cf. Theorem 3.5) such that the component of the solution which corresponds to $\mathcal{P}(pX, \mathcal{C}_1 \, \mathcal{U} \, \mathcal{C}_2)$ is less than or equal to ϱ. Obviously, the minimal solution (i.e., the probability of $\mathcal{P}(pX, \mathcal{C}_1 \, \mathcal{U} \, \mathcal{C}_2)$) can only be smaller. Hence, the existential fragment is sufficient for deciding whether $\mathcal{P}(pX, \mathcal{C}_1 \, \mathcal{U} \, \mathcal{C}_2) \leq \varrho$, and similarly we can use the universal fragment to decide whether $\mathcal{P}(pX, \mathcal{C}_1 \, \mathcal{U} \, \mathcal{C}_2) \geq \varrho$. To sum up, the problem whether $\mathcal{P}(pX, \mathcal{C}_1 \, \mathcal{U} \, \mathcal{C}_2) \sim \varrho$, where $\sim \in \{<, \leq, >, \geq, =\}$, is decidable in polynomial space.

Recently, deeper results concerning the complexity of the reachability problem for pPDA and pBPA have been presented by Etessami and Yannakakis in [EY05]. In particular, they show that the *qualitative* reachability problem for pBPA processes (i.e., the question whether a given configuration is visited with probability 1) is decidable in polynomial time. It is also shown that the SQUARE-ROOT-SUM problem (i.e., the question whether $\sum_{i=1}^{n} \sqrt{a_i} \leq c$ for a given tuple $(a_1, \ldots, a_n, c)$ of natural numbers) is polynomially reducible to the *quantitative* reachability problem for pBPA, and to the *qualitative* reachability problem for pPDA. The complexity of the SQUARE-ROOT-SUM problem is a famous open problem in the area of exact numerical algorithms. It is known that the problem is solvable in polynomial space, but no lower bound (like NP or co-NP hardness) is known. This means that the PSPACE upper bound for the quantitative pBPA reachability and the qualitative pPDA reachability cannot be improved without achieving an improvement in the complexity of the SQUARE-ROOT-SUM problem.

Some of the problems which were left open in [EKM04] were solved later in [BKS05]. It was shown that the model-checking problems for PCTL and pPDA, and for PCTL* and pBPA, are undecidable (PCTL* is the probabilistic extension of CTL*). On the other hand, the decidability result about qualitative/quantitative model-checking pPDA against deterministic Büchi specifications was extended to Muller automata. In the qualitative case, the algorithm runs in time which is singly exponential in the size of a given pPDA and a given Muller automaton. In the quantitative case, the algorithm needs exponential space. Finally, it was shown that the model-checking problem for the qualitative fragment of the logic PECTL* and pPDA processes is also decidable. The complexity bounds are essentially the same as for Muller properties.

The complexity of model-checking ω-regular properties (encoded by Büchi automata) for pPDA and pBPA processes was studied also in [EY]. The complexity bounds improve the ones given in [BKS05]. In particular, it is shown that the qualitative model-checking problem for pPDA and Büchi specifications is EXPTIME-complete.

An interesting open problem is the decidability of the model-checking problem for PCTL and pBPA processes, i.e., whether there is an "exact" algorithm apart from the error-tolerant one given in Section 4.2. Another area of open problems is generated by considering model-checking problems for a more general class of pushdown automata whose

underlying semantics is defined in terms of Markov decision processes (this model combines the paradigms of non-deterministic and probabilistic choice).

7. Acknowledgments

References

[ABIJ05] P.A. Abdulla, C. Baier, S.P. Iyer, and B. Jonsson. Simulating perfect channels with probabilistic channel systems. *Information and Computation*, 197(1–2):22–40, 2005.

[AEY01]   R. Alur, K. Etessami, and M. Yannakakis. Analysis of recursive state machines. In *Proceedings of CAV 2001*, volume 2102 of *Lecture Notes in Computer Science*, pages 207–220. Springer, 2001.

[AMP99]   A. Abney, D. McAllester, and F. Pereira. Relating probabilistic grammars and automata. In *Proceedings of ACP'99*, pages 542–549, 1999.

[AR03]   P.A. Abdulla and A. Rabinovich. Verification of probabilistic systems with faulty communication. In *Proceedings of FoSSaCS 2003*, volume 2620 of *Lecture Notes in Computer Science*, pages 39–53. Springer, 2003.

[ASB+95] A. Aziz, V. Singhal, F. Balarin, R. Brayton, and A. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In *Proceedings of CAV'95*, volume 939 of *Lecture Notes in Computer Science*, pages 155–165. Springer, 1995.

[BE99]   C. Baier and B. Engelen. Establishing qualitative properties for probabilistic lossy channel systems: an algorithmic approach. In *Proceedings of 5th International AMAST Workshop on Real-Time and Probabilistic Systems (ARTS'99)*, volume 1601 of *Lecture Notes in Computer Science*, pages 34–52. Springer, 1999.

[BEM97]   A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: application to model checking. In *Proceedings of CONCUR'97*, volume 1243 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 1997.

[BKS05]   T. Brázdil, A. Kučera, and O. Stražovský. On the decidability of temporal properties of probabilistic pushdown automata. In *Proceedings of STACS'2005*, volume 3404 of *Lecture Notes in Computer Science*, pages 145–157. Springer, 2005.

[BS97]   O. Burkart and B. Steffen. Model checking the full modal mu-calculus for infinite sequential processes. In *Proceedings of ICALP'97*, volume 1256 of *Lecture Notes in Computer Science*, pages 419–429. Springer, 1997.

[BS03]   N. Bertrand and Ph. Schnoebelen. Model checking lossy channel systems is probably decidable. In *Proceedings of FoSSaCS 2003*, volume 2620 of *Lecture Notes in Computer Science*, pages 120–135. Springer, 2003.

[BW90]   J.C.M. Baeten and W.P. Weijland. *Process Algebra*. Number 18 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1990.

[Can88]   J. Canny. Some algebraic and geometric computations in PSPACE. In *Proceedings of STOC'88*, pages 460–467. ACM Press, 1988.

[CSS03]   J.M. Couvreur, N. Saheb, and G. Sutre. An optimal automata approach to LTL model checking of probabilistic systems. In *Proceedings of LPAR 2003*, volume 2850 of *Lecture Notes in Computer Science*, pages 361–375. Springer, 2003.

[CY88]   C. Courcoubetis and M. Yannakakis. Verifying temporal properties of finite-state probabilistic programs. In *Proceedings of FOCS'88*, pages 338–345. IEEE Computer Society Press, 1988.

[CY95]   C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *Journal of the Association for Computing Machinery*, 42(4):857–907, 1995.

[EHRS00] J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon. Efficient algorithms for model checking pushdown systems. In *Proceedings of CAV 2000*, volume 1855 of *Lecture Notes in Computer Science*, pages 232–247. Springer, 2000.

[EKM04]   J. Esparza, A. Kučera, and R. Mayr. Model-checking probabilistic pushdown automata. In *Proceedings of LICS 2004*, pages 12–21. IEEE Computer Society Press, 2004.

[EKS03]   J. Esparza, A. Kučera, and S. Schwoon. Model-checking LTL with regular valuations for pushdown systems. *Information and Computation*, 186(2):355–376, 2003.

[EY]   K. Etessami and M. Yannakakis. Algorithmic verification of recursive probabilistic systems. Technical Report, School of Informatics, U. of Edinburgh, 2005.

[EY05]   K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proceedings of STACS'2005*, volume 3404 of *Lecture Notes in Computer Science*, pages 340–352. Springer, 2005.

[Fel66]   W. Feller. *An Introduction to Probability Theory and Its Applications*. Wiley & Sons, 1966.

[Gri88]   D. Grigoriev. Complexity of deciding Tarski algebra. *Journal of Symbolic Computation*, 5(1–2):65–108, 1988.

[HJ94]   H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.

[HK97]   M. Huth and M.Z. Kwiatkowska. Quantitative analysis and model checking. In *Proceedings of LICS'97*, pages 111–122. IEEE Computer Society Press, 1997.

[HS84]   S. Hart and M. Sharir. Probabilistic temporal logic for finite and bounded models. In *Proceedings of POPL'84*, pages 1–13. ACM Press, 1984.

[IN97]   S.P. Iyer and M. Narasimha. Probabilistic lossy channel systems. In *Proceedings of TAPSOFT'97*, volume 1214 of *Lecture Notes in Computer Science*, pages 667–681. Springer, 1997.

[LS82]   D. Lehman and S. Shelah. Reasoning with time and chance. *Information and Control*, 53:165–198, 1982.

[MO98]   I. Macarie and M. Ogihara. Properties of probabilistic pushdown automata. *Theoretical Computer Science*, 207:117–130, 1998.

[Rab03]   A. Rabinovich. Quantitative analysis of probabilistic lossy channel systems. In *Proceedings of ICALP 2003*, volume 2719 of *Lecture Notes in Computer Science*, pages 1008–1021. Springer, 2003.

[Tar51]   A. Tarski. *A Decision Method for Elementary Algebra and Geometry*. Univ. of California Press, Berkeley, 1951.

[Var85]   M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of FOCS'85*, pages 327–338. IEEE Computer Society Press, 1985.

[Wal01]   I. Walukiewicz. Pushdown processes: Games and model-checking. *Information and Computation*, 164(2):234–263, 2001.

INSTITUTE FOR FORMAL METHODS IN COMPUTER SCIENCE, UNIVERSITY OF STUTTGART, UNIVERSITÄT STR. 38, 70569 STUTTGART, GERMANY.
   E-mail address: esparza@informatik.uni-stuttgart.de

FACULTY OF INFORMATICS, MASARYK UNIVERSITY, BOTANICKÁ 68A, CZ-60200 BRNO, CZECH REPUBLIC
   E-mail address: tony@fi.muni.cz

DEPARTMENT OF COMPUTER SCIENCE, NORTH CAROLINA STATE UNIVERSITY, 900 MAIN CAMPUS DRIVE, CAMPUS BOX 8207, RALEIGH NC 27695, USA.
   E-mail address: mayr@csc.ncsu.edu