Masaryk University
Faculty of Informatics

Verification of Probabilistic Recursive Sequential Programs

Tomáš Brázdil

Ph.D. Thesis

2007

Abstract

This work studies algorithmic verification of infinite-state probabilistic systems generated by probabilistic pushdown automata (pPDA). Probabilistic pushdown automata are obtained as a probabilistic variant of pushdown automata that proved to be a successful abstract model of recursive sequential programs.

The main aim of this work is to study decidability and complexity of the problem whether a given probabilistic system generated by a pPDA satisfies a given property expressed in a suitable formalism. There are plenty of formalisms available for specifying properties of probabilistic systems. In this work we consider various temporal properties expressed by finite-state automata on infinite words and formulae of temporal logics, long-run average properties, and properties connected with expected behavior.

Concerning temporal logics, we consider both linear and branching time ones. Among others we consider linear temporal logic (LTL) and probabilistic computation tree logic (PCTL), which is a probabilistic variant of the well-known logic CTL. We also consider a general logic PECTL*, which combines automata based specifications with the structure of branching-time temporal logics. We provide a systematic presentation of results on decidability and complexity of the model-checking problem for these logics together with a detailed presentation of some verification techniques. Possibly the most important original results of this work concerning temporal logics are decidability of the model-checking problem for the qualitative fragment of PECTL* and undecidability of the same problem for PCTL.

The class of long-run average properties is very useful in performance and reliability analysis of systems. These properties are tightly connected with service cycles and can be used to answer questions like: Is the average time of servicing a request lower than a given bound? In this work we define a whole class of such properties, and study decidability and complexity of the problem whether a given system satisfies these properties. We also study a notion of predictability of long-run average properties based on observing a finite part of a computation of the system. We show that all properties from our class are effectively predictable up to a given error tolerance.

Acknowledgements

First and foremost, I want to thank Antonín Kučera, my supervisor, for his exceptional guidance, outstanding collaboration, friendly approach, and for reading a preliminary draft of this thesis. I would also like to thank all other collaborators with whom I worked during my PhD study: Javier Esparza, Olřich Stražovský, Václav Brožek, and Vojtěch Forejt. Special thanks go to Javier Esparza for his kind approach and for his willingness to share his great ideas with me, and to Václav Brožek for his careful reading of a preliminary draft of this thesis.

I would like to thank Mojmír Křetínský, Jan Obdržálek, and Vojtěch Řehák for their moral support and also for many thought provoking discussions during lunch time. Last but not least I would like to thank my wife Luisa for her encouragement and impatience which made me work at least twice as effectively as usual.

Tomáš Brázdil

Contents

Chapter 1

Introduction

Probabilistic methods are widely used in the design, analysis, and verification of computer systems that exhibit some kind of "quantified uncertainty" such as coin-tossing in randomized algorithms, subsystem failures (caused, e.g., by communication errors or bit flips with an empirically evaluated probability), or underspecification in some components of the system [40]. The underlying semantic models are usually Markov chains or Markov decision processes, depending mainly on whether the systems under consideration are sequential or parallel.

Verification methods have been developed mainly for finite-state Markov chains and finite-state Markov decision processes [22, 50, 35, 21, 23, 24]. This is certainly a limitation, because many implementations use unbounded data structures (counters, queues, stacks, etc.) that cannot always be faithfully abstracted into finite-state models. The question whether one can go beyond this limit has been rapidly gaining importance and attention in recent years. Positive results exist mainly for probabilistic lossy channel systems [7, 12, 36, 44, 2]. Examples of more generic results are [1, 45, 16]. Very recently, probabilistic aspects of recursive sequential programs have also been taken into account [27, 17, 33, 31, 28, 32, 34, 15]. A part of this thesis is based on results of these papers.

In the non-probabilistic setting, the literature offers two natural models for recursive sequential programs:

- *pushdown automata (PDA)*, see, e.g., [26, 30, 53, 5], where the stack symbols correspond to individual procedures and their local data, and the global data is modeled in the finite-state control;
- *recursive state machines (RSM)*, see, e.g., [4, 3], where the behavior of each procedure is specified by a finite-state automaton which can possibly invoke the computation of another automaton in a recursive fashion.

Since PDA and RSM are fully equivalent (in a well-defined sense) and there are linear-time translations between them, the results achieved for one model immediately apply to the other. A practical impact of these results can be documented by successful applications of software tools [8, 9].

Formal models for probabilistic recursive programs are obtained as probabilistic variants of PDA (probabilistic pushdown automata, pPDA) and RSM (recursive Markov chains, RMCs). The underlying semantics is given in terms of infinite-state Markov chains [39], and the two models are equivalent with respect to this semantics (see Section 3.3). We formulate all results in terms of probabilistic pushdown automata; however, all of these results can easily be translated into the recursive Markov chains. Both pPDA and RMCs are defined in Chapter 3 together with a formal proof of equivalence of their expressive power.

Properties of probabilistic systems can be specified using various formalisms. Some of them, e.g., various temporal logics and automata based specifications, originated in the non-probabilistic setting. In addition, there are some specification formalisms developed specifically for probabilistic systems such as, e.g., the expected accumulated reward and long-run average properties.

Temporal logic specifications

Into this class of specification formalisms we count not only temporal logics but also automata based specifications and reachability. Typical properties that one can express using these formalisms are "the probability of termination is at least 0.98", "the probability that each request will eventually be granted is 1", etc. Depending on their treatment of time, the specification formalisms can roughly be divided into linear-time and branching-time ones. In the case of linear time, each moment in time has a unique possible future, while in branching time, each moment in time may split into several possible futures. In this work, we consider both linear and branching time specifications.

The simplest example of a linear-time specification is reachability. In the probabilistic setting, we are interested in the probability of reaching a state (a set of states) of a Markov chain from another one. Among more sophisticated linear-time formalisms, the linear temporal logic (LTL), and ω-regular properties (i.e. properties expressible by finite state automata over infinite words) are the most successful ones. Similarly to the case of reachability, we are interested in the probability measure of all runs that satisfy a given LTL formula or are accepted by a given automaton. This probabilistic interpretation can be seen as a refinement of the usual problem whether all runs satisfy a given specification (we ask not only whether all/no runs satisfy a given specification but also "how many" of them satisfy the specification). Verification of linear-time properties of probabilistic recursive programs was studied in [27, 17] for pPDA and in [31, 32, 33] for RMCs. Results of these papers are surveyed in Chapter 4 in the unifying framework of pPDA. Some of these results are only mentioned without a proof. However, we formally prove all results that were either co-authored by the author of this thesis, or are needed in the model-checking of branching-time properties.

A popular branching-time temporal logic suitable for specification of probabilistic systems is PCTL [35]. The logic PCTL is a probabilistic variant of the well-known temporal logic CTL [20], which substitutes path quantifiers "A" and

"E" of CTL with bounds on the probability measure of all paths that satisfy a given subformula. In [17] we considered a strong generalization of PCTL called PECTL* that combines branching-time operators with ω-regular properties. The logic PECTL* subsumes not only PCTL, but even PCTL* which is a probabilistic variant of the temporal logic CTL* [25]. Decidability of the model-checking problem for probabilistic recursive programs and branching-time logics was first studied in [27]. In [17] results of [27] were substantially extended and some complexity estimations were provided. Many of these results were further improved after publishing these papers. The aim of this work is to give as complete picture as possible of known results connected with the model-checking problem for probabilistic PDA and branching-time temporal logics.

Expectations and limit properties

Besides properties expressible in temporal logics, we study expected behavior of Markov chains and certain *limit properties* of runs that are explained below.

By the expected behavior of Markov chains we mean the expected accumulated reward along paths between a given state and a given set of states. The reward is expressed by a reward function that assigns a value to every state of the chain. A special case of the expected accumulated reward is the expected number of steps needed to reach a set of states from a given state, a value well-known from basic theory of Markov chains. Besides this, general reward functions can model for example values of certain variables, time spent in particular states, height of the stack of recursive calls in procedural programs, etc. For recursive probabilistic programs, the problem of determining the expected accumulated reward was first studied in [28] for the so called simple and linear reward functions. Simple reward functions assign rewards to configurations of a pPDA based only on their control states, while linear reward functions take into account the actual stack content in a "linear" manner. We generalize results of this paper to well-defined reward functions that have both simple and linear reward functions as their special cases.

The limit properties studied in this work express the average behavior of a given probabilistic sequential program (pPDA) along a long run (long-run average properties). An example of such a property is "What is the probability that the average time for performing a given procedure/service is less than 20 seconds?". These properties are useful especially in reliability and performance evaluation of systems. Long-run properties of pPDA were first studied in [28], and then in [15]. In this work, we follow [15], and consider an important class of long-run properties related to service cycles together with ways to efficiently *predict* them after performing (and observing) a bounded initial prefix of a run. Intuitive description of these properties together with their formal definitions are given at the beginning of Chapter 7.

1.1 Aims of the Thesis

The main goal of this thesis is to study the decidability and complexity of the verification problem for probabilistic recursive sequential programs modeled as pPDA and various specification formalisms. Besides a survey of known results, the work presents several original contributions of the author to the field:

1. We prove the decidability of the quantitative model-checking problem for ω-regular properties. Moreover, we show that the set of all configurations that satisfy a given qualitative ω-regular property with probability 1 or 0 is effectively regular (i.e., accepted by a finite state automaton) and we provide an upper bound on the size of the finite automaton accepting this set. We also show that in general, the set of all configurations satisfying a given ω-regular property with a prescribed probability (different from 0 and 1) is not regular (not even context-free).

2. We show that the model-checking problem for pPDA and PCTL is undecidable, and for the qualitative fragment of PCTL we show that this problem is **EXPTIME**-complete. We also show that the model-checking problem for pPDA and the qualitative fragment of the logic PECTL* is in **2-EXPTIME**.

3. We show that the expected accumulated reward is expressible in the existential fragment of the first-order theory of reals for a broad class of well-defined reward functions (see Section 6.1). Consequently, the problem whether the expected accumulated reward is greater than (less than, equal to) a given rational number is in **PSPACE**.

4. We define a wide class of long-run properties of Markov chains (that are practically motivated) and show that these properties can effectively be analyzed for pPDA with well-defined reward functions. We also estimate the complexity of the analysis for a subclass of simple reward functions. In particular, we show that the problem whether the average reward per service is greater than (less than, equal to) a given rational number ϱ with the probability greater than (less than, equal to) a given rational number x is in **PSPACE** for simple reward functions. This improves on the results of [28] where the problem was shown to be in **EXPTIME** only for a special case of the average reward per service called gain per transition.

1.2 Outline of the Thesis

The thesis is divided into seven chapters. The main body of the thesis is contained within Chapters 4, 5, 6, and 7.

Chapter 2 contains some preliminary material on probabilistic spaces, Markov chains, and the first-order theory of real numbers.

Chapter 3 introduces the probabilistic pushdown automata together with their basic properties. In this chapter we also formally define the recursive Markov chains and prove their equivalence to pPDA.

Chapter 4 presents a collection of results concerning the model-checking of linear-time properties for pPDA. In particular, we present results concerning reachability, LTL model-checking, and model-checking of ω-regular properties. At the end of this chapter we discuss the regularity problem for the set of configurations of a pPDA that satisfy a given linear-time property with a prescribed probability.

Chapter 5 deals with the decidability and complexity of the model-checking problem for branching-time temporal logics (PCTL, PCTL*, and PECTL*).

Chapter 6 is devoted to a study of the expected behavior of Markov chains generated by pPDA. The main objective is an effective analysis of the expected accumulated reward.

Chapter 7 studies long-run average behavior of Markov chains generated by pPDA. In this chapter we introduce a specific class of long-run properties and study several algorithmic problems related to them.

1.3 How to Read This Thesis

All readers are advised to read at least basic definitions and results from Chapter 2 and the definition of pPDA from Section 3.1. The main body of the thesis is divided into four parts (Chapters 4–7): Linear-time model-checking, branching-time model-checking, expectations, and long-run properties. These four parts can, in principle, be studied separately. However, there are some dependencies as indicated below:

Linear-time model-checking: For a survey of results (without some of the technical proofs) one should read Sections 4.1, 4.2, 4.3, and 4.4. The proofs of Chapter 4 that are technically more demanding are presented in Section 4.6.

Branching-time model-checking: Sufficient prerequisites for reading the first part of Chapter 5 up to Section 5.2 (inclusive) are only the basic definitions and elementary properties of Markov chains and pPDA together with some notions from Section 4.4.2. In order to fully understand the results and proofs of Section 5.3, it is necessary to know the contents of Section 4.4 and Section 4.5.

Expected Behavior: In order to understand Section 6.1 and Section 6.2, no special knowledge is needed except for notation and results of Section 4.1. However, formal proofs presented in Section 6.3 use some additional notation from Section 4.6.

Long-run properties: Reading Chapter 7 up to Section 7.2 (inclusive) should not demand more than the basic knowledge mentioned above together with some knowledge learnt in Section 4.1 and Section 4.3.1. In order to read Section 7.3 and Section 7.4, one should know the notation and results introduced in Chapter 6 and Section 4.6.

Chapter 2

Mathematical Preliminaries

In this work we use $\mathbb{R}$ and $\mathbb{R}^+$ to denote the sets of real numbers and non-negative real numbers, respectively. We also use $\mathbb{R}_{\pm\infty}$ to denote $\mathbb{R} \cup \{-\infty, \infty\}$, and $\mathbb{R}^+_\infty$ to denote $\mathbb{R}^+ \cup \{\infty\}$. The symbols ∞ and $-\infty$ are treated according to the standard conventions. In this thesis we use the following simple properties of limits. The proof can be found in standard textbooks of mathematical analysis (e.g., [46]).

Proposition 2.0.1. *Given $a_0, a_1, \ldots \in \mathbb{R}$ and $b_0, b_1, \ldots \in \mathbb{R}$ such that $\lim_{n\to\infty} a_n = A$ and $\lim_{n\to\infty} b_n = B$ for $A, B \in \mathbb{R}_{\pm\infty}$, we have*

- $\lim_{n\to\infty}(a_n + b_n) = A + B$;
- $\lim_{n\to\infty}(a_n \cdot b_n) = A \cdot B$;
- $\lim_{n\to\infty} \frac{a_n}{b_n} = \frac{A}{B}$, *provided $b_n \neq 0$ for $n \geq 0$;*

whenever the expression on the right hand side is defined.

As a consequence we obtain the following proposition:

Proposition 2.0.2. *Given $k \geq 0$ and $a_1, a_2, \ldots \in \mathbb{R}$ we have*

$$\lim_{n\to\infty} \frac{\sum_{i=1}^n a_i}{n} = \lim_{n\to\infty} \frac{\sum_{i=1}^n a_{i+k}}{n}$$

whenever one of the limits exists.

2.1 Probability Spaces

A *σ-field* over a set T is a set $\mathcal{F} \subseteq 2^T$ that includes T and is closed under complement and countable union. A *measurable space* is a pair $(T, \mathcal{F})$ where T is a set called *sample space* and $\mathcal{F}$ is a σ-field over T. A *probability measure* over a measurable space $(T, \mathcal{F})$ is a function $\mathcal{P} : \mathcal{F} \to [0, 1]$ such that for each countable collection $A_1, A_2, \ldots$ of pairwise disjoint elements[1] of $\mathcal{F}$, $\mathcal{P}(\biguplus_{i=1}^\infty A_i) = \sum_{i=1}^\infty \mathcal{P}(A_i)$, and moreover $\mathcal{P}(T) = 1$. A *probability space* is

[1] We use $\biguplus_{i=1}^\infty A_i$ instead of $\bigcup_{i=1}^\infty A_i$ to stress the fact that the sets $A_1, A_2, \ldots$ are disjoint.

a triple $(T, \mathcal{F}, \mathcal{P})$ where $(T, \mathcal{F})$ is a measurable space and $\mathcal{P}$ is a probability measure over $(T, \mathcal{F})$.

For basic properties of the probability spaces see, e.g., [13]. The following basic assertion is widely used throughout this work without an explicit reference:

Given $A_1, A_2, \ldots \in \mathcal{F}$ such that $\mathcal{P}(A_i) = 1$ for $i \geq 1$, we have $\mathcal{P}(\bigcap_{i=1}^{\infty} A_i) = 1$.

Conditional probability: Given $A, B \in \mathcal{F}$ such that $\mathcal{P}(B) > 0$, we denote $\mathcal{P}(A \mid B) = \frac{\mathcal{P}(A \cap B)}{\mathcal{P}(B)}$ the probability of A conditional on B. Note that for a fixed B satisfying $\mathcal{P}(B) > 0$, the function, which assigns to every $A \in \mathcal{F}$ the value $\mathcal{P}(A \mid B)$, is a probability measure. Moreover, it is easy to prove that given $A, B \in \mathcal{F}$ and pairwise disjoint sets $B_1, B_2, \ldots \in \mathcal{F}$ such that $\biguplus_{i=1}^{\infty} B_i \subseteq B$ and $\mathcal{P}(B) = \mathcal{P}(\biguplus_{i=1}^{\infty} B_i) > 0$, we have

$$\mathcal{P}(A \mid B) = \sum_{i=1}^{\infty} \mathcal{P}(B_i \mid B) \cdot \mathcal{P}(A \mid B_i)$$

We say that almost all elements of a set $A \in \mathcal{F}$ satisfy a given property (or that this property holds a.s. over A) if there is a set $B \in \mathcal{F}$ such that $\mathcal{P}(B \mid A) = 1$ and all elements of B satisfy this property.

Random variables and expectations: Generally, a *random variable* over the probability space $(T, \mathcal{F}, \mathcal{P})$ is a function $X : T \to \mathbb{R}_{\pm\infty} \cup \{\perp\}$ (here $\perp$ stands for "undefined") such that for all $x \in \mathbb{R}_{\pm\infty}$ we have $\{\omega \in T \mid X(\omega) \geq x\} \in \mathcal{F}$. The following basic fact is proved, e.g., in [13].

Proposition 2.1.1. *Let $X_i : T \to \mathbb{R}$ be random variables for all $i \geq 1$. Then $X : T \to \mathbb{R}_{\pm\infty} \cup \{\perp\}$ defined by*

$$X(\omega) = \begin{cases} \lim_{i \to \infty} X_i(\omega) & \text{if the limit exists;} \\ \perp & \text{otherwise.} \end{cases}$$

is also a random variable.

A random variable X is *discrete* if it takes values only in a countable subset of $\mathbb{R}$. Given a non-negative discrete random variable X, we define its expectation $EX = \sum_{x \in R} x \cdot \mathcal{P}(X = x)$ where $R = \{x \in \mathbb{R}^+ \mid \mathcal{P}(X = x) > 0\}$. Note that the non-negativity of X ensures that the sum $\sum_{x \in R} x \cdot \mathcal{P}(X = x)$ is always defined (possibly ∞) and that the sum does not depend on the order of summands (see, e.g., [46]).

One of the most important properties of the expectation, which we often use without an explicit reference, is the following linearity property (for a proof see, e.g., [13]): Given random variables $X_1, X_2, \ldots, X_n$ such that $X_i : T \to \mathbb{R}$ for $1 \leq i \leq n$, and $a_1, a_2, \ldots, a_n \in \mathbb{R}$, the sum $\sum_{i=1}^{n} a_i \cdot X_i$ is a random variable. Moreover, given non-negative discrete random variables $X_1, X_2, \ldots, X_n$ and $a_1, a_2, \ldots, a_n \in \mathbb{R}^+$,

$$E(\sum_{i=1}^{n} a_i \cdot X_i) = \sum_{i=1}^{n} a_i \cdot EX_i.$$

Given a measurable set $A \in \mathcal{F}$ such that $\mathcal{P}(A) > 0$ and a non-negative discrete random variable X, we denote $E(X \mid A) = \sum_{x \in R} x \cdot \mathcal{P}(X = x \mid A)$ where $R = \{x \in \mathbb{R}^+ \mid \mathcal{P}(X = x) > 0\}$, the expectation of X conditional on A. The following important property allows us to decompose the expectation into "simpler" cases. Given $B \in \mathcal{F}$, pairwise disjoint sets $B_1, B_2, \ldots \in \mathcal{F}$ such that $\biguplus_{i=1}^{\infty} B_i \subseteq B$ and $\mathcal{P}(B) = \mathcal{P}(\biguplus_{i=1}^{\infty} B_i) > 0$, and a non-negative discrete random variable X, we have

$$E(X \mid B) = \sum_{i=1}^{\infty} \mathcal{P}(B_i \mid B) \cdot E(X \mid B_i)$$

Independence and strong law of large numbers: A collection of discrete random variables $\mathcal{C}$ is *independent* [2] if for all finite subcollections $\{X_1, \ldots, X_n\} \subseteq \mathcal{C}$ and for all $x_1, \ldots, x_n \in \mathbb{R}$ we have $\mathcal{P}(\bigwedge_{i=1}^{n} X_i = x_i) = \prod_{i=1}^{n} \mathcal{P}(X_i = x_i)$. Discrete random variables X and Y defined over the same probability space are *identically distributed* if for all $x \in \mathbb{R}$ we have $\mathcal{P}(X = x) = \mathcal{P}(Y = x)$. In Section 7.3.1 we use the following special case of a basic theorem of the probability theory (proved, e.g., in [13]).

Theorem 2.1.2. *Given an independent collection $X_1, X_2, \ldots$ of identically distributed non-negative discrete random variables, we have*

$$\mathcal{P}(\lim_{n \to \infty} \frac{\sum_{i=1}^{n} X_i}{n} = EX_1) = 1$$

2.2 Markov Chains

The underlying semantics of probabilistic sequential programs is defined in terms of discrete Markov chains.

Definition 2.2.1. *A (discrete)* Markov chain *is a triple $M = (S, \to, Prob)$ where S is a finite or countably infinite set of* states, *$\to \subseteq S \times S$ is a* transition relation, *and $Prob$ is a function which to each transition $s \to t$ of M assigns its probability $Prob(s \to t) \in (0, 1]$ so that for every $s \in S$ we have $\sum_{s \to t} Prob(s \to t) = 1$.*

If M is finite and all transition probabilities are rational, then the size $|M|$ of M is defined to be the sum $|\to| + |Prob|$ where $|Prob|$ equals the sum of sizes of binary representations of all values of $Prob$ (rational values are represented as fractions where both numerator and denominator are represented as binary numbers).

In the rest of this work we also write $s \xrightarrow{x} t$ instead of $Prob(s \to t) = x$. A *path* in M is a finite or infinite sequence $w = s_0, s_1, \ldots$ of states such that

[2] A collection $\mathcal{D} \subseteq \mathcal{F}$ of measurable sets is independent if the collection of characteristic functions of sets of $\mathcal{D}$ is independent.

$s_i \to s_{i+1}$ for every i. The *length* of a given path w (denoted $|w|$) is the number of transitions in w. In particular, the length of an infinite path is ∞, and the length of a path s, where $s \in S$, is zero. 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). Given a finite path v we denote $last(v) = v(|v|)$, the last state of the path v. A state t is *reachable* from a state s if there is a finite path v such that $v(0) = s$ and $last(v) = t$. Given a state s, we denote M^s the Markov chain obtained from M by restricting the set of states to those states reachable from s (transitions are also appropriately restricted). The prefix $s_0, \ldots, s_i$ and the suffix $s_i, s_{i+1}, \ldots$ of w are denoted by w^i and w_i, respectively. A *run* is an infinite path.

The sets of all paths, all finite paths and all runs of M are denoted $Path$, $FPath$ and Run, respectively. Similarly, the sets of all paths, finite paths and runs that start with a given $w \in FPath$ are denoted $Path(w)$, $FPath(w)$ and $Run(w)$, respectively. In particular, $Run(s)$, where $s \in S$, is the set of all runs initiated in s. Given a set $A \subseteq FPath$, we denote $Run(A) = \bigcup_{v \in A} Run(v)$. When needed, we write $Path[M]$, $FPath[M]$, $Run[M]$, etc., to stress that the paths and runs are in the Markov chain M.

The transition system (graph) $(S, \to)$ is called the *underlying transition system* of M. We say that a set $C \subseteq S$ is a *bottom strongly connected component* (BSCC) of M if for every $s, t \in C$ there is a path from s to t in M, and whenever there is a path from $s \in C$ to $t \in S$, then $t \in C$. Note that when we restrict the set of states of M to a BSCC C, we obtain a Markov chain. The chain M is strongly connected if S is the only BSCC of M. Note that M is strongly connected if and only if the underlying transition system $(S, \to)$ is strongly connected in the usual graph theoretic sense.

In this work we are interested in probabilities of certain events that are associated with runs. To every $s \in S$ we associate the probability space $(Run(s), \mathcal{F}, \mathcal{P})$ where $\mathcal{F}$ is the σ-field generated by all *basic cylinders* $Run(w)$ where $w \in FPath(s)$, and $\mathcal{P} : \mathcal{F} \to [0, 1]$ is the unique probability measure (see [39, 38]) such that $\mathcal{P}(Run(w)) = \prod_{i=0}^{m-1} x_i$ 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 $\mathcal{P}(Run(w)) = 1$). A set of runs $A \subseteq Run(s)$ is *measurable* if $A \in \mathcal{F}$.

Given $v \in FPath$ and $v' \in Path(last(v))$, we denote $v \odot v' = v(0), \ldots, v(|v| - 1), v'$ the path obtained by concatenating v and v' through the last state of v. Given a state s, a set $A \subseteq FPath$ such that for every $v \in A$ we have $last(v) = s$, and a set $B \subseteq Path(s)$, we denote $A \odot B = \{v \odot v' \mid v \in A, v' \in B\}$. We usually write $v \odot B$ instead of $\{v\} \odot B$.

Definition 2.2.2. *A set of finite paths $A \subseteq FPath$ is* prefix-free *if for all $u, v \in A$ such that $u \neq v$, we have $Run(u) \cap Run(v) = \emptyset$.*

Lemma 2.2.3. *Let $s, t \in S$, let $A \subseteq FPath(s)$ be a prefix-free set of paths from s to t, and let $B \subseteq Run(t)$ be a measurable set of runs. Then $A \odot B$ is measurable and*

$$\mathcal{P}(A \odot B) = \mathcal{P}(Run(A)) \cdot \mathcal{P}(B)$$

Proof. Let $u \in A$. We show that $u \odot B$ is measurable and that $\mathcal{P}(u \odot B) = \mathcal{P}(Run(u)) \cdot \mathcal{P}(B)$. Let $\mathcal{G}$ be the set of all $D \subseteq Run(t)$ such that $u \odot D$ is measurable. We show that $\mathcal{G}$ is a σ-field over $Run(t)$ which contains all basic cylinders. Indeed, $Run(v) \in \mathcal{G}$ for all $v \in FPath(t)$ because $u \odot Run(v) = Run(u \odot v)$ is measurable, and given a countable collection $D_1, D_2 \ldots$ of elements of $\mathcal{G}$, we have that both $u \odot \bigcup_{i=1}^{\infty} D_i = \bigcup_{i=1}^{\infty} u \odot D_i$ and $u \odot (Run(t) \setminus D_1) = Run(u) \setminus (u \odot D_1)$ are measurable. It follows that $\mathcal{G}$ contains all measurable subsets of $Run(t)$ because the set of all measurable subsets of $Run(t)$ is the least σ-field which contains all basic cylinders. Hence, $u \odot B$ is measurable.

Now let us denote $\mathcal{P}'$ the function which assigns to every measurable set $D \subseteq Run(t)$ the value $\mathcal{P}(u \odot D \mid Run(u))$. We show that $\mathcal{P}'$ is a probability measure. Clearly, $\mathcal{P}'(Run(t)) = 1$, and given a countable collection $D_1, D_2, \ldots$ of pairwise disjoint measurable sets, we have $\mathcal{P}'(\biguplus_{i=1}^{\infty} D_i) = \mathcal{P}(u \odot \biguplus_{i=1}^{\infty} D_i \mid Run(u)) = \sum_{i=1}^{\infty} \mathcal{P}(u \odot D_i \mid Run(u)) = \sum_{i=1}^{\infty} \mathcal{P}'(D_i)$. Moreover, given $v \in FPath(t)$ we have $\mathcal{P}'(Run(v)) = \mathcal{P}(u \odot Run(v) \mid Run(u)) = \frac{\mathcal{P}(Run(u)) \cdot \mathcal{P}(Run(v))}{\mathcal{P}(Run(u))} = \mathcal{P}(Run(v))$. It follows that $\mathcal{P}'(D) = \mathcal{P}(D)$ for all measurable sets $D \subseteq Run(t)$ by the uniqueness of $\mathcal{P}$, and hence $\mathcal{P}(u \odot B) = \mathcal{P}(Run(u)) \cdot \mathcal{P}(u \odot B \mid Run(u)) = \mathcal{P}(Run(u)) \cdot \mathcal{P}(B)$.

Finally, note that $\mathcal{P}(A \odot B) = \sum_{u \in A} \mathcal{P}(u \odot B) = \sum_{u \in A} \mathcal{P}(Run(u)) \cdot \mathcal{P}(B) = \mathcal{P}(Run(A)) \cdot \mathcal{P}(B)$ where the first and the last equation follow from the fact that A is prefix-free. □

Quotient Markov Chain

In this subsection we define a "simulation" quotient for Markov chains which constitutes a formal basis for many constructions in this thesis.

Definition 2.2.4. *Let $M' = (S', \to', Prob')$ be a Markov chain and let $\Theta : S \to S'$ be a* surjective *function mapping states of M onto states of M'. We say that Θ is a* quotient of M onto M' *if the following condition is satisfied: For every $s \in S$ and every $s' \in S'$ such that $\Theta(s) \xrightarrow{x} s'$, there is $t \in \Theta^{-1}(s')$ satisfying $s \xrightarrow{x} t$. If, in addition, Θ is one-to-one, then we say that Θ is an isomorphism of M onto M'.*

Each quotient Θ extends naturally to $v \in Path[M]$ by $\Theta(v)(i) = \Theta(v(i))$. Given $A \subseteq Path[M]$, we denote $\Theta(A) = \{\Theta(v) \mid v \in A\}$.

Proposition 2.2.5. *Let Θ be a quotient of M onto M' and let $s_0 \in S$. Then Θ maps $Run(s_0)$ isomorphically to $Run(\Theta(s_0))$, and $A \subseteq Run(s_0)$ is measurable if and only if $\Theta(A) \subseteq Run(\Theta(s_0))$ is measurable, in which case $\mathcal{P}(A) = \mathcal{P}(\Theta(A))$.*

Proof. We show that Θ maps $Path(s_0)$ isomorphically to $Path(\Theta(s_0))$. First, we show that $s \to t$ implies $\Theta(s) \to \Theta(t)$ for every $s, t \in S$. Assume the opposite, i.e., $\Theta(s) \not\to \Theta(t)$. Then for every s' such that $\Theta(s) \xrightarrow{x} s'$ (here $x > 0$) there is

$t' \neq t$ such that $s \xrightarrow{x} t'$. However, then $s \not\rightarrow t$ because the sum of probabilities of transitions outgoing from s is 1. It follows immediately that $\Theta(Path(s_0)) \subseteq Path(\Theta(s_0))$.

Now let $v \in Path(\Theta(s_0))$. We define inductively a unique path $v' \in Path(s_0)$ such that $\Theta(v') = v$. We put $v'(0) = s_0$. Let us assume that we have already defined $v'(i)$ where $i < |v|$ and let $v(i) \rightarrow v(i+1)$. It is easy to show, using a very similar argument as above, that there is exactly one state $t \in S$ such that $\Theta(t) = v(i+1)$ and $v'(i) \rightarrow t$. We put $v'(i+1) = t$. Observe that for all $0 \leq i < |v|$ holds $v'(i) \xrightarrow{x} v'(i+1)$ where $v(i) \xrightarrow{x} v(i+1)$.

Now let $u \in FPath(s_0)$. Note that Θ maps $Run(u)$ isomorphically to $Run(\Theta(u))$ because $\Theta(Run(u)) \subseteq Run(\Theta(u))$, $\Theta(Run(s_0) \setminus Run(u)) \subseteq Run(\Theta(s_0)) \setminus Run(\Theta(u))$, and Θ maps $Run(s_0)$ isomorphically to $Run(\Theta(s_0))$.

It follows that both Θ (restricted to $Run(s_0)$) and Θ^{-1} (restricted to $Run(\Theta(s_0))$) are measurable mappings (see [13], Theorem 13.1), which implies that $A \subseteq Run(s_0)$ is measurable if and only if $\Theta(A)$ is measurable. The equality of probabilities follows from the uniqueness of $\mathcal{P}$ over $Run(s_0)$ because for each basic cylinder $Run(u)$ we have $\mathcal{P}(Run(u)) = \mathcal{P}(Run(\Theta(u))) = \mathcal{P}(\Theta(Run(u)))$ and $\mathcal{P} \circ \Theta$ is clearly a probability measure. □

2.3 First-order Theory of Reals

In this work we are interested in various numerical features that characterize the behavior of Markov chains (the probability of reaching a given state, the expected accumulated reward along a path, etc.). For finite Markov chains, these values are usually rational and can effectively be computed explicitly. On the other hand, in the case of Markov chains generated by pPDA these values can be irrational (as we show in appropriate places) and thus cannot be computed explicitly. However, we are still able to show that many of these values can effectively be expressed in *the existential fragment of the first-order theory of reals* (denoted $ExTh(\mathbb{R})$). This expressibility result brings important consequences because $ExTh(\mathbb{R})$ is decidable in polynomial space. For example, we may effectively approximate the expressible values, decide whether they lie within given bounds, etc. The decidability of $ExTh(\mathbb{R})$ is a highly non-trivial result whose proof uses methods of algebraic geometry. Its complete treatment can be found, e.g., in [18, 10]. In this section we only review some basic definitions and results related to $ExTh(\mathbb{R})$, and explain what is meant by the expressibility of a value in $ExTh(\mathbb{R})$.

We start with a formal definition of the first order theory of reals (denoted $Th(\mathbb{R})$). Let $Vars$ be a countable set of "real" variables. Formulae of $Th(\mathbb{R})$ are built up from atoms using Boolean connectives and first-order quantifiers. Atoms are of the form $P \sim 0$ where P is a multivariate polynomial in variables of $Vars$ with rational coefficients, and $\sim \in \{=, \neq, \leq, \geq, <, >\}$. More formally, formulae of $Th(\mathbb{R})$ are defined as follows:

- Atoms $P \sim 0$ are formulae of $Th(\mathbb{R})$;
- If Φ_1 and Φ_2 are formulae of $Th(\mathbb{R})$, then $\Phi_1 \wedge \Phi_2$, $\neg\Phi_1$, and $(\exists x)\Phi_1$ (where $x \in Vars$) are formulae of $Th(\mathbb{R})$.

The semantics of formulae of $Th(\mathbb{R})$ is defined as follows: Let $e : Vars \to \mathbb{R}$ be a valuation of variables. We define

- $e \models P \sim 0$ if and only if $P(e) \sim 0$, where $P(e)$ denotes the value of P when every variable x in P is assigned the value $e(x)$;
- $e \models \Phi_1 \wedge \Phi_2$ (or $e \models \neg\Phi_1$) if and only if $e \models \Phi_1$ and $e \models \Phi_2$ (or $e \not\models \Phi_1$, resp.);
- $e \models (\exists x)\Phi_1$ if and only if there is $a \in \mathbb{R}$ such that $e[a/x] \models \Phi_1$, where $e[a/x]$ denotes the valuation which equals e over all variables (possibly) except x and $e[a/x](x) = a$.

If $e \models \Phi$, then we say that the formula Φ is satisfied by the valuation e. We say that Φ is true if it is satisfied by all valuations. A variable x is free in Φ if there is an occurrence of x in Φ which is not bound by any quantifier. Formulae without free variables are called *sentences*. We write $\Phi(x_1, \ldots, x_n)$ to indicate that $x_1, \ldots, x_n$ are *exactly* the free variables in Φ. It is easy to show that the truth value of $\Phi(x_1, \ldots, x_n)$ depends only on the valuation of the variables $x_1, \ldots, x_n$. Hence, we say that $\vec{a} \in \mathbb{R}^n$ satisfies $\Phi(x_1, \ldots, x_n)$ (or that $\Phi(\vec{a})$ is true) if $e \models \Phi$ for all valuations that satisfy $e(x_i) = a_i$.

The *existential fragment* of $Th(\mathbb{R})$, denoted $ExTh(\mathbb{R})$, consists of formulae of the form $(\exists x_1) \cdots (\exists x_n)\Phi$ where Φ is a formula of $Th(\mathbb{R})$ without quantifiers.

Theorem 2.3.1 ([18, 10])**.** *Given a sentence Φ of $Th(\mathbb{R})$, the problem whether Φ is true is decidable in exponential time. If Φ is in $ExTh(\mathbb{R})$, then the problem is decidable in polynomial space.*

Let $\Phi(x)$ be a formula of $ExTh(\mathbb{R})$ whose only free variable is x, and let us assume that there is *exactly one* $c \in \mathbb{R}$ such that $\Phi(c)$ is true. Then we say that the value c is *expressed* by the formula $\Phi(x)$.

Several key theorems in this work state that some values of $\mathbb{R}_{\pm\infty} \cup \{\bot\}$, associated to instances of particular problems, are effectively expressible. The precise meaning of this statement is the following:

Definition 2.3.2. *Let P be an instance of a problem (typically a pPDA together with some additional structure such as a reward function, etc.) and let $c(P) \in \mathbb{R}_{\pm\infty} \cup \{\bot\}$ be a value associated to P (e.g. an expected accumulated reward). We say that the value $c(P)$ is* effectively expressible in $ExTh(\mathbb{R})$ *if there is an effective procedure which for the input P decides whether $c(P) = \bot$ or $c(P) = \pm\infty$, and if $c(P) \in \mathbb{R}$, then the procedure computes a formula $\Phi(x)$ of $ExTh(\mathbb{R})$ expressing $c(P)$. (We also say that $c(P)$ is expressible by a formula of $ExTh(\mathbb{R})$ of size $|\Phi|$ computable with the complexity of the above procedure.)*

The following proposition shows how solutions of a given system of polynomial equations, with coefficients expressed by formulae of $ExTh(\mathbb{R})$, can be expressed in $ExTh(\mathbb{R})$.

Proposition 2.3.3. *Let $P = \{x_i = P_i \mid 1 \leq i \leq k\}$ be a system of equations where each P_i is a multivariate polynomial in variables $x_1, \ldots, x_k, y_1, \ldots, y_\ell$ with rational coefficients. Let $\Phi_1(y_1), \ldots, \Phi_\ell(y_\ell)$ be formulae of $ExTh(\mathbb{R})$ expressing values $c_1, \ldots, c_\ell \in \mathbb{R}$, respectively. Let $\Phi(x_1, \ldots, x_k)$ be a formula of $ExTh(\mathbb{R})$. If there is* exactly one *vector $(a_1, \ldots, a_k) \in \mathbb{R}^k$ such that $\Phi(a_1, \ldots, a_k)$ is true and the vector $a_1, \ldots, a_k, c_1, \ldots, c_\ell$ solves the system P, then the values $a_1, \ldots, a_k$ are expressible in $ExTh(\mathbb{R})$ by formulae effectively computable in time polynomial in $|P| + |\Phi| + \sum_{i=1}^{\ell} |\Phi_i|$.*

We use Proposition 2.3.3 in the following way: Let P be a system of polynomial equations with coefficients expressed by formulae of $ExTh(\mathbb{R})$ (these coefficients are in P symbolically represented by the variables $y_1, \ldots, y_\ell$). Let us assume that there is exactly one solution of P satisfying a given constraint Φ. Then the components of the solution are effectively expressible in $ExTh(\mathbb{R})$. As a simple consequence we obtain that whenever e is an arithmetical expression over rational numbers and values expressed by formulae $\Phi_1, \ldots, \Phi_\ell$, then a formula expressing the value of e is computable in time polynomial in $|e| + \sum_{i=1}^{\ell} |\Phi_i|$.

Proof of Proposition 2.3.3. Consider

$$\Psi(x_1, \ldots, x_k) = (\exists y_1) \cdots (\exists y_\ell) \Psi'$$

where $\Psi' = \Phi(x_1, \ldots, x_k) \wedge \bigwedge_{i=1}^{\ell} \Phi_i(y_i) \wedge \bigwedge_{i=1}^{k} x_i = P_i$. It is easy to see that a formula $(\exists x_1) \cdots (\exists x_{i-1})(\exists x_{i+1}) \cdots (\exists x_k) \Psi(x_1, \ldots, x_k)$ expresses a_i and that this formula can be transformed to a formula of $ExTh(\mathbb{R})$ (i.e., to the prenex form). □

Chapter 3

Models of Probabilistic Programs

In this chapter we introduce two models of probabilistic recursive sequential programs: probabilistic pushdown automata (pPDA) and recursive Markov chains (RMC). We show that these two models are expressively equivalent, and moreover, that they can be translated between each other in linear time. In this work we deal almost exclusively with probabilistic pushdown automata. However, all our results can be easily translated to the language of recursive Markov chains.

3.1 Probabilistic PDA

In this section we define probabilistic pushdown automata and explain their basic features.

Definition 3.1.1. *A* probabilistic PDA (pPDA) *is a tuple* $\Delta = (Q, \Gamma, \delta, Prob)$ *where* Q *is a finite set of* control states, Γ *is a finite* stack alphabet, $\delta \subseteq Q \times \Gamma \times Q \times \Gamma^{\leq 2}$ *(here* $\Gamma^{\leq 2} = \{\alpha \in \Gamma^*, |\alpha| \leq 2\}$*) is a* transition relation, *and* $Prob$ *is a function which to each transition* $pX \to q\alpha$ *assigns a rational probability* $Prob(pX \to q\alpha) \in (0, 1]$ *so that for all* $p \in Q$ *and* $X \in \Gamma$ *we have that* $\sum_{pX \to q\alpha} Prob(pX \to q\alpha) = 1$.

In the rest of this thesis we adopt a more intuitive notation, writing $pX \to q\alpha$ instead of $(p, X, q, \alpha) \in \delta$, and $pX \xrightarrow{x} q\alpha$ instead of $Prob(pX \to q\alpha) = x$. The set $Q \times \Gamma^*$ of all configurations of Δ is denoted by $\mathcal{C}(\Delta)$. Given a configuration $pX\alpha$, we call pX the *head* and α the *tail* of $pX\alpha$ (we also write $head(pX\alpha)$ and $tail(pX\alpha)$ to denote the head pX and the tail α of $pX\alpha$, respectively). The head of $p\varepsilon$ is p and the tail is ε.

We denote pBPA the class of all pPDA with just one control state. The class pBPA can be seen either as a probabilistic variant of the well-known basic process algebra (BPA, see [11]), or as the class of 1-exit RMCs defined in [33]. In what follows, configurations of pPBA are usually written without the control state (i.e., we write only α instead of $p\alpha$).

We define the size $|\Delta|$ of the pPDA Δ to be the sum $|\delta| + |Prob|$ where $|Prob|$ equals the sum of sizes of binary representations of values of $Prob$.

To Δ we associate the Markov chain M_Δ where $\mathcal{C}(\Delta)$ is the set of states and the transitions are determined as follows:

- $p\varepsilon \xrightarrow{1} p\varepsilon$ for each $p \in Q$ (here ε denotes the empty stack);
- $pX\beta \xrightarrow{x} q\alpha\beta$ is a transition of M_Δ if and only if $pX \xrightarrow{x} q\alpha$ is a transition of Δ.

We demonstrate the concept of pPDA on some examples. The examples are chosen to illustrate some interesting properties of Markov chains generated by pPDA that are not present in the case of finite chains.

The class of Markov chains generated by pPDA subsumes many examples which are interesting from the standpoint of the general theory of denumerable Markov chains. The following example resembles a simple one dimensional random walk.

Example 3.1.2. *Let us consider a pBPA Δ_w with three stack symbols X, Y, Z and the following transitions:*

$$\begin{array}{lll} Z \xrightarrow{x} YZ, & X \xrightarrow{x} \varepsilon, & Y \xrightarrow{1-x} \varepsilon \\ Z \xrightarrow{1-x} XZ, & X \xrightarrow{1-x} XX, & Y \xrightarrow{x} YY, \end{array}$$

where $0 < x < 1$ is a rational number. The underlying Markov chain of Δ_w is shown in Figure 3.1 (only states reachable from Z are drawn).

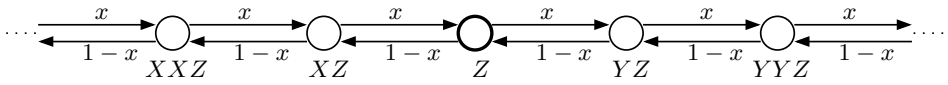

Figure 3.1: The Markov chain M_{Δ_w}

This example is interesting because (as we show later) the probability of eventually returning back to Z after leaving Z depends on x even though the chain M_{Δ_w} is strongly connected. This phenomenon sharply contrasts with the case of finite chains where the return probability is always 1 in strongly connected chains.

Another important feature of this chain is connected with the expected number of steps for returning back to Z after leaving Z (expected return time). We show later (see Example 4.1.6 and Example 6.2.5) that if $x = \frac{1}{2}$, then the return probability is 1, but the expected return time is infinite. Once more, this cannot happen in finite chains where the expected return time is finite in strongly connected chains.

The following example illustrates another important property of Markov chains generated by pPDA: The probability of reaching a state from another state can be irrational (again, this cannot happen in the finite case).

Example 3.1.3. *Let us define a pPDA $\bar{\Delta}$ with two control states s, p, three stack symbols I, D, Z, and the following transitions:*

$$\begin{array}{llll} sZ \xrightarrow{0.75} sZ, & sZ \xrightarrow{0.25} pIZ, & sI \xrightarrow{1} sI, & sD \xrightarrow{1} sD, \\ pI \xrightarrow{0.5} pID, & pI \xrightarrow{0.5} p\varepsilon, & pD \xrightarrow{0.5} pI, & pD \xrightarrow{0.5} pDD, \\ pZ \xrightarrow{1} pZ & & & \end{array}$$

The underlying Markov chain of $\bar{\Delta}$ is shown in Figure 3.2 (only states reachable from sZ are drawn).

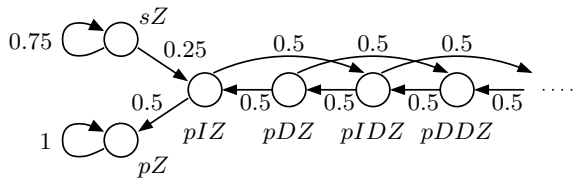

Figure 3.2: The Markov chain $M_{\bar{\Delta}}$

We show later that the probability of reaching the configuration pZ from sZ is $\frac{\sqrt{5}-1}{2}$ (the golden ratio), which is irrational.

3.2 Recursive Markov Chains

The following definition was taken from [33]. A *Recursive Markov Chain* (RMC) is a tuple $A = (A_1, \ldots, A_k)$, where each *component graph* $A_i = (N_i, B_i, Y_i, En_i, Ex_i, \delta_i)$ consists of:

- A set N_i of nodes.
- A subset of entry nodes $En_i \subseteq N_i$, and a subset of exit nodes $Ex_i \subseteq N_i$.
- A set B_i of boxes. Let $B = \bigcup_{i=1}^{k} B_i$ be a union of all boxes of A.
- A mapping $Y_i : B_i \to \{1, \ldots, k\}$ that assigns to every box (the index of) one of the components, $A_1, \ldots, A_k$. Let $Y = \bigcup_{i=1}^{k} Y_i$ be $Y : B \to \{1, \ldots, k\}$ where the restriction of Y to B_i is equal to Y_i for $1 \leq i \leq k$.
- To each box $b \in B_i$ we associate a set of call ports $Call_b = \{(b, en) \mid en \in En_{Y(b)}\}$, and a set of return ports $Return_b = \{(b, ex) \mid ex \in Ex_{Y(b)}\}$.
- A transition relation δ_i where transitions are of the form $(u, p_{u,v}, v)$ where
  1. the source u is either a non-exit node $u \in N_i \backslash Ex_i$, or a return port $u = (b, ex) \in Return_b$ where $b \in B_i$;
  2. the destination v is either a non-entry node $v \in N_i \backslash En_i$, or a call port $u = (b, en) \in Call_b$ where $b \in B_i$;
  3. $p_{u,v} \in (0, 1]$ is a rational transition probability from u to v;
  4. for each u we have $\sum_{\{v' \mid (u, p_{u,v'}, v') \in \delta_i\}} p_{u,v'} = 1$, unless u is a call port or an exit node.

Let us denote V_i the set of all nodes, call ports and return ports of A_i. Let $V = \bigcup_{i=1}^{k} V_i$ be the set of *vertices* of A.

The RMC A defines a Markov chain $M_A = (S_A, \rightarrow, Prob)$ where $S_A \subseteq B^* \times V$ is a set of states. Elements of S_A are denoted $\langle \beta, u \rangle$ where $\beta \in B^*$ and $u \in V$. Formally, the set S_A and transitions of M_A are defined as follows:

1. $\langle \varepsilon, u \rangle \in S_A$ for $u \in V$.
2. If $\langle \beta, u \rangle \in S_A$ and $(u, p_{u,v}, v) \in \bigcup_{i=1}^{k} \delta_i$, then $\langle \beta, v \rangle \in S_A$ and $\langle \beta, u \rangle \stackrel{p_{u,v}}{\rightarrow} \langle \beta, v \rangle$.
3. If $\langle \beta, (b, en) \rangle \in S_A$ and $(b, en) \in Call_b$, then $\langle \beta b, en \rangle \in S_A$ and $\langle \beta, (b, en) \rangle \stackrel{1}{\rightarrow} \langle \beta b, en \rangle$.
4. If $\langle \beta b, ex \rangle \in S_A$ and $(b, ex) \in Return_b$, then $\langle \beta, (b, ex) \rangle \in S_A$ and $\langle \beta b, ex \rangle \stackrel{1}{\rightarrow} \langle \beta, (b, ex) \rangle$.
5. If $\langle \varepsilon, ex \rangle \in S_A$ where $ex \in Ex_i$ for some i, then $\langle \varepsilon, ex \rangle \stackrel{1}{\rightarrow} \langle \varepsilon, ex \rangle$.
6. Nothing else is in S_A and nothing else is a transition of M_A.

Various subclasses of RMCs are defined in the literature (see e.g. [33, 31]). One of these classes is the class of 1-exit RMCs that consists of all RMCs whose components have one exit. The class of 1-exit RMCs corresponds to the class pBPA. Other classes defined by Etessami and Yannakakis in [33] are the classes of bounded RMCs, linear RMCs, etc.

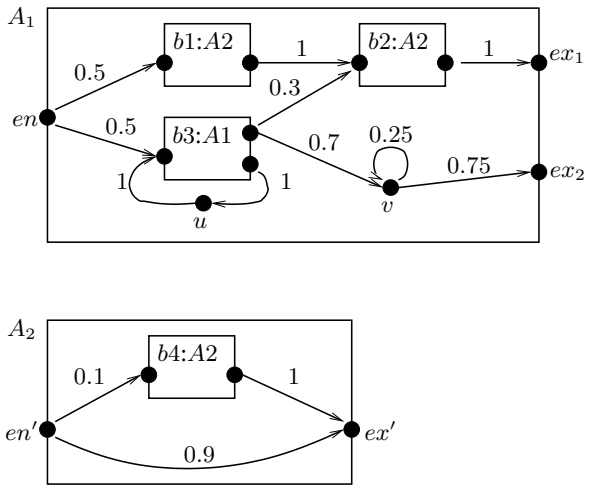

Figure 3.3: RMC A

To illustrate how a RMC works, let us consider the RMC A depicted in Figure 3.3. The RMC A consists of two components A_1 and A_2. The component A_1 has three boxes b_1, b_2, and b_3, where b_1 and b_2 are associated to the component A_2, and b_3 is associated to A_1. The component A_2 has one box b_4 associated to A_2. There are five nodes in the component A_1: one entry node en, two exist nodes ex_1 and ex_2, and two "inner" nodes u and v. The component A_2 contains only two nodes en' and ex'. The bullets drawn on the left-hand (right-hand) side of boxes label call ports (return ports). Let us describe several steps of a run of M_A. Let us

start in the configuration (ε, en). There is a transition from the node en to the call port $(b1, en')$ with the probability 0.5. After taking this transition, the run proceeds to $(\varepsilon, (b1, en'))$, and then with the probability 1 to $(b1, en')$. This means that the component A_2 was recursively called, $b1$ pushed to the stack, and the run continues in the node en'. Let us assume that from here the run proceeds immediately to $(b1, ex')$ by taking the transition from en' to ex' (which occurs with probability 0.9). From $(b1, ex')$ the run proceeds with probability 1 to $(\varepsilon, (b1, ex'))$, which means that $b1$ was popped from the stack (i.e., the recursive call of A_2 finished), and the run returns to the component A_1 through the return port $(b1, ex')$. From this return port there is a transition to the call port $(b2, en')$ with probability 1, and so on.

The above example illustrates that RMCs behave very similarly as pPDA. There is, however, a small technical difficulty that complicates the equivalence proof (see below). This difference lies within the mid steps added by the semantics of RMCs whenever a call or a return occurs (i.e., the transitions from exit nodes to return ports and from call ports to entry nodes). It follows that for some pPDA there are no RMCs generating isomorphic Markov chains. Hence, we have to be more careful in formulating the equivalence between RMCs and pPDA.

3.3 Equivalence of pPDA and RMC

In this section we show how pPDA can be efficiently translated to RMCs and vice versa. Note that both these translations run in linear time.

From pPDA to RMC

Let $\Delta = (Q, \Gamma, \Delta, Prob)$ be a pPDA. Let us define a RMC A consisting of exactly one component $A_1 = (N_1, B_1, Y_1, En_1, Ex_1, \delta_1)$ defined as follows:

- $N_1 = (Q \times \Gamma) \cup Q \cup \{[pX] \mid pX \in Q \times \Gamma\}$;
- $En_1 = \{[qY] \mid pX \to qYZ\}$ and $Ex_1 = Q$;
- $B_1 = \Gamma$;
- transitions of A_1 are defined as follows:
  1. $(pX, x, qY) \in \delta_1$ if $pX \xrightarrow{x} qY$;
  2. $([pX], 1, pX) \in \delta_1$ for all $[pX] \in En_1$;
  3. $(pX, x, (Z, [qY])) \in \delta_1$ if $pX \xrightarrow{x} qYZ$;
  4. $(pX, x, q) \in \delta_1$ if $pX \xrightarrow{x} q\varepsilon$;
  5. $((Z, q), 1, qZ) \in \delta_1$;
  6. All transitions of A_1 are defined by the above rules.

We claim that for every state $\langle \beta, pX \rangle$ of M_A we have

- $\langle \beta, pX \rangle \xrightarrow{x} \langle \beta, qY \rangle$ iff $pX\beta^R \xrightarrow{x} qY\beta^R$;
- $\langle \beta Y, pX \rangle \xrightarrow{x} \langle \beta Y, q \rangle \xrightarrow{1} \langle \beta, (Y, q) \rangle \xrightarrow{1} \langle \beta, qY \rangle$ iff $pXY\beta^R \xrightarrow{x} qY\beta^R$;

- $\langle \varepsilon, pX \rangle \xrightarrow{x} \langle \varepsilon, q \rangle$ iff $pX \xrightarrow{x} q\varepsilon$;
- $\langle \beta, pX \rangle \xrightarrow{x} \langle \beta, (Z, [qY]) \rangle \xrightarrow{1} \langle \beta Z, [qY] \rangle \xrightarrow{1} \langle \beta Z, qY \rangle$ iff $pX\beta^R \xrightarrow{x} qYZ\beta^R$.

We show that for every state $\langle \beta, pX \rangle$ of M_A, the chain $M_A^{\langle \beta, pX \rangle}$ (i.e., the part of M_A reachable from $\langle \beta, pX \rangle$) and the chain $M_\Delta^{pX\beta^R}$ "behave" similarly once we ignore the mid steps in M_A. Let Θ be a partial function assigning to a state of the form $\langle \beta, pX \rangle$ (where $pX \in Q \times \Gamma$) the configuration $pX\beta^R$, and to a state of the form $\langle \varepsilon, q \rangle$ (where $q \in Q$) the configuration $q\varepsilon$ (and is undefined on all other states of M_A). Let us extend the function Θ to paths of $Path(\langle \beta, pX \rangle)$ as follows: Let v be a path of M_A, and let $i_1, i_2, \ldots$ be all indexes of states of $Dom(\Theta) = \Theta^{-1}(\mathcal{C}(\Delta))$ in v. We put $\Theta(v) = \Theta(v(i_1)), \Theta(v(i_2)), \ldots$.

Proposition 3.3.1. *The function Θ maps $Run(\langle \beta, pX \rangle)$ isomorphically onto $Run(pX\beta^R)$, and a set $A \subseteq Run(\langle \beta, pX \rangle)$ is measurable if and only if the set $\Theta(A)$ is measurable, in which case $\mathcal{P}(\Theta(A)) = \mathcal{P}(A)$.*

Proof. The fact that Θ is an isomorphism follows from above properties of transitions of M_A and from the fact that Θ is one-to-one.

Let $v \in FPath(\langle \beta, pX \rangle)$. If $last(v) \in Dom(\Theta)$, then $\Theta(Run(v)) = Run(\Theta(v))$, and clearly $\mathcal{P}(\Theta(Run(v))) = \mathcal{P}(Run(\Theta(v))) = \mathcal{P}(Run(v))$. On the other hand, if $last(v) \notin Dom(\Theta)$, then there is a path u (of length at most 2) from $last(v)$ to a state of $Dom(\Theta)$ such that $\mathcal{P}(Run(u)) = 1$, $\Theta(Run(v)) = Run(\Theta(v \odot u))$ and $\mathcal{P}(\Theta(Run(v))) = \mathcal{P}(Run(\Theta(v \odot u))) = \mathcal{P}(Run(v))$. In both cases the map Θ maps a cylinder set to a cylinder set of the same probability. It follows that both Θ (restricted to $Run(\langle \beta, pX \rangle)$), and Θ^{-1} (restricted to $Run(pX\beta^R)$) are measurable (see [13], Theorem 13.1), and hence $A \subseteq Run(\langle \beta, pX \rangle)$ is measurable if and only if $\Theta(A) \subseteq Run(pX\beta^R)$ is measurable. The equivalence of probabilities follows from the uniqueness of $\mathcal{P}$ and the fact that $\mathcal{P} \circ \Theta$ is a probability measure. □

From RMCs to pPDA

Let $A = (A_1, \ldots, A_k)$ be a RMC where $A_i = (N_i, B_i, Y_i, En_i, Ex_i, \delta_i)$, and let us assume that all sets Ex_i are ordered (we denote $ex_{i,j}$ the j'th exit of A_i). Let us define a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$ such that

- $Q = \{q_1, \ldots, q_\zeta\}$ for $\zeta = \max\{|Ex_i| \mid 1 \leq i \leq k\} \geq 1$ ($Q = \{q_1\}$ for $\zeta = 0$);
- $\Gamma = V \cup \bigcup_{i=1}^k B_i$ (here V is the set of vertices of A);
- transitions of Δ are defined as follows:
  1. $q_1 u \xrightarrow{x} q_1 v$ if $(u, x, v) \in \bigcup_{i=1}^k \delta_i$ and $v \notin \bigcup_{i=1}^k Ex_i$;
  2. $q_1 u \xrightarrow{x} q_j$ if $(u, x, ex_{i,j}) \in \bigcup_{i=1}^k \delta_i$;
  3. $q_1(b, v) \xrightarrow{1} q_1 vb$ if $(b, v) \in Call_b$ is a call port;
  4. $q_j b \xrightarrow{1} q_1(b, ex_{i,j})$ if $Y(b) = i$;
  5. all transitions of Δ are defined by the above rules.

Lemma 3.3.2. *The function Θ assigning to a state of the form $\langle \beta, u \rangle$, where u is not an exit node, the configuration $q_1 u \beta^R$, and to a state of the form $\langle \beta, ex_{i,j}) \rangle$ the configuration $q_j \beta^R$, satisfies the following condition: For every state of the form $\langle \beta, u \rangle$, the function Θ is an isomorphism of $M_A^{\langle \beta, u \rangle}$ onto $M_\Delta^{q_1 u \beta^R}$.*

Chapter 4

Linear-time Properties

In this chapter we study various model-checking problems for pPDA and linear-time properties. We start with reachability (Section 4.1), then quickly survey results on LTL model-checking (Section 4.2), and finally present a complete treatment of model-checking ω-regular properties.

4.1 Reachability

The reachability is the simplest and the most fundamental problem we consider. It is formally defined as follows. Let $M = (S, \to, Prob)$ be a Markov chain. Given a state s_0 and a set L of states of M, we denote

$$Run(s_0 \to^* L) = \{w \in Run(s_0) \mid \exists i \geq 0 : w(i) \in L\}$$

Clearly, $Run(s_0 \to^* L) = Run(\{v \in FPath(s_0) \mid last(v) \in L\})$, and hence is measurable. We denote $\mathcal{P}(s_0 \to^* L) = \mathcal{P}(Run(s_0 \to^* L))$, the probability of reaching a state of L from the state s_0. (For $t \in S$, we write $\mathcal{P}(s_0 \to^* t)$ instead of $\mathcal{P}(s_0 \to^* \{t\})$.)

We consider the following reachability problems:

- *qualitative reachability*: Is $\mathcal{P}(s_0 \to^* L) = 1$? (Is $\mathcal{P}(s_0 \to^* L) = 0$?)
- *quantitative reachability*:

  Given $\varrho \in [0, 1]$ and $\sim \in \{<, \leq, >, \geq, =\}$, is $\mathcal{P}(s_0 \to^* L) \sim \varrho$?

Ideally, we would like to *compute* the probability $\mathcal{P}(s_0 \to^* L)$. However, unlike for finite-state chains, the probability can be irrational even for very simple Markov chains generated by pBPA (see Example 4.1.7).

Finite Markov Chains

To motivate our approach to the reachability problem for pPDA, we first describe a classical solution of this problem for finite Markov chains. Let us fix a set L of states of a finite Markov chain $M = (S, \to, Prob)$. Let us assume, without

the loss of generality, that for each $t \in L$ holds $t \xrightarrow{1} t$. The problem whether $\mathcal{P}(s_0 \rightarrow^* L) = 1$ (i.e., the qualitative reachability problem) can be solved just by examining the topology of the underlying transition system of M. More concretely, it can be shown that $\mathcal{P}(s_0 \rightarrow^* L) = 1$ if and only if L is reachable from all states that are reachable from s_0 (see, e.g., [38]).

Now we show that the tuple of all $\mathcal{P}(s \rightarrow^* L)$ values is a unique solution of an effectively computable system of linear equations, which will solve the quantitative reachability problem. Let us denote S^{YES} and S^{NO} the sets of states $s \in S$ such that $\mathcal{P}(s \rightarrow^* L) = 1$ and $\mathcal{P}(s \rightarrow^* L) = 0$, respectively. Both sets S^{YES} and S^{NO} are computable in polynomial time using the above assertion for deciding $\mathcal{P}(s \rightarrow^* L) = 1$, and the simple fact that $\mathcal{P}(s \rightarrow^* L) = 0$ if and only if L is not reachable from s. To every $s \in S$ we associate a real variable x_s (i.e., each x_s corresponds to the probability $\mathcal{P}(s \rightarrow^* L)$). Consider the following system of equations:

$$\begin{array}{ll} x_s = 1 & \text{if } s \in S^{YES}; \\ x_s = 0 & \text{if } s \in S^{NO}; \\ x_s = \sum_{s \rightarrow t} x_t \cdot Prob(s \rightarrow t) & \text{otherwise.} \end{array}$$

Proposition 4.1.1 ([22])**.** *The tuple of all $\mathcal{P}(s \rightarrow^* L)$ values is exactly the* unique *solution of the above system in $\mathbb{R}$.*

It follows that the probabilities $\mathcal{P}(s \rightarrow^* L)$ are always rational and explicitly computable in polynomial time.

Probabilistic Pushdown Automata

It is obvious that in the case of pPDA the reachability problem is undecidable for general sets L. Hence, in this section, we consider the reachability problem only for *simple* sets[1] L defined as follows.

Definition 4.1.2. *A set of configurations L of a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$ is* simple *if there is a set $\mathcal{H} \subseteq (Q \times \Gamma) \cup Q$ of heads such that for all configurations $p\alpha \in \mathcal{C}(\Delta)$ we have $p\alpha \in L$ if and only if the head of $p\alpha$ is in $\mathcal{H}$ (we also say that $\mathcal{H}$* determines *L).*

However, even for simple sets, results from finite case do not easily generalize to pPDA as illustrated below. First, it turns out that the solution to the qualitative reachability problem may depend not only on the topology of the underlying transition system but also on concrete transition probabilities (see Example 4.1.6). Second, the probability $\mathcal{P}(s_0 \rightarrow^* L)$ can be irrational as illustrated by the chain $\bar{\Delta}$ from Example 3.1.3. Indeed, we show in Example 4.1.7 that the probability $\mathcal{P}(sZ \rightarrow^* pZ)$ equals $(\sqrt{5}-1)/2$ (the "golden ratio"), which is irrational. Another example of this phenomenon is given in [33] in the language of

[1] In Section 4.5 we show how to extend our results to deal with regular sets L.

RMCs. This example straightforwardly translates to a pBPA with six stack symbols $X_1, \ldots, X_4, X, Y$ and the following transition rules: $X \stackrel{1/6}{\to} X_1X$, $X \stackrel{1/2}{\to} \varepsilon$, $X \stackrel{1/3}{\to} Y$, $X_i \stackrel{1}{\to} X_{i+1}X$ for $1 \leq i < 4$, $X_4 \stackrel{1}{\to} X$, $Y \stackrel{1}{\to} Y$. One can show (using Lemma 4.1.5) that the probability of reaching ε from X is a root of the polynomial $x^5 - 6x + 3$. However, the Galois theorem (see , e.g., [48]) implies that all roots of this polynomial are irrational, and moreover, not solvable by radicals.

In what follows we show that although the reachability problem is apparently more delicate in the case of pPDA than in the finite case, it is still decidable (in polynomial space).

4.1.1 Reachability and pPDA

The content of this section is based on results of [27, 33]. Throughout this section we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$. First, we simplify the reachability problem in the sense of the following lemma (its proof is presented in Section 4.6.4).

Lemma 4.1.3. *Given $p\alpha \in \mathcal{C}(\Delta)$ and a simple set $L \subseteq \mathcal{C}(\Delta)$, there are effectively computable (in polynomial time) a pPDA Δ' and a configuration q_0Z_0 of Δ', such that $\mathcal{P}(p\alpha \to^* L) = \mathcal{P}(q_0Z_0 \to^* q_0\varepsilon)$.*

Hence, in order to solve the reachability problem, it suffices to consider only the probabilities of the form $\mathcal{P}(pX \to^* q\varepsilon)$. The qualitative and quantitative reachability problems restricted to the probabilities of the form $\mathcal{P}(pX \to^* q\varepsilon)$ are called the qualitative and quantitative *termination* problems, respectively.

Let us introduce some abbreviations to simplify our presentation. Given $pX \in Q \times \Gamma$ and $q \in Q$, we denote $[pXq] = \mathcal{P}(pX \to^* q\varepsilon)$. Furthermore, we denote $[pX\uparrow] = 1 - \sum_{q \in Q}[pXq]$, the probability that a run initiated in pX never erases the stack. In this notation, the quantitative termination problem is to decide whether $[pXq] \sim \varrho$ for given $\sim \in \{=, <, >, \leq, \geq\}$ and $\varrho \in [0, 1]$.

Termination Problem and pPDA

We show that the quantitative termination problem is in **PSPACE**, which implies that also the quantitative reachability problem is in **PSPACE** due to Lemma 4.1.3. We start with the following simple observation.

Proposition 4.1.4. *Given $p, q \in Q$ and $X \in \Gamma$, the problem whether $[pXq] = 0$ is decidable in polynomial time. Hence, given $p\alpha \in \mathcal{C}(\Delta)$ and a simple set $L \subseteq \mathcal{C}(\Delta)$, the problem whether $\mathcal{P}(p\alpha \to^* L) = 0$ is in **P**.*

Proof. By definition, the probability $[pXq]$ equals 0 if and only if $q\varepsilon$ is not reachable from pX in the underlying transition system of M_Δ. The latter problem can be solved in polynomial time using standard algorithms for PDA (see, e.g., [14]). □

Now we show that the probabilities $[pXq]$ can be expressed as the least non-negative solution of an effectively computable system of quadratic equations. Let $\mathcal{V} := \{\langle pXq\rangle \mid p, q \in Q, X \in \Gamma\}$ be a set of variables over $\mathbb{R}$. That is, for every $[pXq]$ there is the associated variable $\langle pXq\rangle$. Consider the following system of recursive equations:

$$\langle pXq\rangle = \sum_{pX \xrightarrow{x} q\varepsilon} x + \sum_{pX \xrightarrow{x} rY} x\langle rYq\rangle + \sum_{pX \xrightarrow{x} rYZ, s\in Q} x\langle rYs\rangle\langle sZq\rangle \qquad (4.1)$$

The following lemma is proved in Section 4.6.1.

Lemma 4.1.5. *The tuple of all $[pXq]$ values is exactly the* least non-negative *solution of the above system of equations with respect to component-wise ordering.*

The following examples illustrate how the termination probabilities can be computed using the system (4.1) and Lemma 4.1.5.

Example 4.1.6. *Let us consider the pBPA Δ_w from Example 3.1.2. We denote p the only control state of Δ_w. Let us compute the probability $[pYp]$. By Lemma 4.1.5, $[pYq]$ is the least non-negative solution of $\langle pYp\rangle = (1-x) + x \cdot \langle pYp\rangle^2$. This equation has two solutions: 1 and $\frac{1-x}{x}$. Hence, if $x \leq \frac{1}{2}$, then $[pYp] = 1$, else $[pYp] = \frac{1-x}{x}$. Using similar arguments, one can show that $[pXp] = \frac{x}{1-x}$ for $x \leq \frac{1}{2}$, and $[pXp] = 1$ otherwise.*

Example 4.1.7. *Let us consider the pPDA $\bar{\Delta}$ from Example 3.1.3. The set $\mathcal{V}$ contains 12 variables. However, by Lemma 4.1.5, only two of them, $\langle pIp\rangle$ and $\langle pDp\rangle$, can get non-zero value in the least non-negative solution of the system (4.1). The equations for these two variables look as follows (we have already eliminated all variables except $\langle pIp\rangle$ and $\langle pDp\rangle$):*

$$\langle pIp\rangle = 0.5 + 0.5\langle pIp\rangle\langle pDp\rangle$$
$$\langle pDp\rangle = 0.5\langle pIp\rangle + 0.5\langle pDp\rangle\langle pDp\rangle$$

Using some elementary transformations, one obtains that any solution must satisfy $\langle pDp\rangle^3 - 4\langle pDp\rangle^2 + 4\langle pDp\rangle - 1 = 0$. It is easy to show that roots of this polynomial are 1, $\frac{3+\sqrt{5}}{2}$, and $\frac{3-\sqrt{5}}{2}$. Assigning $\langle pDp\rangle = \frac{3-\sqrt{5}}{2}$, the first equation yields $\langle pIp\rangle = \frac{\sqrt{5}-1}{2}$, which is the least solution of the above system. Hence, by Lemma 4.1.5, the probabilities $[pDp]$ and $[pIp]$ are equal to $\frac{3-\sqrt{5}}{2}$ and $\frac{\sqrt{5}-1}{2}$, respectively. It follows that the probability of reaching pZ from sZ is indeed $\frac{\sqrt{5}-1}{2}$ as we have announced in Example 3.1.3.

Let $x_1, \ldots, x_n$ be all variables of $\mathcal{V}$ (ordered arbitrarily). Given $p, q \in Q$ and $X \in \Gamma$, we denote $i(pXq)$ the index of the variable $\langle pXq\rangle$ among $x_1, \ldots, x_n$ (i.e., $x_{i(pXq)}$ is the variable $\langle pXq\rangle$). For every x_i there is an equation of the system (4.1) of the form $x_i = P_i$ where P_i is a multivariate polynomial in variables of $\mathcal{V}$. Note that the system (4.1) also defines a unique function $P : \mathbb{R}^n \to \mathbb{R}^n$ by

$P(a_1, \ldots, a_n) = (P_1(a_1, \ldots, a_n), \ldots, P_n(a_1, \ldots, a_n))$ for all $(a_1, \ldots, a_n) \in \mathbb{R}^n$ (here every $P_i(a_1, \ldots, a_n)$ is the value of P_i when each variable x_i is assigned the value a_i).

Lemma 4.1.5 now says that the tuple of all $[pXq]$ values is exactly the *least non-negative* fixed-point of P. Let us consider the following formula of $ExTh(\mathbb{R})$:

$$\Phi(x_1, \ldots, x_n) \equiv \left(\bigwedge_{i=1}^{n} x_i \geq 0 \right) \wedge \left(\bigwedge_{i=1}^{n} x_i = P_i \right)$$

Let us denote $\Psi(x_1) \equiv (\exists x_2) \cdots (\exists x_n) \Phi(x_1, \ldots, x_n)$ and let us assume, without the loss of generality, that $i(pXq) = 1$. Clearly, if $\Psi(a_1)$ is true, then a_1 is the first component of a non-negative fixed-point of P, and hence $a_1 \geq [pXq]$. Conversely, $\Psi([pXq])$ is true. Thus, in order to decide whether $[pXq] \sim \varrho$ for a given $\varrho \in [0, 1]$ and $\sim \in \{<, \leq\}$, it suffices to decide whether the formula of the form $(\exists x_1)(\Psi(x_1) \wedge x_1 \sim \varrho)$ is true. This problem is in **PSPACE** due to Theorem 2.3.1, because the formula $\Psi(x_1)$ is in $ExTh(\mathbb{R})$. The problem whether $[pXq] \sim \varrho$ for $\varrho \in [0, 1]$ and $\sim \in \{>, \geq\}$ can easily be reduced to the previous problem. Hence, we have proved that the quantitative termination problem is decidable in polynomial space. Applying Lemma 4.1.3 we obtain the following theorem.

Theorem 4.1.8. *For pPDA, the* quantitative *reachability problem is in* ***PSPACE****.*

It is easy to show, using the formula $\Phi(x_1, \ldots, x_n)$, that the "termination" probabilities $[pXq]$ are effectively expressible in $Th(\mathbb{R})$ by formulae computable in polynomial time. As several numerical features of pPDA (e.g., the average reward per service) are expressible using the termination probabilities, we would obtain **EXPTIME** upper bound on the complexity of algorithmic problems connected with these features (due to Theorem 2.3.1). However, we can do better. We show that the termination probabilities $[pXq]$ are expressible in $ExTh(\mathbb{R})$ by formulae of polynomial size computable in polynomial space. The benefit of this result is that the above mentioned **EXPTIME** upper bound drops to **PSPACE** due to Theorem 2.3.1.

The core result which allows for the expressibility of termination probabilities in $ExTh(\mathbb{R})$ is the following uniqueness theorem proved in [31].

Theorem 4.1.9 ([31])**.** *The tuple of all $[pXq]$ values is equal to the* unique *non-negative fixed point $(a_1, \ldots, a_n) = P(a_1, \ldots, a_n)$ of P that satisfies: $\sum_{q \in Q} a_{i(pXq)} < 1$ whenever $[pX\uparrow] > 0$, and $\sum_{q \in Q} a_{i(pXq)} \leq 1$, otherwise.*

Let us denote $\mathbf{S}_\Delta = \{pX \in Q \times \Gamma \mid [pX\uparrow] > 0\}$.

Corollary 4.1.10. *The set $\mathbf{S}_\Delta$ is computable in polynomial space. Moreover, given $\mathbf{S}_\Delta$ (i.e., assuming that $\mathbf{S}_\Delta$ has already been computed), $p, q \in Q$ and $X \in \Gamma$, the values $[pXq]$ and $[pX\uparrow]$ are effectively expressible in $ExTh(\mathbb{R})$ by formulae computable in polynomial time.*

Proof. The set $\mathbf{S}_\Delta$ is computable in polynomial space by Theorem 4.1.8. Now consider the following formula:

$$\Psi(x_1,\ldots,x_n) \equiv \bigwedge_{[pX\uparrow]>0} \sum_{q\in Q} x_{i(pXq)} < 1 \;\wedge\; \bigwedge_{[pX\uparrow]=0} \sum_{q\in Q} x_{i(pXq)} \leq 1 \;\wedge\; \bigwedge_{i=1}^{n} x_i \geq 0$$

By Theorem 4.1.9, the tuple of all $[pXq]$ values is the unique solution of the system (4.1) which satisfies $\Psi(x_1,\ldots,x_n)$. Now the effective expressibility of $[pXq]$ follows from Proposition 2.3.3. Another application of Proposition 2.3.3 shows that the value $[pX\uparrow] = 1 - \sum_{q\in Q}[pXq]$ is also effectively expressible in $ExTh(\mathbb{R})$. □

Lower Bound: Theorem 4.1.8 establishes the **PSPACE** upper bound on the reachability problem. There remains a question whether the bound is tight. A partial answer to this question is given in [33] where the Square Root Sum (SRS) problem is shown to be polynomially reducible to the reachability (termination) problem. This means that any improvement in the upper bound on the reachability problem would bring a major breakthrough in the complexity theory, because SRS is not known to be contained in **NP** for a long time. As noted in [33], the SRS problem has many applications in computational geometry and elsewhere.

Formally, the SRS problem is defined as follows: Given a vector $(a_1,\ldots,a_n) \in \mathbb{N}^n$ of natural numbers and $k \in \mathbb{N}$, decide whether $\sum_{i=1}^{n} \sqrt{a_i} \leq k$. The following proposition is proved in [33][2].

Proposition 4.1.11 ([33])**.** *The SRS problem is polynomially reducible to the quantitative termination problem for pBPA, and to the qualitative termination problem for pPDA (where only two control states are needed).*

pBPA: For completeness, let us mention that for pBPA, the *qualitative* termination problem can be solved in polynomial time. More formally,

Proposition 4.1.12 ([33])**.** *Given a pBPA Δ and a configuration X of Δ, it is decidable in polynomial time which one of the following three possibilities holds: $\mathcal{P}(X \to^* \varepsilon) = 0$, or $\mathcal{P}(X \to^* \varepsilon) = 1$, or $0 < \mathcal{P}(X \to^* \varepsilon) < 1$.*

As a corollary of this deep result and Theorem 4.3.14 we will obtain that even the qualitative reachability problem is decidable in polynomial time (observe that this does not follow from Lemma 4.1.3 because the proof of Lemma 4.1.3 adds new control states to the pPDA). In fact, we obtain even more general result which says that the qualitative model-checking problem for pBPA and ω-regular properties specified by deterministic Rabin automata is decidable in polynomial time (see Theorem 4.3.15).

[2]The proof given in [33] is written in the language of RMCs. Therefore, to obtain Proposition 4.1.11, one has to apply the translation procedure from Section 3.3.

Approximation: We have proved above that for pPDA the probability $\mathcal{P}(p\alpha \rightarrow^* L)$ of reaching the simple set of configurations L from the configuration $p\alpha$ can be irrational. Hence, $\mathcal{P}(p\alpha \rightarrow^* L)$ cannot always be effectively computed and one has to be satisfied with an approximate value. In the rest of this section we survey some approaches to approximating the probability $\mathcal{P}(p\alpha \rightarrow^* L)$.

The easiest way is to use the binary search algorithm. Let us describe how this algorithm can be applied to compute the probability $[pXq]$. The algorithm iteratively computes a sequence of approximations $x_0, x_1, x_2, \ldots$ where $x_0 = 0$ and $x_n = \frac{k}{2^n}$ satisfies $|[pXq] - x_n| \leq \frac{1}{2^n}$. For each $n \geq 1$, the approximation x_{n+1} is computed based on $x_n = \frac{k}{2^n}$ as follows: First, decide whether $x_n \leq [pXq]$ using Theorem 4.1.8. If $x_n \leq [pXq]$, then put $x_{n+1} = \frac{2k+1}{2^{n+1}}$, else put $x_{n+1} = \frac{2k-1}{2^{n+1}}$.

Another way how the values $[pXq]$ can be approximated is to iterate the function P defined by the system 4.1. As we note in the end of the proof of Lemma 4.1.5, the function P is monotonic, and $\lim_{n\to\infty} P^n(\vec{0})$ is equal to the vector of all $[pXq]$ values. However, this approach possesses two serious shortcomings. First, there is no way how to measure precision of the approximations (what we only know is that the sequence of the approximations converges but we do not know how fast). Second, there are examples where these approximations converge rather slowly (see [33]). For example, consider a pBPA Δ_{slow} with one stack symbol X and the following transition rules: $X \stackrel{1/2}{\rightarrow} XX$ and $X \stackrel{1/2}{\rightarrow} \varepsilon$. It is argued in [33] that the probability $\mathcal{P}(X \rightarrow^* \varepsilon)$ equals 1, but for each n, $P^n(\vec{0}) \geq 1 - \frac{1}{2^{k+2}}$ implies $n \geq 2^{k-1}$. Hence, it takes exponentially many iterations to obtain one bit of precision.

Possibly the best known method (called *decomposed Newton's method*) for approximating the probabilities $[pXq]$ is described in [33], and is based on a multivariable version of Newton's method. It is proved in [33] that this method always performs at least as well as the simple approximation algorithm which computes $P^n(\vec{0})$ for $n \rightarrow \infty$. There are also some evidences, given in [33], that the decomposed Newton's method should approximate the probabilities $[pXq]$ much faster than the simple approximation. For example, the decomposed Newton's method approximates the probability $\mathcal{P}(X \rightarrow \varepsilon)$ in the above example pBPA Δ_{slow} exponentially faster than the simple approximation algorithm (i.e., it gives one bit of precision per iteration). Moreover, as noted in [33], for finite Markov chains (and even for a larger class of linear RMCs) the decomposed Newton's method computes the fixed point in one iteration. For a detailed description of the decomposed Newton's method and related results see [33].

4.2 Linear Temporal Logic

Linear temporal logic (LTL) is one of the most successful temporal logics in the area of specification and verification of reactive systems. A lot of work has been devoted to the model-checking problem for LTL both in non-probabilistic setting (see, e.g., [19, 14, 30]) and probabilistic setting (see, e.g., [22, 50, 32]).

LTL is defined as follows. Let $Ap = \{a, b, c, \dots\}$ be a countably infinite set of *atomic propositions*. The syntax of LTL formulae is defined by the following abstract syntax equation:

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

Here $a \in Ap$. Formulae of the form $\mathtt{tt}\,\mathcal{U}\,\varphi$ are usually abbreviated as $\Diamond(\varphi)$. The size $|\varphi|$ of the formula φ is defined to be the number of connectives in φ. Let $M = (S, \to, Prob)$ be a Markov chain. The semantics of LTL formulae is defined over runs of M. Let $\nu : Ap \to 2^S$ be a valuation which assigns to each atomic proposition a set of states that satisfy this atomic proposition. Given a run w of M, the semantics of LTL formulae is defined inductively as follows.

$$\begin{array}{lll}
w \models^\nu \mathtt{tt} & & \\
w \models^\nu a & \text{iff} & w(0) \in \nu(a) \\
w \models^\nu \neg\varphi & \text{iff} & w \not\models^\nu \varphi \\
w \models^\nu \varphi_1 \wedge \varphi_2 & \text{iff} & w \models^\nu \varphi_1 \text{ and } w \models^\nu \varphi_2 \\
w \models^\nu \mathcal{X}\varphi & \text{iff} & w_1 \models^\nu \varphi \\
w \models^\nu \varphi_1 \,\mathcal{U}\, \varphi_2 & \text{iff} & \exists j \geq 0 : w_j \models^\nu \varphi_2 \text{ and } w_i \models^\nu \varphi_1 \text{ for all } 0 \leq i < j
\end{array}$$

Given an LTL formula φ, a valuation ν, and a state s_0 of M, we denote $Run(s_0, \varphi, \nu) = \{w \in Run(s_0) \mid w \models^\nu \varphi\}$. It was shown in [50] that the set $Run(s_0, \varphi, \nu)$ is measurable. We denote $\mathcal{P}(s_0, \varphi, \nu) = \mathcal{P}(Run(s_0, \varphi, \nu))$. We write $\mathcal{P}(s_0, \varphi)$ instead of $\mathcal{P}(s_0, \varphi, \nu)$ whenever the valuation ν is clear from the context. We consider the following problems.

- *qualitative LTL model-checking*: Is $\mathcal{P}(s_0, \varphi) = 1$?
- *quantitative LTL model-checking*:

$$\text{Given } \varrho \in [0, 1] \text{ and } \sim \in \{=, <, >, \leq, \geq\}, \text{ is } \mathcal{P}(s_0, \varphi) \sim \varrho \,?$$

Note that $\mathcal{P}(s_0, \varphi) = 1 - \mathcal{P}(s_0, \neg\varphi)$, and hence the problem whether $\mathcal{P}(s_0, \varphi) = 0$ is equivalent to the qualitative LTL model-checking problem.

4.2.1 LTL and pPDA

Clearly, in the case of Markov chains generated by pPDA the LTL model-checking problem is undecidable for general valuations. Hence, in this section we consider only *simple* valuations which assign simple sets of configurations to all atomic propositions (see Section 4.5 for an extension of our results to more general valuations).

The decidability of the LTL model-checking problem for pPDA was first proved in [17] together with some complexity estimates. The algorithm presented in [17] is based on a reduction of the LTL model-checking problem to the model-checking of ω-regular properties, and is not optimal. Better complexity estimates for the LTL model-checking problem were obtained in [32] for RMCs using a more direct approach. In the rest of this section we show how the results of [32] can be used to solve the LTL model-checking problem for pPDA.

From pPDA to RMC

Let $\Delta = (Q, \Gamma, \delta, Prob)$ be a pPDA, let τ be an LTL formula, and let ν be a valuation which assigns to each $a \in Ap$ a simple set $\nu(a)$ determined by a set of heads $\mathcal{H}_a$ (see Definition 4.1.2). Let us fix an initial configuration $q_0 Z_0 \in Q \times \Gamma$ such that $[q_0 Z_0 \uparrow] = 1$. The following lemma shows that the initial configuration is chosen without the loss of generality.

Lemma 4.2.1. *Given $p\alpha \in \mathcal{C}(\Delta)$, there are effectively computable (in polynomial time) a pPDA Δ', a configuration $q_0 Z_0$ of Δ' satisfying $[q_0 Z_0 \uparrow] = 1$, an LTL formula τ', and a simple valuation ν', such that $\mathcal{P}(p\alpha, \tau, \nu) = \mathcal{P}(q_0 Z_0, \tau', \nu')$.*

We reduce the LTL model-checking problem for pPDA to the same problem for RMCs, and then apply results of [32]. Let A be the RMC obtained from Δ using the translation procedure described in Section 3.3 and let V be the set of vertices of A. There is a small technical difficulty caused by the fact that the translation procedure between pPDA and RMCs (presented in Section 3.3) does not yield isomorphic chains M_Δ and M_A. There are some mid steps in M_A with probability one whenever a call or a return occurs. In order to avoid this problem we have to modify the formula τ before the model-checking algorithm of [32] is applied to A.

Let a_{skip} be a fresh atomic proposition which does not occur in τ. Let us define a transformation $\mathcal{T}$, which transforms subformulae of τ as follows: $\mathcal{T}(a) = a$ for $a \in Ap$, $\mathcal{T}(\varphi_1 \wedge \varphi_2) = \mathcal{T}(\varphi_1) \wedge \mathcal{T}(\varphi_2)$, $\mathcal{T}(\neg\varphi) = \neg\mathcal{T}(\varphi)$, $\mathcal{T}(\mathcal{X}\varphi) = \mathcal{X}(a_{skip}\ \mathcal{U}\ (\mathcal{T}(\varphi_1) \wedge \neg a_{skip}))$, and finally $\mathcal{T}(\varphi_1\ \mathcal{U}\ \varphi_2) = (\mathcal{T}(\varphi_1) \vee a_{skip})\ \mathcal{U}\ (\mathcal{T}(\varphi_2) \wedge \neg a_{skip})$.

We define a new valuation $\nu' : Ap \to 2^{S_A}$ (here S_A is the set of states of M_A) as follows: For $a \in Ap \setminus \{a_{skip}\}$ we put $\nu'(a) = \{\langle \beta, pX\rangle \in S_A \mid pX \in \mathcal{H}_a\}$, and we put $\nu'(a_{skip}) = \{\langle \beta, v\rangle \in S_A \mid v \in V \setminus (Q \times \Gamma)\}$ (here V is the set of vertices of A).

Let Θ be the mapping from Proposition 3.3.1. It is easy to verify that for all $w \in Run(\langle \varepsilon, q_0 Z_0\rangle)$, we have $w \models^{\nu'} \mathcal{T}(\tau)$ if and only if $\Theta(w) \models^{\nu} \tau$. It follows from Proposition 3.3.1 that $\mathcal{P}(\langle \varepsilon, q_0 Z_0\rangle, \mathcal{T}(\tau), \nu') = \mathcal{P}(q_0 Z_0, \tau, \nu)$.

Now the results of [32] can be applied and we obtain the following theorem.

Theorem 4.2.2 ([32])**.**

1. *For pPDA, the* qualitative *LTL model-checking problem is in **PSPACE** in the size of pPDA, and in **EXPTIME** in the size of formulae.*

2. *For pBPA, the* qualitative *LTL model-checking problem is **EXPTIME**-complete, and in **P** in the size of pBPA.*

3. *For pPDA, the* quantitative *LTL model-checking problem is in **PSPACE** in the size of pPDA, and in **EXPSPACE** in the size of formulae.*

4.3 ω-regular Properties

In this section we study the model-checking problem for pPDA and properties specified by finite state automata over infinite words (ω-regular properties). Similarly to the previous sections, we start with a formal definition of the problem for Markov chains.

Definition 4.3.1. *An* ω-automaton *is a tuple* $\mathcal{R} = (\Sigma, R, \rho, s_I, F)$*, where* Σ *is a finite* alphabet*,* R *is a finite set of* states*,* $\rho \subseteq R \times \Sigma \times R$ *is a* transition relation *(we write* $s \xrightarrow{a} t$ *instead of* $(s, a, t) \in \rho$*),* s_I *is the* initial state*, and* F *specifies an acceptance condition (see below). The* ω*-automaton* $\mathcal{R}$ *is* deterministic *if* ρ *is a (total) function.*

We assume, without the loss of generality, that for each $a \in \Sigma$ and each $s \in R$ there is $t \in R$ such that $(s, a, t) \in \rho$. We define the size $|\mathcal{R}|$ of the ω-automaton $\mathcal{R}$ to be the sum $|\rho| + |F|$.

The symbol Σ^ω denotes the set of all infinite words over the alphabet Σ. A *computation* of $\mathcal{R}$ on a word $w = w(0)w(1)\cdots \in \Sigma^\omega$ is a sequence $\omega = t_0, t_1, \ldots$ of states of $\mathcal{R}$ such that $t_0 = s_I$ and for all $i \geq 0$ we have $t_i \xrightarrow{w(i)} t_{i+1}$. Note that if $\mathcal{R}$ is deterministic, then there is exactly one computation on a given infinite word. A computation ω of $\mathcal{R}$ is *accepting* if it satisfies the acceptance condition specified by F. The automaton $\mathcal{R}$ accepts a word $w \in \Sigma^\omega$ if there exists an accepting computation of $\mathcal{R}$ on w. The set of all $w \in \Sigma^\omega$ accepted by $\mathcal{R}$ is denoted $\mathcal{L}(\mathcal{R})$.

Depending on the way the acceptance condition is specified, various types of ω-automata can be distinguished (see, e.g., [47, 49]). In this work we restrict our attention only to the following two types of ω-automata: deterministic Rabin automata and non-deterministic Büchi automata. Both these types of ω-automata have the same expressive power as they both determine the class of ω-regular languages (see [47]). The acceptance conditions of both Büchi and Rabin automata are defined in terms of states that occur infinitely often in computations. Hence, for every computation ω of $\mathcal{R}$, we denote $Inf(\omega)$ the set of states of $\mathcal{R}$ that occur infinitely often in ω. If $\mathcal{R}$ is a Rabin automaton, then F is of the form $\{(A_1, B_1), \ldots, (A_n, B_n)\}$ where $A_1, \ldots, A_n, B_1, \ldots, B_n \subseteq R$, and a computation ω is accepting if and only if there is $(A_i, B_i) \in F$ such that $Inf(\omega) \cap A_i \neq \emptyset$ and $Inf(\omega) \cap B_i = \emptyset$. If $\mathcal{R}$ is a Büchi automaton, then F is a subset of R, and a computation ω is accepting if and only if $Inf(\omega) \cap F \neq \emptyset$.

Now let us describe how the ω-automata can be used to specify properties of Markov chains. Let $M = (S, \rightarrow, Prob)$ be a Markov chain. Let $\nu : S \rightarrow \Sigma$ be a valuation which assigns to each state of M a letter of the alphabet Σ. Each run w of M determines a unique word $w_\nu \in \Sigma^\omega$ by $w_\nu(i) = \nu(w(i))$ for all $i \geq 0$. Given a run w of M, we write $w \in \mathcal{L}(\mathcal{R})$ (and say that $\mathcal{R}$ accepts w) instead of $w_\nu \in \mathcal{L}(\mathcal{R})$ whenever the valuation ν is known from the context.

Given a state s_0 of M, we denote $Run(s_0, \mathcal{R}, \nu) = \{w \in Run(s_0) \mid w_\nu \in \mathcal{L}(\mathcal{R})\}$. It was shown in [50] that the set $Run(s_0, \mathcal{R}, \nu)$ is measurable

whenever $\mathcal{R}$ is either a Büchi or Rabin automaton. We denote $\mathcal{P}(s_0, \mathcal{R}, \nu) = \mathcal{P}(Run(s_0, \mathcal{R}, \nu))$. We write $Run(s_0, \mathcal{R})$ and $\mathcal{P}(s_0, \mathcal{R})$ instead of $Run(s_0, \mathcal{R}, \nu)$ and $\mathcal{P}(s_0, \mathcal{R}, \nu)$, respectively, whenever the valuation ν is known from the context. We consider the following problems:

- *qualitative ω-regular model-checking*: Is $\mathcal{P}(s_0, \mathcal{R}) = 1$? Is $\mathcal{P}(s_0, \mathcal{R}) = 0$?
- *quantitative ω-regular model-checking*:

$$\text{Given } \varrho \in [0, 1], \text{ and } \sim \in \{<, \leq, >, \geq, =\}, \text{ is } \mathcal{P}(s_0, \mathcal{R}) \sim \varrho \text{ ?}$$

Whenever $\mathcal{P}(s_0, \mathcal{R}) \sim \varrho$, we say that the state s_0 of M satisfies the given ω-regular property (specified by $\mathcal{R}$) with the prescribed probability (specified by " $\sim \varrho$ ").

Finite Markov chains

To motivate our approach to the ω-regular model-checking for pPDA, we shortly sketch a model-checking algorithm for finite Markov chains and deterministic Rabin automata. The algorithm relies on the following basic result from the theory of finite Markov chains (see, e.g., [38]):

Proposition 4.3.2. *Let M be a finite Markov chain and let s be a state of M. Almost surely, a run of $Run(s)$ enters all states of a BSCC of M infinitely often.*

Let $M = (S, \to, Prob)$ be a finite Markov chain and let $\mathcal{R} = (\Sigma, R, \rho, s_I, F)$ be a deterministic Rabin automaton where $F = \{(A_1, B_1), \ldots, (A_n, B_n)\}$. Let us fix a valuation $\nu : S \to \Sigma$. We define a "product" Markov chain $M \times \mathcal{R}$ whose set of states is $S \times R$ and transitions are of the form $(s, t) \xrightarrow{x} (s', t')$ where $s \xrightarrow{x} s'$ in M and $t \xrightarrow{\nu(s)} t'$ in $\mathcal{R}$. In other words, $M \times \mathcal{R}$ corresponds to a synchronous parallel composition of M and $\mathcal{R}$. We show that the ω-regular model-checking problem for M and $\mathcal{R}$ reduces to the reachability problem for $M \times \mathcal{R}$.

We say that a BSCC C of $M \times \mathcal{R}$ is *accepting* if there is $(A_i, B_i) \in F$ such that $O_C \cap A_i \neq \emptyset$ and $O_C \cap B_i = \emptyset$ where $O_C = \{t \in R \mid (s, t) \in C\}$.

Lemma 4.3.3. *Given $s_0 \in S$, the probability $\mathcal{P}(s_0, \mathcal{R})$ is equal to the probability of reaching an accepting BSCC of $M \times \mathcal{R}$ from (s_0, s_I).*

Proof. Note that a function Θ, which maps each $(s, t) \in S \times R$ to s, is a quotient of $M \times \mathcal{R}$ onto M. Let us denote $Accept$ the set of all runs $w = (s_0, t_0), (s_1, t_1), \ldots$ of $Run[M \times \mathcal{R}]((s_0, s_I))$ such that $t_0, t_1, \ldots$ is an accepting computation of $\mathcal{R}$ (note that $t_0, t_1, \ldots$ is the unique computation of $\mathcal{R}$ over $\nu(s_0), \nu(s_1), \ldots$). Then $\Theta(Accept) = Run(s_0, \mathcal{R})$, and hence, by Proposition 2.2.5, $\mathcal{P}(Accept) = \mathcal{P}(\Theta(Accept)) = \mathcal{P}(s_0, \mathcal{R})$.

Now let us denote Bot the set of all runs of $Run[M \times \mathcal{R}]((s_0, s_I))$ that enter all states of some BSCC of $M \times \mathcal{R}$ infinitely many times. It follows from Proposition 4.3.2 that $\mathcal{P}(Bot) = 1$. Moreover, for all $w \in Bot$, we have $w \in Accept$ if and only if the BSCC entered by w is accepting. It follows that $\mathcal{P}(Accept)$, and hence also $\mathcal{P}(s_0, \mathcal{R})$, is equal to the probability of reaching an accepting BSCC of $M \times \mathcal{R}$ from (s_0, s_I). □

It follows from Proposition 4.1.1 that the probability of reaching an accepting BSCC of $M \times \mathcal{R}$ can be evaluated explicitly in polynomial time, which solves both the quantitative and qualitative ω-regular model-checking problems in polynomial time. Moreover, the answer to the qualitative model-checking problem does not depend on concrete transition probabilities in $M \times \mathcal{R}$ but only on the topology of $M \times \mathcal{R}$. In particular, the probability $\mathcal{P}(s_0, \mathcal{R})$ is equal to 1 (or 0) if and only if all BSCCs of $M \times \mathcal{R}$ reachable from (s_0, s_I) are accepting (or not accepting, resp.).

Probabilistic Pushdown Automata

Decidability of the ω-regular model-checking problem for pPDA was first proved in [27] for deterministic Büchi automata. This result was extended to general ω-regular properties (expressed by deterministic Muller automata) in [17] where **EXPSPACE** and **2-EXPTIME** upper bounds were derived for the qualitative and quantitative model-checking problem, respectively. Finally, these complexity bounds were improved in [31].

In this work we follow the approach of [27, 17] while including some crucial ideas from [31]. Although the presented solution of the ω-regular model-checking problem was not completely invented by the author of this thesis, there are some reasons for its detailed presentation: First, the original decidability proof for ω-regular properties was co-authored by the author of this thesis. Second, the model-checking algorithm for ω-regular properties introduces the finite Markov chain $\mathbf{X}_\Delta$ which turns out to be the crucial tool in the verification of pPDA. Finally, the effective verification of ω-regular properties is the main building block in the model-checking algorithms for qualitative branching-time logics studied in Chapter 5. In addition, we study the regularity problem for the set of configurations that satisfy a given ω-regular property with a prescribed probability, which was not studied in [31] (for a precise formulation of this problem see Section 4.4).

How can the ω-regular model-checking problem be solved for pPDA? One may try to generalize the techniques described above for finite Markov chains. However, a direct generalization meets several obstacles not present in the finite case. First, similarly to the reachability problem, the probability measure of all runs accepted by a given ω-automaton can be irrational, and hence not explicitly computable in an effective way. This obstacle is avoided by expressing this probability in $ExTh(\mathbb{R})$. A more serious problem is that the Markov chain generated by a given pPDA may have infinitely many (or none) bottom strongly connected components, and moreover, some of them can be infinite. The topological argument for the qualitative model-checking problem does not work for pPDA either: Consider the pPDA Δ_w from Example 3.1.2. For $x = \frac{1}{2}$, almost all runs of $Run(Z)$ enter the configuration Z infinitely many times. On the other hand, if $x \neq \frac{1}{2}$, then the probability of all runs of $Run(Z)$ that never return to Z is greater than 0.

The above problems are solved in [27] by introducing a finite Markov chain that suitably abstracts the behavior of the infinite chain generated by a given pPDA. This abstraction allows us to reduce problems for chains generated by pPDA to

problems for finite chains. The only problem is that transition probabilities of the finite chain cannot (in general) be computed explicitly (they can be irrational), but merely effectively expressed in $ExTh(\mathbb{R})$. We follow the same approach in this thesis, and for a given pPDA Δ define a finite Markov chain $\mathbf{X}_\Delta$ that abstracts M_Δ (the finite chain abstracting M_Δ is defined in a slightly different way in [27], but the differences are only syntactical). It turns out that $\mathbf{X}_\Delta$ is a very powerful tool and it is heavily used in the rest of this thesis.

The rest of this section is organized as follows. In Section 4.3.1 we define the Markov chain $\mathbf{X}_\Delta$ and discus its basic properties. In Section 4.3.2 we present several results on the decidability and complexity of the model-checking problem for pPDA and deterministic Rabin automata. In Section 4.3.3 we comment on extensions of these results to the model-checking problem for pPDA and nondeterministic Büchi automata. Finally, in Section 4.4 we study the regularity problem for the set of configurations that satisfy a given ω-regular property with a prescribed probability. Complete formal proofs of crucial propositions are presented in Section 4.6.

4.3.1 The Markov Chain $\mathbf{X}_\Delta$

In this subsection we define the finite Markov chain $\mathbf{X}_\Delta$ for a given pPDA Δ, and prove its basic properties. Let us fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$. The intuitive meaning of the following definition is explained below.

Definition 4.3.4. *The Markov chain $\mathbf{X}_\Delta$ is a finite chain with the set of states $\mathbf{S}_\Delta = \{pX \in Q \times \Gamma \mid [pX\uparrow] > 0\}$, whose transitions (denoted $pX \hookrightarrow qY$) are defined as follows: Given $pX, qY \in \mathbf{S}_\Delta$, there is a transition $pX \hookrightarrow qY$ in $\mathbf{X}_\Delta$ with a probability $x \in (0, 1]$ if and only if*

$$x = \sum_{pX \xrightarrow{y} qY} \frac{y[qY\uparrow]}{[pX\uparrow]} + \sum_{pX \xrightarrow{y} qYZ} \frac{y[qY\uparrow]}{[pX\uparrow]} + \sum_{pX \xrightarrow{y} rZY} \frac{y[rZq][qY\uparrow]}{[pX\uparrow]} > 0$$

The following lemma justifies that $\mathbf{X}_\Delta$ is indeed a Markov chain. It is formally proved in Section 4.6.2.

Lemma 4.3.5. *For every $pX \in \mathbf{S}_\Delta$ we have*

$$\sum_{qY \in \mathbf{S}_\Delta} \left(\sum_{pX \xrightarrow{y} qY} \frac{y[qY\uparrow]}{[pX\uparrow]} + \sum_{pX \xrightarrow{y} qYZ} \frac{y[qY\uparrow]}{[pX\uparrow]} + \sum_{pX \xrightarrow{y} rZY} \frac{y[rZq][qY\uparrow]}{[pX\uparrow]} \right) = 1$$

Now we show how the chain $\mathbf{X}_\Delta$ characterizes the behavior of Δ. We need some additional definitions. Let $pX\alpha$ be a configuration of Δ, where $X \in \Gamma$ and $\alpha \in \Gamma^*$. We say that a run $w \in Run(pX\alpha)$ is *clean* if all configurations in w are

of the form $q\beta\alpha$, where $\beta \in \Gamma^+$. In other words, α is never accessed in a clean run of $pX\alpha$. We use $Clean(pX\alpha)$ to denote the set of all clean runs of $Run(pX\alpha)$.

Let $w = p_0\alpha_0, p_1\alpha_1, \ldots$ be a clean run of $Run(pX\alpha)$ (i.e., $p_0\alpha_0 = pX\alpha$). A configuration $p_i\alpha_i$ of w, where $i \geq 0$, is *minimal* if $|\alpha_i| \leq |\alpha_j|$ for all $j > i$. The *k-th minimum* of w, denoted $\min_k(w)$, is the k-th minimal configuration of w. The *index* of the k-th minimum, denoted $ind_k(w)$, is the i such that $w(i)$ is the k-th minimum of w. Intuitively, the minimal configurations of a given run are exactly the positions where one can forget about the stack content below the top-of-the-stack symbol, because these symbols are never accessed in the future.

An intuitive explanation of the relationship between the behavior of $\mathbf{X}_\Delta$ and Δ is following: Observing heads of minima of runs of $Clean(pX\alpha)$ is "the same" as observing runs of $\mathbf{X}_\Delta$ initiated in pX. This is caused by the fact, that the probability of each transition $qY \hookrightarrow rZ$ is equal to the probability that in a run $w \in Clean(pX\alpha)$ the head of $i + 1^{th}$ minimum is rZ under the condition that the head of i^{th} minimum is qY (here we assume that qY is the i^{th} minimum of some run, with a positive probability).

Let us formulate this intuitive idea precisely. To every $w \in Clean(pX\alpha)$ we associate its *footprint*: $fp(w) = \mathbf{X}^1_\Delta(w), \mathbf{X}^2_\Delta(w), \ldots$ where $\mathbf{X}^i_\Delta(w)$ is equal to the head of $\min_i(w)$ for every $i \geq 1$. Observe that there can be clean runs whose footprints are *not* paths in $\mathbf{X}_\Delta$. Indeed, consider the pPDA Δ_w from Example 3.1.2 with $x = \frac{1}{2}$. It is easy to see that $\mathbf{X}_{\Delta_w}$ has only one state Z. Now the run $Z, XZ, X^2Z, X^3Z, \ldots$ has the footprint $Z, X, X, \ldots$ which is not a path in X_{Δ_w}.

We denote $Good(pX)$ the set of all runs of $Clean(pX)$ whose footprints are runs of $\mathbf{X}_\Delta$ (runs of $Good(pX)$ are called *good*). The following crucial lemma shows (among others) that footprints of almost all runs of $Clean(pX)$ are runs of $\mathbf{X}_\Delta$. A formal proof of this proposition is deferred to Section 4.6.2.

Lemma 4.3.6. *Given $pX \in \mathbf{S}_\Delta$ and a measurable set A of runs of $\mathbf{X}_\Delta$ initiated in pX (i.e., $A \subseteq Run[\mathbf{X}_\Delta](pX)$), the set $fp^{-1}(A) \subseteq Good(pX)$ is measurable, and moreover, $\mathcal{P}(fp^{-1}(A) \mid Clean(pX)) = \mathcal{P}(A)$, where the probability $\mathcal{P}(A)$ is measured in $\mathbf{X}_\Delta$. In particular, $\mathcal{P}(Good(pX) \mid Clean(pX)) = 1$.*

Let us illustrate the concept of $\mathbf{X}_\Delta$ on the following simple example.

Example 4.3.7. *Consider again the pPDA $\bar{\Delta}$ from Example 3.1.3. We have shown in Example 4.1.7 that the probability $[pIp]$ is equal to $\frac{\sqrt{5}-1}{2}$, which means that $[pI\uparrow] = \frac{3-\sqrt{5}}{2}$. Similarly, $[pDp] = \frac{3-\sqrt{5}}{2}$ which implies $[pD\uparrow] = \frac{\sqrt{5}-1}{2}$. The Markov chain $\mathbf{X}_{\bar{\Delta}}$ is depicted in Figure 4.1 (only states reachable from sZ are drawn).*

By Lemma 4.3.6 and Proposition 4.3.2, almost all runs of $Run(sZ)$ contain infinitely many minimal configurations with the head in $\{pZ, pI\}$.

The following two lemmas impose some complexity bounds on basic algorithmic problems concerning $\mathbf{X}_\Delta$ (equivalent forms of these lemmas were proved also in [31]).

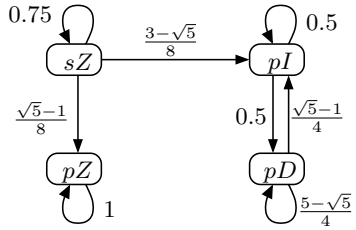

Figure 4.1: The Markov chain $X_{\bar{\Delta}}$

Lemma 4.3.8. *$\mathbf{S}_\Delta$ is computable in polynomial space. If Δ is a pBPA, then $\mathbf{S}_\Delta$ is computable in polynomial time. Given $\mathbf{S}_\Delta$ (i.e., assuming that $\mathbf{S}_\Delta$ has already been computed), the underlying transition system of $\mathbf{X}_\Delta$ is computable in polynomial time, and transition probabilities of $\mathbf{X}_\Delta$ are effectively expressible in $ExTh(\mathbb{R})$ by formulae computable in polynomial time.*

Proof. The complexity estimates for computing $\mathbf{S}_\Delta$ follow from Corollary 4.1.10 and Proposition 4.1.12, respectively.

Now for all $pX, qY \in \mathbf{S}_\Delta$ we have $pX \hookrightarrow qY$ if and only if either $pX \to qY$, or $pX \to qYZ$ for some $Z \in \Gamma$, or there is $rZ \in Q \times \Gamma$ such that $pX \to rZY$ and $[rZq] > 0$. The problem whether $[rZq] > 0$ is in **P** by Proposition 4.1.4, and thus the problem whether $pX \hookrightarrow qY$ is also in **P**. It follows that the underlying transition system of $\mathbf{X}_\Delta$ is computable in polynomial time (once $\mathbf{S}_\Delta$ is given).

Finally, the effective expressibility of transition probabilities of $\mathbf{X}_\Delta$ follows from the definition of $\mathbf{X}_\Delta$, Corollary 4.1.10, and Proposition 2.3.3. □

Lemma 4.3.9. *Given $\mathbf{S}_\Delta$ and a set $A \subseteq \mathbf{S}_\Delta$, for each $pX \in \mathbf{S}_\Delta$, the probability $\mathcal{P}(pX \hookrightarrow^* A)$ is effectively expressible in $ExTh(\mathbb{R})$ by a formula computable in polynomial time. Hence, the problem whether $\mathcal{P}(pX \hookrightarrow^* A) \sim \varrho$ for a given $\varrho \in [0,1]$ and $\sim \in \{<, >, =, \leq, \geq\}$ is decidable in polynomial space.*

Proof. By Lemma 4.3.8, the underlying transition system of $\mathbf{X}_\Delta$ can be computed in polynomial time (once $\mathbf{S}_\Delta$ is given). We showed in the beginning of Section 4.1 (Finite Markov Chains) that the problem whether $\mathcal{P}(pX \hookrightarrow^* A) = 1$ or $\mathcal{P}(pX \hookrightarrow^* A) = 0$ is decidable in polynomial time just by considering the topology of $\mathbf{X}_\Delta$. It follows that a system of linear equations (with symbolically represented coefficients) whose unique solution is the vector of probabilities $\mathcal{P}(pX \hookrightarrow^* A)$, can be computed in polynomial time (see Proposition 4.1.1). Observe that coefficients of the system are transition probabilities of $\mathbf{X}_\Delta$ that are effectively expressible in $ExTh(\mathbb{R})$ by formulae computable in polynomial time due to Lemma 4.3.8 (once $\mathbf{S}_\Delta$ is given). Hence, by Proposition 2.3.3, the probabilities $\mathcal{P}(pX \hookrightarrow^* A)$ are effectively expressible in $ExTh(\mathbb{R})$ by formulae computable in polynomial time, which proves the first part of this lemma. The rest follows from Theorem 2.3.1. □

Finally, let us introduce some notation used in connection with $\mathbf{X}_\Delta$. We denote BSCC_Δ the set of all bottom strongly connected components of $\mathbf{X}_\Delta$. To each $C \in \text{BSCC}_\Delta$ we associate the set $Run(pX, C)$ consisting of all $w \in Good(pX)$ such that the footprint of w enters the component C. The next proposition follows immediately from Lemma 4.3.6 and Proposition 4.3.2.

Proposition 4.3.10. *For all $pX \in \mathbf{S}_\Delta$*

$$\sum_{C \in BSCC_\Delta} \mathcal{P}(Run(pX, C) \mid Clean(pX)) = 1$$

Moreover, $\mathcal{P}(Run(pX, C) \mid Clean(pX)) = \mathcal{P}(pX \hookrightarrow^* C)$ for $C \in BSCC_\Delta$.

4.3.2 Deterministic Rabin Automata and pPDA

In this section we show how to solve the ω-regular model-checking problem for pPDA and deterministic Rabin automata. Let us fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$, and a *deterministic* Rabin automaton $\mathcal{R} = (\Sigma, R, \rho, s_I, F)$ where $F = \{(A_1, B_1), \ldots, (A_n, B_n)\}$ and $\Sigma = (Q \times \Gamma) \cup Q$. We also fix the valuation $\nu : \mathcal{C}(\Delta) \to \Sigma$ such that $\nu(p\alpha)$ is the head of $p\alpha$ for every $p\alpha \in \mathcal{C}(\Delta)$. Later on, we show how to extend our results to more general regular valuations (see Section 4.5). Let us fix an initial configuration $q_0 Z_0 \in Q \times \Gamma$ such that all runs of $Run(q_0 Z_0)$ are clean. The following lemma shows that the initial configuration is chosen without the loss of generality:

Lemma 4.3.11. *Given $p\alpha \in Q \times \Gamma^+$, there are effectively computable (in polynomial time) a pPDA Δ', a configuration $q_0 Z_0$ of Δ' satisfying $[q_0 Z_0 \uparrow] = 1$, and a Rabin automaton $\mathcal{R}'$, such that $\mathcal{P}(p\alpha, \mathcal{R}) = \mathcal{P}(q_0 Z_0, \mathcal{R}')$. For $q \in Q$, the problem whether $\mathcal{R}$ accepts q^ω is decidable in time polynomial in $|\mathcal{R}|$.*

Now we present a solution of the ω-regular model-checking problem. Our approach resembles the one described above for finite chains.

First, we define a *product* pPDA $\Delta \times \mathcal{R} = (Q \times R, \Gamma, \delta', Prob')$ whose transitions are defined as follows: $(p, s)X \xrightarrow{x} (q, t)\alpha$ if and only if $pX \xrightarrow{x} q\alpha$ and $s \xrightarrow{pX} t$. We show that the ω-regular model-checking problem for Δ and $\mathcal{R}$ reduces to the reachability problem for $\mathbf{X}_{\Delta \times \mathcal{R}}$.

Given $s \in R$ and a path v in $M_{\Delta \times \mathcal{R}}$, we say that s occurs in v if $v(i) = (q, s)\alpha$ for some $i \geq 0$, q and α. Given a run w of $M_{\Delta \times \mathcal{R}}$, we denote $Inf_{\mathcal{R}}(w)$ the set of all states of $\mathcal{R}$ that occur in all w_i for $i \geq 0$ (in other words, $Inf_{\mathcal{R}}(w)$ consists of all states of $\mathcal{R}$ that occur infinitely many times in w).

Given $C \in \text{BSCC}_{\Delta \times \mathcal{R}}$, we denote O_C the set of all $s \in R$ that satisfy the following condition: There is $(p, t)X \in C$ and a path $v \in FPath[M_{\Delta \times \mathcal{R}}]((p, t)X)$ such that $head(last(v)) = (p, t)X$ and s occurs in v. The following lemma is proved in Section 4.6.

Lemma 4.3.12. *Let C be a BSCC of $\mathbf{X}_{\Delta \times \mathcal{R}}$ reachable from $(q_0, s_I)Z_0$. For almost all $w \in Run((q_0, s_I)Z_0, C)$ holds $Inf_{\mathcal{R}}(w) = O_C$.*

We say that $C \in \text{BSCC}_{\Delta\times\mathcal{R}}$ is *accepting* if there is $(A_i, B_i) \in F$ such that $O_C \cap A_i \neq \emptyset$ and $O_C \cap B_i = \emptyset$.

Lemma 4.3.13. *The probability $\mathcal{P}(q_0 Z_0, \mathcal{R})$ is equal to the probability of reaching an accepting BSCC of $\mathbf{X}_{\Delta\times\mathcal{R}}$ from $(q_0, s_I)Z_0$.*

Proof. Let Θ be a function that assigns to every configuration $(p, s)\alpha$ of $\Delta \times \mathcal{R}$ the configuration $p\alpha$. It is easy to verify that Θ is a quotient of $M_{\Delta\times\mathcal{R}}^{(q_0,s_I)Z_0}$ onto $M_\Delta^{q_0 Z_0}$. Let us denote $Accept$ the set of all $w \in Run((q_0, s_I)Z_0)$ for which there is $(A_i, B_i) \in F$ satisfying $Inf_{\mathcal{R}}(w) \cap A_i \neq \emptyset$ and $Inf_{\mathcal{R}}(w) \cap B_i = \emptyset$. It is easy to verify that for all $w \in Run((q_0, s_I)Z_0)$, we have $w \in Accept$ if and only if $\Theta(w)$ is accepted by $\mathcal{R}$. It follows from Proposition 2.2.5 that $\Theta(Accept) = Run(q_0 Z_0, \mathcal{R})$, and hence, $\mathcal{P}(Accept) = \mathcal{P}(q_0 Z_0, \mathcal{R})$.

Given $C \in \text{BSCC}_{\Delta\times\mathcal{R}}$, we denote Ω_C the of all runs $w \in Run((q_0, s_I)Z_0, C)$ such that $Inf_{\mathcal{R}}(w) = O_C$. Let $Bot = \biguplus_{C\in\text{BSCC}_{\Delta\times\mathcal{R}}} \Omega_C$. By Proposition 4.3.10 and Lemma 4.3.12,

$$\begin{aligned}\mathcal{P}(Bot) &= \sum_{C\in\text{BSCC}_{\Delta\times\mathcal{R}}} \mathcal{P}(\Omega_C \mid Run((q_0, s_I)Z_0, C)) \cdot \mathcal{P}(Run((q_0, s_I)Z_0, C)) \\ &= \sum_{C\in\text{BSCC}_{\Delta\times\mathcal{R}}} \mathcal{P}(Run((q_0, s_I)Z_0, C)) \;=\; 1\end{aligned}$$

Here we used the fact that all runs of $Run((q_0, s_I)Z_0)$ are clean. Moreover, by definition, for each $w \in Bot$, we have $w \in Accept$ if and only if the footprint $fp(w)$ enters an accepting BSCC of $\mathbf{X}_{\Delta\times\mathcal{R}}$. It follows that the probability $\mathcal{P}(Accept)$, and hence also $\mathcal{P}(q_0 Z_0, \mathcal{R})$, is equal to the probability of reaching an accepting BSCC of $\mathbf{X}_{\Delta\times\mathcal{R}}$ from $(q_0, s_I)Z_0$. □

It follows that the qualitative and quantitative ω-regular model-checking problem can be solved using the following algorithm:

1. Compute $\Delta \times \mathcal{R}$ and the set $\mathbf{S}_{\Delta\times\mathcal{R}}$;
2. Compute the underlying transition system of $\mathbf{X}_{\Delta\times\mathcal{R}}$;
3. Compute the set of all accepting BSCCs of $\mathbf{X}_{\Delta\times\mathcal{R}}$;
4. Qualitative ω-regular model-checking: Decide whether $\mathcal{P}(q_0 Z_0, \mathcal{R}) = 1$ (or $\mathcal{P}(q_0 Z_0, \mathcal{R}) = 0$) by deciding whether all (no) BSCCs of $\mathbf{X}_{\Delta\times\mathcal{R}}$ reachable from $(q_0, s_I)Z_0$ are accepting.
5. Quantitative model-checking: Decide whether $\mathcal{P}(q_0 Z_0, \mathcal{R}) \sim \varrho$ by deciding whether $\mathcal{P}((q_0, s_I)Z_0 \hookrightarrow^* \mathcal{U}) \sim \varrho$ where $\mathcal{U}$ is the union of all accepting BSCCs of $\mathbf{X}_{\Delta\times\mathcal{R}}$.

Let us analyze the complexity of the above algorithm.

1. By Lemma 4.3.8, the set $\mathbf{S}_{\Delta\times\mathcal{R}}$ can be computed in space polynomial in $|\Delta \times \mathcal{R}|$. However, we can do better as follows: Let us assume that $\mathbf{S}_\Delta$ has already been computed. Then observe that the set $\mathbf{S}_{\Delta\times\mathcal{R}}$ can be computed from $\mathbf{S}_\Delta$ in polynomial time. Indeed, it is easy to see that $[(p,t)X\uparrow] = [pX\uparrow]$, and hence that $(p,t)X \in \mathbf{S}_{\Delta\times\mathcal{R}}$ if and only if $pX \in \mathbf{S}_\Delta$.

2. By Lemma 4.3.8, the underlying transition system of $\mathbf{X}_{\Delta\times\mathcal{R}}$ can be computed in time polynomial in $|\Delta \times \mathcal{R}|$, because $\mathbf{S}_{\Delta\times\mathcal{R}}$ has already been computed.

3. The set of all BSCCs of $\mathbf{X}_{\Delta\times\mathcal{R}}$ can be computed in time polynomial in $|\Delta\times\mathcal{R}|$ using standard algorithms for computing bottom strongly connected components of finite directed graphs. Now for each BSCC C the set O_C can be computed using standard methods for model-checking (non-probabilistic) PDA in time polynomial in $|\Delta| \cdot |\mathcal{R}|$ (see, e.g., [14]). Hence, the step 3. can be performed in time polynomial in $|\Delta| \cdot |\mathcal{R}|$.

4., 5. The qualitative problem can be solved in time linear in $|\Delta \times \mathcal{R}|$ by using a standard algorithm for the reachability in finite directed graphs. The quantitative case can be solved in space polynomial in $|\Delta \times \mathcal{R}|$ due to Lemma 4.3.9.

We have proved the following result.

Theorem 4.3.14. *Given $\mathbf{S}_\Delta$, the* qualitative *ω-regular model-checking problem is decidable in time polynomial in $|\Delta| \cdot |\mathcal{R}|$.*

Applying Lemma 4.3.8 and Proposition 4.1.12 (for pBPA), we obtain

Theorem 4.3.15. *For pPDA and deterministic Rabin automata, the* qualitative *model-checking problem is in **PSPACE** in the size of pPDA, and in **P** in the size of Rabin automata. Moreover, for pBPA the problem is in **P**.*

Theorem 4.3.16. *For pPDA and deterministic Rabin automata, the* quantitative *model-checking problem is in **PSPACE**.*

4.3.3 Non-deterministic Büchi Automata and pPDA

It was shown in [47] that every non-deterministic Büchi automaton can effectively be translated to a deterministic Rabin automaton with only a singly exponential blowup in its size. More precisely

Proposition 4.3.17 ([47])**.** *Given a Büchi automaton $\mathcal{B}$ with a set of states B, there is an effectively computable deterministic Rabin automaton $\mathcal{R} = (R, \Sigma, \rho, s_I, F)$ such that $\mathcal{L}(\mathcal{R}) = \mathcal{L}(\mathcal{B})$, $|R| = 2^{\mathcal{O}(|B|\log|B|)}$ and $|F| = \mathcal{O}(|B|^2)$.*

Proposition 4.3.17 allows us to derive the following corollaries of Theorem 4.3.15 and Theorem 4.3.16 for non-deterministic Büchi automata:

Table 4.1: Qualitative ω-regular model-checking

	det. Rabin	non-det. Büchi
pBPA	**P**	**EXPTIME**-complete, **P** in pBPA, **EXPTIME** in Büchi
pPDA	**PSPACE**	**PSPACE** in pPDA, **EXPTIME** in Büchi

Table 4.2: Quantitative ω-regular model-checking

	det. Rabin	non-det. Büchi
pBPA	**PSPACE**	**PSPACE** in pBPA, **EXPSPACE** in Büchi
pPDA	**PSPACE**	**PSPACE** in pPDA, **EXPSPACE** in Büchi

Theorem 4.3.18. *For pPDA and non-deterministic Büchi automata, the* qualitative *ω-regular model-checking problem is in **PSPACE** in the size of pPDA, and in **EXPTIME** in the size of Büchi automata. Moreover, for pBPA the problem is in **P** in the size of pBPA.*

Theorem 4.3.19. *For pPDA and non-deterministic Büchi automata, the* quantitative *ω-regular model-checking problem is in **PSPACE** in the size of pPDA, and in **EXPSPACE** in the size of Büchi automata.*

Remark 4.3.20. *Results of both Theorem 4.3.18 and Theorem 4.3.19 were proved for the first time in [31] (for RMCs) using a slightly more optimal procedure. The procedure of [31] works with deterministic automata of size in $2^{\mathcal{O}(|\mathcal{B}|)}$ (instead of $2^{\mathcal{O}(|\mathcal{B}| \log |\mathcal{B}|)}$), and hence the overall complexity does not involve the logarithmic factor in the exponent. However, the procedure is much more involved, and thus we decided not to present it here in detail. An interested reader can consult [31].*

There is also a matching lower bound for the qualitative ω-regular model-checking problem proved in [31].

Theorem 4.3.21 ([31])**.** *For pBPA, the* qualitative *ω-regular model-checking problem is **EXPTIME**-hard.*

All results concerning the qualitative and quantitative ω-regular model-checking are summarized in Table 4.1 and Table 4.2, respectively.

4.4 Regularity Issues

In the previous section we presented a solution of the problem, whether a given configuration $p\alpha$ of some pPDA Δ satisfies $\mathcal{P}(p\alpha, \mathcal{R}) \sim \varrho$, for a given Rabin automaton $\mathcal{R}$. This problem is usually called the *local* model-checking problem.

However, one may also want to solve the *global* model-checking problem: Compute (a finite representation of) the set of all configurations $p\alpha$ that satisfy the above condition. We show that in the *qualitative* case (i.e., for $\varrho \in \{0,1\}$) this set is effectively regular. On the other hand, we show that, in general, this set is not even context-free.

Let $\Delta = (Q, \Gamma, \delta, Prob)$ be a pPDA. Regular sets of configurations of Δ are formally defined as follows:

Definition 4.4.1. *Given a* deterministic *finite state automaton (DFA)* [3] $\mathcal{A}$ *whose alphabet is* $Q \cup \Gamma$*, we denote* $\mathcal{C}(\mathcal{A})$ *the set of all configurations* $p\alpha \in \mathcal{C}(\Delta)$ *satisfying* $(p\alpha)^R \in \mathcal{L}(\mathcal{A})$ *(here* $(p\alpha)^R$ *denotes the reverse image of the word* $p\alpha$*). A set of configurations* $A \subseteq \mathcal{C}(\Delta)$ *is* regular *if there is a DFA* $\mathcal{A}$ *with the alphabet* $\Gamma \cup Q$*, such that* $\mathcal{C}(\mathcal{A}) = A$*. (We also say that* $\mathcal{A}$ *accepts the set of configurations* A*.)*

4.4.1 Qualitative Properties

In this subsection we show that the set of all configurations satisfying a given qualitative ω-regular property, is effectively regular.

Let us fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$. Let us fix a deterministic Rabin automaton $\mathcal{R} = (\Sigma, R, \rho, s_I, F)$ where $F = \{(A_1, B_1), \ldots, (A_n, B_n)\}$. Similarly to the previous section, we assume that $\Sigma = (Q \times \Gamma) \cup Q$ and consider the valuation $\nu : \mathcal{C}(\Delta) \to \Sigma$ such that $\nu(p\alpha)$ is the head of $p\alpha$ for every $p\alpha \in \mathcal{C}(\Delta)$. Given $0 \leq \varrho \leq 1$ and $\sim \in \{<, >, =, \leq, \geq\}$, we denote $L_{\sim\varrho} = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{R}) \sim \varrho\}$.

We show that for $\varrho \in \{0,1\}$ and $\sim \in \{=, \leq, \geq, <, >\}$, the set of configurations $L_{\sim\varrho}$ is accepted by an effectively computable DFA of size at most exponential in $|\Delta| \cdot |\mathcal{R}|$. Clearly, it suffices to show this claim only for $L_{=1}$ and $L_{=0}$, because DFA are easily complemented by switching accepting and non-accepting states.

In order to simplify our presentation, we introduce some additional notation. The symbol $[s, pX\uparrow]$ denotes the probability that a run of $Clean(pX)$ is accepted by $\mathcal{R}$ where the initial state of $\mathcal{R}$ is changed to s. Furthermore, let us denote $Run(s, pXq, t)$ the set of all runs $w \in Run(pX)$ satisfying the following condition: there is $i \geq 0$ such that $w(i+1) = q\varepsilon$, $w(i) \neq q\varepsilon$, and $\mathcal{R}$ initiated in s moves to t after reading the heads of all configurations in w^i. The symbol $[s, pXq, t]$ denotes the probability $\mathcal{P}(Run(s, pXq, t))$.

We define DFA $\mathcal{A}_1$ and $\mathcal{A}_0$ accepting $L_{=1}$ and $L_{=0}$, respectively. Both $\mathcal{A}_1$ and $\mathcal{A}_0$ have the set $2^{Q \times R}$ as their set of states, and the set $Q \cup \Gamma$ as their alphabet. We define $\mathcal{A}_1 = (2^{Q \times R}, Q \cup \Gamma, \gamma_1, I_1, \{\emptyset\})$ where the initial state $I_1 \subseteq Q \times R$ consists

[3] A DFA is a tuple $\mathcal{A} = (K, \Sigma, \gamma, q_{init}, Acc)$ where K is a finite set of states, Σ is a finite alphabet, $\gamma : K \times \Sigma \to K$ is a (total) transition function, $q_{init} \in K$ is an initial state, and $Acc \subseteq K$ is a set of accepting states. We denote $|\mathcal{A}| = |\gamma| + |Acc|$ the size of the DFA $\mathcal{A}$. A language $\mathcal{L}(\mathcal{A}) \subseteq \Sigma^*$ of $\mathcal{A}$ is defined as follows: a word $v \in \Sigma^*$ of length k is in $\mathcal{L}(\mathcal{A})$ iff there is a sequence of states $q_1, \ldots, q_{k+1} \in K$ such that $q_1 = q_{init}$, $q_{k+1} \in Acc$, and for $1 \leq i \leq k$ holds $\gamma(q_i, v(i)) = q_{i+1}$ (here $v(i)$ is the i'th letter of v). For more details and basic results on DFA see [37].

of all pairs $(p,s) \in Q \times R$ such that $\mathcal{R}$ initiated in s accepts p^ω, the set $\emptyset \subseteq Q \times R$ is the only accepting state, and the transition function γ_1 is defined as follows:

- For all $A \in 2^{Q \times R}$, $X \in \Gamma$ and all $(p,s) \in Q \times R$, we put $(p,s) \in \gamma_1(A,X)$ if and only if $[s, pX\uparrow] + \sum_{(q,t)\in A}[s, pXq, t] = 1$.
- For all $A \in 2^{Q \times R}$ and all $q \in Q$, if $(q, s_I) \in A$, then we put $\gamma_1(A,q) = \emptyset$, else we put $\gamma_1(A,q) = Q \times R$ (note that the set $Q \times R$ was chosen arbitrarily; any non-empty set would do).

Intuitively, the automaton $\mathcal{A}_1$ reads the stack of Δ "bottom-up" and remembers (in its states) all tuples $(p,s) \in Q \times R$ that satisfy the following condition: if α is the already read portion of the stack, then almost all runs of $Run(p\alpha)$ are accepted by $\mathcal{R}$ initiated in s.

The automaton $\mathcal{A}_0$ is defined as follows: Let $\mathcal{A}_0 = (2^{Q\times R}, \Gamma \cup Q, \gamma_0, I_0, \{\emptyset\})$ where the initial state $I_0 \subseteq Q \times R$ consists of all pairs $(p,s) \in Q \times R$ such that $\mathcal{R}$ initiated in s does *not* accept p^ω, and the transition function γ_0 is defined as follows:

- For all $A \in 2^{Q \times R}$, $X \in \Gamma$ and all $(p,s) \in Q \times R$, we put $(p,s) \in \gamma_0(A,X)$ if and only if $[s, pX\uparrow] + \sum_{(q,t)\in(Q\times\Gamma)\setminus A}[s, pXq, t] = 0$.
- For all $A \in 2^{Q \times R}$ and all $q \in Q$, if $(q, s_I) \in A$, then we put $\gamma_0(A,q) = \emptyset$, else we put $\gamma_0(A,q) = Q \times R$.

The following lemma is proved in Section 4.6.3.

Lemma 4.4.2. *$\mathcal{C}(\mathcal{A}_1) = L_{=1}$ and $\mathcal{C}(\mathcal{A}_0) = L_{=0}$.*

Now we show that the automata $\mathcal{A}_1$ and $\mathcal{A}_0$ can be computed in exponential time. We start with the following auxiliary lemma.

Lemma 4.4.3. *Given $\mathbf{S}_\Delta$, $B \subseteq Q \times R$, $X \in \Gamma$, $(p,s) \in Q \times R$, and $\varrho \in \{0,1\}$, the problem whether $[s, pX\uparrow] + \sum_{(q,t)\in B}[s, pXq, t] = \varrho$ is decidable in time polynomial in $|\Delta| \cdot |\mathcal{R}|$.*

Proof. We reduce this problem to the qualitative ω-regular model-checking problem by modifying $\mathcal{R}$ as follows:

1. Delete (from $\mathcal{R}$) all transitions of the form $t \xrightarrow{q} t'$ where $q \in Q$;
2. add (to $\mathcal{R}$) fresh new states s_a and s_r (together with self loops $s_a \xrightarrow{\zeta} s_a$ and $s_r \xrightarrow{\zeta} s_r$ for every $\zeta \in (Q \times \Gamma) \cup Q$);
3. for every $t \in R\setminus\{s_a, s_r\}$ and $q \in Q$, add a transition either of the form $t \xrightarrow{q} s_a$ or of the form $t \xrightarrow{q} s_r$ depending on whether $(q,t) \in B$ or not, respectively;
4. add the pair $(\{s_a\}, \emptyset)$ to the Rabin condition F;
5. change the initial state to s.

Let us denote $\mathcal{R}'$ the resulting Rabin automaton. It is easy to verify that $\mathcal{P}(pX, \mathcal{R}') = [s, pX\uparrow] + \sum_{(q,t)\in B}[s, pXq, t]$. Moreover, $\mathcal{R}'$ is computable from $\mathcal{R}$ in polynomial time. The rest follows from Theorem 4.3.14. □

Lemma 4.4.4. *Given $\mathbf{S}_\Delta$, the automata $\mathcal{A}_{=1}$ and $\mathcal{A}_{=0}$ are computable in time $2^{\mathcal{O}(|Q|\cdot|R|)} \cdot (|\Delta| \cdot |\mathcal{R}|)^{\mathcal{O}(1)}$, and in polynomial space.*

Proof. Let us fix $\varrho \in \{0, 1\}$. The set I_ϱ can easily be computed, using Lemma 4.3.11, in time polynomial in $|\Delta| \cdot |\mathcal{R}|$. The transition function γ_ϱ can be computed as follows:

1. For each $A \subseteq Q \times R$, $X \in \Gamma$ and every $(p, s) \in Q \times R$, decide whether $(p, s) \in \gamma_\varrho(A, X)$ using the procedure of Lemma 4.4.3. (This step involves $|Q \times R| \cdot 2^{|Q\times R|} \cdot |\Gamma|$ invocations of the procedure of Lemma 4.4.3 where each invocation runs in time polynomial in $|\Delta| \cdot |\mathcal{R}|$. Moreover, all these invocations can be performed in the same space, i.e., the space can be reused).

2. For each $A \subseteq Q \times R$ and every $q \in Q$, decide whether $\gamma_\varrho(A, q) = \emptyset$ by deciding whether $(q, s_I) \in A$. This is done in time $(2^{|Q\times R|}\cdot|Q|\cdot|Q\times R|)^{\mathcal{O}(1)}$ and in polynomial space.

Hence, the overall time complexity is $2^{\mathcal{O}(|Q|\cdot|R|)} \cdot (|\Delta| \cdot |\mathcal{R}|)^{\mathcal{O}(1)}$ and the space complexity is polynomial. □

Theorem 4.4.5. *Given $\mathbf{S}_\Delta$, a DFA accepting $L_{\sim\varrho}$ for some $\sim \in \{<, >, =, \leq, \geq\}$ and $\varrho \in \{0, 1\}$, is computable in time $2^{\mathcal{O}(|Q|\cdot|R|)} \cdot (|\Delta| \cdot |\mathcal{R}|)^{\mathcal{O}(1)}$, and in polynomial space. The number of states of the resulting automaton equals $2^{|Q|\cdot|R|}$.*

The following theorem is a direct consequence of Theorem 4.4.5 and Corollary 4.3.8.

Theorem 4.4.6. *Let $\varrho \in \{0, 1\}$ and $\sim \in \{<, >, =, \leq, \geq\}$.*

- *For a deterministic Rabin automaton $\mathcal{R}$, a DFA accepting $L_{\sim\varrho} = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{R}) \sim \varrho\}$ is computable in polynomial space and its size is at most exponential in $|\Delta| \cdot |\mathcal{R}|$.*
- *For a non-deterministic Büchi automaton $\mathcal{B}$, a DFA accepting $L_{\sim\varrho} = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{B}) \sim \varrho\}$ is computable in time exponential in $|\Delta|$ and doubly exponential in $|\mathcal{B}|$.*

4.4.2 Quantitative Properties and pPDA

In this subsection we show that for $0 < \varrho < 1$, the set $L_{\sim\varrho} = \{q\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(q\alpha, \mathcal{R}) \sim \varrho\}$ does not have to be regular even if the Rabin automaton $\mathcal{R}$ expresses the *termination* property (i.e., $\mathcal{R}$ accepts exactly those runs that eventually reach a configuration of the form $t\varepsilon$ for a given control state t).

Let us assume that Δ is a pPDA, which has three stack symbols A, B, and $\#$, and the following transition rules (we assume that transition probabilities are

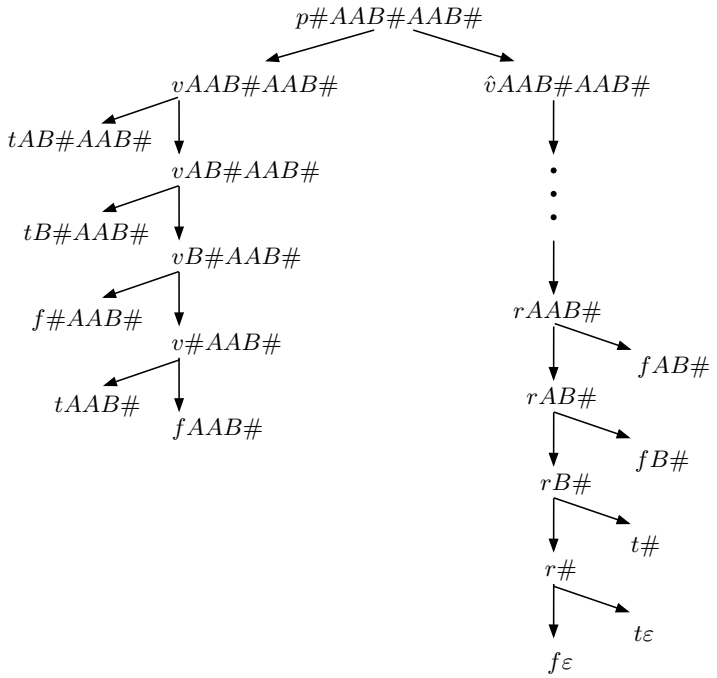

Figure 4.2: The Markov chain M_Δ (only a part of the state space reachable from the configuration $p\#AAB\#AAB\#$ is drawn). Note that transitions in the left-hand branch are determined by the word, which occurs between the first and the second occurrence of the symbol $\#$, while transitions in the right-hand branch are determined by the word between the second and the third occurrence of $\#$.

distributed uniformly, and thus we do not write these probabilities explicitly; the symbol "|" separates alternatives)[4]:

$$\begin{array}{lllll} pX \to v\varepsilon \mid \hat{v}\varepsilon, & vA \to t\varepsilon \mid v\varepsilon, & \hat{v}A \to \hat{v}\varepsilon, & rA \to f\varepsilon \mid r\varepsilon, & tX \to t\varepsilon, \\ & vB \to f\varepsilon \mid v\varepsilon, & \hat{v}B \to \hat{v}\varepsilon, & rB \to t\varepsilon \mid r\varepsilon, & fX \to f\varepsilon, \\ & v\# \to t\varepsilon \mid f\varepsilon, & \hat{v}\# \to r\varepsilon, & r\# \to t\varepsilon \mid f\varepsilon & \end{array}$$

Here X ranges over $\{A, B, \#\}$.

We show that for arbitrary $\sim \in \{<, >, =, \leq, \geq\}$ the set

$$L_{\sim\frac{1}{2}} = \{q\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(q\alpha \to^* t\varepsilon) \sim \frac{1}{2}\}$$

is *not context-free* (see [37]). Later we show how to modify the pPDA Δ to allow other probabilities $\varrho \in (0, 1)$ instead of $\frac{1}{2}$.

To reveal the subtlety of the construction, let us evaluate the probability $\mathcal{P}(p\#AAB\#AAB\# \to^* t\varepsilon)$. By inspecting the above rules, and Figure 4.2 where the relevant part of the Markov chain M_Δ is drawn, one can easily confirm that the

[4]Hence, here we have $vA \stackrel{1/2}{\to} t\varepsilon$, $vA \stackrel{1/2}{\to} v\varepsilon$, etc.

probability is equal to

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

which can be written in binary as follows: $\frac{1}{2}(0.1101{+}0.0011)$. The binary numbers 0.1101 and 0.0011 are "complementary" and their sum is equal to 1. On the other hand, this complementarity breaks down for $p\#\alpha\#\beta\#$ (here $\alpha,\beta\in\{A,B\}^*$) if and only if $\alpha\neq\beta$, in which case the probability is different from $\frac{1}{2}$.

We show that there is even deeper connection between the probability $\mathcal{P}(p\#\alpha\#\beta\#\rightarrow^* t\varepsilon)$ and the words $\alpha,\beta\in\{A,B\}^*$. In order to formalize this connection we define an "alphabetical" ordering on words of $\{A,B\}^*$ as follows: Given two words $\alpha,\beta\in\{A,B\}^*$, we put $\beta\leq\alpha$ if and only if either β is a prefix of α, or there is γ such that $\beta=\gamma A\delta_1$ and $\alpha=\gamma B\delta_2$ for some $\delta_1,\delta_2\in\{A,B\}^*$. We show that for arbitrary $\sim\in\{=,<,>,\leq,\geq\}$ we have $\mathcal{P}(p\#\alpha\#\beta\#\rightarrow^* t\varepsilon)\sim\frac{1}{2}$ if and only if $\beta B\sim\alpha B$. First, we prepare some tools.

Definition 4.4.7. *Given $\alpha\in\{A,B\}^+$ where $\alpha=X_1\cdots X_k$, we define $Num^+(\alpha)=\sum_{i=1}^k\xi_i\cdot\frac{1}{2^i}+\frac{1}{2^{k+1}}$ and $Num^-(\alpha)=\sum_{i=1}^k\xi'_i\cdot\frac{1}{2^i}+\frac{1}{2^{k+1}}$ where*

$$\xi_i=\begin{cases}1 & \text{if } X_i=A;\\ 0 & \text{if } X_i=B.\end{cases}\qquad \xi'_i=\begin{cases}1 & \text{if } X_i=B;\\ 0 & \text{if } X_i=A.\end{cases}$$

We also define $Num^+(\varepsilon)=Num^-(\varepsilon)=\frac{1}{2}$.

Lemma 4.4.8. *Given $\alpha,\beta\in\{A,B\}^*$,*

1. *$\alpha=\beta$ if and only if $Num^+(\alpha)+Num^-(\beta)=1$;*
2. *$\beta B<\alpha B$ if and only if $Num^+(\alpha)+Num^-(\beta)<1$;*
3. *$\beta B>\alpha B$ if and only if $Num^+(\alpha)+Num^-(\beta)>1$.*

Proof. We take the advantage of the binary representation, and consider the binary numbers $Num^+(\alpha)=0.\xi_1\cdots\xi_k1$ and $Num^-(\beta)=0.\xi'_1\cdots\xi'_\ell1$. Moreover, given a number $\xi\in\{0,1\}$, we denote $\bar{\xi}=1-\xi$.

1. If $\alpha=\beta$, then $k=\ell$ and $\xi_i\neq\xi'_i$ for all $1\leq i\leq k$. Using the algorithm for adding binary numbers we find out that $Num^+(\alpha)+Num^-(\beta)=1$.

2. Assume that $\beta B<\alpha B$. First suppose that βB is a strict prefix of αB. Then βB is a prefix of α, and thus $Num^+(\alpha)=0.\xi_1\cdots\xi_\ell0\xi_{\ell+2}\cdots\xi_k1$ and $Num^-(\beta)=0.\bar{\xi_1}\cdots\bar{\xi_\ell}1$. However, then $Num^+(\alpha)+Num^-(\beta)=0.1\cdots1\xi_{\ell+2}\cdots\xi_k1<1$.

   Now let us assume that $\beta=\alpha A\delta_1$ for some $\delta_1\in\{A,B\}^*$. Then $Num^+(\alpha)=0.\xi_1\cdots\xi_k1$ and $Num^-(\beta)=0.\bar{\xi_1}\cdots\bar{\xi_k}0\xi'_{k+2}\cdots\xi'_\ell1$. However, then $Num^+(\alpha)+Num^-(\beta)=0.1\cdots1\xi'_{k+2}\cdots\xi'_\ell1<1$.

Now let us assume that $\beta = \gamma A \delta_1$ and $\alpha = \gamma B \delta_2$ for some $\delta_1, \delta_2 \in \{A, B\}^*$. Then $Num^+(\alpha) = 0.\xi_1 \cdots \xi_j 0 \xi_{j+2} \cdots \xi_k 1$ and $Num^-(\beta) = 0.\bar{\xi}_1 \cdots \bar{\xi}_j 0 \xi'_{j+2} \cdots \xi'_\ell 1$ where $j = |\gamma|$, and thus $Num^+(\alpha) + Num^-(\beta) = 0.z_1 \cdots z_j c_1 \cdots c_r$ where $z_1 = \cdots = z_j = 1$. It follows that $Num^+(\alpha) + Num^-(\beta) < 1$.

3. Assume that $\beta B > \alpha B$. First suppose that αB is a strict prefix of βB. Then αB is a prefix of β, and thus $Num^+(\alpha) = 0.\xi_1 \cdots \xi_k 1$ and $Num^-(\beta) = 0.\bar{\xi}_1 \cdots \bar{\xi}_k 1 \xi'_{k+2} \cdots \xi'_\ell 1$. However, then $Num^+(\alpha) + Num^-(\beta) = 1.0 \cdots 0 \xi'_{k+2} \cdots \xi'_\ell 1 > 1$.

   Now let us assume that $\alpha = \beta A \delta_1$ for some $\delta_1 \in \{A, B\}^*$. Then $Num^+(\alpha) = 0.\xi_1 \cdots \xi_\ell 1 \xi_{\ell+2} \cdots \xi_k 1$ and $Num^-(\beta) = 0.\bar{\xi}_1 \cdots \bar{\xi}_\ell 1$. However, then $Num^+(\alpha) + Num^-(\beta) = 1.0 \cdots 0 \xi_{\ell+2} \cdots \xi_k 1 > 1$.

   Now let us assume that $\beta = \gamma B \delta_1$ and $\alpha = \gamma A \delta_2$ for some $\delta_1, \delta_2 \in \{A, B\}^*$. Then $Num^+(\alpha) = 0.\xi_1 \cdots \xi_j 1 \xi_{j+2} \cdots \xi_k 1$ and $Num^-(\beta) = 0.\bar{\xi}_1 \cdots \bar{\xi}_j 1 \xi'_{j+2} \cdots \xi'_\ell 1$ where $j = |\gamma|$, and thus $Num^+(\alpha) + Num^-(\beta) = 1.z_1 \cdots z_j c_1 \cdots c_r$ where $z_1 = \cdots = z_j = 0$ and at least one of $c_1, \ldots, c_r$ is not 0. It follows that $Num^+(\alpha) + Num^-(\beta) > 1$.

□

Lemma 4.4.9. *Given $\alpha, \beta \in \{A, B\}^*$ and $\sim \in \{=, <, >, \leq, \geq\}$, we have $\mathcal{P}(p\#\alpha\#\beta\# \rightarrow^* t\varepsilon) \sim \frac{1}{2}$ if and only if $\beta B \sim \alpha B$.*

Proof. We show that $\mathcal{P}(p\#\alpha\#\beta\# \rightarrow^* t\varepsilon)$ equals $\frac{1}{2} \cdot (Num^+(\alpha) + Num^-(\beta))$. The rest follows from Lemma 4.4.8.

We show that $\mathcal{P}(v\alpha\#\beta\# \rightarrow^* t\varepsilon) = Num^+(\alpha)$ by induction on $|\alpha|$. For $|\alpha| = 0$, this follows immediately from the definition of Δ. Now let $\alpha = A\alpha'$. We have $vA\alpha'\#\beta\# \xrightarrow{\frac{1}{2}} t\alpha'\#\beta\#$ where $\mathcal{P}(t\alpha'\#\beta\# \rightarrow^* t\varepsilon) = 1$, and $vA\alpha'\#\beta\# \xrightarrow{\frac{1}{2}} v\alpha'\#\beta\#$. By induction, $\mathcal{P}(vA\alpha'\#\beta\# \rightarrow^* t\varepsilon) = \frac{1}{2} + \frac{1}{2} \cdot Num^+(\alpha') = Num^+(\alpha)$. The case $\alpha = B\alpha'$ is treated similarly.

One can show, using similar arguments as above, that $\mathcal{P}(r\beta\# \rightarrow^* t\varepsilon) = Num^-(\beta)$. Observe also that $\mathcal{P}(\hat{v}\alpha\#\beta\# \rightarrow^* r\beta\#) = 1$. Hence, by Lemma 2.2.3, $\mathcal{P}(\hat{v}\alpha\#\beta\# \rightarrow^* t\varepsilon) = Num^-(\beta)$. Finally, by Lemma 2.2.3 and definition of Δ, $\mathcal{P}(p\#\alpha\#\beta\# \rightarrow^* t\varepsilon) = \frac{1}{2} \cdot (\mathcal{P}(v\alpha\#\beta\# \rightarrow^* t\varepsilon) + \mathcal{P}(\hat{v}\alpha\#\beta\# \rightarrow^* t\varepsilon)) = \frac{1}{2} \cdot (Num^+(\alpha) + Num^-(\beta))$. □

Let us denote $L = \{p\#\alpha\#\beta\# \mid \alpha, \beta \in \{A, B\}^*\}$. By Lemma 4.4.9 we have $L_{\sim \frac{1}{2}} \cap L = \{p\#\alpha\#\beta\# \mid \alpha, \beta \in \{A, B\}^*, \beta B \sim \alpha B\}$, which is not context-free (this can easily be verified using Pumping Lemma for context-free languages, see [37]). Because L is regular and the class of context-free languages is closed under intersection with regular languages, we obtain the following result:

Proposition 4.4.10. *The set $L_{\sim \frac{1}{2}}$ is not context-free.*

Finally, we show that given $0 < \varrho < 1$, the pPDA Δ can be modified in such a way that the set $L_{\sim\varrho} = \{q\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(q\alpha \rightarrow^* t\varepsilon) \sim \varrho\}$ is not context-free. If $\varrho < \frac{1}{2}$, then it suffices to modify the transitions outgoing from $p\#$ as follows: $p\# \xrightarrow{\varrho} v\varepsilon$, $p\# \xrightarrow{\varrho} \hat{v}\varepsilon$, and $p\# \xrightarrow{1-2\varrho} f\varepsilon$. An inspection of the proof of Lemma 4.4.9 reveals that for all $\alpha, \beta \in \{A, B\}^*$ we have $\mathcal{P}(p\#\alpha\#\beta\# \rightarrow^* t\varepsilon) \sim \varrho$ if and only if $\beta B \sim \alpha B$. If $\varrho > \frac{1}{2}$, then the same effect is achieved by putting $p\# \xrightarrow{1-\varrho} v\varepsilon$, $p\# \xrightarrow{1-\varrho} \hat{v}\varepsilon$, and $p\# \xrightarrow{2\varrho-1} t\varepsilon$. In both cases $L_{\sim\varrho} \cap L = \{p\#\alpha\#\beta\# \mid \alpha, \beta \in \{A, B\}^*, \beta B \sim \alpha B\}$, which is not context-free, and thus $L_{\sim\varrho}$ is not context-free either. We have proved the following theorem:

Theorem 4.4.11. *For arbitrary $\sim \in \{<, >, =, \leq, \geq\}$ and $0 < \varrho < 1$, there is a pPDA Δ and a control state t of Δ, such that the set of all configurations $q\alpha \in \mathcal{C}(\Delta)$ that satisfy $\mathcal{P}(q\alpha \rightarrow^* t\varepsilon) \sim \varrho$, is* not context-free.

4.4.3 Quantitative Properties and pBPA

We prove an analogy of Theorem 4.4.11 for pBPA. We show that for each $\varrho \in (0, 1)$ and $\sim \in \{=, <, >, \leq, \geq\}$, there exist a pBPA Δ, an LTL formula φ, and a simple valuation ν, such that the set $L_{\sim\varrho} = \{\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(\alpha, \varphi, \nu) \sim \varrho\}$ is not context-free. Note that this result is valid even if we use non-deterministic Büchi automata instead of LTL formulae, because each LTL formula can be translated to an equivalent non-deterministic Büchi automaton (see, e.g., [51, 19]). In this section we use LTL instead of automata to simplify our presentation.

Let us assume that Δ is a pBPA with the set of stack symbols $\Gamma = \{P, V, \hat{V}, A, B, A', B', T, F, T', F', \#, \#'\}$ and the following transition rules (transition probabilities are again distributed uniformly):

$$\begin{array}{lllll} P \rightarrow V \mid \hat{V}, & A \rightarrow T \mid \varepsilon, & A' \rightarrow F' \mid \varepsilon, & T \rightarrow \varepsilon, & T' \rightarrow \varepsilon, \\ V \rightarrow \varepsilon, & B \rightarrow F \mid \varepsilon, & B' \rightarrow T' \mid \varepsilon, & F \rightarrow \varepsilon, & F' \rightarrow \varepsilon, \\ \hat{V} \rightarrow \varepsilon, & \# \rightarrow T \mid F, & \#' \rightarrow T' \mid F', & & \end{array}$$

Let $\varphi \equiv ((\neg\hat{V} \wedge \neg F)\,\mathcal{U}\; T) \vee ((\neg V \wedge \neg F')\,\mathcal{U}\; T')$ be an LTL formula. Let us fix a valuation ν which assigns to each $X \in \Gamma$ the set of all configurations with the head X.

We show that the set $L_{\sim\frac{1}{2}} = \{\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(\alpha, \varphi) \sim \frac{1}{2}\}$ is not context-free for arbitrary $\sim \in \{<, >, =, \leq, \geq\}$. The trick is essentially the same as in the case of pPDA. However, because there is only one control state in Δ, we need the power of LTL (ω-regular properties) instead of the simple termination property.

Given a word $\beta \in \{A, B\}^*$, we denote $\bar{\beta}$ the word of $\{A', B'\}^*$ obtained from β by adding a prime to each letter of β (e.g., if $\beta = ABA$, then $\bar{\beta} = A'B'A'$).

Lemma 4.4.12. *Given $\alpha, \beta \in \{A, B\}^*$ and $\sim \in \{=, <, >, \leq, \geq\}$, we have $\mathcal{P}(P\alpha\#\bar{\beta}\#', \varphi) \sim \frac{1}{2}$ if and only if $\beta B \sim \alpha B$.*

Proof. We show that $\mathcal{P}(P\alpha\#\bar{\beta}\#', \varphi)$ is equal to $\frac{1}{2} \cdot (Num^+(\alpha) + Num^-(\beta))$. The rest follows from Lemma 4.4.8.

Let us denote $\psi_1 \equiv (\neg\hat{V} \wedge \neg F)\mathcal{U}\ T$ and $\psi_2 \equiv (\neg V \wedge \neg F')\mathcal{U}\ T'$. We show that $\mathcal{P}(\alpha\#\bar{\beta}\#', \psi_1) = Num^+(\alpha)$ by induction on $|\alpha|$. For $|\alpha| = 0$, this follows immediately from the definition of Δ and ψ_1. Now let $\alpha = A\zeta$. We have $A\zeta\#\bar{\beta}\#' \xrightarrow{\frac{1}{2}} T\zeta\#\bar{\beta}\#'$ and $A\zeta\#\bar{\beta}\#' \xrightarrow{\frac{1}{2}} \zeta\#\bar{\beta}\#'$. Hence, by induction, $\mathcal{P}(A\zeta\#\bar{\beta}\#', \psi_1) = \frac{1}{2} + \frac{1}{2} \cdot \mathcal{P}(\zeta\#\bar{\beta}\#', \psi_1) = \frac{1}{2} + \frac{1}{2} \cdot Num^+(\zeta) = Num^+(\alpha)$. The case $\alpha = B\zeta$ is treated similarly.

Similarly as above $\mathcal{P}(\bar{\beta}\#', \psi_2) = Num^-(\beta)$. Let $\mathcal{W}$ be the set of all paths $u \in FPath(\alpha\#\bar{\beta}\#')$ such that $last(u) = \bar{\beta}\#'$. Let $\mathcal{V}$ be the set of all runs of $Run(\bar{\beta}\#')$ that satisfy ψ_2. Clearly, $\mathcal{W} \odot \mathcal{V}$ is the set of all runs of $FPath(\alpha\#\bar{\beta}\#')$ that satisfy ψ_2. Because $\mathcal{W}$ is prefix-free and $\mathcal{P}(Run(\mathcal{W})) = 1$, we obtain, by Lemma 2.2.3, that $\mathcal{P}(\alpha\#\bar{\beta}\#', \psi_2) = \mathcal{P}(Run(\mathcal{W})) \cdot \mathcal{P}(\bar{\beta}\#', \psi_2) = Num^-(\bar{\beta})$.

By Lemma 2.2.3, $\mathcal{P}(P\alpha\#\bar{\beta}\#', \varphi) = \frac{1}{2} \cdot (\mathcal{P}(\alpha\#\bar{\beta}\#', \psi_1) + \mathcal{P}(\alpha\#\bar{\beta}\#', \psi_2)) = \frac{1}{2} \cdot (Num^+(\alpha) + Num^-(\beta))$. □

Let us denote $L = \{P\alpha\#\bar{\beta}\#' \mid \alpha, \beta \in \{A, B\}^*\}$. It follows from Lemma 4.4.12 that $L_{\sim\frac{1}{2}} \cap L = \{P\alpha\#\bar{\beta}\#' \mid \alpha, \beta \in \{A, B\}^*, \beta B \sim \alpha B\}$, which is not context free. Hence,

Proposition 4.4.13. *$L_{\sim\frac{1}{2}}$ is not context-free.*

Finally, we show that the pBPA Δ and the formula φ can be modified to allow other probabilities $\varrho \in (0, 1)$ instead of $\frac{1}{2}$. The trick is very similar to the one we have already used for pPDA. If $\varrho < \frac{1}{2}$, then we modify the transitions outgoing from P as follows: $P \xrightarrow{\varrho} V$, $P \xrightarrow{\varrho} \hat{V}$, and $P \xrightarrow{1-2\varrho} F$. Then, clearly, for all $\alpha, \beta \in \{A, B\}^*$ holds $\mathcal{P}(P\alpha\#\bar{\beta}\#', \varphi \wedge \neg\mathcal{X}F) \sim \varrho$ if and only if $\beta B \sim \alpha B$. If $\varrho > \frac{1}{2}$, then the same result is achieved by defining $P \xrightarrow{1-\varrho} V$, $P \xrightarrow{1-\varrho} \hat{V}$, and $P \xrightarrow{2\varrho-1} T$, and considering $\varphi \vee \mathcal{X}T$ instead of φ. In both cases $L_{\sim\varrho} \cap L = \{P\#\alpha\#\bar{\beta}\#' \mid \alpha, \beta \in \{A, B\}^*, \beta B \sim \alpha B\}$, which is not context-free, and thus $L_{\sim\varrho}$ is not context-free either.

Now because each LTL formula can effectively be translated to an equivalent non-deterministic Büchi automaton, we obtain the following analogy of Theorem 4.4.11.

Theorem 4.4.14. *For arbitrary $\sim \in \{<, >, =, \leq, \geq\}$ and $0 < \varrho < 1$, there is a pBPA Δ and an LTL formula φ (or a Büchi automaton $\mathcal{B}$), such that the set of all configurations $\alpha \in \mathcal{C}(\Delta)$ that satisfy $\mathcal{P}(\alpha, \varphi) \sim \varrho$ (or $\mathcal{P}(\alpha, \mathcal{B}) \sim \varrho$, resp.), is* not context-free.

4.5 Regular Valuations

In this section we extend some results from previous sections to deal with regular sets of configurations (see Definition 4.4.1) and regular valuations. We follow the approach of [30, 27] with some modifications.

Let us fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$ and let us fix DFA $\mathcal{A}_1, \ldots, \mathcal{A}_n$ where each $\mathcal{A}_i = (K_i, \Gamma \cup Q, \gamma_i, q_i^0, F_i)$ (i.e., the alphabet of each $\mathcal{A}_i$ consists of control states and stack symbols of Δ). We show that the regular sets $\mathcal{C}(\mathcal{A}_1), \ldots, \mathcal{C}(\mathcal{A}_n)$ can effectively be transformed to simple sets (see Definition 4.1.2) by augmenting the stack alphabet of Δ with vectors of states of the DFA $\mathcal{A}_1, \ldots, \mathcal{A}_n$.

Definition 4.5.1. *We define a pPDA $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n] = (Q, \Gamma', \delta', Prob')$ where Q is the set of control states of Δ, $\Gamma' = \Gamma \times \prod_{i=1}^{n} K_i$ (here $\prod_{i=1}^{n} K_i$ is a shorthand for $K_1 \times \cdots \times K_n$), and transitions of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ are defined as follows:*

1. *$p(X, \vec{s}) \xrightarrow{x} q\varepsilon$ if and only if $pX \xrightarrow{x} q\varepsilon$;*
2. *$p(X, \vec{s}) \xrightarrow{x} q(Y, \vec{s})$ if and only if $pX \xrightarrow{x} qY$;*
3. *$p(X, \vec{s}) \xrightarrow{x} q(Y, \vec{t})(Z, \vec{s})$ if and only if $pX \xrightarrow{x} qYZ$ and $\gamma_i(\vec{s}(i), Z) = \vec{t}(i)$ for all $1 \leq i \leq n$;*
4. *nothing else is a transition of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$.*

Intuitively, the pPDA $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ simulates the pPDA Δ and at the same time simulates (on its stack) the computations of the DFA $\mathcal{A}_1, \ldots, \mathcal{A}_n$. Observe that not every configuration of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ contains a correct simulation of the DFA $\mathcal{A}_1, \ldots, \mathcal{A}_n$. We say that a configuration $p(X_1, \vec{s}_1) \cdots (X_k, \vec{s}_k)$ of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ is *consistent* if $\vec{s}_k = (q_1^0, \ldots, q_n^0)$ and for all $1 < j \leq k$ and all $1 \leq i \leq n$ we have $\vec{s}_{j-1}(i) = \gamma_i(\vec{s}_j(i), X_j)$ (each configuration of the form $p\varepsilon$ is also consistent).

Note that given a consistent configuration $p(X_1, \vec{s}_1) \cdots (X_k, \vec{s}_k)$, the configuration $pX_1 \cdots X_k$ is in $\mathcal{C}(\mathcal{A}_i)$ if and only if $\mathcal{A}_i$ initiated in $\vec{s}_1(i)$ accepts the word $X_1 p$. Hence, the problem whether $pX_1 \cdots X_k$ is in $\mathcal{C}(\mathcal{A}_i)$ can be decided based solely on the head $p(X_1, \vec{s}_1)$. Thus, for each $1 \leq i \leq n$, we denote $\mathcal{S}[\mathcal{A}_i]$ the simple set of configurations of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ determined by the set of heads $\mathcal{H}[\mathcal{A}_i]$ which is defined as follows:

- For $p(X_1, \vec{s}_1) \in Q \times (\Gamma \times \prod_{i=1}^{n} K_i)$, we put $p(X_1, \vec{s}_1) \in \mathcal{H}[\mathcal{A}_i]$ if and only if $\mathcal{A}_i$ initiated in $\vec{s}_1(i)$ accepts the word $X_1 p$;
- For $p \in Q$, we put $p \in \mathcal{H}[\mathcal{A}_i]$ if and only if $p\varepsilon \in \mathcal{C}(\mathcal{A}_i)$.

The following lemma shows that the pPDA $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ faithfully simulates the pPDA Δ, and also connects the simple sets $\mathcal{S}[\mathcal{A}_1], \ldots, \mathcal{S}[\mathcal{A}_n]$ with the regular sets $\mathcal{C}(\mathcal{A}_1), \ldots, \mathcal{C}(\mathcal{A}_n)$.

Lemma 4.5.2. *Let Θ be the function which assigns to each configuration $p(X_1, \vec{s}_1) \cdots (X_k, \vec{s}_k)$ of $\mathcal{C}(\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n])$ the configuration $pX_1 \cdots X_k$. Let $\mathcal{K}$ be the function which assigns to each $p\alpha \in \mathcal{C}(\Delta)$ the unique consistent configuration $\mathcal{K}(p\alpha)$ of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ satisfying $\Theta(\mathcal{K}(p\alpha)) = p\alpha$.*

1. *Given $p\alpha \in \mathcal{C}(\Delta)$, all configurations $q\beta$ reachable from $\mathcal{K}(p\alpha)$ are consistent, and thus for all $1 \leq i \leq n$ we have $q\beta \in \mathcal{S}[\mathcal{A}_i]$ if and only if $\Theta(q\beta) \in \mathcal{C}(\mathcal{A}_i)$.*

2. *For every $p\alpha \in \mathcal{C}(\Delta)$, Θ is an isomorphism of $M^{\mathcal{K}(p\alpha)}_{\Delta[\mathcal{A}_1,\ldots,\mathcal{A}_n]}$ onto $M^{p\alpha}_{\Delta}$.*

Moreover, the pPDA $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ together with the sets $\mathcal{H}[\mathcal{A}_1], \ldots, \mathcal{H}[\mathcal{A}_n]$, are computable (from Δ and $\mathcal{A}_1, \ldots, \mathcal{A}_n$) in time $(|\Delta| \cdot n \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$, and the size of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ is in $\mathcal{O}(|\Delta| \cdot \prod_{i=1}^{n} |K_i|)$.

Proof. 1. and 2. follow easily from definitions. Let us concentrate on the complexity estimate. Computing one transition of Δ requires searching through transition functions of all $\mathcal{A}_1, \ldots, \mathcal{A}_n$. Hence, $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ can be computed in time $(|\Delta| \cdot \prod_{i=1}^{n} |K_i| \cdot \sum_{i=1}^{n} |\mathcal{A}_i|)^{\mathcal{O}(1)}$. Observe that $\sum_{i=1}^{n} |\mathcal{A}_i|$ is in $\mathcal{O}(|Q \cup \Gamma| \cdot \sum_{i=1}^{n} |K_i|) \subseteq \mathcal{O}(|\Delta| \cdot n \cdot \prod_{i=1}^{n} |K_i|)$. It follows that $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ is computable in time $(|\Delta| \cdot n \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$.

Each set $\mathcal{H}[\mathcal{A}_i]$ can be computed in time $(|\Delta| \cdot \prod_{i=1}^{n} |K_i| \cdot \sum_{i=1}^{n} |\mathcal{A}_i|)^{\mathcal{O}(1)} \subseteq (|\Delta| \cdot n \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$. Hence, all sets $\mathcal{H}[\mathcal{A}_1], \ldots, \mathcal{H}[\mathcal{A}_n]$ can be computed in time $\mathcal{O}(n \cdot (|\Delta| \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}) \subseteq (|\Delta| \cdot n \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$. □

We also show how to project regular sets of configurations of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ to regular sets of configurations of Δ. The following lemma is used in Section 5.3.

Lemma 4.5.3. *Let $\mathcal{B}$ be a DFA with a set of states K. There is a DFA $\mathcal{B}'$ such that $\mathcal{C}(\mathcal{B}') = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{K}(p\alpha) \in \mathcal{C}(\mathcal{B})\}$ (here $\mathcal{K}$ is from Lemma 4.5.2). Moreover, $\mathcal{B}'$ is computable (from $\mathcal{B}$, Δ, $\mathcal{A}_1, \ldots, \mathcal{A}_n$) in time $(|K| \cdot |\Delta| \cdot n \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$, and $\mathcal{B}'$ has $|K| \cdot \prod_{i=1}^{n} |K_i|$ states.*

Proof. Let us assume, w.l.o.g, that $\mathcal{B} = (K, (\Gamma \times \prod_{i=1}^{n} K_i) \cup Q, \gamma, q_0, F)$. We define a DFA $\mathcal{B}' = (K \times \prod_{i=1}^{n} K_i, \Gamma \cup Q, \gamma', (q_0, \vec{q}_0), F')$ where $\vec{q}_0 = (q_1^0, \ldots, q_n^0)$, $F' = F \times \prod_{i=1}^{n} K_i$, and the transition function γ' is defined as follows:

- For $X \in \Gamma$ and $(s, \vec{s}) \in K \times \prod_{i=1}^{n} K_i$, we define $\gamma'((s, \vec{s}), X) = (t, \vec{t})$ if and only if $\gamma(s, (X, \vec{s})) = t$ and $\gamma_i(\vec{s}(i), X) = \vec{t}(i)$ for all $1 \leq i \leq n$.
- For $p \in Q$ and $(s, \vec{s}) \in K \times \prod_{i=1}^{n} K_i$, we define $\gamma'((s, \vec{s}), p) = (t, \vec{t})$ if and only if $\gamma(s, p) = t$ and $\gamma_i(\vec{s}(i), p) = \vec{t}(i)$ for all $1 \leq i \leq n$.

It is easy to verify that $\mathcal{C}(\mathcal{B}') = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{K}(p\alpha) \in \mathcal{C}(\mathcal{B})\}$. The DFA $\mathcal{B}'$ can be computed in time $(|\mathcal{B}| \cdot \sum_{i=1}^{n} |\mathcal{A}_i|)^{\mathcal{O}(1)}$. Note that $|\mathcal{B}|$ is in $\mathcal{O}(|K| \cdot |\Delta| \cdot \prod_{i=1}^{n} |K_i|)$. Now using similar arguments as in the proof of Lemma 4.5.2, one can show that $\mathcal{B}'$ can be computed in time $(|K| \cdot |\Delta| \cdot n \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$. □

Now let us discuss how Lemma 4.5.2 can be used to extend our results on model-checking linear-time properties to deal with regular sets.

The reachability problem: Let us assume that L is a regular set of configurations of Δ accepted by a DFA $\mathcal{A}$ with a set of states K. Let $p\alpha$ be a configuration of Δ. It follows immediately from Lemma 4.5.2 and Proposition 2.2.5 that $\mathcal{P}(p\alpha \rightarrow^* L) = \mathcal{P}(\mathcal{K}(p\alpha) \rightarrow^* \mathcal{S}[\mathcal{A}])$ (where the latter probability is evaluated in $M_{\Delta[\mathcal{A}]}$). Hence, all results for the reachability problem remain valid if we multiply the size of Δ with the factor $|K|$.

The LTL model-checking problem: A valuation ν is regular if for all $a \in Ap$ the set $\nu(a)$ is regular. Let τ be an LTL formula and let $\{a_1, \ldots, a_n\}$ be the set of all atomic propositions occurring in τ. Let ν be a regular valuation such that for every a_i we have a DFA $\mathcal{A}_i$ accepting $\nu(a_i)$. Let $K_1, \ldots, K_n$ be the sets of states of $\mathcal{A}_1, \ldots, \mathcal{A}_n$, respectively. Let $\nu' : Ap \rightarrow 2^{\mathcal{C}(\Delta[\mathcal{A}_1,\ldots,\mathcal{A}_n])}$ be a simple valuation satisfying $\nu'(a_i) = \mathcal{S}[\mathcal{A}_i]$ for all $1 \leq i \leq n$. By Lemma 4.5.2 and Proposition 2.2.5, for all $p\alpha$ we have $\mathcal{P}(p\alpha, \tau, \nu) = \mathcal{P}(\mathcal{K}(p\alpha), \tau, \nu')$. Hence all complexity estimates of Theorem 4.2.2 are valid if we multiply the size of Δ with the factor $\prod_{i=1}^{n} |K_i|$.

ω-regular properties: Let $\mathcal{R} = (\Sigma, R, \rho, s_I, F)$ be a deterministic Rabin automaton. Let us denote $\Sigma = \{a_1, \ldots, a_n\}$. Let $\nu : \mathcal{C}(\Delta) \rightarrow \Sigma$ be a valuation such that for every $a_i \in \Sigma$ the set $\nu^{-1}(a_i)$ is accepted by a DFA $\mathcal{A}_i$. Let $K_1, \ldots, K_n$ be the sets of states of $\mathcal{A}_1, \ldots, \mathcal{A}_n$, respectively.

Let us transform $\mathcal{R}$ to a Rabin automaton $\mathcal{R}'$ as follows: Substitute every transition of $\mathcal{R}$ of the form $t \xrightarrow{a_i} t'$ with transitions of the form $t \xrightarrow{h} t'$ for all heads $h \in \mathcal{H}[\mathcal{A}_i]$. Let ν' be the valuation which assigns to each configuration $q\beta$ of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ the head of $q\beta$.

It is easy to verify, using Lemma 4.5.2 and Proposition 2.2.5, that for every $p\alpha \in \mathcal{C}(\Delta)$ we have $\mathcal{P}(p\alpha, \mathcal{R}, \nu) = \mathcal{P}(\mathcal{K}(p\alpha), \mathcal{R}', \nu')$. It follows from Lemma 4.5.2 that the Rabin automaton $\mathcal{R}'$ can be computed in time $(|\mathcal{R}| \cdot |\Delta| \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$. It follows that Theorem 4.3.14 remains valid if we multiply $|\Delta|$ with $\prod_{i=1}^{n} |K_i|$. Hence, all results of Section 4.3.2 and Section 4.3.3 for ω-regular properties generalize to regular valuations after multiplying $|\Delta|$ with $\prod_{i=1}^{n} |K_i|$.

However, this is not the best result we could obtain because, e.g., the qualitative ω-regular model-checking problem would have been solved in space polynomial in $\prod_{i=1}^{n} |K_i|$, which is *not* optimal. A better solution follows from the following observation: The set $\mathbf{S}_{\Delta[\mathcal{A}_1,\ldots,\mathcal{A}_n]}$ can easily be computed from $\mathbf{S}_{\Delta}$ (see Definition 4.3.4) in polynomial time because for every $p(X, \vec{s})$ we have $[p(X, \vec{s})\uparrow] = [pX\uparrow]$. Hence, by Theorem 4.3.14, the qualitative ω-regular model-checking problem is decidable in *time* polynomial in $\prod_{i=1}^{n} |K_i|$ (for both deterministic Rabin and non-deterministic Büchi automata).

4.6 Formal Proofs

This section contains some technical proofs that were omitted in previous sections. For the whole section we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$.

4.6.1 Proofs for Reachability

Our goal is to prove Lemma 4.1.5. Let us first introduce some technical tools for manipulating the runs.

Definition 4.6.1. *Given a configuration $p\alpha \in \mathcal{C}(\Delta)$ and $\beta \in \Gamma^*$, we denote $p\alpha\lfloor\beta = p\alpha\beta$. Given a path $v \in Path[M_\Delta]$ and $\beta \in \Gamma^*$, we denote $v\lfloor\beta$ the string of configurations defined by $(v\lfloor\beta)(i) = v(i)\lfloor\beta$ (note that $v\lfloor\beta$ does not have to be a path). Given a set of paths A, we denote $A\lfloor\beta = \{v\lfloor\beta \mid v \in A\}$.*

Given $pX \in Q \times \Gamma$, we denote $FPath(pX\bullet)$ the set of all $v \in FPath(pX)$ such that for every $0 \leq i < |v|$ holds $v(i) \notin Q \times \{\varepsilon\}$ (i.e., $last(v)$ can be of the form $q\varepsilon$).

Lemma 4.6.2. *Given $\alpha \in \Gamma^*$ and $A \subseteq FPath(pX\bullet)$, we have $\mathcal{P}(Run(A\lfloor\alpha)) = \mathcal{P}(Run(A))$.*

Proof. Let A' be the set of all paths $u \in A$ that satisfy the following condition: If $v \in A$ is a prefix of u, then $v = u$. It is easy to prove that $Run(A') = Run(A)$ and that the set A' is prefix-free (see Definition 2.2.2). Moreover, $Run(A'\lfloor\alpha) = Run(A\lfloor\alpha)$ and $A'\lfloor\alpha$ is prefix-free. Thus we have $\mathcal{P}(Run(A)) = \mathcal{P}(Run(A')) = \sum_{v\in A'} \mathcal{P}(Run(v)) = \sum_{v\in A'} \mathcal{P}(Run(v\lfloor\alpha)) = \mathcal{P}(Run(A'\lfloor\alpha)) = \mathcal{P}(Run(A\lfloor\alpha))$ where the third equation follows from the definition of $\mathcal{P}$. □

Definition 4.6.3. *We denote $Run(pXq) = Run(pX \to^* q\varepsilon)$ and $FPath(pXq) = \{v \in FPath(pX) \mid last(v) = q\varepsilon, \forall j < |v| : v(j) \neq q\varepsilon\}$.*

Proof of Lemma 4.1.5

We start by proving that the tuple of all $[pXq]$ values solves the system (4.1). By definition, $[pXq] = \mathcal{P}(Run(pXq))$. Let $W = \{w \in Run(pX) \mid w(1) = q\varepsilon\}$. The set $Run(pXq)$ can be decomposed as follows [5]:

$$\begin{aligned} Run(pXq) \;=\; & W \uplus \left(\biguplus_{pX\to rY} pX \to rY \odot Run(rYq) \right) \uplus \\ & \uplus \biguplus_{pX\to rYZ, s\in Q} pX \to rYZ \odot FPath(rYs)\lfloor Z \odot Run(sZq) \end{aligned}$$

[5] We use $r\alpha \to r'\alpha'$ to denote a path of length 1 of the form $r\alpha, r'\alpha'$

Using Lemma 2.2.3 and Lemma 4.6.2, we obtain that

$$[pXq] \;=\; \sum_{pX \xrightarrow{x} q\varepsilon} x + \sum_{pX \xrightarrow{x} rY} x[rYq] + \sum_{pX \xrightarrow{x} rYZ, s \in Q} x[rYs][sZq]$$

and we are done.

Now let us prove that the tuple of all $[pXq]$ values is the least solution. For every $n \geq 0$, we denote $Run^n(pXq) = \{w \in Run(pXq) \mid \exists j \leq n : w(j) = q\varepsilon\}$ and $FPath^n(pXq) = \{v \in FPath(pXq) \mid |v| \leq n\}$. Clearly $Run^n(pXq) = Run(FPath^n(pXq))$. Let us denote $[pXq]_n = \mathcal{P}(Run^n(pXq))$. Observe that $Run^0(pXq) \subseteq Run^1(pXq) \subseteq \cdots \subseteq Run(pXq)$ and $\bigcup_{n \geq 0} Run^n(pXq) = Run(pXq)$. By the monotonicity of the probability and by [13] (Theorem 10.2), $[pXq]_0 \leq [pXq]_1 \leq \cdots \leq [pXq]$ and $\lim_{n\to\infty}[pXq]_n = [pXq]$. Thus $[pXq] = \sup_{n\geq 0}[pXq]_n$.

Let $V \in (\mathbb{R}^+)^{|\mathcal{V}|}$ be a non-negative solution of the system (4.1). We denote $V[pXq]$ the component of the vector V corresponding to the variable $\langle pXq \rangle$. We show that $[pXq]_n \leq V[pXq]$ for all $n \geq 0$, which implies $[pXq] = \sup_{n\geq 0}[pXq]_n \leq V[pXq]$. It follows that the tuple of all $[pXq]$ values is indeed the least non-negative solution of the system (4.1).

For $n = 0$ we have $Run^n(pXq) = \emptyset$, and thus $[pXq]_n = 0 \leq V[pXq]$ by the non-negativity of V. Now let us fix $n \geq 1$. By Lemma 2.2.3, Lemma 4.6.2, and induction hypothesis,

$$\begin{aligned}
[pXq]_n &= \mathcal{P}(Run^n(pXq)) \\
&\leq \mathcal{P}(W \uplus \left(\biguplus_{pX \to rY} pX \to rY \odot Run^{n-1}(rYq) \right) \uplus \\
&\uplus \biguplus_{pX \to rYZ, s \in Q} pX \to rYZ \odot FPath^{n-1}(rYs) \lfloor Z \odot Run^{n-1}(sZq)) \\
&= \sum_{pX \xrightarrow{x} q\varepsilon} x + \sum_{pX \xrightarrow{x} rY} x[rYq]_{n-1} + \sum_{pX \xrightarrow{x} rYZ, s \in Q} x[rYs]_{n-1}[sZq]_{n-1} \\
&\leq \sum_{pX \xrightarrow{x} q\varepsilon} x + \sum_{pX \xrightarrow{x} rY} xV[rYq] + \sum_{pX \xrightarrow{x} rYZ, s \in Q} xV[rYs]V[sZq] \\
&= V[pXq]
\end{aligned}$$

Note that the above inequality also proves that $[pXq]_n \leq (P^n(\vec{0}))(i(pXq)) \leq V[pXq]$ (by induction). Because P is clearly (point-wise) monotone on $(\mathbb{R}^+)^{|\mathcal{V}|}$, we obtain that $P^n(\vec{0})$ converges to the vector of all $[pXq]$ values monotonically from below. □

4.6.2 Proofs for $\mathbf{X}_\Delta$

Our primary goal in this section is to prove Lemma 4.3.6. As a byproduct, we introduce some additional tools and notation for dealing with clean runs and the chain $\mathbf{X}_\Delta$, that will prove useful especially in Section 7.3.

Lemma 4.6.4. *Given a measurable set $A \subseteq Clean(pX)$ and $\alpha \in \Gamma^*$, the set $A\lfloor\alpha$ is measurable and $\mathcal{P}(A) = \mathcal{P}(A\lfloor\alpha)$. In particular, $Clean(pX\alpha)$ is measurable and $\mathcal{P}(Clean(pX\alpha)) = \mathcal{P}(Clean(pX)) = [pX\uparrow]$.*

Proof. Let us define a function $\Theta : Run(pX) \to 2^{Run(pX\alpha)}$ as follows: Given $w \in Clean(pX)$, we put $\Theta(w) = \{w\lfloor\alpha\}$. Given $w \in Run(u)$ where $u \in FPath(pXq)$ for some $q \in Q$, we put $\Theta(w) = Run(u\lfloor\alpha)$. Observe that given two (distinct) runs $v, w \in Run(pX)$, the sets $\Theta(v)$ and $\Theta(w)$ are disjoint. The function Θ extends straightforwardly to sets $A \subseteq Run(pX)$ by $\Theta(A) = \biguplus_{w\in A} \Theta(w)$. One can easily show that for all $u \in FPath(pX\bullet)$ holds $\Theta(Run(u)) = Run(u\lfloor\alpha)$.

Let us denote $\mathcal{G}$ the set of all measurable sets $A \subseteq Run(pX)$ such that $\Theta(A)$ is measurable. We show that $\mathcal{G}$ is a σ-field containing all basic cylinders. Clearly, if $A_1, A_2, \ldots \in \mathcal{G}$, then $\Theta(\bigcup_{i=1}^{\infty} A_i) = \biguplus_{w\in\bigcup_{i=1}^{\infty} A_i} \Theta(w) = \bigcup_{i=1}^{\infty} \biguplus_{w\in A_i} \Theta(w) = \bigcup_{i=1}^{\infty} \Theta(A_i)$ is measurable, and hence $\bigcup_{i=1}^{\infty} A_i \in \mathcal{G}$. Similarly, if $A \in \mathcal{G}$, then $\Theta(Run(pX)\setminus A) = \biguplus_{w\in Run(pX)\setminus A} \Theta(w) = \biguplus_{w\in Run(pX)} \Theta(w) \setminus \biguplus_{w\in A} \Theta(w) = Run(pX\alpha)\setminus\Theta(A)$ which is measurable, and thus $Run(pX)\setminus A \in \mathcal{G}$. Finally, let us consider a basic cylinder $Run(u)$ where $u \in FPath(pX)$. We may safely assume that $u \in FPath(pX\bullet)$, and hence $\Theta(Run(u)) = Run(u\lfloor\alpha)$ which is measurable. It follows that each measurable set $A \subseteq Run(pX)$ is in $\mathcal{G}$ because the set of all measurable subsets of $Run(pX)$ is the least σ-field containing basic cylinders. In particular, for measurable $A \subseteq Clean(pX)$, the set $\Theta(A) = A\lfloor\alpha$ is measurable.

Let us define a function $\mathcal{P}'$ which assigns to each measurable set $A \subseteq Run(pX)$ the value $\mathcal{P}(\Theta(A))$. We claim that $\mathcal{P}'$ is a probability measure such that $\mathcal{P}'(Run(u)) = \mathcal{P}(Run(u))$ for every $u \in FPath(pX)$. Indeed, given a collection $A_1, A_2, \ldots \subseteq Run(pX)$ of pairwise disjoint measurable sets, we have $\Theta(\biguplus_{i=1}^{\infty} A_i)) = \biguplus_{w\in\biguplus_{i=1}^{\infty} A_i} \Theta(w) = \biguplus_{i=1}^{\infty} \biguplus_{w\in A_i} \Theta(w) = \biguplus_{i=1}^{\infty} \Theta(A_i)$, and hence, $\mathcal{P}'(\biguplus_{i=1}^{\infty} A_i) = \mathcal{P}(\Theta(\biguplus_{i=1}^{\infty} A_i)) = \sum_{i=1}^{\infty} \mathcal{P}(\Theta(A_i)) = \sum_{i=1}^{\infty} \mathcal{P}'(A_i)$. Also, given $u \in FPath(pX\bullet)$ we have $\mathcal{P}'(Run(u)) = \mathcal{P}(\Theta(Run(u))) = \mathcal{P}(Run(u\lfloor\alpha)) = \mathcal{P}(Run(u))$ due to Lemma 4.6.2. Thus, by the uniqueness of $\mathcal{P}$, we obtain $\mathcal{P}' = \mathcal{P}$. In particular, given a measurable set $A \subseteq Clean(pX)$, we have $\mathcal{P}(A\lfloor\alpha) = \mathcal{P}(\Theta(A)) = \mathcal{P}'(A) = \mathcal{P}(A)$.

□

In the rest of this section we use the following notation:

Given $qY \in \mathbf{S}_\Delta$ and a set of runs A, we denote $\mathcal{P}_{qY}(A) = \mathcal{P}(A \mid Clean(qY))$.

Definition 4.6.5. *Given a path $u \in FPath[M_\Delta]$ such that $last(u) = qY\beta$ for some $qY\beta \in \mathcal{C}(\Delta)$, and $v \in Path(qY)$, we denote $u * v = u \odot (v\lfloor\beta)$ (note that $u * v$ does not have to be a path). Given a set $A \subseteq FPath[M_\Delta]$ such that for all $u \in A$ holds $head(last(u)) = qY$, and a set $B \subseteq Path(qY)$, we denote $A * B = \{u * v \mid u \in A, v \in B\}$. We write $u * B$ instead of $\{u\} * B$. We also write $Clean(A)$ and $Clean(u)$ instead of $A * Clean(qY)$ and $u * Clean(qY)$, respectively.*

Let us illustrate the above definition using a simple example: Assume that $u = pX, pXZ, qXZZ$ and that $v = qX, qXY, qXYY, \ldots$. Then $u * v = pX, pXZ, qXZZ, qXYZZ, qXYYZZ, \ldots$.

Definition 4.6.6. *A set $A \subseteq FPath$ is* clean-prefix-free *if for all $u, v \in A$, where $u \neq v$, holds $Clean(u) \cap Clean(v) = \emptyset$.*

Lemma 4.6.7. *Let $pX, qY \in \mathbf{S}_\Delta$, let $A \subseteq FPath(pX)$ be a clean-prefix-free set of paths such that for all $v \in A$ we have $head(last(v)) = qY$, and let $B \subseteq Clean(qY)$ be a measurable set. Then $A * B$ is measurable and*

$$\mathcal{P}(A * B) = \sum_{u \in A} \mathcal{P}(Run(u)) \cdot \mathcal{P}(B) = \mathcal{P}(Clean(A)) \cdot \mathcal{P}_{qY}(B)$$

Hence, $\mathcal{P}_{pX}(A * B) = \mathcal{P}_{pX}(Clean(A)) \cdot \mathcal{P}_{qY}(B)$

Proof. First, note that $A * B = \biguplus_{u \in A} u * B = \biguplus_{u \in A} u \odot (B \lfloor tail(last(u)))$ is measurable due to Lemma 2.2.3 and Lemma 4.6.4. Note that the union is disjoint because A is clean-prefix-free and $B \subseteq Clean(qY)$. Hence,

$$\begin{aligned}
\mathcal{P}(A * B) &= \mathcal{P}\Big(\biguplus_{u \in A} u \odot (B \lfloor tail(last(u)))\Big) = \sum_{u \in A} \mathcal{P}(u \odot (B \lfloor tail(last(u)))) \\
&= \sum_{u \in A} \mathcal{P}(Run(u)) \cdot \mathcal{P}(B \lfloor tail(last(u))) = \sum_{u \in A} \mathcal{P}(Run(u)) \cdot \mathcal{P}(B)
\end{aligned}$$

where the third equation follows from Lemma 2.2.3, and the last one from Lemma 4.6.4. Now

$$\mathcal{P}(A * B) = \sum_{u \in A} \mathcal{P}(Run(u)) \cdot [qY\uparrow] \cdot \frac{\mathcal{P}(B)}{[qY\uparrow]} = \mathcal{P}(Clean(A)) \cdot \mathcal{P}_{qY}(B)$$

Finally, $\mathcal{P}_{pX}(A * B) = \frac{\mathcal{P}(Clean(A))}{[pX\uparrow]} \cdot \mathcal{P}_{qY}(B) = \mathcal{P}_{pX}(Clean(A)) \cdot \mathcal{P}_{qY}(B)$.

□

Proof of Lemma 4.3.5.

It is easy to see that

$$\begin{aligned}
Clean(pX) &= \left(\biguplus_{pX \to qY} pX \to qY \odot Clean(qY) \right) \uplus \\
&\uplus \left(\biguplus_{pX \to qYZ} pX \to qYZ * Clean(qY) \right) \uplus \\
&\uplus \biguplus_{pX \to rZY, q \in Q} pX \to rZY \odot FPath(rZq) \lfloor Y \odot Clean(qY)
\end{aligned}$$

Applying Lemma 2.2.3, Lemma 4.6.2 and Lemma 4.6.7 we obtain

$$\begin{aligned}[pX\uparrow] &= \sum_{pX \xrightarrow{y} qY} y[qY\uparrow] + \sum_{pX \xrightarrow{y} qYZ} y[qY\uparrow] + \sum_{pX \xrightarrow{y} rZY, q\in Q} y[rZq][qY\uparrow] \\ &= \sum_{qY \in \mathbf{S}_\Delta} \left(\sum_{pX \xrightarrow{y} qY} y[qY\uparrow] + \sum_{pX \xrightarrow{y} qYZ} y[qY\uparrow] + \sum_{pX \xrightarrow{y} rZY} y[rZq][qY\uparrow] \right)\end{aligned}$$

which implies the lemma. □

Definition 4.6.8. *Given two states $pX, qY \in Q \times \Gamma$, we denote $MPath(pX, qY)$ the set of* all *paths $v \in FPath(pX)$ that satisfy the following condition: Either $v = pX \to qYZ$ for some $Z \in \Gamma$, or $v = pX, p_1\alpha_1 Y, \ldots, p_k\alpha_k Y, qY$, where $k \geq 0$ ($k = 0$ implies $v = pX \to qY$) and $\alpha_1 \ldots, \alpha_k \in \Gamma^+$.*

Given a path $v = p_1X_1, \ldots, p_kX_k \in FPath[\mathbf{X}_\Delta](pX)$, where $k \geq 2$, we define $MPath(v) = MPath(p_1X_1, p_2X_2) * \cdots * MPath(p_{k-1}X_{k-1}, p_kX_k)$. For convenience, we define $MPath(pX) = \{pX\}$. Given a set of paths $A \subseteq FPath[\mathbf{X}_\Delta]$, we define $MPath(A) = \bigcup_{v\in A} MPath(v)$.

Lemma 4.6.9.

1. *Let $v \in FPath[\mathbf{X}_\Delta]$, let $u_{(i)} \in MPath(v(i-1), v(i))$ for $1 \leq i \leq |v|$, and let $w \in Clean(u_{(1)} * u_{(2)} * \cdots * u_{|v|})$ (i.e., $w \in Clean(MPath(v))$). Then $w^{ind_1(w)} = v(0)$, and for all $1 \leq i \leq |v|$ holds $w^{ind_{i+1}(w)} = u_{(1)} * \cdots * u_{(i)} \in MPath(v^i)$.*

2. *Let $w \in Clean(pX)$, where $pX \in Q \times \Gamma$. Then $w^{ind_1(w)} = pX$, and there are finite paths $u_{(1)}, u_{(2)}, \ldots$, such that for all $i \geq 1$ we have that $u_{(i)} \in MPath(\mathbf{X}^i_\Delta(w), \mathbf{X}^{i+1}_\Delta(w))$ and*

$$w^{ind_{i+1}(w)} = u_{(1)} * \cdots * u_{(i)} \in MPath(\mathbf{X}^1_\Delta(w), \ldots, \mathbf{X}^{i+1}_\Delta(w))$$

3. *Given $A \subseteq FPath[\mathbf{X}_\Delta](pX)$, where $pX \in \mathbf{S}_\Delta$, we have*

$$Clean(MPath(A)) = \bigcup_{v\in A} \{w \in Clean(pX) \mid \bigwedge_{i=0}^{|v|} \mathbf{X}^{i+1}_\Delta(w) = v(i)\}$$

4. *If $A \in FPath[\mathbf{X}_\Delta](pX)$ is prefix-free, then $MPath(A)$ is clean-prefix-free.*

Proof. 1. and 2. follow immediately from definitions. 3. follows from 1. and 2. and the fact that $Clean(MPath(A)) = \bigcup_{v\in A} Clean(MPath(v))$. It remains to prove 4. Let us assume that A is prefix-free. Let $z, z' \in A$ such that $z \neq z'$, and let $u \in MPath(z)$ and $v \in MPath(z')$. If $w \in Clean(u) \cap Clean(v)$, then, by 3., either z is a prefix of z' or vice versa, which contradicts the assumption

that A is prefix-free. Let us fix $z \in FPath[\mathbf{X}_\Delta](pX)$. Let $u, v \in MPath(z) = MPath(z(0), z(1)) * MPath(z_1)$, and let us assume that $Clean(u) \cap Clean(v) \neq \emptyset$. We show that $u = v$. Let us denote $u = u' * u''$ and $v = v' * v''$ where $u', v' \in MPath(z(0), z(1))$ and $u'', v'' \in MPath(z_1)$. We show that $u' = v'$. Assume, without the loss of generality, that u' is a proper prefix of v'. Then, by definition, $u' = pX \rightarrow qYZ$ and $v' = pX, qYZ, \ldots, qY$. However, then $Clean(u') \cap Clean(v') = \emptyset$, and thus also $Clean(u) \cap Clean(v) = \emptyset$, a contradiction. Hence, $u' = v'$, and thus $Clean(u'') \cap Clean(v'') \neq \emptyset$. However, then $u'' = v''$ (by induction), which implies $u = v$. □

Now we have all tools needed to prove Lemma 4.3.6.

Proof of Lemma 4.3.6.

First, we prove the following claim:

Claim (1). *Given* $v \in FPath[\mathbf{X}_\Delta](pX)$,

$$\mathcal{P}_{pX}(\bigwedge_{j=0}^{|v|} \mathbf{X}_\Delta^{j+1} = v(j)) = \prod_{j=0}^{|v|-1} Prob(v(j) \hookrightarrow v(j+1)) = \mathcal{P}(Run[\mathbf{X}_\Delta](v))$$

Proof. Given $qY, rZ \in \mathbf{S}_\Delta$, the probability $\mathcal{P}_{qY}(Clean(MPath(qY, rZ)))$ is equal to $Prob(qY \hookrightarrow rZ)$ (this can be proved using arguments similar to the proof of Lemma 4.3.5). Hence, by Lemma 4.6.9 and Lemma 4.6.7,

$$\begin{aligned}
\mathcal{P}_{pX}(\bigwedge_{j=0}^{|v|} \mathbf{X}_\Delta^{j+1} = v(j)) &= \mathcal{P}_{pX}(Clean(MPath(v))) \\
&= \prod_{j=0}^{|v|-1} \mathcal{P}_{v(j)}(Clean(MPath(v(j), v(j+1)))) \\
&= \prod_{j=0}^{|v|-1} Prob(v(j) \hookrightarrow v(j+1))
\end{aligned}$$

◊

Now we prove that the set $Good(pX)$ is measurable and $\mathcal{P}_{pX}(Good(pX)) = 1$. Let us denote

$$D_k = \biguplus_{u \in FPath[\mathbf{X}_\Delta](pX), |u|=k} \{w \in Clean(pX) \mid \bigwedge_{j=0}^{k} \mathbf{X}_\Delta^{j+1}(w) = u(j)\}$$

It is easy to see that $Good(pX) = \bigcap_{k=1}^{\infty} D_k$, and hence $Good(pX)$ is measurable by Lemma 4.6.9. Moreover, by Claim (1),

$$\mathcal{P}_{pX}(D_k) \;=\; \sum_{u \in FPath[\mathbf{X}_\Delta](pX), |u|=k} \mathcal{P}(Run[\mathbf{X}_\Delta](u)) = 1 \qquad (4.2)$$

Since $D_1 \supseteq D_2 \supseteq \cdots$, we have $\mathcal{P}_{pX}(Good(pX)) = \lim_{k\to\infty} \mathcal{P}_{pX}(D_k) = 1$ by [13] (Theorem 10.2).

Now we finish the proof of the lemma. Let $\mathcal{F}'$ be the set of all sets of the form $A \cap Good(pX)$, where $A \subseteq Run(pX)$ is a measurable set. Clearly, all sets of $\mathcal{F}'$ are measurable. By [13] (Theorem 10.1), the set $\mathcal{F}'$ is a σ-field. Let $v \in FPath[\mathbf{X}_\Delta](pX)$ and let $B = \{w \in Clean(pX) \mid \bigwedge_{j=0}^{|v|} \mathbf{X}_\Delta^{j+1}(w) = v(j)\}$. It follows from Lemma 4.6.9 that B is measurable, which implies that $fp^{-1}(Run[\mathbf{X}_\Delta](v)) = Good(pX) \cap B \in \mathcal{F}'$. It follows that for *all* measurable sets $A \subseteq Run[\mathbf{X}_\Delta](pX)$, we have that $fp^{-1}(A) \in \mathcal{F}'$, and hence that $fp^{-1}(A)$ is measurable (see [13], Theorem 13.1). Now for all measurable sets $A \subseteq Run[\mathbf{X}_\Delta](pX)$ we define $\mathcal{P}'(A) = \mathcal{P}_{pX}(fp^{-1}(A))$. By Claim (1),

$$\mathcal{P}_{pX}(fp^{-1}(Run[\mathbf{X}_\Delta](v))) = \mathcal{P}_{pX}(Good(pX) \cap B) = \mathcal{P}(Run[\mathbf{X}_\Delta](v))$$

It follows that $\mathcal{P}'$ is a probability measure, which coincides with $\mathcal{P}$ on basic cylinders in $\mathbf{X}_\Delta$. Hence, $\mathcal{P}_{pX}(fp^{-1}(A)) = \mathcal{P}'(A) = \mathcal{P}(A)$, by the uniqueness of $\mathcal{P}$.

□

4.6.3 Proofs for ω-regular Properties

The goal of this section is to prove Lemma 4.3.12 and Lemma 4.4.2.

Definition 4.6.10. *Let w be a clean run in M_Δ, let $i \geq 1$, and let $(w^{ind_{i+1}(w)})_{ind_i(w)} = qY\beta, r_1\alpha_1\beta, \ldots, r_k\alpha_k\beta$ be the subword of w between the i'th and $i+1$'th minima of w (both inclusive). We denote $\wp_i(w)$ the path $qY, r_1\alpha_1, \ldots, r_k\alpha_k$ (i.e., $\wp_i(w)$ is the path $qY\beta, r_1\alpha_1\beta, \ldots, r_k\alpha_k\beta$ where β is "cut off").*

Lemma 4.6.11. *Let $pX, qY \in \mathbf{S}_\Delta$ such that almost all runs of $Run[\mathbf{X}_\Delta](pX)$ contain qY infinitely many times. Let $\zeta : FPath(qY) \to \mathbb{R}$ be a function. Let $X_1, X_2, \ldots$ be random variables defined over $Run(pX)$ such that for each $i \geq 1$ and each $w \in Clean(pX)$ we have $X_i(w) = \zeta(\wp_k(w))$, where k is the index of i^{th} occurrence of qY in $fp(w)$ (if either $w \notin Clean(pX)$, or there are less than i occurrences of qY in $fp(w)$, then we put $X_i(w) = 0$). Then the sequence $X_1, X_2, \ldots$ is independent w.r.t. $\mathcal{P}_{pX}$, and moreover, for all $x \in \mathbb{R}$ we have $\mathcal{P}_{pX}(X_i = x) = \mathcal{P}_{qY}(\zeta \circ \wp_1 = x)$.*

Proof. By [13], it suffices to show that for all $k \geq 1$ and all $x_1, \ldots, x_k \in \mathbb{R}$ holds $\mathcal{P}_{pX}(\bigwedge_{i=1}^{k} X_i = x_i) = \prod_{i=1}^{k} \mathcal{P}_{qY}(\zeta \circ \wp_1 = x_i)$, because the variables $X_1, X_2, \ldots$ are discrete.

Let us first assume that $pX = qY$. We denote $\mathcal{V}$ the set of all paths $v \in FPath[\mathbf{X}_\Delta](qY)$ that satisfy $|v| > 0$, $last(v) = qY$, and $v(i) \neq qY$ for $0 < i < |v|$. Note that $\mathcal{P}_{qY}(Clean(MPath(\mathcal{V}))) = 1$ due to Lemma 4.6.9, Lemma 4.3.6, and the fact that almost all runs of $Run[\mathbf{X}_\Delta](qY)$ reach qY infinitely many times.

Observe that given $u \in MPath(\mathcal{V})$, the function $\wp_1$ is constant over $Clean(u)$. For each $1 \leq i \leq k$ we define B_i to be the set of all finite paths $u \in MPath(\mathcal{V})$ such that $\zeta(\wp_1(w)) = x_i$ for all $w \in Clean(u)$. One can easily verify, using Lemma 4.6.9, that $\mathcal{P}_{qY}(\bigwedge_{i=1}^{k} X_i = x_i) = \mathcal{P}_{qY}(Clean(B_1 * \cdots * B_k))$. Moreover, $\mathcal{P}_{qY}(Clean(B_i)) = \mathcal{P}_{qY}(\zeta \circ \wp_1 = x_i)$ for all $1 \leq i \leq k$. Also, by Lemma 4.6.9, each set B_i is clean-prefix-free. Hence, by Lemma 4.6.7,

$$\mathcal{P}_{qY}(\bigwedge_{i=1}^{k} X_i = x_i) = \mathcal{P}_{qY}(Clean(B_1 * \cdots * B_k)) = \prod_{i=1}^{k} \mathcal{P}_{qY}(\zeta \circ \wp_1 = x_i)$$

Now let us assume that $pX \neq qY$. Let $\mathcal{U}$ be the set of all paths $v \in FPath[\mathbf{X}_\Delta](pX)$ that satisfy $last(v) = qY$, and $v(i) \neq qY$ for $0 \leq i < |v|$. Let $B_0 = MPath(\mathcal{U})$. One can easily verify that $\mathcal{P}_{pX}(\bigwedge_{i=1}^{k} X_i = x_i) = \mathcal{P}_{pX}(Clean(B_0 * B_1 * \cdots * B_k))$. Also, note that $\mathcal{P}_{pX}(Clean(B_0)) = 1$ due to Lemma 4.6.9, Lemma 4.3.6, and the fact that almost all runs of $Run[\mathbf{X}_\Delta](pX)$ reach qY infinitely many times. Hence, by Lemma 4.6.7,

$$\mathcal{P}_{pX}(\bigwedge_{i=1}^{k} X_i = x_i) = \mathcal{P}_{pX}(Clean(B_0 * B_1 * \cdots * B_k)) = \prod_{i=1}^{k} \mathcal{P}_{qY}(\zeta \circ \wp_1 = x_i)$$

□

Corollary 4.6.12. *Let $pX, qY \in \mathbf{S}_\Delta$ such that almost all runs of $Run[\mathbf{X}_\Delta](pX)$ contain qY infinitely many times, let $rZ \in \mathbf{S}_\Delta$ such that $qY \hookrightarrow rZ$, and let $A \subseteq MPath(qY, rZ)$ be a* non-empty *set. Then for almost all runs $w \in Clean(pX)$, there are infinitely many $i \geq 0$ such that $\wp_i(w) \in A$.*

Proof. Given $i \geq 1$, we denote D_i the set of all runs $w \in Clean(pX)$ such that $\wp_k(w) \in A$, where k is the index of i^{th} occurrence of qY in $fp(w)$. By Lemma 4.6.11, the sets $D_1, D_2, \ldots$ are independent (w.r.t. $\mathcal{P}_{pX}$), and $\mathcal{P}_{pX}(D_i) = \mathcal{P}_{qY}(\wp_1 \in A) > 0$ for all $i \geq 1$. (To apply Lemma 4.6.11 define $\zeta(v) = 1$ for $v \in A$, and $\zeta(v) = 0$ for $v \in FPath(qY) \backslash A$. Then the variables $X_1, X_2, \ldots$ become characteristic functions of the sets $D_1, D_2, \ldots$, resp.). Now $\sum_{i=1}^{\infty} \mathcal{P}_{pX}(D_i) = \sum_{i=1}^{\infty} \mathcal{P}_{qY}(\wp_1 \in A) = \infty$, and hence, by the second Borel-Cantelli lemma (see, e.g., [13]), $\mathcal{P}_{pX}(\bigcap_{i=1}^{\infty} \bigcup_{j=i}^{\infty} D_j) = 1$. □

Proof of Lemma 4.3.12

Let us assume that $s \in O_C$, i.e., that there is $(p,t)X \in C$ and a path $v \in FPath[M_{\Delta \times \mathcal{R}}]((p,t)X)$ such that $head(last(v)) = (p,t)X$ and s occurs in v. Let $w \in Clean(v)$ be a good run (such a good run exists due to

Proposition 4.3.6, because $\mathcal{P}(Clean(v)) = \mathcal{P}(Run(v)) \cdot [(p,t)X\uparrow] > 0$ by Lemma 4.6.7). There is i such that s occurs in $\wp_i(w)$. Since w is good, we have that $\mathbf{X}^i_{\Delta\times\mathcal{R}}(w) \in C$, which implies that almost all runs of $Run[\mathbf{X}_{\Delta\times\mathcal{R}}]((p,t)X)$ contain $\mathbf{X}^i_{\Delta\times\mathcal{R}}(w)$ infinitely many times. Moreover, $\mathbf{X}^i_{\Delta\times\mathcal{R}}(w) \hookrightarrow \mathbf{X}^{i+1}_{\Delta\times\mathcal{R}}(w)$. Hence, by Corollary 4.6.12, almost all runs of $Clean((p,t)X)$ follow the path $\wp_i(w) \in MPath(\mathbf{X}^i_{\Delta\times\mathcal{R}}(w), \mathbf{X}^{i+1}_{\Delta\times\mathcal{R}}(w))$ infinitely many times (with possibly different contexts), and thus for almost all runs $w \in Clean((p,t)X)$ holds $s \in Inf_{\mathcal{R}}(w)$.

Let us denote $A = \{w \in Good((p,t)X) \mid s \in Inf_{\mathcal{R}}(w)\}$. We have proved above that $\mathcal{P}_{(p,t)X}(A) = 1$. Let B be the set of all paths $v \in FPath[\mathbf{X}_{\Delta\times\mathcal{R}}]((q_0,s_I)Z_0)$, such that $last(v) = (p,t)X$ and for all $j < |v|$ holds $v(j) \neq (p,t)X$. Observe that $MPath(B) * A \subseteq Run((q_0,s_I)Z_0, C)$, and that for all $w \in MPath(B) * A$ we have $s \in Inf_{\mathcal{R}}(w)$. Moreover, by Lemma 4.6.9, $Clean(MPath(B)) \cap Good((q_0,s_I)Z_0) = Run((q_0,s_I)Z_0, C)$, which implies that $\mathcal{P}(Clean(MPath(B))) = \mathcal{P}(Run((q_0,s_I)Z_0, C))$ because almost all runs of $Clean(MPath(B))$ are good by Lemma 4.3.6. Hence, by Lemma 4.6.7,

$$\begin{aligned} \mathcal{P}(MPath(B) * A) &= \mathcal{P}(Clean(MPath(B))) \cdot \mathcal{P}_{(p,t)X}(A) \\ &= \mathcal{P}(Run((q_0 s_I)Z_0, C)) \end{aligned}$$

It follows that for almost all runs $w \in Run((q_0,s_I)Z_0, C)$ holds $s \in Inf_{\mathcal{R}}(w)$.

Now let us assume that $s \in R\backslash O_C$. If w is a run of $Run((q_0,s_I)Z_0, C)$ such that $s \in Inf_{\mathcal{R}}(w)$, then there are $i < k < j$ such that $\mathbf{X}^i_{\Delta\times\mathcal{R}}(w) = \mathbf{X}^j_{\Delta\times\mathcal{R}}(w) \in C$ and s occurs in $\wp_k(w)$. Then, however, the path $\wp_i(w) * \cdots * \wp_{j-1}(w)$ contradicts the assumption that $s \notin O_C$. Hence, no run $w \in Run((q_0,s_I)Z_0, C)$ satisfies $s \in Inf_{\mathcal{R}}(w)$.

□

Proof of Lemma 4.4.2

First, we introduce some notation. For every $s \in R$, the symbol $\mathcal{R}^s$ denotes the Rabin automaton obtained from $\mathcal{R}$ by changing the initial state to s. Moreover, for all $s,t \in R$ we denote $\mathcal{L}^{s\to t}$ the set of all finite paths v satisfying the following condition: the Rabin automaton $\mathcal{R}$ initiated in s moves to t after reading the heads of all configurations in v.

Let us fix $\varrho \in \{0,1\}$. Let $\alpha \in \Gamma^*$, and let us denote $\gamma_\varrho(\alpha^R) = A$. We show the following assertion by induction on $|\alpha|$: for all $(p,s) \in Q \times R$ we have $(p,s) \in A$ if and only if $\mathcal{P}(p\alpha, \mathcal{R}^s) = \varrho$. The lemma then follows immediately from the definition of γ_ϱ.

If $|\alpha| = 0$, then the assertion follows immediately from the definition of I_ϱ. Let us assume that $\alpha = X\beta$ where $X \in \Gamma$, and let us denote $B = \gamma_\varrho(\beta^R)$.

By Lemma 4.6.7, Lemma 4.6.2, and Lemma 2.2.3,

$$\begin{aligned}
\mathcal{P}(p\alpha, \mathcal{R}^s) &= \mathcal{P}(\{w \in \mathit{Clean}(pX\beta) \mid w \in \mathcal{L}(\mathcal{R}^s)\}) + \\
&+ \sum_{(q,t)\in Q\times R} \mathcal{P}(\mathit{Run}(\{v \in \mathit{FPath}(pXq)\lfloor\beta \mid v \in \mathcal{L}^{s\to t}\})) \cdot \mathcal{P}(q\beta, \mathcal{R}^t) \\
&= \mathcal{P}(\{w \in \mathit{Clean}(pX) \mid w \in \mathcal{L}(\mathcal{R}^s)\}) + \\
&+ \sum_{(q,t)\in Q\times R} \mathcal{P}(\mathit{Run}(\{v \in \mathit{FPath}(pXq) \mid v \in \mathcal{L}^{s\to t}\})) \cdot \mathcal{P}(q\beta, \mathcal{R}^t) \\
&= [s, pX\uparrow] + \sum_{(q,t)\in Q\times R} [s, pXq, t] \cdot \mathcal{P}(q\beta, \mathcal{R}^t)
\end{aligned}$$

Now we distinguish two cases depending on ϱ:

- $\varrho = 1$: Suppose that $(p, s) \in A$. Then by the definition of γ_1, we have $[s, pX\uparrow] + \sum_{(q,t)\in B} [s, pXq, t] = 1$. Moreover, by induction, $\mathcal{P}(q\beta, \mathcal{R}^t) = 1$ for all $(q, t) \in B$. The rest follows from the above equation.
  Now suppose that $\mathcal{P}(p\alpha, \mathcal{R}^s) = 1$. It follows from the above equation that if $[s, pXq, t] > 0$, then $\mathcal{P}(q\beta, \mathcal{R}^t) = 1$, which implies $(q, t) \in B$ by induction. But then $[s, pX\uparrow] + \sum_{(q,t)\in B} [s, pXq, t] = 1$, and hence $(p, s) \in A$.
- $\varrho = 0$: Suppose that $(p, s) \in A$. Then by the definition of γ_0, we have $[s, pX\uparrow] + \sum_{(q,t)\in(Q\times R)\setminus B} [s, pXq, t] = 0$. Moreover, by induction, $\mathcal{P}(q\beta, \mathcal{R}^t) = 0$ for all $(q, t) \in B$. The rest follows from the above equation.
  Now suppose that $\mathcal{P}(p\alpha, \mathcal{R}^s) = 0$. It follows from the above equation that if $[s, pXq, t] > 0$, then $\mathcal{P}(q\beta, \mathcal{R}^t) = 0$, which implies $(q, t) \in B$ by induction. But then $[s, pX\uparrow] + \sum_{(q,t)\in(Q\times R)\setminus B} [s, pXq, t] = 0$, and hence $(p, s) \in A$.

□

4.6.4 Miscellaneous Proofs

Proof of Lemma 4.1.3

Let $\mathcal{H}$ be the set of heads which determines L. Let us denote $\alpha = X_n \cdots X_1$. We define a new pPDA Δ' with the set of states $Q \uplus \{q_0, \ldots, q_n\} \uplus \{q_e\}$, the stack alphabet $\Gamma \uplus \{Z_0\}$, and transitions defined as follows:

1. $q_0 Z_0 \xrightarrow{1} q_1 X_1 Z_0$, $q_k X_k \xrightarrow{1} q_{k+1} X_{k+1} X_k$ for $1 \leq k < n$ and $q_n X_n \xrightarrow{1} pX_n$ (for $n = 0$, we put $q_0 Z_0 \xrightarrow{1} pZ_0$);
2. For $rX \in \mathcal{H}$, we define $rX \xrightarrow{1} q_e\varepsilon$ in Δ';
3. For $rX \in (Q \times \Gamma)\setminus\mathcal{H}$, we put $rX \xrightarrow{x} q\alpha$ in Δ' iff $rX \xrightarrow{x} q\alpha$ in Δ;
4. For $r \in \mathcal{H} \cap Q$, we define $rZ_0 \xrightarrow{1} q_0\varepsilon$ in Δ';
5. For $r \in Q\setminus\mathcal{H}$, we define $rZ_0 \xrightarrow{x} rZ_0$ in Δ';
6. $q_e Z_0 \xrightarrow{1} q_0\varepsilon$ and $q_e X \xrightarrow{1} q_e\varepsilon$ for all $X \in \Gamma$.

Intuitively, the pPDA Δ' first pushes the word αZ_0 to the stack, and then simulates transitions of Δ until a configuration with a head of $\mathcal{H}$ is reached. Then Δ' clears its stack and changes its control state to q_0. It is easy to verify that $\mathcal{P}(p\alpha \rightarrow^* L) = \mathcal{P}(q_0 Z_0 \rightarrow^* q_0\varepsilon)$ and that the size of Δ' is in $\mathcal{O}(|\Delta| + |p\alpha|)$. □

Proof of Lemma 4.2.1

Let us denote $\alpha = X_n \cdots X_1$. We define Δ' with the set of states $Q \uplus \{q_0, \ldots, q_n\}$, the stack alphabet $\Gamma \uplus \{Z_0\}$, and transitions:

1. $q_0 Z_0 \xrightarrow{1} q_1 X_1 Z_0$, $q_k X_k \xrightarrow{1} q_{k+1} X_{k+1} X_k$ for $1 \leq k < n$ and $q_n X_n \xrightarrow{1} p X_n$ (for $n = 0$, we put $q_0 Z_0 \xrightarrow{1} p Z_0$);
2. For $rX \in Q \times \Gamma$ we define: $rX \xrightarrow{x} q\alpha$ in Δ' iff $rX \xrightarrow{x} q\alpha$ in Δ;
3. For $q \in Q \uplus \{q_1, \ldots, q_n\}$ we define: $qZ_0 \xrightarrow{1} qZ_0$.

The pPDA Δ' first pushes the word αZ_0 to the stack, and then simulates Δ. We define $\tau' = \mathcal{X}^{n+1}(\tau)$ (here $\mathcal{X}^{n+1}$ is a shorthand for $n+1$ applications of the operator $\mathcal{X}$). Let us assume that for each $a \in Ap$, the simple set $\nu(a)$ is determined by a set $\mathcal{H}_a$ of heads. Given $a \in Ap$, the set $\nu'(a)$ is determined by the set of heads $(\mathcal{H}_a \backslash Q) \cup \{pZ_0 \mid p \in \mathcal{H}_a\}$. It is easy to verify that $\mathcal{P}(p\alpha, \tau, \nu) = \mathcal{P}(q_0 Z_0, \tau', \nu')$. □

Proof of Lemma 4.3.11

Let us denote $\alpha = X_n \cdots X_1$. Let Δ' be exactly the same pPDA as in the previous proof of Lemma 4.2.1. The automaton $\mathcal{R}'$ is obtained from $\mathcal{R}$ by adding a new initial state s'_I and transitions: $s'_I \xrightarrow{q_0 Z_0} s'_I$, $s'_I \xrightarrow{q_i X_i} s'_I$ for all $1 \leq i < n$, $s'_I \xrightarrow{q_n X_n} s_I$, and $s \xrightarrow{qZ_0} t$ for $q \in Q$ whenever $s \xrightarrow{q} t$. It is easy to verify that $\mathcal{P}(p\alpha, \mathcal{R}) = \mathcal{P}(q_0 Z_0, \mathcal{R}')$.

Finally, we present a simple procedure which decides whether $\mathcal{R}$ accepts q^ω. The procedure follows the only computation of $\mathcal{R}$ over q^ω until a state s of $\mathcal{R}$ repeats twice in the computation. Let D be the set of all states of $\mathcal{R}$ visited between the two occurrences of s. If there is $1 \leq i \leq n$ such that $D \cap A_i \neq \emptyset$ and $D \cap B_i = \emptyset$, then clearly $q^\omega \in \mathcal{L}(\mathcal{R})$, else $q^\omega \notin \mathcal{L}(\mathcal{R})$. □

Chapter 5

Branching-time Properties

In this chapter we study the model-checking problem for pPDA and the following branching-time temporal logics: PCTL, PCTL* and PECTL*. We show that this problem is undecidable for pPDA and a very simple fragment of PCTL. We also prove that this problem is undecidable for pBPA and a fragment of PCTL* (called PCTL$^+$). On the other hand, we show that the model-checking problem is decidable for pPDA and *qualitative* fragments of the above logics, and also provide some complexity bounds. This chapter is based on the extended and updated version of the paper [17].

5.1 Basic Definitions

We start with a definition of the temporal logic PCTL*. The syntax of PCTL* *state* and *path* formulae is given by the following abstract syntax equations.

$$\begin{array}{rcl} \Phi & ::= & \mathtt{tt} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathcal{P}^{\sim\varrho}\varphi \\ \varphi & ::= & \Phi \mid \neg\varphi \mid \varphi_1 \wedge \varphi_2 \mid \mathcal{X}\varphi \mid \varphi_1 \,\mathcal{U}\, \varphi_2 \end{array}$$

Here a ranges over the countably infinite set Ap of atomic propositions, $\varrho \in [0,1]$, and $\sim \in \{\leq, <, \geq, >, =\}$. Formulae of the form $\mathcal{P}^{\sim\varrho}(\mathtt{tt}\,\mathcal{U}\,\varphi)$ are sometimes abbreviated as $\Diamond^{\sim\varrho}\varphi$. The size $|\Phi|$ of a formula Φ is defined to be the number of connectives in Φ plus the sum of sizes of binary representations of all numbers occurring in the formula.

The logic PCTL is a fragment of PCTL* where state formulae are defined as for PCTL* and path formulae are given by the equation $\varphi ::= \mathcal{X}\Phi \mid \Phi_1 \,\mathcal{U}\, \Phi_2$. We also denote PCTL$^+$ the fragment of PCTL* where the $\mathcal{X}$ and $\mathcal{U}$ operators in path formulae can be combined using Boolean connectives, but they cannot be nested (for example, $\mathcal{P}^{=\frac{1}{2}}((a\,\mathcal{U}\,b) \vee (b\,\mathcal{U}\,c))$ is a PCTL$^+$ formula while $\mathcal{P}^{=\frac{1}{2}}(a\,\mathcal{U}\,(b\,\mathcal{U}\,c))$ is not).

Finally, the logic PECTL* is an extension of PCTL* (see Remark 5.1.1 below) where only state formulae are introduced and have the following syntax:

$$\Phi \quad ::= \quad \mathtt{tt} \mid a \mid \neg\Phi \mid \Phi_1 \wedge \Phi_2 \mid \mathcal{P}^{\sim\varrho}\mathcal{B}(\Phi_1, \cdots, \Phi_n)$$

Here $n \geq 1$, $\mathcal{B}$ is a Büchi automaton over the alphabet $2^{\{1,\ldots,n\}}$, and each Φ_i is a PECTL* formula.

The semantics of PCTL* formulae is defined below. Let $M = (S, \to, Prob)$ be a Markov chain and let $\nu : Ap \to 2^S$ be a valuation. State formulae are interpreted over S, and path formulae are interpreted over Run. Hence, for a given $s \in S$ we define

$$
\begin{array}{lll}
s \models^\nu \mathtt{tt} & & \\
s \models^\nu a & \text{iff} & s \in \nu(a) \\
s \models^\nu \neg\Phi & \text{iff} & s \not\models^\nu \Phi \\
s \models^\nu \Phi_1 \wedge \Phi_2 & \text{iff} & s \models^\nu \Phi_1 \text{ and } s \models^\nu \Phi_2 \\
s \models^\nu \mathcal{P}^{\sim\varrho}\varphi & \text{iff} & \mathcal{P}(\{w \in Run(s) \mid w \models^\nu \varphi\}) \sim \varrho
\end{array}
$$

and for a given $w \in Run$ we define

$$
\begin{array}{lll}
w \models^\nu \Phi & \text{iff} & w(0) \models^\nu \Phi \\
w \models^\nu \neg\varphi & \text{iff} & w \not\models^\nu \varphi \\
w \models^\nu \varphi_1 \wedge \varphi_2 & \text{iff} & w \models^\nu \varphi_1 \text{ and } w \models^\nu \varphi_2 \\
w \models^\nu \mathcal{X}\varphi & \text{iff} & w_1 \models^\nu \varphi \\
w \models^\nu \varphi_1 \,\mathcal{U}\, \varphi_2 & \text{iff} & \exists j \geq 0 : w_j \models^\nu \varphi_2 \text{ and } w_i \models^\nu \varphi_1 \text{ for all } 0 \leq i < j
\end{array}
$$

For PCTL, the semantics of path formulae is redefined to

$$
\begin{array}{lll}
w \models^\nu \mathcal{X}\Phi & \text{iff} & w(1) \models^\nu \Phi \\
w \models^\nu \Phi_1 \,\mathcal{U}\, \Phi_2 & \text{iff} & \exists j \geq 0 : w(j) \models^\nu \Phi_2 \text{ and } w(i) \models^\nu \Phi_1 \text{ for all } 0 \leq i < j
\end{array}
$$

The semantics of a PECTL* formula $\Phi \equiv \mathcal{P}^{\sim\varrho}\mathcal{B}(\Phi_1, \cdots, \Phi_n)$, where $\mathcal{B}$ is a Büchi automaton with the alphabet $2^{\{1,\ldots,n\}}$, is defined as follows: First, we can assume that the semantics of the PECTL* formulae $\Phi_1, \ldots, \Phi_n$ has already been defined. This means that for each $w \in Run$ we can define an infinite word w_Φ over the alphabet $2^{\{1,\ldots,n\}}$ by $w_\Phi(i) = \{k \in \{1, \ldots, n\} \mid w(i) \models^\nu \Phi_k\}$ for all $i \geq 0$. For every state s of M, let $Run(s, \Phi) = \{w \in Run(s) \mid w_\Phi \in \mathcal{L}(\mathcal{B})\}$. We stipulate that $s \models^\nu \Phi$ if and only if $\mathcal{P}(Run(s, \Phi)) \sim \varrho$.

Remark 5.1.1. *For the sake of completeness, we prove that PECTL* is indeed an extension of PCTL*. We show that every PCTL* state formula of the form $\mathcal{P}^{\sim\varrho}\varphi$ can be translated to an equivalent PECTL* formula. Let $\Phi_1, \ldots, \Phi_n$ be all maximal state subformulae of φ (i.e. for every i, there is no state subformula of φ containing Φ_i as its strict subformula).*

Let $\bar{\varphi}$ be the LTL formula obtained from φ by substituting formulae $\Phi_1, \ldots, \Phi_n$ with fresh new atomic propositions $[\Phi_1], \ldots, [\Phi_n]$, respectively. Applying well-known algorithms for translating LTL formulae to Büchi automata (see, e.g., [51, 19]), one obtains a Büchi automaton $\mathcal{B}$ (where $|\mathcal{B}| = 2^{\mathcal{O}(|\varphi|)}$) with the alphabet $2^{\{1,\ldots,n\}}$ that satisfies the following condition: for every run w and a valuation ν, we have $w \models^\nu \bar{\varphi}$ if and only if $w_{\bar{\varphi}} \in \mathcal{L}(\mathcal{B})$ where $w_{\bar{\varphi}}$ is a word defined by $w_{\bar{\varphi}}(i) = \{k \in \{1, \ldots, n\} \mid w(i) \models^\nu [\Phi_k]\}$ for all $i \geq 0$. It is easy to show, by induction, that also $w \models^\nu \varphi$ if and only if $w_\varphi \in \mathcal{L}(\mathcal{B})$ where w_φ is a word defined by $w_\varphi(i) = \{k \in \{1, \ldots, n\} \mid w(i) \models^\nu \Phi_k\}$ for all $i \geq 0$.

Now let us assume that the formulae $\Phi_1, \ldots, \Phi_n$ have already been translated to PECTL* formulae $\Phi'_1, \ldots, \Phi'_n$, respectively. It is easy to verify that for each state s of M and each valuation ν, we have $s \models^\nu \mathcal{P}^{\sim\varrho}\varphi$ if and only if $s \models^\nu \mathcal{P}^{\sim\varrho}\mathcal{B}(\Phi'_1, \ldots, \Phi'_n)$.

The *qualitative fragments* of PCTL, PCTL*, and PECTL*, denoted qPCTL, qPCTL*, and qPECTL*, resp., are obtained by restricting the allowed operator/number combinations in $\mathcal{P}^{\sim\varrho}\varphi$ and $\mathcal{P}^{\sim\varrho}\mathcal{B}(\Phi_1, \cdots, \Phi_n)$ subformulae to '≤ 0' and '≥ 1', which can also be written as '$= 0$' and '$= 1$', resp. (Observe that '< 1', '> 0' are definable from '≤ 0', '≥ 1', and negation.)

The model-checking problem for the above logics is stated as follows.

- *(branching-time) model-checking*:

  Given a state s_0 of M, a valuation ν, and a *state* formula Φ, does $s_0 \models^\nu \Phi$ hold?

Finite Markov chains

The model-checking problem for finite Markov chains and branching-time probabilistic logics was intensively studied in the past (see, e.g., [35, 23, 6]). Many model-checking algorithms were proposed and also implemented (see, e.g., []). It is quite instructive to look at one of these algorithms, because model-checking algorithms for pPDA and qualitative branching-time logics, presented in Section 5.3, follow a similar pattern.

We outline an algorithm for model-checking PECTL* formulae: Let τ be a PECTL* state formula, and let M be a finite Markov chain. The algorithm proceeds "bottom up" through τ, and computes for every state subformula Φ of τ the set of states that satisfy Φ. The only non-trivial part of this algorithm is to compute the set of states that satisfy a subformula of the form $\mathcal{P}^{\sim\varrho}\mathcal{B}(\Phi_1, \ldots, \Phi_n)$. Assume that the sets $A_1, \ldots, A_n$ of states satisfying subformulae $\Phi_1, \ldots, \Phi_n$, resp., have already been computed. Then the problem of computing the set of states that satisfy $\mathcal{P}^{\sim\varrho}\mathcal{B}(\Phi_1, \ldots, \Phi_n)$ reduces to the following problem: Compute the set of all states s of M that satisfy $\mathcal{P}(s, \mathcal{B}, \nu) \sim \varrho$ (see Section 4.3) for a valuation ν defined by $\nu(t) = \{i \in \{1, \ldots, n\} \mid t \in A_i\}$ for all states t of M (remember that the alphabet of $\mathcal{B}$ is $2^{\{1,\ldots,n\}}$). Note that the latter problem is solved by solving the ω-regular model-checking problem for each state of M, which can be done in exponential time (see [22]). Hence the set of states that satisfy τ can be computed in exponential time, which gives the **EXPTIME** upper bound on the model-checking problem for finite Markov chains and PECTL*.

Probabilistic pushdown automata

In the case of pPDA, we consider only regular valuations (a valuation ν is regular if for every $a \in Ap$, the set of configurations $\nu(a)$ is regular, see Definition 4.4.1). Our results for pPDA can naturally be divided into two parts. The first part (Section 5.2) deals with the undecidability of the model-checking problem for (quanti-

tative) branching-time logics, and the second part (Section 5.3) is concerned with the decidability and complexity of the same problem for the *qualitative* fragments of branching-time logics.

In Section 5.2 we prove that the model-checking problem is undecidable for pPDA and PCTL (Section 5.2.1), and for pBPA and PCTL$^+$ (Section 5.2.2). Both these results were proved in [17] using reductions from the halting problem for two-counter machines. In this work we give more elegant and shorter proofs using reductions from Post's correspondence problem (see [43]). Note that the decidability of model-checking PCTL formulae and pBPA still remains an open problem.

In Section 5.3 we show that the model-checking problem is decidable for pPDA and the qualitative fragment of PECTL*, and we also provide **2-EXPTIME** upper bound for this problem. As a consequence of this result and Remark 5.1.1, we obtain **3-EXPTIME** upper bound on the model-checking problem for pPDA and PCTL*. The decidability of the model-checking problem for pPDA and qualitative PCTL was first proved in [27]. Here, we complete this result by showing that the problem is **EXPTIME**-complete.

5.2 Undecidability of Quantitative Model-checking

All undecidability results of this section are proved by reduction from (a slightly modified version of) Post's correspondence problem (PCP): An instance of PCP consists of two sequences $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ of words over the alphabet $\Sigma = \{A, B, \bullet\}$ such that $|x_i| = |y_i|$ for each $1 \leq i \leq n$. The question is whether there is a finite sequence $i_1, \cdots, i_k$ of indexes such that $x_{i_1} \cdots x_{i_k}$ and $y_{i_1} \cdots y_{i_k}$ are the same words after erasing all occurrences of "$\bullet$".

In order to simplify our notation in this section, we restrict ourselves to atomic propositions of the form pX (where p is a control state and X is a stack symbol of a given pPDA) and to the only valuation which assigns to each pX the simple set of all configurations with the head pX. One of the advantages is that we can write $\models$ instead of $\models^\nu$. In order to further simplify our presentation, we sometimes do not explicitly write down all transitions of a concrete pPDA which is being defined. We implicitly assume that these undefined transitions are of the form $pX \xrightarrow{1} pX$.

5.2.1 pPDA and PCTL

Let us consider an instance of PCP consisting of sequences $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ of words over the alphabet $\Sigma = \{A, B, \bullet\}$. We construct a pPDA Δ and a configuration gZ of Δ, such that $gZ \models \Diamond^{>0}(cZ \wedge \Diamond^{=1/2}(tY))$ (here cZ and tY are atomic propositions) if and only if the PCP instance has a solution.

¿From the initial configuration gZ, the automaton Δ tries to "guess" a solution to our PCP instance by storing pairs of words (x_i, y_i) successively to the stack. Since x_i and y_i have the same length, this is implemented by pushing pairs of letters from Σ. For example, if $x_i = AAB$ and $y_i = BA\bullet$, then the pair (x_i, y_i) is

stored as a sequence of three stack symbols $(A,B),(A,A),(B,\bullet)$. After storing a chosen pair of words, the automaton can either go on with guessing another pair of words, or enter a *checking* configuration. This is done by changing the control state from g to c and by pushing the symbol Z on top of the stack. The transition probabilities do not matter here, and hence we assume that these probabilities are distributed uniformly. This "generating" phase is formalized as follows (since the probability distribution is always uniform, we do not write the transition probabilities explicitly; the symbol "|" separates alternatives):

$$\begin{array}{l} gX \to g_1^1 X \mid \cdots \mid g_n^1 X, \\ g_\ell^i X \to g_\ell^{i+1}(x_\ell(i), y_\ell(i))X, \\ g_\ell^{|x_\ell|+1} X \to cZX \mid gX \end{array}$$

Here $1 \leq \ell \leq n$, $1 \leq i \leq |x_\ell|$, $x_\ell(i)$ and $y_\ell(i)$ denote i'th letters in x_ℓ and y_ℓ, respectively, X ranges over the stack alphabet. The following lemma is easily proved by induction.

Lemma 5.2.1. *$gZ \to^* cZ\alpha$ if and only if α has the form $(a_1,b_1)\cdots(a_\ell,b_\ell)Z$ where $a_1,\ldots,a_\ell,b_1,\ldots,b_\ell \in \Sigma$ and there is a sequence $i_1,\ldots,i_k$ such that $a_\ell \cdots a_1 = x_{i_1}\cdots x_{i_k}$ and $b_\ell \cdots b_1 = y_{i_1}\cdots y_{i_k}$.*

The crucial part of the construction is the next phase which verifies that the guess was correct, i.e., that the words stored in the first and the second component of stack symbols are the same (when "$\bullet$" is disregarded). For this we use the following transitions (again the transition probabilities are distributed uniformly):

$$\begin{array}{lll} cZ \to v\varepsilon \mid \hat{v}\varepsilon, & v(A,z) \to tY \mid v\varepsilon, & \hat{v}(z,A) \to rY \mid \hat{v}\varepsilon, \\ & v(B,z) \to rY \mid v\varepsilon, & \hat{v}(z,B) \to tY \mid \hat{v}\varepsilon, \\ & v(\bullet,z) \to v\varepsilon, & \hat{v}(z,\bullet) \to \hat{v}\varepsilon, \\ & vZ \to tY \mid rY, & \hat{v}Z \to tY \mid rY \end{array}$$

Here z ranges over Σ (for completeness, we define $tY \to tY$ and $rY \to rY$). We claim that the checking configuration satisfies the formula $\Diamond^{=1/2}(tY)$ if and only if the previous guess was correct. To enhance one's intuition, let us evaluate the probability of reaching a configuration with the head tY for, e.g., a configuration $cZ(A,A)(A,\bullet)(\bullet,A)(B,B)Z$. By inspecting the above rules, one can easily confirm that the probability is equal to

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

which can be written in binary as follows: $\frac{1}{2}(0.1101 + 0.0011)$. The binary numbers 0.1101 and 0.0011 are "complementary" and their sum is equal to 1. This "complementarity" breaks down if and only if the words stored in the first and the second component of stack symbols are *not* the same, in which case the probability is different from $\frac{1}{2}$. This intuition is formalized in the next lemma which is proved using similar arguments as Lemma 4.4.9. Given a word $v \in \Sigma^*$, we denote $P(v)$ the word obtained from v by erasing all occurrences of the symbol "$\bullet$".

Lemma 5.2.2. *Given $\alpha = (a_1, b_1) \cdots (a_\ell, b_\ell)Z$ where $a_1, \ldots, a_\ell, b_1, \ldots, b_\ell \in \Sigma$, we have $cZ\alpha \models \Diamond^{=\frac{1}{2}}(tY)$ if and only if $P(a_1 \cdots a_\ell) = P(b_1 \cdots b_\ell)$.*

Proof. We denote $[tY]$ the set of all configurations of $\mathcal{C}(\Delta)$ with the head tY. We show that $\mathcal{P}(cZ\alpha \rightarrow^* [tY]) = \frac{1}{2} \cdot (Num^+(P(a_1 \cdots a_\ell)) + Num^-(P(b_1 \cdots b_\ell)))$ (see Definition 4.4.7). The rest follows from Lemma 4.4.8.

Observe that $\mathcal{P}(cZ\alpha \rightarrow^* [tY]) = \frac{1}{2} \cdot (\mathcal{P}(v\alpha \rightarrow^* [tY]) + \mathcal{P}(\hat{v}\alpha \rightarrow^* [tY]))$ due to Lemma 2.2.3.

Now we show that $\mathcal{P}(v\alpha \rightarrow^* [tY]) = Num^+(P(a_1 \cdots a_\ell))$ by induction on ℓ. For $\ell = 0$, this follows immediately from the definition of Δ. Now let us assume that $\ell \geq 1$ and let us denote $\alpha' = (a_2, b_2) \cdots (a_\ell, b_\ell)Z$. There are three cases, depending on a_1. Let us consider the case where $a_1 = A$. We have $v(a_1, b_1)\alpha' \xrightarrow{\frac{1}{2}} tY\alpha'$ and $v(a_1, b_1)\alpha' \xrightarrow{\frac{1}{2}} v\alpha'$. Hence, by induction, $\mathcal{P}(v(a_1, b_1)\alpha' \rightarrow^* [tY]) = \frac{1}{2} + \frac{1}{2} \cdot Num^+(P(a_2 \cdots a_\ell)) = Num^+(P(a_1 \cdots a_\ell))$. The cases where $a_1 = B$ or $a_1 = \bullet$ are treated similarly.

One can also show, using similar arguments as above, that $\mathcal{P}(\hat{v}\alpha \rightarrow^* [tY]) = Num^-(P(b_1 \cdots b_\ell))$, which finishes the proof.

□

Putting Lemma 5.2.1 and Lemma 5.2.2 together we obtain that $gZ \models \Diamond^{>0}(cZ \wedge \Diamond^{=1/2}(tY))$ if and only if the PCP instance has a solution. Finally, observe that the subformula $\Diamond^{=1/2}(tY)$ can be substituted with a formula $\neg(\Diamond^{>1/2}(tY) \vee \Diamond^{>1/2}(rY))$.

Theorem 5.2.3. *The model-checking problem for pPDA and PCTL is undecidable. Moreover, the undecidability result holds even for the fragment of PCTL where the nesting depth of $\mathcal{U}$ is at most two, and for all subformulae of the form $\mathcal{P}^{\sim\varrho}\varphi$ we have that $\sim = >$.*

Note that the above proof can be modified, using a similar trick as in the proof of Theorem 4.4.11, to allow arbitrary formula of the form $\Diamond^{=\varrho}(tY)$, where $0 < \varrho < 1$, instead of $\Diamond^{=1/2}(tY)$ (it suffices to appropriately change the transitions outgoing from cZ).

5.2.2 pBPA and PCTL$^+$

As in the previous subsection, we consider an instance of PCP consisting of sequences $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$. We construct a pBPA Δ, a configuration Z of Δ, and a PCTL$^+$ state formula Φ such that $Z \models \Diamond^{>0}(\Phi)$ if and only if the PCP instance has a solution. Similarly as in the previous subsection, Δ works in two phases. In the first phase, the pBPA Δ tries to guess a solution to the PCP instance by storing pairs of words (x_i, y_i) to the stack. This phase is formalized by the following rules:

$$\begin{aligned} &Z \rightarrow G^1_1 Z \mid \cdots \mid G^1_n Z, \\ &G^i_\ell \rightarrow G^{i+1}_\ell (x_\ell(i), y_\ell(i)), \\ &G^{|x_\ell|+1}_\ell \rightarrow C \mid G^1_1 \mid \cdots \mid G^1_n. \end{aligned}$$

where $1 \leq \ell \leq n$ and $1 \leq i \leq |x_\ell|$. The following lemma justifies that the above rules work as expected.

Lemma 5.2.4. *$Z \to^* C\alpha$ if and only if α has the form $(a_1, b_1) \cdots (a_\ell, b_\ell)Z$ where $a_1, \ldots, a_\ell, b_1, \ldots, b_\ell \in \Sigma$ and there is a sequence $i_1, \ldots, i_k$ such that $a_\ell \cdots a_1 = x_{i_1} \cdots x_{i_k}$ and $b_\ell \cdots b_1 = y_{i_1} \cdots y_{i_k}$.*

The second phase differs slightly from the second phase for pPDA and PCTL. Observe that in pPDA, we used control states to distinguish between runs checking the first component of stack symbols and those checking the second component. As there is only one control state in pBPA, this trick can no longer be used, and runs have to be distinguished by the formula Φ. The following transitions perform the second phase:

$$\begin{array}{ll} C \to V \mid \hat{V}, & (z, z') \to X_{(z,z')} \mid \varepsilon, \\ V \to \varepsilon, & Z \to X_{(A,B)} \mid X_{(B,A)}, \\ \hat{V} \to \varepsilon & X_{(z,z')} \to \varepsilon \end{array}$$

Here z and z' range over Σ. We define

$$\varphi_1 = \left(\neg \hat{V} \wedge \bigwedge_{z \in \Sigma} (\neg X_{(A,z)} \wedge \neg X_{(B,z)}) \right) \mathcal{U} \left(\bigvee_{z \in \Sigma} X_{(A,z)} \right)$$

$$\varphi_2 = \left(\neg V \wedge \bigwedge_{z \in \Sigma} (\neg X_{(z,A)} \wedge \neg X_{(z,B)}) \right) \mathcal{U} \left(\bigvee_{z \in \Sigma} X_{(z,B)} \right)$$

and $\Phi = \mathcal{P}^{=\frac{1}{2}}(\varphi_1 \vee \varphi_2)$.

Lemma 5.2.5. *Given $\alpha = (a_1, b_1) \cdots (a_\ell, b_\ell)Z$ where $a_1, \ldots, a_\ell, b_1, \ldots, b_\ell \in \Sigma$, we have $C\alpha \models \Phi$ if and only if $P(a_1 \cdots a_\ell) = P(b_1 \cdots b_\ell)$.*

Proof. We prove that

$$\mathcal{P}(C\alpha, \varphi_1 \vee \varphi_2) = \frac{1}{2} \cdot (Num^+(P(a_1 \cdots a_\ell)) + Num^-(P(b_1 \cdots b_\ell)))$$

The rest follows from Lemma 4.4.8.

First, we show, by induction on ℓ, that $\mathcal{P}(\alpha, \varphi_1) = Num^+(P(a_1 \cdots a_\ell))$ (see Definition 4.4.7). If $\ell = 0$, then the result follows immediately from definitions. Let us assume that $\ell \geq 1$, and denote $\alpha' = (a_2, b_2) \cdots (a_\ell, b_\ell)Z$. By definition, $\alpha \xrightarrow{\frac{1}{2}} X_{(a_1,b_1)}\alpha' \xrightarrow{1} \alpha'$ and $\alpha \xrightarrow{\frac{1}{2}} \alpha'$. We have the following consequences of Lemma 2.2.3 and the induction hypothesis: If $a_1 = \bullet$, then $\mathcal{P}(\alpha, \varphi_1) = \mathcal{P}(\alpha', \varphi_1) = Num^+(P(a_2 \cdots a_\ell)) = Num^+(P(a_1 \cdots a_\ell))$. If $a_1 = A$, then $\mathcal{P}(\alpha, \varphi_1) = \frac{1}{2} + \frac{1}{2} \cdot \mathcal{P}(\alpha', \varphi_1) = \frac{1}{2} + \frac{1}{2} \cdot Num^+(P(a_2 \cdots a_\ell)) = Num^+(P(a_1 \cdots a_\ell))$. Finally, if $a_1 = B$, then $\mathcal{P}(\alpha, \varphi_1) = \frac{1}{2} \cdot \mathcal{P}(\alpha', \varphi_1) = Num^+(P(a_1 \cdots a_\ell))$.

Similarly, one can show that $\mathcal{P}(\alpha, \varphi_2) = Num^-(P(b_1 \cdots b_\ell))$. Now observe that the sets of runs of $Run(C\alpha)$ satisfying φ_1 and φ_2, respectively, are disjoint, and thus $\mathcal{P}(C\alpha, \varphi_1 \vee \varphi_2) = \frac{1}{2} \cdot (\mathcal{P}(C\alpha, \varphi_1) + \mathcal{P}(C\alpha, \varphi_2)) = \frac{1}{2} \cdot (\mathcal{P}(\alpha, \varphi_1) + \mathcal{P}(\alpha, \varphi_2)) = \frac{1}{2} \cdot (Num^+(P(a_1 \cdots a_\ell)) + Num^-(P(b_1 \cdots b_\ell)))$. □

Putting Lemma 5.2.4 and Lemma 5.2.5 together we obtain that $Z \models \Diamond^{>0}(\Phi)$ if and only if the PCP instance has a solution. Finally, note that Φ is equivalent to $\neg(\mathcal{P}^{>1/2}\varphi \vee \mathcal{P}^{>1/2}\neg\varphi)$ where $\varphi \equiv \varphi_1 \vee \varphi_2$.

Theorem 5.2.6. *The model-checking problem for pBPA and PCTL$^+$ is undecidable. More precisely, the undecidability result holds even for a fragment of PCTL$^+$ where the nesting depth of $\mathcal{U}$ is at most two, and for all subformulae of the form $\mathcal{P}^{\sim\varrho}\varphi$ we have that $\sim = >$.*

5.3 Qualitative Model-checking

We start by showing that the model checking problem for pPDA and qPECTL* is in **2-EXPTIME**. Let us fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$. Our model-checking algorithm for pPDA follows similar pattern as the algorithm for finite Markov chains outlined in Section 5.1. The main difference is that the sets of configurations satisfying particular subformulae of a given formula does not have to be finite. However, as we show, these sets are always effectively regular.

Lemma 5.3.1. *Let $\mathcal{R} = (2^{\{1,\ldots,n\}}, R, \rho, s_I, F)$ be a deterministic Rabin automaton, let $\mathcal{A}_1, \ldots, \mathcal{A}_n$ be DFA with the alphabet $Q \cup \Gamma$ and sets of states $K_1, \ldots, K_n$, respectively. Let $\nu : \mathcal{C}(\Delta) \to 2^{\{1,\ldots,n\}}$ be a valuation such that for every $p\alpha \in \mathcal{C}(\Delta)$ we have $\nu(p\alpha) = \{1 \leq i \leq n \mid p\alpha \in \mathcal{C}(\mathcal{A}_i)\}$. If $\mathbf{S}_\Delta$ is known in advance, then for arbitrary $\sim \in \{<, >, =, \leq, \geq\}$ and $\varrho \in \{0, 1\}$, a DFA accepting $L_{\sim\varrho} = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{P}(p\alpha, \mathcal{R}, \nu) \sim \varrho\}$ is computable in time $2^{\mathcal{O}(|Q|\cdot|R|)} \cdot (|\mathcal{R}| \cdot |\Delta| \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$. The number of states of the resulting automaton is equal to $2^{|Q|\cdot|R|} \cdot \prod_{i=1}^{n} |K_i|$.*

Proof. Throughout this proof we use the notation and results introduced in Section 4.5. Let $\mathcal{R}'$ be a Rabin automaton obtained from $\mathcal{R}$ by substituting each transition of the form $s \xrightarrow{A} s'$ (here $A \in 2^{\{1,\ldots,n\}}$) with all transitions of the form $s \xrightarrow{h} s'$ where $h \in \bigcup_{k\in A} \mathcal{H}[\mathcal{A}_k] \setminus \bigcup_{k\notin A} \mathcal{H}[\mathcal{A}_k]$. Let ν' be the valuation which assigns to each configuration of $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ its head. It follows from Lemma 4.5.2 that for all $p\alpha \in \mathcal{C}(\Delta)$, we have $\mathcal{P}(p\alpha, \mathcal{R}, \nu) = \mathcal{P}(\mathcal{K}(p\alpha), \mathcal{R}', \nu')$ (here $\mathcal{K}$ is from Lemma 4.5.2).

A DFA accepting the set $L_{\sim\varrho}$ can be computed using the following procedure:

1. Compute $\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]$ together with the sets $\mathcal{H}[\mathcal{A}_1], \ldots, \mathcal{H}[\mathcal{A}_n]$;
2. compute the Rabin automaton $\mathcal{R}'$;
3. using Theorem 4.4.5, compute a DFA $\mathcal{B}$ such that
$$\mathcal{C}(\mathcal{B}) = \{q\beta \in \mathcal{C}(\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]) \mid \mathcal{P}(q\beta, \mathcal{R}', \nu') \sim \varrho\}$$
4. using Lemma 4.5.3, compute a DFA $\mathcal{B}'$ such that
$$\mathcal{C}(\mathcal{B}') = \{p\alpha \in \mathcal{C}(\Delta) \mid \mathcal{K}(p\alpha) \in \mathcal{C}(\mathcal{B})\} = L_{\sim\varrho}$$

We analyze the complexity of the above procedure (we use the fact that $n \leq |\mathcal{R}|$):

1. By Lemma 4.5.2, this step can be treated in time $(|\Delta| \cdot |\mathcal{R}| \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$.
2. $\mathcal{R}'$ can be computed in time $(|\mathcal{R}| \cdot |\Delta| \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$.
3. By Theorem 4.4.5, the DFA $\mathcal{B}$ can be computed in time

$$2^{\mathcal{O}(|Q| \cdot |R|)} \cdot \left(|\mathcal{R}| \cdot |\Delta| \cdot \prod_{i=1}^{n} |K_i| \right)^{\mathcal{O}(1)}$$

   Here we use the fact that the set $\mathbf{S}_{\Delta[\mathcal{A}_1,\ldots,\mathcal{A}_n]}$ can be computed in time polynomial in $|\Delta[\mathcal{A}_1, \ldots, \mathcal{A}_n]|$ (i.e., in time $(|\Delta| \cdot \prod_{i=1}^{n} |K_i|)^{\mathcal{O}(1)}$) once the set $\mathbf{S}_\Delta$ is given. The number of states of $\mathcal{B}$ is equal to $2^{|Q| \cdot |R|}$.
4. By Lemma 4.5.3, the DFA $\mathcal{B}'$ can be computed in time

$$2^{\mathcal{O}(|Q| \cdot |R|)} \cdot \left(|\Delta| \cdot |\mathcal{R}| \cdot \prod_{i=1}^{n} |K_i| \right)^{\mathcal{O}(1)}$$

   and the number of states of $\mathcal{B}'$ is equal to $2^{|Q| \cdot |R|} \cdot \prod_{i=1}^{n} |K_i|$.

Hence, the overall complexity of the above procedure is as claimed. □

In what follows we describe an algorithm, which for a given qPECTL* formula τ computes a DFA accepting the set of all configurations that satisfy τ. We also provide a detailed complexity analysis of this algorithm. Let us fix a qPECTL* formula τ and a valuation ν. Let us assume that for each atomic proposition $a \in Ap$ which occurs in τ, the set $\nu(a)$ is accepted by a given DFA $\mathcal{A}_a$.

Let $\varphi_1, \ldots, \varphi_\ell$ be all subformulae of τ ordered in such a way, that for $j < i$ the formula φ_i is *not* a subformula of φ_j. Clearly $\varphi_\ell = \tau$. Given a subformula φ_i, we denote $\mathcal{C}(\varphi_i)$ the set of all configurations that satisfy φ_i. In particular, $\mathcal{C}(\tau)$ is the set of all configurations that satisfy τ.

We propose an algorithm which successively computes a sequence of DFA $\mathcal{A}_1, \ldots, \mathcal{A}_\ell$ accepting $\mathcal{C}(\varphi_1), \ldots, \mathcal{C}(\varphi_\ell)$, respectively. We denote $K_1, \ldots, K_\ell$ the sets of states of DFA $\mathcal{A}_1, \ldots, \mathcal{A}_\ell$, respectively. Let us assume that $\mathcal{A}_1, \ldots, \mathcal{A}_{i-1}$ has already been computed. Let us consider φ_i.

- If φ_i is an atomic proposition, then we put $\mathcal{A}_i = \mathcal{A}_{\varphi_i}$.
- If $\varphi_i = \varphi_j \wedge \varphi_k$ for some $j, k < i$, then a standard construction from automata theory yields a DFA $\mathcal{A}_i$ such that $\mathcal{L}(\mathcal{A}_i) = \mathcal{L}(\mathcal{A}_j) \cap \mathcal{L}(\mathcal{A}_k)$ and whose number of states is equal to $|K_j| \cdot |K_k|$. Clearly, $\mathcal{C}(\mathcal{A}_i) = \mathcal{C}(\varphi_j \wedge \varphi_k)$.
- If $\varphi_i = \neg\varphi_j$ for some $j < i$, then by switching accepting and non-accepting states in $\mathcal{A}_j$, we obtain the automaton $\mathcal{A}_i$ accepting the complement of $\mathcal{L}(\mathcal{A}_j)$ (and hence $\mathcal{C}(\mathcal{A}_i) = \mathcal{C}(\neg\varphi_j)$).

- Now assume that φ_i is of the form $\mathcal{P}^{\sim\varrho}\mathcal{B}(\varphi_{i_1},\ldots,\varphi_{i_n})$ where $i_1,\ldots,i_n < i$ and $\mathcal{B}$ has the alphabet $2^{\{1,\ldots,n\}}$. Let $\mathcal{R}$ be the Rabin automaton obtained from $\mathcal{B}$ using the procedure of Theorem 4.3.17. We apply Lemma 5.3.1 to $\mathcal{R}$ and DFA $\mathcal{A}_{i_1},\ldots,\mathcal{A}_{i_n}$, and obtain a DFA $\mathcal{A}_i$ such that $\mathcal{C}(\mathcal{A}_i) = \mathcal{C}(\varphi_i)$. The complexity of this step is analyzed below.

Complexity: In what follows, subformulae of τ of the form $\mathcal{P}^{\sim\varrho}\mathcal{B}(\varphi_{i_1},\ldots,\varphi_{i_n})$ are called *temporal*. Given i, we denote $Sub[i], At[i] \subseteq \{1,\ldots,i\}$ the sets of indexes of all temporal and atomic subformulae of φ_i, respectively. For every i such that φ_i is of the form $\mathcal{P}^{\sim\varrho}\mathcal{B}(\varphi_{i_1},\ldots,\varphi_{i_n})$, we denote $\mathcal{R}_i$ the Rabin automaton obtained from $\mathcal{B}$ using the procedure of Theorem 4.3.17 (we denote R_i the set of states of $\mathcal{R}_i$). Observe that $|\mathcal{R}_i|$ is in $2^{\mathcal{O}(|\tau|\log|\tau|)}$), due to Theorem 4.3.17.

Now a straightforward induction using Lemma 5.3.1 reveals that for all $1 \leq i \leq \ell$ the number of states of $\mathcal{A}_i$ is equal to $\prod_{j\in Sub[i]} 2^{|Q|\cdot|R_j|} \cdot \prod_{j\in At[i]} |K_j|$. Hence, given $\mathbf{S}_\Delta$, the time complexity of computing the DFA $\mathcal{A}_i$ for the temporal subformula φ_i of the form $\mathcal{P}^{\sim\varrho}\mathcal{B}(\varphi_{i_1},\ldots,\varphi_{i_n})$ is in

$$2^{\mathcal{O}(|Q|\cdot|R_i|)} \cdot \left(|\Delta|\cdot|\mathcal{R}_i|\cdot\prod_{m=1}^{n}|K_{i_m}|\right)^{\mathcal{O}(1)} \subseteq$$

$$2^{\mathcal{O}(|Q|\cdot|R_i|)} \cdot \left(|\Delta|\cdot|\mathcal{R}_i|\cdot\prod_{m=1}^{n}\left(\prod_{j\in Sub[i_m]} 2^{\mathcal{O}(|Q|\cdot|R_j|)}\cdot\prod_{j\in At[i_m]}|K_j|\right)\right)^{\mathcal{O}(1)} \subseteq$$

$$\prod_{j\in Sub[i]} 2^{\mathcal{O}(|Q|\cdot|R_j|)} \cdot \left(|\Delta|\cdot|\mathcal{R}_i|\cdot\prod_{j\in At[i]}|K_j|\right)^{\mathcal{O}(1)} \subseteq$$

$$2^{|\Delta|\cdot 2^{\mathcal{O}(|\tau|\log|\tau|)}} \cdot \left(\prod_{j\in At[i]}|\mathcal{A}_j|\right)^{\mathcal{O}(1)}$$

This formula expresses the time complexity of one iteration of the above algorithm (for one subformula φ_i). However, in order to compute the automaton $\mathcal{A}_\ell$ accepting the set of configurations $\mathcal{C}(\tau)$, the procedure has to be iterated at most $\ell = \mathcal{O}(|\tau|)$ times. Because the set $\mathbf{S}_\Delta$ can be computed in polynomial space (and hence in time exponential in $|\Delta|$) due to Lemma 4.3.8, it follows that the overall time complexity is in $2^{|\Delta|\cdot 2^{\mathcal{O}(|\tau|\log|\tau|)}} \cdot (\prod_{j\in At[\ell]}|\mathcal{A}_j|)^{\mathcal{O}(1)}$. In other words, a DFA accepting $\mathcal{C}(\tau)$ is computable in time doubly exponential in $|\tau|$, singly exponential in $|\Delta|$, and polynomial in $\prod_{j\in At[\ell]}|\mathcal{A}_j|$. Observe also that if Δ is a pBPA, then one iteration of the above algorithm runs in time polynomial in $|\Delta|$. Hence, in the case of pBPA,

the set $\mathcal{C}(\tau)$ is computable in time polynomial in $|\Delta|$, because $\mathbf{S}_\Delta$ is computable in polynomial time by Lemma 4.3.8.

Now let us assume that τ is a qPCTL formula. Note that each of the Rabin automata $\mathcal{R}_i$ is either the automaton corresponding to the $\mathcal{X}$ operator, or the automaton corresponding to the $\mathcal{U}$ operator. Hence there are, in fact, only two Rabin automata, whose size is fixed, and hence constant. The above time complexity estimate for one step computing $\mathcal{A}_i$, reduces to $2^{\mathcal{O}(|\Delta|\cdot|\tau|)} \cdot (\prod_{j \in At[\ell]} |\mathcal{A}_j|)^{\mathcal{O}(1)}$. Thus, if τ is a qPCTL formula, then a DFA accepting $\mathcal{C}(\tau)$ is computable in time singly exponential in $|\Delta| \cdot |\tau|$.

Finally, let us assume that τ is a qPCTL* formula. In order to compute a DFA accepting $\mathcal{C}(\tau)$, we use the procedure outlined in Remark 5.1.1 to translate the formula τ to qPECTL* (with a singly exponential blowup in its size). Then apply the above procedure to the resulting qPECTL* formula and compute the desired DFA. Thus, if τ is a qPCTL* formula, then a DFA accepting $\mathcal{C}(\tau)$ is computable in time triply exponential in $|\tau|$, and singly exponential in $|\Delta|$.

Because the membership problem is easily decidable (in polynomial time) for DFA, we obtain the following theorem.

Theorem 5.3.2. *The model-checking problem for pPDA and* qualitative *PCTL, PECTL*, PCTL* is in **EXPTIME**, **2-EXPTIME**, **3-EXPTIME**, respectively. Moreover, for pBPA, this model-checking problem is in **P** in the size of pBPA.*

Lower Bound

Finally, let us note that the construction presented in [52] which shows **EXPTIME**-hardness of the model-checking problem for non-probabilistic PDA and CTL can be adapted so that it works for (non-probabilistic) BPA[1]. This idea carries over to the probabilistic case after some trivial modifications. Thus, we obtain the following:

Theorem 5.3.3. *The model-checking problem for pBPA and* qualitative *PCTL is **EXPTIME**-complete.*

Proof. We reduce the acceptance problem for alternating LBA (which is known to be **EXPTIME**-complete [42]). An *alternating LBA* is a tuple $\mathcal{T} = (T, \Sigma, \Lambda, t_0, \vdash, \dashv, \zeta, P)$ where T is a finite set of *control states*, Σ is a finite *input alphabet*, $\Lambda \supseteq \Sigma$ is a finite *tape alphabet*, $t_0 \in T$ is the *initial* control state, $\vdash, \dashv \in \Lambda$ are the left-end and the right-end markers, $\zeta : T \times \Lambda \to 2^{T \times \Lambda \times \{left, right\}}$ is a *transition function*, and $P = (T_\forall, T_\exists, T_{acc}, T_{rej})$ is a partition of T into *universal*, *existential*, *accepting*, and *rejecting* states, respectively. We can safely assume that $T \cap \Lambda = \emptyset$, $t_0 \in T_\exists$, $\zeta(t, a) = \emptyset$ for all $t \in T_{acc} \cup T_{rej}$, and that $\zeta(t, a)$ has exactly two elements (t_1, a_1, D_1), (t_2, a_2, D_2), where $t_1 \neq t_2$, for all $t \in T_\forall \cup T_\exists$. A *computational tree* for $\mathcal{T}$ on a word $u \in \Sigma^*$ is a tree Tr satisfying the following: the root of Tr is

[1] This observation is due to Mayr (Private communication, July 2004.)

(labeled by) the initial configuration for u, and if N is a node of Tr labeled by a configuration with a control state t, then the following holds:

- if t is accepting or rejecting, then N is a leaf;
- if t is existential, then N has one successor labeled by a configuration reachable from the configuration of N in one step.
- if t is universal, then N has two successors labeled by the two configurations reachable from the configuration of N in one step.

$\mathcal{T}$ *accepts* u iff there is a finite computational tree Tr such that all leaves of Tr are accepting configurations. We can safely assume that *all* computational trees for $\mathcal{T}$ are finite.

Let $\mathcal{T} = (T, \Sigma, \Lambda, t_0, \vdash, \dashv, \zeta, P)$ be an alternating LBA and $u \in \Sigma^*$ an input word. We construct (in polynomial time) a pBPA Δ, a configuration α of Δ, and a qPCTL formula φ, such that $\mathcal{T}$ accepts u if and only if $\alpha \models \varphi$.

Configurations of $\mathcal{T}$ are written as words over the alphabet $\Xi = T \cup \Lambda$ in the standard way; for example, the initial configuration for u is written as $t_0 \vdash u \dashv$. Configurations with control states in $T_\forall$ ($T_\exists$) are called universal (existential). In our proof we have to distinguish between the two successors of each universal configuration. Hence, we fix an ordering on the set T of control states, which naturally distinguishes the left-hand side successor (with the lesser control state) and the right-hand side successor of each universal configuration.

Using a standard algorithm, one can efficiently compute the sets $\mathcal{S}_L(\mathcal{T}), \mathcal{S}_R(\mathcal{T}), \mathcal{S}_E(\mathcal{T}) \subseteq \Xi^6$ of *compatible 6-tuples* such that for each configuration c (written as a word over Ξ) and each $c' \in \Xi^*$ of the same length as c, we have that

- c is universal and c' is the left-hand side (one-step) successor of c if and only if $(c(i), c(i{+}1), c(i{+}2), c'(i), c'(i{+}1), c'(i{+}2)) \in \mathcal{S}_L(\mathcal{T})$ for all $1 \leq i \leq |c|{-}2$;
- c is universal and c' is the right-hand side successor of c if and only if $(c(i), c(i{+}1), c(i{+}2), c'(i), c'(i{+}1), c'(i{+}2)) \in \mathcal{S}_R(\mathcal{T})$ for all $1 \leq i \leq |c|{-}2$;
- c is an existential configuration and c' is one of the successors of c if and only if $(c(i), c(i{+}1), c(i{+}2), c'(i), c'(i{+}1), c'(i{+}2)) \in \mathcal{S}_E(\mathcal{T})$ for all $1 \leq i \leq |c|{-}2$;

Let $n = |u|{+}3$. The stack alphabet of Δ is equal to $\Lambda \cup T \cup \{G, M, A, F, L, E, R\}$, where G, M, A, F, L, E, R are special symbols that are not in $\Lambda \cup T$. Transitions of Δ are given by the following rules (the probability distribution over transitions is not significant, we only need that each transition is assigned a non-zero proba-

bility).

$$\begin{array}{lcl} G & \to & McL \\ G & \to & McE \\ G & \to & A \end{array} \qquad \begin{array}{lcl} L & \to & McR \\ L & \to & F \\ E & \to & \varepsilon \\ E & \to & F \\ R & \to & \varepsilon \\ R & \to & F \end{array}$$

$$\begin{array}{lcl} M & \to & F \\ M & \to & G \end{array} \qquad \begin{array}{lcl} t & \to & \varepsilon \\ a & \to & \varepsilon \\ A & \to & \varepsilon \\ F & \to & \varepsilon \end{array}$$

where $a \in \Lambda$, $t \in T$, and c stands for a configuration of $\mathcal{T}$ (a string of length n). Here $G \to McL$, $G \to McE$, $L \to McR$ are in fact abbreviations for a family of transitions that guess a new configuration c on the top of the stack symbol by symbol.

We define $\alpha = Gt_0 \vdash u \dashv$. The simulation works as follows: It begins in the configuration α and starts guessing a computation tree for $\mathcal{T}$ on u. Each of the configurations is guessed in n steps, because we need Δ to be polynomial in the size of $\mathcal{T}$. Each time a new configuration is guessed, the run of Δ has to pass through a configuration with M on the top of the stack, and the correctness is checked by the formula $Move$ (defined below). When an accepting configuration is guessed (by putting A on the stack), and checked (by the formula $Accept$), all the symbols up to L are erased from the stack, and guessing of the right branch of the corresponding universal move is started. This continues until the stack becomes empty. The special symbol F is used to distinguish runs that perform the above described simulation from those that are used to check correctness of guesses by the formula $Move$.

Now we describe the formula φ in detail. First, let us introduce some shorthands that will simplify our notation: Because all subformulae of φ are either Boolean combinations, or of the form $\mathcal{P}^{>0}\psi$, we leave out the symbol $\mathcal{P}^{>0}$ in explicitly written formulae. Hence, we write $\psi_1 \,\mathcal{U}\, \psi_2$ and $\mathcal{X}\psi$ instead of $\mathcal{P}^{>0}(\psi_1 \,\mathcal{U}\, \psi_2)$ and $\mathcal{P}^{>0}\mathcal{X}\psi$, respectively.

We put

$$\varphi \equiv \big(\neg F \wedge (A \Rightarrow Accept) \wedge (M \Rightarrow Move)\big) \,\mathcal{U}\, \varepsilon$$

The subformula $Accept$ checks whether the configuration stored on the top of the stack is accepting:

$$Accept \equiv \bigvee_{t \in T_{acc}} (\neg E \wedge \neg L \wedge \neg R) \,\mathcal{U}\, t$$

The subformula $Move$ checks whether the configuration, stored on the top of the stack, is a correct successor of the preceding one:

$$Move \equiv \mathcal{X}\left(F \wedge \bigvee_{K\in\{L,R,E\}}\left((\mathcal{X}^{n+1}K) \wedge \bigwedge_{i=1}^{n-2}\psi_{K,i}\right)\right)$$

Here $\mathcal{X}^{n+1}K$ denotes the formula obtained from K by $n+1$ applications of the operator $\mathcal{X}$, and the formulae $\psi_{K,i}$ check the consistence of the successive transitions using the compatible 6-tuples:

$$\psi_{K,i} \equiv \bigvee_{(X_1,X_2,X_3,Y_1,Y_2,Y_3)\in\mathcal{S}_K(\mathcal{T})} Pos(i,X_1,X_2,X_3,Y_1,Y_2,Y_3)$$

where $Pos(i,X_1,X_2,X_3,Y_1,Y_2,Y_3)$ stands for

$$\mathcal{X}^i\big(Y_1 \wedge (\mathcal{X}\,Y_2) \wedge (\mathcal{X}^2\,Y_3)\big) \wedge \mathcal{X}^{n+2}\big(F \wedge \mathcal{X}^i\big(X_1 \wedge (\mathcal{X}\,X_2) \wedge (\mathcal{X}^2\,X_3)\big)\big)$$

It is easy to verify that $\alpha = Gt_0 \vdash u \dashv$ satisfies φ if and only if $\mathcal{T}$ accepts u. □

Chapter 6

Expected Behavior

Our object of study in this chapter is the (conditional) expected accumulated reward. This value can be intuitively described as follows: Let us assume that each state of a given Markov chain is assigned a number (a reward). Whenever a run of the chain enters a state, the reward assigned to this state is collected. Given a state s and a set of states L, we define a random variable which assigns to each run of $Run(s)$ the reward accumulated along this run before it enters L for the first time (runs that do not enter L are assigned 0). The expected accumulated reward is the expectation of this variable under the condition that L is reached. This value can be seen as a generalization of the mean time for reaching a given set of states (see, e.g., [41]) and as such provides some additional information to the reachability problem. One can, for example, verify not only that a procedure terminates almost surely but also that the mean time to the termination is less than a given limit.

In this chapter we give a precise definition of the expected accumulated reward, and show how it can effectively be analyzed for pPDA and so called well-defined reward functions. To achieve this goal, we prove that the expected accumulated reward is effectively expressible in $ExTh(\mathbb{R})$. Results of this chapter are heavily used in the next chapter, especially in Section 7.3.

6.1 Expected Accumulated Reward

Let $M = (S, \to, Prob)$ be a Markov chain and let $f : S \to \mathbb{R}^+$ be a non-negative *reward function*. The function f extends to paths as follows: for every path v we define $f(v) = \sum_{i=0}^{|v|} f(v(i))$. In what follows we denote $\mathbf{1} : S \to \mathbb{R}^+$ the reward function which assigns 1 to all states.

Given a set of states $L \subseteq S$ and a state $s \in S \backslash L$, we define a random variable $R^f_{s,L} : Run(s) \to \mathbb{R}^+$ as follows:

$$R^f_{s,L}(w) = \begin{cases} f(w^{\ell-1}) & w(\ell) \in L \text{ and } \forall j < \ell : w(j) \notin L; \\ 0 & \forall j \geq 0 : w(j) \notin L. \end{cases}$$

Clearly, $R^f_{s,L}$ is a non-negative discrete random variable. If $Prob(s \to^* L) > 0$, then $E(R^f_{s,L} \mid Run(s \to^* L))$ is the (conditional) *expected accumulated reward* along a path from s (inclusive) to a state of L (not inclusive) under the condition that L is reached. We write $[E(s \to^* L), f]$ instead of $E(R^f_{s,L} \mid Run(s \to^* L))$. (For $t \in S$, we write $[E(s \to^* t), f]$ instead of $[E(s \to^* \{t\}), f]$.) We are interested in the following problem:

• *Expected accumulated reward problem*:

Given $\varrho \in [0,1]$ and $\sim \in \{<, >, =, \leq, \geq\}$, is $[E(s \to^* L), f] \sim \varrho$?

Finite Markov chains

Similarly to the previous chapters, we outline a solution of the expected accumulated reward problem for finite Markov chains because it nicely motivates the solution for pPDA. Let us fix a set L of states of a finite Markov chain $M = (S, \to, Prob)$ and a (rational valued) reward function $f : S \to \mathbb{R}^+$. We show that the tuple of all $[E(s \to^* L), f]$ values is the unique solution of an effectively computable system of linear equations in $\mathbb{R}$.

Let $\bar{S}$ be the set of all states of $S \backslash L$ that satisfy $\mathcal{P}(s \to^* L) > 0$, and let us assume, without the loss of generality, that all states t of $S \backslash \bar{S}$ are absorbing (i.e., $t \xrightarrow{1} t$). Moreover, let us denote $S^{>0} \subseteq \bar{S}$, the set of all states $s \in \bar{S}$ that satisfy $[E(s \to^* L), f] > 0$, and let $S^{=0} = \bar{S} \backslash S^{>0}$. It is easy to show that $S^{>0}$ consists of all states $s \in \bar{S}$ that satisfy the following condition: there is a path from s to L through a state $t \notin L$ such that $f(t) > 0$. Hence, both sets $S^{>0}$ and $S^{=0} = \bar{S} \backslash S^{>0}$ can be computed in polynomial time using standard techniques of the graph theory.

To every state $s \in \bar{S}$ we associate a real variable x_s. Consider the following system of equations: For $s \in S^{=0}$, we put $x_s = 0$. Otherwise, for $s \in S^{>0}$, we put

$$x_s = \frac{1}{\mathcal{P}(s \to^* L)} \cdot \left(\sum_{s \xrightarrow{y} t, t \in L} y \cdot f(s) + \sum_{s \xrightarrow{y} t, t \in \bar{S}} y \cdot \mathcal{P}(t \to^* L) \cdot (f(s) + x_t) \right)$$

By Lemma 6.2.4 below (see also Remark 6.2.7), the tuple of all $[E(s \to^* L), f]$ values is the least solution of the above system in $\mathbb{R}^+_\infty$. However, we show that the values $[E(s \to^* L), f]$ are always finite (unlike the case of Markov chains generated by pPDA, see Example 6.2.6 below). Intuitively, we use the fact that the expected time spent in transient states (i.e., states that do not belong to a bottom strongly connected component) is finite which is proved, e.g., in [38].

It follows from our assumptions and Proposition 4.3.2 that $\mathcal{P}(s \to^* S \backslash \bar{S}) = 1$ for each $s \in \bar{S}$ because BSCCs of M are precisely all singleton sets containing states of $S \backslash \bar{S}$. Moreover, by [38] (Theorem 3.2.4), the expected number of steps for reaching $S \backslash \bar{S}$ from s is finite. In our notation, $[E(s \to^* S \backslash \bar{S}), \mathbf{1}] < \infty$. However,

then because $L \subseteq S \backslash \bar{S}$, we have

$$\begin{aligned}[E(s \to^* L), f] &\leq \max(f) \cdot [E(s \to^* L), \mathbf{1}] \\ &\leq \frac{\max(f)}{\mathcal{P}(s \to^* L)} \cdot [E(s \to^* S \backslash \bar{S}), \mathbf{1}] < \infty\end{aligned}$$

Here $\max(f) = \max\{f(t) \mid t \in S\}$.

Thus, as a consequence of Lemma 6.2.4 (and Remark 6.2.7), we obtain that the tuple of all $[E(s \to^* L), f]$ values is the *unique* solution of the above system in $\mathbb{R}$. Hence, the values $[E(s \to^* L), f]$ are rational and computable in polynomial time.

Probabilistic pushdown automata

In the case of pPDA, we consider the expected accumulated reward problem only for simple sets L (see Definition 4.1.2). Note that all results of this chapter can easily be generalized to deal with regular sets using similar arguments as for the reachability (see Section 4.5).

It is easy to show that in the case of pPDA the expected accumulated reward problem is undecidable for general reward functions. Thus we restrict ourselves to the subclass of well-defined reward functions defined as follows. Let us fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$.

Definition 6.1.1. *A reward function $f : \mathcal{C}(\Delta) \to \mathbb{R}^+$ is* well-defined *if there are non-negative functions $g, h : Q \to \mathbb{R}^+$ and $c : \Gamma \to \mathbb{R}^+$ such that $f(p\alpha) = g(p) + h(p) \cdot (\sum_{Y \in \Gamma} c(Y) \cdot \#_Y(\alpha))$ for all $p\alpha \in \mathcal{C}(\Delta)$, where $\#_Y(\alpha)$ denotes the number of occurrences of Y in α. We say that f is* simple *(or* linear*) if and only if $h(p) = 0$ (or $h(p) = 1$, resp.) for all $p \in Q$.*

In the rest of this work we use $c(\alpha)$ to denote $\sum_{Y \in \Gamma} c(Y) \cdot \#_Y(\alpha)$. Sometimes we abuse our notation by considering g and h as stand-alone simple reward functions. We define the size $|f|$ of the well-defined reward function f to be the sum of sizes of binary representations of all values of functions g, h, and c (here we assume that these values are rational).

Simple reward functions can model gains and costs which do not depend on the history of activation records (i.e., the stack of procedure calls that have not terminated yet). A simple example is execution time—one can reasonably assume that the time spent in a given procedure for given input data does not depend on the current stack of activation records. On the other hand, if one is interested in, e.g., memory consumptions, then the total amount of allocated memory in a given configuration does depend on the amount of memory allocated in the individual procedures stored in the stack, and here one can use linear reward functions. The main reason why we also introduced the function h in Definition 6.1.1 is that in certain situations we wish not to "count" some configurations. For example, if we want to model an unbounded integer variable which is used in a given procedure, we might encode its value in unary by pushing a special symbol to the stack.

Bounded changes to the variable (such as increment or decrement) can easily be implemented as single pPDA transitions. However, unbounded changes such as setting the variable back to 1 cannot be modeled as a single pPDA transition—the previously pushed symbols must be removed one by one. The artificially-added intermediate configurations can influence the properties of our interest, and hence the obtained results can become irrelevant. However, using h one can "switch off" the intermediate configurations so that they do not contribute to the accumulated reward.

How the expected accumulated reward problem can be solved for pPDA and well-defined reward functions? Similarly to the reachability problem, the methods for finite Markov chains cannot be easily generalized to pPDA even if we restrict ourselves to simple reward functions. The main reason is that the expected accumulated reward can be both infinite and irrational (see Example 6.2.5 and Example 6.2.6 below).

We solve the expected accumulated reward problem using similar tools as for the reachability problem. Namely, we show that the expected accumulated reward is effectively expressible in $ExTh(\mathbb{R})$ (in the sense of Definition 2.3.2) by a formula of polynomial size computable in polynomial space. Note that this result improves on results of [28] where only the expressibility in $Th(\mathbb{R})$ was proved (and only for linear reward functions). The expressibility in $ExTh(\mathbb{R})$ brings several improvements in complexity estimates for the analysis of the long-run average properties (see Chapter 7).

6.2 Expected Accumulated Reward and pPDA

For the rest of this section we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$, and a well-defined reward function $f : \mathcal{C}(\Delta) \to \mathbb{R}^+$ such that for every $p\alpha \in \mathcal{C}(\Delta)$ we have $f(p\alpha) = g(p) + h(p) \cdot c(\alpha)$ where $g, h : Q \to \mathbb{R}^+$ and $c : \Gamma \to \mathbb{R}^+$ are rational valued functions.

First, we simplify the problem in the sense of the following lemma which can easily be proved using the construction from the proof of Lemma 4.1.3.

Lemma 6.2.1. *Given $p\alpha \in \mathcal{C}(\Delta)$ and a simple set $L \subseteq \mathcal{C}(\Delta)$ (where $p\alpha \notin L$), there are effectively computable (in polynomial time) a pPDA Δ', a configuration $q_0 Z_0$ of Δ', and a well-defined reward function $f' : \mathcal{C}(\Delta') \to \mathbb{R}^+$, such that $[E(p\alpha \to^* L), f] = [E(q_0 Z_0 \to^* q_0\varepsilon), f']$.*

Lemma 6.2.1 implies that we may safely concentrate on values of the form $[E(pX \to^* q\varepsilon), f]$ where $[pXq] > 0$. From now on we write $[E(pXq), f]$ instead of $[E(pX \to^* q\varepsilon), f]$. In this notation, the expected accumulated reward problem is to decide whether $[E(pXq), f] \sim \varrho$ for given $\sim \in \{=, <, >, \leq, \geq\}$ and $\varrho \in [0, 1]$.

We show that the expected accumulated reward problem is in **PSPACE**. We start with the following simple observation.

Proposition 6.2.2. *Given $p, q \in Q$ and $X \in \Gamma$, the problem whether $[E(pXq), f] = 0$ is decidable in polynomial time. Hence, given $p\alpha \in \mathcal{C}(\Delta)$ and a simple set $L \subseteq \mathcal{C}(\Delta)$ (where $p\alpha \notin L$), the problem whether $[E(p\alpha \to^* L), f] = 0$ is in **P**.*

Proof. It follows directly from definitions that $[E(pXq), f] > 0$ if and only if there is a path from pX to $q\varepsilon$ which contains a configuration $r\beta$ (distinct from $q\varepsilon$) such that $f(r\beta) > 0$. Hence, it is a mere technicality to reduce the problem whether $[E(pXq), f] > 0$ to the reachability problem (with probability greater than 0), which can be decided in polynomial time due to Proposition 4.1.4. □

Now we show that the values $[E(pXq), f]$ can be expressed as the *least* solution of a system of linear equations in $\mathbb{R}^+_\infty$. Note that if f is linear, then this system of equations corresponds to the system described in [28].

Let $\mathcal{V} := \{\langle E(pXq)\rangle \mid p, q \in Q, X \in \Gamma, [pXq] > 0\}$ be a set of variables. That is, for every $[E(pXq), f]$ there is the associated variable $\langle E(pXq)\rangle$. Consider the following system of equations: If $[E(pXq), f] = 0$, then we put $\langle E(pXq)\rangle = 0$. Otherwise, if $[E(pXq), f] > 0$, then we put

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

where

$$H^{pX}_{rY} = [rYq](f(pX) + \langle E(rYq)\rangle)$$

$$H^{pX}_{rYZ} = \sum_{s \in Q} [rYs][sZq] \left(f(pX) + \langle E(rYs)\rangle + c(Z) \cdot [E(rYs), h] + \langle E(sZq)\rangle\right)$$

Here we use the convention that if $[rYq] = 0$ (or $[rYs][sZq] = 0$ or $c(Z) = 0$), then H_{rY} (or the corresponding summand of H_{rYZ}, resp.) is removed. Using this convention, we avoid problems with undefined terms like $0 \cdot \infty$. The summands that should be removed can be identified in polynomial time using Proposition 4.1.4.

We denote $P[\mathcal{V}]$ the above system of equations. Moreover, given a set of variables $\mathcal{V}' \subseteq \mathcal{V}$, we denote $P[\mathcal{V}']$ the system of equations obtained from the system $P[\mathcal{V}]$ by eliminating all equations for variables *not* in $\mathcal{V}'$.

Let us denote $\mathcal{V}^0$, $\mathcal{V}^r$, and $\mathcal{V}^{<\infty}$ the sets of all variables $\langle E(pXq)\rangle$ such that $[E(pXq), f] = 0$, $0 < [E(pXq), f] < \infty$, and $0 \leq [E(pXq), f] < \infty$, respectively.

We show that the tuple of all $[E(pXq), f]$ values is exactly the least solution of the system $P[\mathcal{V}]$ in $\mathbb{R}^+_\infty$, and moreover, that the tuple of all $[E(pXq), f]$ values where $\langle E(pXq)\rangle \in \mathcal{V}^{<\infty}$ is the *unique* solution of $P[\mathcal{V}^{<\infty}]$ in $\mathbb{R}$. In order to prove the uniqueness part of this result, we use the following lemma.

Lemma 6.2.3. *Let $A \cdot \vec{y} = \vec{b}$ be a linear system of n equations in n variables (i.e., A is a $n \times n$ matrix, $\vec{y}$ is a n-dimensional vector of real variables, and $\vec{b} \in \mathbb{R}^n$), and let us assume that there is the* least non-negative *solution $\vec{e} \in \mathbb{R}^n$ of the system (i.e., among all non-negative real solutions the vector $\vec{e}$ is the (point-wise) least one). If $\vec{e}$ is positive (i.e., $\vec{e}(i) > 0$ for $1 \leq i \leq n$), then $\vec{e}$ is the* unique *solution of the system in $\mathbb{R}$.*

Proof. We show that the matrix A is non-singular. Once we show this we obtain by standard arguments of the linear algebra that $\vec{e} = A^{-1} \cdot \vec{b}$ is the unique solution of the system. Let us assume that A is singular. Then a set V of all $\vec{a}$ satisfying $A \cdot \vec{a} = 0$ is a non-trivial subspace of $\mathbb{R}^n$. Let $\vec{a} \in V$ be an arbitrary vector such that for some $1 \leq i \leq n$ we have $\vec{a}(i) < 0$. Let $\xi = \min_{1 \leq j \leq n} \vec{e}(j)$ and let $\zeta = \max_{1 \leq j \leq n} |\vec{a}(j)|$. Let us denote $\vec{c} = \vec{e} + \frac{\xi}{2\zeta} \cdot \vec{a}$. It is straightforward to verify that $\vec{c}$ is a positive vector and that $\vec{c}(i) < \vec{e}(i)$. However, we have $A \cdot \vec{c} = A \cdot \vec{e} + \frac{\xi}{2\zeta} \cdot A \cdot \vec{a} = \vec{b}$. Thus $\vec{c}$ is a non-negative solution of $A \cdot \vec{y} = \vec{b}$ which is either less than or incomparable with the least non-negative solution $\vec{e}$, a contradiction. □

Lemma 6.2.4. *The tuple of all $[E(pXq), f]$ values is the* least *solution of the system $P[\mathcal{V}]$ in $\mathbb{R}^+_\infty$ with respect to component-wise ordering. Moreover, the tuple of all $[E(pXq), f]$ values, where $\langle E(pXq) \rangle \in \mathcal{V}^{<\infty}$, is the* unique *solution of the system $P[\mathcal{V}^{<\infty}]$ in $\mathbb{R}$.*

Proof Sketch. The proof of the first part of the lemma is deferred to Section 6.3 (the approach is similar to the proof of Theorem 5.1 in [28]). Here we concentrate on the uniqueness part.

Let us consider the system of equations P obtained from $P[\mathcal{V}^r]$ by assigning 0 to all variables of $\mathcal{V}^0$ (thus eliminating these variables from $P[\mathcal{V}^r]$). We claim that P is a system of n linear equations in n variables such that its least non-negative solution is the tuple of all $[E(pXq), f]$ values where $\langle E(pXq) \rangle \in \mathcal{V}^r$. Indeed, observe that no equation of $P[\mathcal{V}]$ for a variable of $\mathcal{V}^{<\infty}$ contains any variable of $\mathcal{V} \setminus \mathcal{V}^{<\infty}$. Note also that each equation of $P[\mathcal{V}]$ for a variable of $\mathcal{V} \setminus \mathcal{V}^{<\infty}$ contains either a variable of $\mathcal{V} \setminus \mathcal{V}^{<\infty}$, or a value of the form $[E(pXq), h]$ such that $[E(pXq), h] = \infty$. It follows that any solution of P can be extended to a solution of $P[\mathcal{V}]$ by assigning 0 to all variables of $\mathcal{V}^0$, and by assigning ∞ to all variables of $\mathcal{V} \setminus \mathcal{V}^{<\infty}$. However, then the tuple of all $[E(pXq), f]$ values where $\langle E(pXq) \rangle \in \mathcal{V}^r$ must be the least non-negative solution of P due to the first part of this lemma. Now, by Lemma 6.2.3, the tuple of all $[E(pXq), f]$ values where $\langle E(pXq) \rangle \in \mathcal{V}^r$ is the *unique* solution of the system P in $\mathbb{R}$.

Because all variables of $\mathcal{V}^0$ have to be assigned 0 in an arbitrary solution of the system $P[\mathcal{V}^{<\infty}]$, we obtain that the tuple of all $[E(pXq), f]$ values where $\langle E(pXq) \rangle \in \mathcal{V}^{<\infty}$ is the unique solution of the system $P[\mathcal{V}^{<\infty}]$ in $\mathbb{R}$. □

The following two examples show that the values $[E(pXq), f]$ can be both infinite and irrational.

Example 6.2.5. *Consider the pBPA from Example 3.1.2 for $x = \frac{1}{2}$. Lemma 6.2.4 implies that $[E(Y \to^* \varepsilon), \mathbf{1}]$ is a solution of the equation $y = 1 + y$ in $\mathbb{R}^+_\infty$, and hence must be equal to ∞.*

Example 6.2.6. *Let us define a pPDA Δ_e with five control states s, p, p', q, r, two stack symbols X, Z, and the following transitions:*

$$sZ \xrightarrow{1} pXZ, \quad pX \xrightarrow{\frac{2}{9}} p'XX, \quad p'X \xrightarrow{1} pXX, \quad pX \xrightarrow{\frac{2}{9}} p\varepsilon, \quad pX \xrightarrow{\frac{5}{9}} q\varepsilon,$$
$$pZ \xrightarrow{1} rZ, \quad rZ \xrightarrow{1} p\varepsilon, \quad qX \xrightarrow{1} q\varepsilon, \quad qZ \xrightarrow{1} p\varepsilon$$

One can easily show that the probability $[sZp] = 1$ (note that from each configuration of the form $pX\alpha Z$ there is a transition to the configuration $q\alpha Z$, which reaches $p\varepsilon$ with the probability 1). On the other hand, using the results of Section 4.1, one can show that the probability $\mathcal{P}(sZ \to^* rZ)$ is equal to $\frac{\sqrt{6}-2}{2}$, which is the least non-negative root of the polynomial $2y^3 - 9y + 2$. Hence, the probability $\mathcal{P}(sZ \to^* rZ)$ is irrational. Now let us define a simple reward function f, which assigns 1 to all configurations with the control state r, and 0 to others. Note that $[E(sZp), f]$ is equal to the probability of reaching rZ from sZ under the assumption that $p\varepsilon$ is reached. However, this probability is equal to $\mathcal{P}(sZ \to^* rZ) = \frac{\sqrt{6}-2}{2}$ because $[sZp] = 1$. Hence, $[E(sZp), f] = \frac{\sqrt{6}-2}{2}$, which is irrational.

Remark 6.2.7. *Now we show that the system of equations for finite Markov chains, presented in the beginning of this chapter, is in fact a special case of the system $P[\mathcal{V}]$. Let $M = (S, \to, Prob)$ be a finite Markov chain, $L \subseteq S$, and let $S' = S \backslash L$. Let us define a pPDA Δ with the control states $S' \cup \{l\}$, one stack symbol X, and the following transitions:*

1. *For $s, t \in S'$, we put $sX \xrightarrow{x} tX$ if and only if $s \xrightarrow{x} t$ in M;*
2. *For $s \in S'$, we put $sX \xrightarrow{x} l\varepsilon$ if and only if $x = \sum_{s \xrightarrow{y} t, t \in L} y > 0$;*
3. *nothing else is a transition of Δ.*

Each reward function $f : S \to \mathbb{R}^+$ can be seen as a simple reward function over $\mathcal{C}(\Delta)$ (let l be assigned 0). Then, clearly, $[E(s \to^* L), f] = [E(sXl), f]$.

If one simplifies the equation of $P[\mathcal{V}]$ for the variable $\langle E(sXl)\rangle$, one obtains the equation for the variable x_s from the linear system for finite chains. It follows that Lemma 6.2.4 applies also to that system for finite chains.

Now we show that the expected accumulated reward is effectively expressible in $ExTh(\mathbb{R})$ (remember that $\mathbf{S}_\Delta = \{pX \in Q \times \Gamma \mid [pX\uparrow] > 0\}$).

Theorem 6.2.8.

1. *The problem whether $[E(pXq), f] = \infty$ is in **PSPACE**.*
2. *The values $[E(pXq), f] \in \mathbb{R}^+_\infty$ are effectively expressible in $ExTh(\mathbb{R})$ by formulae of polynomial size, computable in polynomial space.*

Proof. We proceed as follows: We prove the theorem under an assumption that both 1. and 2. already hold for the simple reward function h. Once we prove this, we obtain that the theorem holds for *all* simple reward functions, because it clearly holds for the reward function that assigns 0 to all configurations. Finally, we obtain that the theorem holds for all well-defined reward functions.

Let us assume that both 1. and 2. hold for h. Let us consider the system $P[\mathcal{V}]$. In order to prove 1., it suffices to show that the set $\mathcal{V}^{<\infty}$ can be computed in polynomial space. We show that the set $\mathcal{V}^{<\infty}$ is equal to the *greatest* subset $\mathcal{V}'$ of $\mathcal{V}$ that satisfies the following conditions:

(a) all equations of $P[\mathcal{V}']$ contain only variables of $\mathcal{V}'$;

(b) each equation of $P[\mathcal{V} \setminus \mathcal{V}']$ contains either at least one variable of $\mathcal{V} \setminus \mathcal{V}'$, or a value of the form $[E(pXq), h]$ where $[E(pXq), h] = \infty$;

(c) the system $P[\mathcal{V}']$ has a solution in $\mathbb{R}^+$.

Indeed, $\mathcal{V}^{<\infty}$ clearly satisfies (a) – (c) due to Lemma 6.2.4. Now let $\mathcal{V}' \subseteq \mathcal{V}$ be a set of variables that satisfies (a) – (c), such that $\mathcal{V}' \not\subseteq \mathcal{V}^{<\infty}$. Observe that the finite solution of $P[\mathcal{V}']$ (which exists due to (c)), can be extended to a solution of $P[\mathcal{V}]$ in $\mathbb{R}^+_\infty$, by assigning ∞ to each variable of $\mathcal{V} \setminus \mathcal{V}'$. However, then all variables of $\mathcal{V}' \setminus \mathcal{V}^{<\infty}$ are assigned finite values in the solution of $P[\mathcal{V}]$, which contradicts Lemma 6.2.4.

We show that for a given set $\mathcal{V}'$ the above conditions (a) – (c) can effectively be checked in polynomial space. Once we show this, we obtain that the set $\mathcal{V}^{<\infty}$ is computable in polynomial space by searching through all subsets of $\mathcal{V}$.

The condition (a) can easily be verified in polynomial time. The condition (b) can be checked in polynomial space due to our assumptions about the function h.

The condition (c) can be checked as follows: First check whether no equation of $P[\mathcal{V}']$ contains a value of the form $[E(pXq), h]$ such that $[E(pXq), h] = \infty$. Clearly, if there is such an equation in $P[\mathcal{V}']$, then the condition (c) is not satisfied. Due to our assumptions about h, this check can be done in polynomial space.

Now let us assume that all values of the form $[E(pXq), h]$ occurring in $P[\mathcal{V}']$ are finite. Let us denote $P[\mathcal{V}'] = \{x_i = P_i \mid 1 \leq i \leq k\}$. Note that each coefficient of $P[\mathcal{V}']$ is an arithmetical expression over rational numbers, probabilities of the form $[pXq]$, and values $[E(sZr), h]$, where $[E(sZr), h] < \infty$. Hence, the coefficients of $P[\mathcal{V}']$ are effectively expressible in $ExTh(\mathbb{R})$ by formulae of polynomial size, computable in polynomial space, due to Corollary 4.1.10, Proposition 2.3.3, and 2. for the function h.

The condition (c) can be checked using the following procedure:

1. compute formulae $\Phi_1(y_1), \ldots, \Phi_\ell(y_\ell)$ expressing all coefficients $c_1, \ldots, c_\ell$, respectively, of the system $P[\mathcal{V}']$, and compute the system $P[\mathcal{V}']$ with each coefficient c_i symbolically represented by the corresponding variable y_i (in polynomial space);

2. compute (in polynomial time) a formula

$$\Psi(x_1, \ldots, x_k) \equiv (\exists y_1) \cdots (\exists y_\ell) \Psi'$$

where $\Psi' \equiv \left(\bigwedge_{i=1}^{k} x_i \geq 0\right) \wedge \left(\bigwedge_{i=1}^{k} x_i = P_i\right) \wedge \left(\bigwedge_{i=1}^{\ell} \Phi_i(y_i)\right)$, which is satisfied exactly by the non-negative solutions of $P[\mathcal{V}']$;

3. Decide whether $(\exists x_1) \cdots (\exists x_k) \Psi(x_1, \ldots, x_k)$ is true (in polynomial space due to Theorem 2.3.1).

Clearly, the above procedure runs in polynomial space and decides whether $P[\mathcal{V}']$ has a solution in $\mathbb{R}^+$. This proves 1.

Now let us prove 2. By 1., the problem whether $[E(pXq), f] = \infty$ can be decided in polynomial space. By Lemma 6.2.4, the tuple of all $[E(pXq), f]$ values, where $\langle E(pXq) \rangle \in \mathcal{V}^{<\infty}$, is the unique solution of the system $P[\mathcal{V}^{<\infty}]$ in $\mathbb{R}$. It follows from the above arguments that the coefficients of $P[\mathcal{V}^{<\infty}]$ are effectively expressible in $ExTh(\mathbb{R})$ by formulae of polynomial size, computable in polynomial space. Hence, by Proposition 2.3.3, also the values $[E(pXq), f] < \infty$ are effectively expressible in $ExTh(\mathbb{R})$ by formulae of polynomial size, computable in polynomial space, which proves 2. □

As a corollary to the above results and Theorem 2.3.1, we obtain

Theorem 6.2.9. *The expected accumulated reward problem for pPDA and well-defined reward functions is in* ***PSPACE****.*

Remark 6.2.10. *[28] also studies the* variance *of the random variable $R^f_{s,L}$ where f is a linear reward function. In [28] the variance is shown to be expressible as the least non-negative solution of a system of polynomial equations with effectively expressible coefficients. It follows that the problem whether the variance is greater than (less than, equal to) a given rational number is in* ***PSPACE****. For further details see [28].*

6.3 Formal Proofs

Throughout this section we use the notation introduced in Section 4.6.1. We also fix a pPDA Δ and a well-defined reward function $f : \mathcal{C}(\Delta) \to \mathbb{R}^+$ such that for every $p\alpha \in \mathcal{C}(\Delta)$ we have $f(p\alpha) = g(p) + h(p) \cdot c(\alpha)$. Our goal is to prove Lemma 6.2.4. The following lemma is an easy consequence of definitions.

Lemma 6.3.1. *Given $v \in FPath$ and $\beta \in \Gamma^*$, $f(v\lfloor\beta) = f(v) + (h(v) \cdot c(\beta))$.*

Proof of Lemma 6.2.4

It remains to show that the tuple of all $[E(pXq), f]$ values is the least solution of the system $P[\mathcal{V}]$ in $\mathbb{R}^+_\infty$. We start by showing that it is a solution of the system. Let us fix $pX \in Q \times \Gamma$ and $q \in Q$ such that $[pXq] > 0$.

Let us introduce some notation. We denote

$$W^{q\varepsilon} = \{w \in Run(pXq) \mid w(1) = q\varepsilon\}$$

Given $rY \in Q \times \Gamma$ such that $pX \to rY$, we denote

$$W^{rY} = pX \to rY \odot Run(rYq)$$

Given $rY \in Q \times \Gamma$ and $Z \in \Gamma$ such that $pX \to rYZ$, and $s \in Q$, we denote

$$W_s^{rYZ} = pX \to rYZ \odot ((FPath(rYs)\lfloor Z) \odot Run(sZq))$$

It is easy to see that

$$Run(pXq) = W^{q\varepsilon} \uplus \biguplus_{pX \to rY} W^{rY} \uplus \biguplus_{pX \to rYZ, s \in Q} W_s^{rYZ}$$

By Lemma 2.2.3 and Lemma 4.6.2, $\mathcal{P}(W^{rY}) = Prob(pX \to rY)[rYq]$, and $\mathcal{P}(W_s^{rYZ}) = Prob(pX \to rYZ)[rYs][sZq]$.

Given a run $w \in W_s^{rYZ}$ of the form $pX \to rYZ \odot (v\lfloor Z) \odot u \odot (q\varepsilon)^\omega$ where $v \in FPath(rYs)$ and $u \in FPath(sZq)$, we denote $S_1^f(w) = f(v^{|v|-1})$, $S_1^h(w) = h(v^{|v|-1})$, $S_2^f(w) = f(u^{|u|-1})$, and $\bar{c}(w) = c(Z)$. Given a run $w \in W^{rY}$ of the form $w = pX \to rY \odot u \odot (q\varepsilon)^\omega$ where $u \in FPath(rYq)$, we denote $S_1^f(w) = f(u^{|u|-1})$, $S_1^h(w) = 0$, $S_2^f(w) = 0$, and $\bar{c}(w) = 0$. Given $w \in Run(pX)$ such that either $w \notin Run(pXq)$ or $w \in W^{q\varepsilon}$, we define $S_1^f(w) = S_1^h(w) = S_2^f(w) = \bar{c}(w) = 0$. To simplify our notation, we write R^f_{pXq} instead of $R^f_{pX,\{q\varepsilon\}}$. By Lemma 6.3.1, for all $w \in Run(pXq)$

$$R^f_{pXq}(w) = f(pX) + S_1^f(w) + \bar{c}(w) \cdot S_1^h(w) + S_2^f(w)$$

It is easy to prove that S_1^f, S_1^h, and S_2^f are discrete random variables. The following equations are straightforwardly proved using Lemma 2.2.3, Lemma 4.6.2, Lemma 4.6.7:

$$\begin{aligned}
E(S_1^f \mid W^{rY}) &= [E(rYq), f] \\
E(S_1^f \mid W_s^{rYZ}) &= [E(rYs), f] \\
E(S_1^h \mid W_s^{rYZ}) &= [E(rYs), h] \\
E(S_2^f \mid W_s^{rYZ}) &= [E(sZq), f]
\end{aligned}$$

and thus, by the linearity of the expectation,

$$\begin{aligned}
E(R^f_{pXq} \mid W^{rY}) &= f(pX) + [E(rYs), f] \\
E(R^f_{pXq} \mid W_s^{rYZ}) &= f(pX) + [E(rYs), f] + c(Z) \cdot [E(rYs), h] + [E(sZq), f]
\end{aligned}$$

Now because

$$\begin{aligned}
[E(pXq), f] &= \mathcal{P}(W^{q\varepsilon} \mid Run(pXq)) \cdot f(pX) + \\
&+ \sum_{pX \xrightarrow{x} rY} \mathcal{P}(W^{rY} \mid Run(pXq)) \cdot E(R^f_{pXq} \mid W^{rY}) + \\
&+ \sum_{pX \xrightarrow{x} rYZ, s \in Q} \mathcal{P}(W^{rYZ}_s \mid Run(pXq)) \cdot E(R^f_{pXq} \mid W^{rYZ}_s) \\
&= \frac{\sum_{pX \xrightarrow{x} q\varepsilon} x}{[pXq]} \cdot f(pX) + \sum_{pX \xrightarrow{x} rY} \frac{x[rYq]}{[pXq]} \cdot E(R^f_{pXq} \mid W^{rY}) + \\
&+ \sum_{pX \xrightarrow{x} rYZ, s \in Q} \frac{x[rYs][sZq]}{[pXq]} \cdot E(R^f_{pXq} \mid W^{rYZ}_s)
\end{aligned}$$

we obtain (by slightly rearranging the last expression) that the tuple of all $[E(pXq), f]$ values is a solution of the system $P[\,\mathcal{V}\,]$.

Now we prove that the tuple of all $[E(pXq), f]$ values is the *least* solution of the system $P[\,\mathcal{V}\,]$ in $\mathbb{R}^+_\infty$. Given $n \geq 0$, we denote $Run^n(pXq)$ the set of all runs $w \in Run(pXq)$ such that $w(n) = q\varepsilon$. Given $pX \in Q \times \Gamma$, $q \in Q$ and $n \geq 0$, we define a random variable $R^{f,n}_{pXq} : Run(pX) \to \mathbb{R}^+$ as follows: Given $w \in Run^n(pXq)$, we put $R^{f,n}_{pXq}(w) = R^f_{pXq}(w)$. Otherwise, we put $R^{f,n}_{pXq}(w) = 0$. If $[pXq] > 0$, then we denote $[E(pXq), f]_n = E(R^{f,n}_{pXq} \mid Run(pXq))$.

It is easy to see that for all $w \in Run(pXq)$ and $i \leq j$, we have $R^{f,i}_{pXq}(w) \leq R^{f,j}_{pXq}(w) \leq R^f_{pXq}(w)$, and $\lim_{n\to\infty} R^{f,n}_{pXq}(w) = R^f_{pXq}(w)$. If $[pXq] > 0$, then $[E(pXq), f]_i \leq [E(pXq), f]_j \leq [E(pXq), f]$ by the monotonicity of the expectation ([13], Theorem 16.1), and $\lim_{n\to\infty} [E(pXq), f]_n = [E(pXq), f]$ by the monotone convergence theorem ([13], Theorem 16.2). It follows that $[E(pXq), f] = \sup\{[E(pXq), f]_n \mid n \geq 0\}$.

Hence, it suffices to prove that for all $n \geq 0$, the tuple of all $[E(pXq), f]_n$ values is (point-wise) less than or equal to any solution of the system $P[\,\mathcal{V}\,]$ in $\mathbb{R}^+_\infty$. Let $V \in (\mathbb{R}^+_\infty)^{|\,\mathcal{V}\,|}$ be a solution of the system $P[\,\mathcal{V}\,]$. We denote $V[pXq]$ the component of V that corresponds to the variable $\langle E(pXq)\rangle$.

Let us fix $pX \in Q \times \Gamma$ and $q \in Q$ such that $[pXq] > 0$. We show, by induction on n, that $[E(pXq), f]_n \leq V[pXq]$ for all $n \geq 0$. If $n = 0$, then for all $w \in Run(pXq)$ we have $R^{f,n}_{pXq}(w) = 0$, and hence $0 = [E(pXq), f]_n \leq V[pXq]$.

Let us fix $n \geq 1$. Given a run $w \in Run^n(pXq)$, we define $S^{f,n}_1(w) = S^f_1(w)$, $S^{h,n}_1(w) = S^h_1(w)$, $S^{f,n}_2(w) = S^f_2(w)$, and $\bar{c}^n(w) = \bar{c}(w)$. For runs $w \in Run(pX) \setminus Run^n(pXq)$ we define $S^{f,n}_1(w) = S^{h,n}_1(w) = S^{f,n}_2(w) = \bar{c}^n(w) = 0$. It follows from Lemma 6.3.1 that for all $w \in Run(pXq)$

$$R^{f,n}_{pXq}(w) \leq f(pX) + S^{f,n}_1(w) + \bar{c}^n(w) \cdot S^{h,n}_1(w) + S^{f,n}_2(w)$$

The following inequalities are straightforwardly proved using Lemma 2.2.3, Lemma 4.6.2, Lemma 4.6.7:

$$\begin{aligned} E(S_1^{f,n} \mid W^{rY}) &\leq [E(rYq), f]_{n-1} \\ E(S_1^{f,n} \mid W_s^{rYZ}) &\leq [E(rYs), f]_{n-1} \\ E(S_1^{h,n} \mid W_s^{rYZ}) &\leq [E(rYs), h]_{n-1} \\ E(S_2^{f,n} \mid W_s^{rYZ}) &\leq [E(sZq), f]_{n-1} \end{aligned}$$

Hence, by induction and the linearity of the expectation,

$$E(R_{pXq}^{f,n} \mid W_s^{rYZ}) \leq f(pX) + V[rYs] + c(Z) \cdot [E(rYs), h] + V[sZq]$$

and similarly $E(R_{pXq}^{f,n} \mid W^{rY}) \leq f(pX) + V[rYq]$.

Thus, finally,

$$\begin{aligned} [E(pXq), f]_n \leq\ & \frac{\sum_{pX \xrightarrow{x} q\varepsilon} x}{[pXq]} \cdot f(pX) + \sum_{pX \xrightarrow{x} rY} \frac{x[rYq]}{[pXq]} \cdot E(R_{pXq}^{f,n} \mid W^{rY}) + \\ & + \sum_{pX \xrightarrow{x} rYZ, s \in Q} \frac{x[rYs][sZq]}{[pXq]} \cdot E(R_{pXq}^{f,n} \mid W_s^{rYZ}) \leq V[pXq] \end{aligned}$$

Chapter 7

Long-run Properties

In this chapter we introduce a family of long-run average properties of Markov chains that are useful for purposes of performance and reliability analysis, and show that these properties can effectively be checked for Markov chains generated by pPDA. This chapter is based on the full version of the paper [15].

An important source of initial inspiration for this study was [24], where de Alfaro convincingly argues that conventional temporal logics cannot express important properties of the long-run average behavior of probabilistic systems. To get some intuition, consider a system which repeatedly services certain requests, like a www server, an answering machine, or a telephone switchboard. Typical performance or reliability questions like "What is the average time of servicing a request?" or "What is the probability that a request will be serviced within 3 seconds?" are not directly expressible in conventional temporal logics. In [24], each run of the system is assigned the average service time defined as $\lim_{n\to\infty}(\sum_{i=1}^{n} T(i))/n$, where $T(i)$ is the service time for the i^{th} request which appears along the run. Then, a special state predicate is introduced which holds in a given state if and only if the total probability of all runs where the average service time is bounded by a given constant is equal to 1. This predicate is then "plugged" into the syntax of temporal logics such as PCTL or PCTL*, and a model-checking algorithm for finite-state Markov decision processes is presented.

Various important reliability and performance properties cannot be deduced just from the average service time. Examples are the average deviation from the average service time, and the probability that a service takes longer than a given bound. To formulate such properties, we introduce a family of random variables that capture certain limit values of runs, and use these variables to define a family of *run-indicators*. A run-indicator classifies each run as "good" or "bad" according to these limit values, and one can thus formulate questions about the probability of good/bad behavior. For example, one can formally express questions like

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

- What is the probability of those runs where the average service time is between 30 and 32 seconds, and the average deviation from 31 seconds is at most 5 seconds?
- What is the probability of all runs satisfying the previous condition and the condition that the percentage of services longer than 37 seconds is at most 20%?

Actually, our treatment is generic in the sense that we use general reward functions to assign numeric values to individual services. These reward functions can also take negative values, and thus arbitrary gains and costs (not only time) can be modeled. For pPDA, we restrict ourselves to the well-defined reward functions (see Definition 6.1.1). We show that the problem whether $\mathcal{P}(I{=}1) \sim \varrho$ is decidable (here I is one of the introduced run-indicators, $\mathcal{P}(I{=}1)$ is the probability that I is satisfied, $\varrho \in [0,1]$ is a rational constant, and $\sim \in \{<, \leq, >, \geq, =\}$). This allows us to approximate $\mathcal{P}(I{=}1)$ by arbitrarily close rational lower and upper bounds.

Another issue addressed in this work is the *prediction* of the aforementioned limit values. Let us explain what is meant by the prediction in greater detail. In ergodic Markov chains, our limit random variables usually take just one value with probability one, regardless of the initial state of the run. For example, the average service time is the same for "almost all" runs, and hence it does not make much sense to predict it because its value is determined from the very beginning. However, in general Markov chains, the average service time can take infinitely many values with a positive probability, and the probability that the average service time stays within given bounds changes along the execution of a run. Hence, one can ask whether it is possible to "predict" the future behavior just by inspecting a bounded prefix of a run. Of course, the answer is negative in general. However, we show that for the subclass of Markov chains that are definable by pPDA, such predictions *are* possible, even though these chains are generally infinite-state and non-ergodic. In fact, one can *efficiently* predict quite complicated run-indicators up to an arbitrarily small given error δ (the smaller δ we choose, the longer prefix of a run must be examined).

We also study the decidability of the model-checking problem for temporal logics (such as LTL, PCTL, and PECTL*) extended with state predicates based on the above mentioned limit properties of runs. We prove that the model-checking problem remains decidable if we only use qualitative variants of these predicates, and derive an undecidability result for general predicates.

7.1 Long-run Properties of Markov Chains

In this section we introduce the family of long-run average properties of Markov chains. We show how to use these properties in performance analysis, and we also explain what is meant by a faithful and efficient prediction of these properties.

For the rest of this section, let us fix a Markov chain $M = (S, \to, Prob)$ and an initial state $s_0 \in S$ (i.e., all probabilities are considered over $Run(s_0)$). We also fix a reward function $f : S \to \mathbb{R}$ (note that here we allow negative values).

The request-service cycles are modeled as follows. Let $T \subseteq S$ be a subset of *triggers*. Let $w \in Run(s_0)$ be a run with infinitely many triggers $w(i_1), w(i_2), \ldots$, and let $w[j]$ denote the subword $w(i_j + 1), \cdots, w(i_{j+1})$ of w. Hence, $w[j]$ is the subword of w consisting of all states in between the j^{th} trigger (not included) and the $j + 1^{th}$ trigger (included). Intuitively, $w[j]$ corresponds to the j^{th} service. According to our definition, a new service starts immediately after finishing the previous service. (This is not a real restriction because the reward function can be set up so that the states visited before the actual start of the service are ignored, i.e., have zero reward.) Similarly to the previous chapter, $f(w[j])$ denotes the total reward accumulated in $w[j]$, i.e., $f(w[j]) = \sum_{k=i_j+1}^{i_{j+1}} f(w(k))$.

The properties of runs we are interested in here are formally defined as *indicators*. An indicator is a random variable $I : Run(s_0) \to \{1, 0\}$ which classifies the runs as "good" or "bad" according to some criterion. For example, the following simple indicator I_{inf} is obviously relevant in our setting:

$$I_{inf}(w) = \begin{cases} 1 & \text{if } w(i) \in T \text{ for infinitely many } i\text{'s;} \\ 0 & \text{otherwise.} \end{cases}$$

It is easy to show that I_{inf} is a random variable. Indeed, observe that for every $x \in \mathbb{R}_{\pm\infty}$, the set $\{w \in Run(s_0) \mid I_{inf}(w) \geq x\}$ is equal either to $\emptyset$ (for $x > 1$), or to $Run(s_0)$ (for $x \leq 0$), or to the measurable set $\bigcap_{i=1}^{\infty} \bigcup_{v \in P_i} Run(v)$, where each P_i is the set of all paths of $FPath(s_0)$, that contain i triggers.

We are primarily interested in those runs w where $I_{inf}(w) = 1$, because only then the limit features introduced below make a good sense. The runs for which I_{inf} equals 0 are those where the service cycle is either eventually terminated, or the last service is never finished. Since this can be seen as an error, $\mathcal{P}(I_{inf}{=}1)$ is an important quantitative information about the behavior of M. For example, the quantitative ω-regular model-checking problem for properties definable via deterministic Büchi automata is obviously reducible to the problem of computing $\mathcal{P}(I_{inf}{=}1)$ in any class of models that is closed under synchronized product with a deterministic finite state automaton (e.g., pPDA). The ω-regular model-checking problem has been already studied in Section 4.3.2 and its decidability for pPDA has been shown by employing non-trivial methods. Hence, even computing $\mathcal{P}(I_{inf}{=}1)$ can be a difficult problem in general.

Before introducing other indicators, let us explain what is meant by "predictability" of an indicator.

Definition 7.1.1. *Let I be an indicator. We say that I is* well-predictable (over $Run(s_0)$) *if for each $\zeta > 0$ there effectively exist $n \geq 0$ and an indicator G^n such that $\mathcal{P}(G^n{\neq}I) \leq \zeta$, and the value of $G^n(w)$ is effectively computable from the prefix of w of length n.*[1]

[1]Due to the Markov property, the last state of the prefix contains a complete information that is relevant for predicting the future behavior; one cannot learn anything "fundamentally new" by inspecting the previous states in the prefix. However, this inspection can make the prediction more *efficient*, as we shall see in Section 7.2.

Hence, G^n efficiently "guesses" the value of I after seeing the first n states of a run, and the "quality" of that guess is measured by ζ.

In general, indicators are rarely well-predictable. An important outcome of our work is that a large class of practically relevant indicators is well-predictable in the class of Markov chains generated by pPDA (at least, for those pPDA that satisfy a mild and effectively checkable condition formulated in Section 7.2).

Now we define other random variables over $Run(s_0)$, and the associated indicators. Let $\kappa \in \mathbb{R}$, $\xi, \lambda \in \mathbb{R}_{\pm\infty}$, and let $B(w[j], \xi, \lambda)$ return either 1 or 0 depending on whether $\xi \leq f(w[j]) \leq \lambda$ or not, respectively. We define the random variables

$$\mathbf{A}(w) = \lim_{n\to\infty} \frac{\sum_{j=1}^{n} f(w[j])}{n}$$

$$\mathbf{D}[\kappa](w) = \lim_{n\to\infty} \frac{\sum_{j=1}^{n} |f(w[j]) - \kappa|}{n}$$

$$\mathbf{R}[\xi, \lambda](w) = \lim_{n\to\infty} \frac{\sum_{j=1}^{n} B(w[j], \xi, \lambda)}{n}$$

If the corresponding limit does not exist or $I_{inf}(w) = 0$, the above variables take the value $\perp$.

Using Proposition 2.1.1, it is easy to show that $\mathbf{A}$, $\mathbf{D}[\kappa]$, and $\mathbf{R}[\xi, \lambda]$ are indeed random variables. Let us consider, e.g., the function $\mathbf{A}$ in greater detail. For every $n \geq 1$ we define a function A_n as follows:

$$A_n(w) = \begin{cases} \frac{\sum_{j=1}^{n} f(w[j])}{n} & \text{if } w \text{ contains at least } n \text{ triggers;} \\ 0 & \text{otherwise.} \end{cases}$$

One can easily show that each A_n is a random variable. For each $w \in Run(s_0)$ we have that

$$\mathbf{A}(w) = \begin{cases} \lim_{n\to\infty} A_n(w) & \text{if } I_{inf}(w) = 1 \text{ and the limit exists;} \\ \perp & \text{otherwise.} \end{cases}$$

It follows that $\mathbf{A}$ is also a random variable by applying Proposition 2.1.1 and the fact that I_{inf} is a random variable. Indeed, for all $x \in \mathbb{R}_{\pm\infty}$ we have

$$\begin{aligned} \{w \in Run(s_0) \mid \mathbf{A}(w) \geq x\} &= \{w \in Run(s_0) \mid \lim_{n\to\infty} A_n(w) \geq x\} \cap \\ &\cap \{w \in Run(s_0) \mid I_{inf}(w) = 1\} \end{aligned}$$

which is measurable.

The variable $\mathbf{A}$ returns the average reward per service in a given run. The variable $\mathbf{D}[\kappa]$ returns the average deviation of the reward per service from a given center κ in a given run. Finally, the variable $\mathbf{R}[\xi, \lambda]$ returns the *ratio* of the services whose rewards are within the bounds ξ, λ, to all services.

Let V be one of the above defined variables, and let $\ell, u \in \mathbb{R}_{\pm\infty}$. The indicator $I[V, \ell, u] : Run(s_0) \to \{0, 1\}$ is defined as follows:

$$I[V, \ell, u](w) = \begin{cases} 1 & \text{if } V(w) \neq \perp \text{ and } \ell \leq V(w) \leq u; \\ 0 & \text{otherwise.} \end{cases}$$

We also consider "Boolean combinations" of the indicators above (where 1 and 0 are interpreted as *true* and *false*, respectively). Thus, we obtain a family $\mathcal{I}$ consisting of I_{inf}, all $I[V, \ell, u]$, and their Boolean combinations. To show the relevance of Boolean combinations, let us formalize the properties mentioned in Section 1 (assume that the reward function corresponds to the time spent in a given state).

- $I[\mathbf{A}, 30, 32]$ defines all runs where the average service time is between 30 and 32.
- $I[\mathbf{A}, 30, 32] \wedge I[\mathbf{D}[31], 0, 5]$ defines all runs where the average service time is between 30 and 32, and the average deviation of service time from 31 is at most 5.
- $I[\mathbf{A}, 30, 32] \wedge I[\mathbf{D}[31], 0, 5] \wedge I[\mathbf{R}[37, \infty], 0, 0.2]$ defines all runs satisfying the previous condition and the condition that the percentage of services longer than 37 is at most 20%.

Let $I \in \mathcal{I}$. We study three basic algorithmic problems:

1. *Long-run average reward problem*:

   Given $\varrho \in [0, 1]$ and $\sim \in \{<, >, =, \leq, \geq\}$, is $\mathcal{P}(I{=}1) \sim \varrho$?

   Of course, we would rather like to compute $\mathcal{P}(I{=}1)$. However, for pPDA, the reachability problem straightforwardly reduces to the long-run average problem even for very simple indicators I. Consequently, the probability $\mathcal{P}(I{=}1)$ can be irrational and we can only hope to solve this problem in the same way as the reachability problem was solved in Section 4.1.1. That is, the task is to show that $\mathcal{P}(I{=}1)$ is effectively expressible in $ExTh(\mathbb{R})$. From this we obtain the decidability of the long-run average reward problem. In particular, $\mathcal{P}(I{=}1)$ can be effectively approximated up to an arbitrarily small error (e.g., by a simple binary search).

2. If I is well-predictable, design a suitable G^n. The indicator G^n should satisfy the "efficiency" requirements discussed after Definition 7.1.1.

3. Since the predicate $\mathcal{P}(I{=}1) \sim \varrho$ is either valid or invalid in each state $s \in S$, it can be "plugged" as an "atomic proposition" into temporal logics such as LTL, PCTL, or PCTL* in the style of [24] (the state predicate which has been introduced and studied in [24] corresponds to $\mathcal{P}(I[\mathbf{A}, \ell, u]{=}1) = 1$). The question is whether there is a model-checking algorithm for these extended logics.

Note that the conditional probability $\mathcal{P}(I{=}1 \mid I_{inf}{=}1)$ (which is relevant in situations when $\mathcal{P}(I_{inf}{=}1){<}1$) is expressible from the probabilities $\mathcal{P}(I{=}1 \wedge I_{inf}{=}1)$ and $\mathcal{P}(I_{inf}{=}1)$. Hence, if we manage to solve the three problems above, our results apply also to $\mathcal{P}(I{=}1 \mid I_{inf}{=}1)$.

7.2 Long-run Properties and pPDA

In this section we examine and solve the three problems given at the end of Section 7.1 for pPDA and the class of well-defined reward functions (see Definition 6.1.1). All these results work under a mild and effectively checkable condition, which is formulated and explained at the beginning of the next subsection.

For the rest of this section we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$ and a subset $T \subseteq Q$ of control states. A configuration $p\alpha \in \mathcal{C}(\Delta)$ is a *trigger* if and only if $p \in T$. The notions introduced in Section 7.1 can now be applied to the chain M_Δ. We also fix a well-defined reward function $f : \mathcal{C}(\Delta) \to \mathbb{R}^+$.

From Section 4.3.1 we borrow the Markov chain $\mathbf{X}_\Delta$ which will prove to be the crucial tool in our investigations of the limit properties. Remember that we denote $\mathbf{S}_\Delta = \{pX \in Q \times \Gamma \mid [pX\uparrow] > 0\}$, the set of states of $\mathbf{X}_\Delta$. Remember also that BSCC_Δ denotes the set of all bottom strongly connected components of the chain $\mathbf{X}_\Delta$.

We define a random variable $Entry$ which for every $w \in Run(pX)$ returns either $w(ind_j(w))$ where $j \geq 1$ is the least number such that $\mathbf{X}^j_\Delta(w) \in C$ for some $C \in \mathrm{BSCC}_\Delta$, or $\perp$ if there is no such j. In other words, if w is a clean run whose footprint hits a BSCC of $\mathbf{X}_\Delta$, then $Entry(w)$ is the configuration which "enters" this component.

Remember that $Run(pX, C)$ denotes the set of all runs of $Good(pX)$ whose footprints enter the component $C \in \mathrm{BSCC}_\Delta$. By Proposition 4.3.10, we have $\sum_{C \in \mathrm{BSCC}_\Delta} \mathcal{P}(Run(pX, C) \mid Clean(pX)) = 1$, provided $pX \in \mathbf{S}_\Delta$. Consequently, $\mathcal{P}(Entry{=}\perp \mid Clean(pX)) = 0$ because for every $C \in \mathrm{BSCC}_\Delta$ we have $\mathcal{P}(Entry{=}\perp \mid Run(pX, C)) = 0$, provided $\mathcal{P}(Run(pX, C)) > 0$. Given $\beta \in \Gamma^*$, we denote $Run(pX, C, \beta)$ the set of all runs $w \in Run(pX, C)$ such that $tail(Entry(w)) = \beta$.

Solving the problems of Section 7.1 for pPDA.

In this subsection we still work with the pPDA Δ which has been fixed at the beginning of Section 7.2 and we also fix a well-defined reward function f. However, we need to adopt one additional assumption about Δ, which is crucial in almost all proofs:

> *"For all $p, q \in Q$ and $X \in \Gamma$, the conditional expected number of transitions needed to reach $q\varepsilon$ from pX, under the condition that $q\varepsilon$ is indeed reached from pX, is finite."*

Formally, we assume that $[E(pXq), \mathbf{1}] < \infty$, whenever $[pXq] > 0$, where $\mathbf{1}$ is the reward function which returns 1 for all configurations. Using Theorem 6.2.8 one can effectively check in polynomial space whether this assumption is satisfied or not for a given pPDA. In terms of recursive sequential programs, the assumption corresponds to the requirement that if we restrict ourselves to terminating computations, then the expected termination time of each procedure is finite. From a practical point of view, this assumption is harmless because its violation indicates a severe design error anyway. From a theoretical point of view, this assumption allows to establish useful connections between the properties of M_Δ and $\mathbf{X}_\Delta$, as we shall see in the forthcoming theorems.

We start by presenting a crucial result which says that the (in)validity of all indicators in our family $\mathcal{I}$ for a given $w \in \mathit{Clean}(pX)$ is essentially determined only by the BSCC of $\mathbf{X}_\Delta$ hit by the footprint of w, and by the stack content in the configuration which enters this component. To formulate this precisely, we need to introduce another indicator $\mathit{Hit}[L]$, where $L \subseteq \Gamma^*$ is a regular language:

$$\mathit{Hit}[L](w) = \begin{cases} 1 & \text{if } \mathit{Entry}(w) \neq \bot \text{ and } \mathit{tail}(\mathit{Entry}(w)) \in L; \\ 0 & \text{otherwise.} \end{cases}$$

The following theorem is the main result of this chapter. Its rather technical proof is deferred to Section 7.3. In order to provide more detailed complexity analysis we denote $\mathcal{I}_{\mathbf{A}}$ a subclass of $\mathcal{I}$, which consists only of I_{inf}, all $I[\mathbf{A}, \ell, u]$, and their Boolean combinations.

Theorem 7.2.1 (Main Theorem)**.** *Let $I \in \mathcal{I}$ be an indicator and $pX \in \mathbf{S}_\Delta$. For every $C \in \mathit{BSCC}_\Delta$, satisfying $\mathcal{P}(\mathit{Run}(pX, C)) > 0$, there effectively exists a regular language $L_C \subseteq \Gamma^*$ such that $\mathcal{P}(I = \mathit{Hit}[L_C] \mid \mathit{Run}(pX, C)) = 1$. If the considered reward function f is simple, then L_C equals either Γ^* or $\emptyset$, and the problem whether $L_C = \Gamma^*$ is in **EXPSPACE** (and in **PSPACE** in $|\Delta|$, i.e., for fixed I and f). Moreover, if f is simple and $I \in \mathcal{I}_{\mathbf{A}}$, then the problem whether $L_C = \Gamma^*$ is in **PSPACE**.*

The following lemma shows that the probability of hitting a given BSCC with a given stack content is effectively expressible in $\mathit{ExTh}(\mathbb{R})$. This lemma is proved in Section 7.4.1.

Lemma 7.2.2. *Let $L \subseteq \Gamma^*$ be a regular language, let $pX \in \mathbf{S}_\Delta$, and $C \in \mathit{BSCC}_\Delta$ such that $\mathcal{P}(\mathit{Run}(pX, C)) > 0$. Then $\mathcal{P}(\mathit{Hit}[L]{=}1 \mid \mathit{Run}(pX, C))$ is effectively expressible in $\mathit{ExTh}(\mathbb{R})$.*

To further simplify our notation, for the rest of this section we fix a distinguished initial configuration $q_0 Z_0$ of Δ such that $[q_0 Z_0 \uparrow] = 1$ (one can show, using similar arguments as in the proof of Lemma 4.3.11, that this assumption is without the loss of generality).

As a corollary to Theorem 7.2.1 and Lemma 7.2.2 we obtain the following (where $q_0 Z_0$ plays the role of the initial configuration):

Corollary 7.2.3. *Let $I \in \mathcal{I}$ be an indicator. The probability $\mathcal{P}(I{=}1)$ is effectively expressible in $ExTh(\mathbb{R})$. Moreover, if f is simple, and the sets $\mathbf{S}_\Delta$ and $\{C \in \mathit{BSCC}_\Delta \mid L_C = \Gamma^*\}$ have already been computed, then a formula expressing $\mathcal{P}(I{=}1)$ is computable in polynomial time.*

Proof. By Theorem 7.2.1 and the fact that all runs of $Run(q_0Z_0)$ are clean we obtain that $\mathcal{P}(I{=}1)$ equals

$$\sum_{C \in \mathrm{BSCC}_\Delta} \mathcal{P}(\mathit{Hit}[L_C]{=}1 \mid \mathit{Run}(q_0Z_0, C)) \cdot \mathcal{P}(\mathit{Run}(q_0Z_0, C))$$

Here the set BSCC_Δ is effectively computable by Lemma 4.3.8, the probability $\mathcal{P}(\mathit{Hit}[L_C]{=}1 \mid \mathit{Run}(q_0Z_0, C))$ is effectively expressible in $ExTh(\mathbb{R})$ due to Lemma 7.2.2, and $\mathcal{P}(\mathit{Run}(q_0Z_0, C))$ is equal to $\mathcal{P}(q_0Z_0 \hookrightarrow^* C)$ (by Proposition 4.3.10), which is effectively expressible in $ExTh(\mathbb{R})$ due to Lemma 4.3.9. Hence, $\mathcal{P}(I{=}1)$ is effectively expressible in $ExTh(\mathbb{R})$ due to Proposition 2.3.3.

Now let us assume that f is simple and let $B = \{C \in \mathrm{BSCC}_\Delta \mid L_C = \Gamma^*\}$. Then, by Theorem 7.2.1,

$$\begin{aligned} \mathcal{P}(I = 1) &= \textstyle\sum_{C \in B} \mathcal{P}(\mathit{Run}(q_0Z_0, C)) = \sum_{C \in B} \mathcal{P}(q_0Z_0 \hookrightarrow^* C) \\ &= \mathcal{P}(q_0Z_0 \hookrightarrow^* \textstyle\bigcup_{C \in B} C) \end{aligned}$$

Hence, the complexity estimate for simple reward functions follows immediately from Lemma 4.3.9. □

The following corollary summarizes the decidability and complexity results that follow immediately from the above results and Theorem 2.3.1.

Theorem 7.2.4. *Let $I \in \mathcal{I}$ be an indicator. The long-run average reward problem (i.e., the problem whether $\mathcal{P}(I{=}1) \sim \varrho$, where ϱ is a rational constant and $\sim \in \{<, \leq, >, \geq, =\}$), is decidable. For simple reward functions, this problem is in **EXPSPACE** (and in **PSPACE** in $|\Delta|$). Moreover, for simple reward functions and $I \in \mathcal{I}_{\mathbf{A}}$, this problem is in **PSPACE**.*

Theorem 7.2.1 can also be used to prove the following:

Theorem 7.2.5. *Each $I \in \mathcal{I}$ is well-predictable (over $Run(q_0Z_0)$).*

Proof sketch. Here we only sketch the intuitive idea behind the proof deferring the complete formal proof to Section 7.4.2. Let $\zeta > 0$. Due to Theorem 7.2.1, it suffices to compute a sufficiently large n satisfying the following property: the probability of all $w \in Run(q_0Z_0)$ such that the position of $Entry(w)$ in w is beyond the prefix of length n, is bounded by ζ. The value of $G^n(w)$ is then defined as follows: we take the first n configurations of w and identify the "developing minimal configurations", i.e., those configurations which become minimal configurations of w under the assumption that the stack length in all configurations $w(n), w(n{+}1), \ldots$ is not smaller than in $w(n{-}1)$. Thus, we also construct the "developing footprint"

of the run. Then we simply check whether this developing footprint hits a BSCC of $\mathbf{X}_\Delta$. If not, $G^n(w)$ returns 0. Otherwise, we identify the $Entry$ configuration $pX\beta$ and check whether $\beta \in L_C$, where C is the corresponding BSCC. If $\beta \in L_C$, then $G^n(w) = 1$. Otherwise, $G^n(w) = 0$.

□

Observe that the algorithm for computing $G^n(w)$ for given n and w is rather efficient, because the developing minima are identified just by comparing the stack length in configurations $w(0), \ldots, w(n-1)$. Of course, we also need to compute the transitions of $\mathbf{X}_\Delta$ (which can be done in space polynomial in the size of Δ due to Lemma 4.3.8), but this expensive computation is performed just once and can be done before starting the on-line analysis of a run initiated in $q_0 Z_0$.

Model-checking temporal logics with state predicates.

Let M be a Markov chain, let T be a set of triggers, let f be a reward function defined over states of M, and let $I \in \mathcal{I}$ be an indicator. For every $\sim \in \{<, \leq, >, \geq, =\}$ and every rational constant ϱ we define a *state predicate* $\mathcal{P}^{\sim\varrho}(I{=}1)$ as follows: a state s of M satisfies $\mathcal{P}^{\sim\varrho}(I{=}1)$ if and only if $\mathcal{P}(I{=}1) \sim \varrho$ (here $\mathcal{P}(I{=}1)$ is considered in the probabilistic space over $Run(s)$ and the indicator I is evaluated using the reward function f and the set of triggers T). State predicates can be plugged into state-based temporal logics (such as PCTL, PCTL*, or PCTL*) as "atomic propositions" in the style of [24], and thus one can combine the expressive power of state predicates with temporal operators.

In the rest of this section we assume that "usual" atomic propositions of temporal logics are interpreted using regular valuations that assign regular sets of configurations to all atomic propositions (see Definition 4.4.1). As we have already proved in Chapter 5, the model-checking problem is decidable for pPDA and the qualitative fragment of PECTL*. Now we investigate the decidability of this problem for pPDA and the above logics extended with the state predicates.

First we show that if the above logics are extended with predicates of the form $\mathcal{P}^{=1/2}(I{=}1)$, then even model checking the simple formula $\Diamond^{>0}(check \wedge \mathcal{P}^{=1/2}(I[\mathbf{A}, 1, 1]{=}1))$ becomes undecidable (the formula says "there is a reachable state satisfying the atomic proposition $check$ and the state predicate $\mathcal{P}^{=1/2}(I[\mathbf{A}, 1, 1]{=}1)$").

Theorem 7.2.6. *The model-checking problem for pPDA and the formula* $\Diamond^{>0}(check \wedge \mathcal{P}^{=1/2}(I[\mathbf{A}, 1, 1]{=}1))$ *is undecidable.*

Proof. Let us consider the pPDA Δ defined in Section 5.2.1 for a given instance of the Post Correspondence Problem. Let us assume that the atomic proposition $check$ is true precisely in configurations with the head cZ. By Lemma 5.2.1 and Lemma 5.2.2, the configuration gZ satisfies the PCTL formula $\Diamond^{>0}(check \wedge \Diamond^{=1/2}(tY))$ (here tY is an atomic proposition which is true precisely in configurations with the head tY) if and only if the instance of PCP has a solution.

Let f be a simple reward function, which assigns 1 to all configurations of Δ with the control state t, and 0 to others. Let us assume that the set of triggers is the set of all configurations with the control state t. Now, for all $w \in Run(cZ\alpha)$, we have $I[\mathbf{A},1,1](w) = 1$ if and only if $w(i) = tY\beta$ for some i and β if and only if $w \models \Diamond(tY)$. Hence, gZ satisfies $\Diamond^{>0}(check \wedge \mathcal{P}^{=1/2}(I[\mathbf{A},1,1]{=}1))$ if and only if gZ satisfies $\Diamond^{>0}(check \wedge \Diamond^{=1/2}(tY))$ if and only if the instance of PCP has a solution. □

The following theorem (proved in Section 7.4.3) indicates that if we only allow qualitative predicates of the form $\mathcal{P}^{=1}(I{=}1)$, the situation becomes different:

Theorem 7.2.7. *The model-checking problem for pPDA and qualitative PECTL* formulae extended with qualitative state predicates of the form $\mathcal{P}^{=1}(I{=}1)$, where $I \in \mathcal{I}$, is decidable.*

In particular, Theorem 7.2.7 applies to the predicate $\mathcal{P}^{=1}(I[\mathbf{A},\ell,u]{=}1)$ which has been considered in [24] for finite-state Markov decision processes.

7.3 Proof of Main Theorem

The goal of this section is to prove Theorem 7.2.1. We rely heavily on the notation and results introduced in Section 4.3.1 and Section 4.6. For the whole section we fix a pPDA $\Delta = (Q, \Gamma, \delta, Prob)$ and a subset $T \subseteq Q$ of control states. A configuration $p\alpha \in \mathcal{C}(\Delta)$ is a *trigger* if and only if $p \in T$. We also fix a well-defined reward function f such that for all $p\alpha \in \mathcal{C}(\Delta)$ we have $f(p\alpha) = g(p) + h(p) \cdot c(\alpha)$ for fixed functions $g, h : Q \to \mathbb{R}^+$ and $c : \Gamma \to \mathbb{R}^+$. Furthermore, for the whole section we fix $p_0X_0 \in \mathbf{S}_\Delta$ and $C \in \mathrm{BSCC}_\Delta$ such that $\mathcal{P}(Run(p_0X_0, C)) > 0$. We denote $\mathbf{X}_\Delta^C$ the restriction of $\mathbf{X}_\Delta$ to C. To simplify our notation, given $qY \in \mathbf{S}_\Delta$ and a set of runs A, we denote $\mathcal{P}_{qY}(A) = \mathcal{P}(A \mid Clean(qY))$.

Let us denote $LV = \{\mathbf{A}, \mathbf{D}[\kappa], \mathbf{R}[\xi,\lambda] \mid \xi,\kappa \in \mathbb{R}^+, \lambda \in \mathbb{R}^+_\infty, \xi \leq \lambda\}$ the collection of all "limit" variables considered in this chapter, which are relevant for non-negative reward functions. Each variable of LV, unless explicitly indexed by another reward function, is always evaluated using the reward function f. Hence, we write $\mathbf{A}^{f'}$, $\mathbf{D}^{f'}[\kappa]$, and $\mathbf{R}^{f'}[\xi,\lambda]$ instead of $\mathbf{A}$, $\mathbf{D}[\kappa]$, and $\mathbf{R}[\xi,\lambda]$, respectively, whenever we want to evaluate these variables using the reward function f' instead of f. Sometimes we work with an "unknown" variable $V \in LV$, in which case we write $V^{f'}$ instead of V to indicate that V is evaluated using the reward function f' instead of f.

Our goal is to show that for every $I \in \mathcal{I}$ there effectively exists a regular language $L_C \subseteq \Gamma^*$ such that $Hit[L_C] = I$ a.s. over $Run(p_0X_0, C)$. Obviously we can concentrate on indicators of the form I_{inf} and $I[V,\ell,u]$ where $V \in LV$ (Boolean connectives are then no problem, because these can be implemented just by performing an appropriate operation on the constructed regular languages).

First, let us consider the indicator I_{inf}. The following lemma follows immediately from results of Section 4.3.2:

Lemma 7.3.1. *There exists $I^C_{inf} \in \{0,1\}$, such that $I_{inf} = I^C_{inf}$ a.s. over $Run(pX, C)$, for all $pX \in \mathbf{S}_\Delta$ satisfying $\mathcal{P}(Run(pX, C)) > 0$. Moreover, the problem whether $I^C_{inf} = 1$ is in **PSPACE**.*

Hence, it suffices to define the language L_C to equal either to Γ^*, or to $\emptyset$, depending on whether $I^C_{inf} = 1$, or $I^C_{inf} = 0$, respectively.

Now let us consider an indicator of the form $I[V, \ell, u]$ where $V \in LV$. In order to prove that the language L_C exists, it suffices to show that for almost all $w \in Run(p_0X_0, C)$ the value $V(w)$ depends only on the *tail* of $Entry(w)$. The next lemma is the first step towards this result.

Remember that given $\beta \in \Gamma^*$, $Run(p_0X_0, C, \beta)$ denotes the set of all runs $w \in Run(p_0X_0, C)$ such that $tail(Entry(w)) = \beta$. We also denote f_β the well-defined reward function such that for all $r\alpha \in \mathcal{C}(\Delta)$ we have $f_\beta(r\alpha) = (g(r) + h(r) \cdot c(\beta)) + h(r) \cdot c(\alpha)$.

Lemma 7.3.2. *Let $V \in LV$ and $\beta \in \Gamma^*$. Let us assume that there is $V^{f_\beta}_C \in \mathbb{R}^+_\infty \cup \{\perp\}$ such that $V^{f_\beta} = V^{f_\beta}_C$ a.s. over $Clean(qY)$, for all $qY \in C$. Then $V = V^{f_\beta}_C$ a.s. over $Run(p_0X_0, C, \beta)$ whenever $\mathcal{P}(Run(p_0X_0, C, \beta)) > 0$.*

Proof. Let us fix $qY \in C$, and let us denote Ω the set of all runs $w \in Run(p_0X_0, C, \beta)$ such that for the least $i \geq 1$ satisfying $\mathbf{X}^i_\Delta(w) \in C$ we have $\mathbf{X}^i_\Delta(w) = qY$. Observe that it suffices to show that $V = V^{f_\beta}_C$ a.s. over Ω, because $qY \in C$ was chosen arbitrarily. Let us denote R the set of all $v \in FPath[\mathbf{X}_\Delta](p_0X_0)$ satisfying $last(v) \in C$ and $v(i) \notin C$ for all $i < |v|$, and let B be the set of all paths $v \in MPath(R)$ (see Definition 4.6.8) such that $last(v) = qY\beta$. By Lemma 4.6.9, each B is clean-prefix-free and $Clean(B) = \Omega$.

Let $v \in B$ and let $u \in Clean(qY)$. It is easy to prove using Proposition 2.0.2 that $V(v * u) = V(u\lfloor\beta)$ for any $V \in LV$. Observe that for all $i \geq 1$ we have $f(u[i]\lfloor\beta) = f(u[i]) + h(u[i]) \cdot c(\beta) = f_\beta(u[i])$ by Lemma 6.3.1, and hence that $V(v * u) = V(u\lfloor\beta) = V^{f_\beta}(u)$.

It follows from the above arguments that if we denote $A = \{u \in Clean(qY) \mid V^{f_\beta}(u) = V^{f_\beta}_C\}$, then for all $w \in \Omega$ we have $V(w) = V^{f_\beta}_C$ if and only if $w \in B{*}A$. Finally, by Lemma 4.6.7,

$$\begin{aligned}
\mathcal{P}(V = V^{f_\beta}_C \mid \Omega) &= \mathcal{P}(B * A \mid \Omega) = \frac{\mathcal{P}(B * A)}{\mathcal{P}(\Omega)} \\
&= \frac{\mathcal{P}(Clean(B)) \cdot \mathcal{P}_{qY}(A)}{\mathcal{P}(\Omega)} \\
&= \mathcal{P}(Clean(B) \mid \Omega) \cdot \mathcal{P}_{qY}(A) = \mathcal{P}_{qY}(A) \\
&= \mathcal{P}_{qY}(V^{f_\beta} = V^{f_\beta}_C) = 1
\end{aligned}$$

□

The above lemma suggests the following plan for the rest of this proof of Theorem 7.2.1: First, we show that for each variable $V \in LV$ there is an effectively expressible constant $V^f_C \in \mathbb{R}^+_\infty \cup \{\perp\}$ such that $V = V^f_C$ a.s. over $Clean(qY)$, for all

$qY \in C$ (Section 7.3.1, Section 7.3.2, and Section 7.3.3). Then by Lemma 7.3.2, the language L_C can be defined as the set of all $\beta \in \Gamma^*$ such that $\ell \leq V_C^{f_\beta} \leq u$. Finally, we use the effective expressibility of V_C^f to show that the language L_C is effectively regular (Section 7.3.4).

7.3.1 Expressibility of Gain

In this section we consider the gain per transition which is in fact a special case of the average reward per service. The gain is a function $\mathbf{G} : Run \to \mathbb{R}^+_\infty \cup \{\bot\}$ where:

$$\mathbf{G}(w) = \begin{cases} \lim_{n\to\infty} \frac{f(w^n)}{n} & \text{if the limit exists;} \\ \bot & \text{otherwise.} \end{cases}$$

Note that $\mathbf{G} = \mathbf{A}$ if all configurations are triggers. According to our notation, we write $\mathbf{G}^{f'}$ instead $\mathbf{G}$, whenever we want to evaluate the gain $\mathbf{G}$ using the reward function f' instead of f.

The main goal of this section is to prove that there is an effectively expressible constant $\mathbf{G}_C^f \in \mathbb{R}^+_\infty$ such that $\mathbf{G} = \mathbf{G}_C^f$ a.s. over $Clean(pX)$, for all $pX \in C$. We proceed as follows: First, for each $qY \in C$ we introduce a value denoted $E^f(qY)$, which expresses the expected accumulated reward between the first two minima of runs of $Clean(qY)$. We show that these values are effectively expressible in $ExTh(\mathbb{R})$ (Lemma 7.3.4). Then we show (Lemma 7.3.7) that $\mathbf{G}_C^f$ is equal to an arithmetical expression over the values $E^f(qY)$ and values of the invariant distribution of $\mathbf{X}_\Delta^C$ (that are also effectively expressible), whenever the reward function f satisfies the following homogeneity condition:

Definition 7.3.3. *A reward function f is* C-homogeneous *if for* all $qY \in C$, almost all $w \in Clean(qY)$, *and* all $i \geq 1$, *holds* $\zeta((w^{ind_{i+1}(w)})_{ind_i(w)}) = \zeta(\wp_i(w))$, *where* $\zeta(v) = f(v^{|v|-1})$ *for all* $v \in FPath[M_\Delta]$ *(*$\wp_i$ *is defined in Definition 4.6.10).*

Intuitively, f is C-homogeneous if for all $qY \in C$ and almost all $w \in Clean(qY)$, the following condition is satisfied: If $u = rZ\beta, r_1\alpha_1\beta, \ldots, r_k\alpha_k\beta$ is the path between the i^{th} minimum of w (inclusive) and $i+1^{th}$ minimum of w (not inclusive), then $f(u) = f(rZ, r_1\alpha_1, \ldots, r_k\alpha_k)$.

Unfortunately, well-defined reward functions are not C-homogeneous in general. However, we give an effectively checkable condition which identifies a subclass of C-homogeneous well-defined reward functions, and show that if f does not satisfy this condition, then $\mathbf{G} = \infty$ a.s. over $Clean(pX)$, for all $pX \in C$.

Gain and C-homogeneous reward functions

Given $qY \in \mathbf{S}_\Delta$ and $w \in Clean(qY)$, we denote $M_{qY}^f(w) = f(w^{ind_2(w)-1})$, and we denote $E^f(qY) = E(M_{qY}^f \mid Clean(qY))$ the expected accumulated reward between the first minimum (inclusive) and the second minimum (not inclusive). The next lemma shows that the values $E^f(qY)$ are effectively expressible in $ExTh(\mathbb{R})$.

Lemma 7.3.4. *Given $qY \in \mathbf{S}_\Delta$, the value $E^f(qY)$ equals*

$$\sum_{qY \xrightarrow{x} rZ} \frac{x[rZ\uparrow]}{[qY\uparrow]} \cdot f(qY) + \sum_{qY \xrightarrow{x} rZU} \frac{x[rZ\uparrow]}{[qY\uparrow]} \cdot f(qY) + \sum_{qY \xrightarrow{x} rZU, s \in Q} \frac{x[rZs][sU\uparrow]}{[qY\uparrow]} \cdot \mathcal{H}$$

where

$$\mathcal{H} = f(qY) + [E(rZs), f] + c(U) \cdot [E(rZs), h]$$

Hence, the value $E^f(qY) \in \mathbb{R}^+_\infty$ is effectively expressible in $ExTh(\mathbb{R})$ by a formula of polynomial size, computable in polynomial space.

Proof. Given $r, s \in Q$ and $Z, U \in \Gamma$, we denote

$$\begin{aligned} W_+^{rZU} &= ((qY \to rZ) \odot \mathit{Clean}(rZ)) \uplus ((qY \to rZU) \odot \mathit{Clean}(rZ)\lfloor U) \\ W_s^{rZU} &= (qY \to rZU) \odot ((\mathit{FPath}(rZs)\lfloor U) \odot \mathit{Clean}(sU)) \end{aligned}$$

It is easy to verify that $\mathit{Clean}(qY) = \biguplus_{r,s \in Q, Z, U \in \Gamma} W_+^{rZU} \uplus W_s^{rZU}$.

Given a run $w \in W_s^{rZU}$ of the form $(qY \to rYZ) \odot (v\lfloor U) \odot u$ where $v \in \mathit{FPath}(rZs)$ and $u \in \mathit{Clean}(sU)$, we denote $S^f(w) = f(v^{|v|-1})$, $S^h(w) = h(v^{|v|-1})$, and $\bar{c}(w) = c(U)$. Given $w \in W_+^{rZU}$, we define $S^f(w) = S^h(w) = \bar{c}(w) = 0$. By Lemma 6.3.1, for all $w \in \mathit{Clean}(qY)$

$$M^f_{qY}(w) = f(qY) + S^f(w) + \bar{c}(w) \cdot S^h(w)$$

Using Lemma 2.2.3, Lemma 4.6.2, and the linearity of the expectation, one can easily show that $E(M^f_{qY} \mid W_+^{rZU}) = f(qY)$ and

$$\begin{aligned} E(M^f_{qY} \mid W_s^{rZU}) &= f(qY) + E(S^f \mid W_s^{rZU}) + c(U) \cdot E(S^h \mid W_s^{rZU}) \\ &= f(qY) + [E(rZs), f] + c(U) \cdot [E(rZs), h] \end{aligned}$$

Hence,

$$\begin{aligned} E^f(qY) &= E(M^f_{qY} \mid \mathit{Clean}(qY)) \\ &= \sum_{qY \xrightarrow{x} rZU} \mathcal{P}(W_+^{rZU} \mid \mathit{Clean}(qY)) \cdot E(M^f_{qY} \mid W_+^{rZU}) + \\ &+ \sum_{qY \xrightarrow{x} rZU, s \in Q} \mathcal{P}(W_0^{rZU,s} \mid \mathit{Clean}(qY)) \cdot E(M^f_{qY} \mid W_0^{rZU,s}) \\ &= \sum_{qY \xrightarrow{x} rZ} \frac{x[rZ\uparrow]}{[qY\uparrow]} \cdot f(qY) + \sum_{qY \xrightarrow{x} rZU} \frac{x[rZ\uparrow]}{[qY\uparrow]} \cdot f(qY) + \\ &+ \sum_{qY \xrightarrow{x} rZU, s \in Q} \frac{x[rZs][sY\uparrow]}{[qY\uparrow]} \cdot (f(qY) + [E(rZs), f] + c(U) \cdot [E(rZs), h]) \end{aligned}$$

Now the effective expressibility of $E^f(qY)$ follows from Corollary 4.1.10, Theorem 6.2.8, and Proposition 2.3.3.

□

Next we show (Lemma 7.3.7, below) that the gain $\mathbf{G}$ can be expressed using the values $E^f(qY)$, whenever f is C-homogeneous. In order to prove this result, we need the following two results (Proposition 7.3.5 and Proposition 7.3.6) from the theory of the finite Markov chains.

Proposition 7.3.5. *Let $M = (S, \to, Prob)$ be a strongly connected* finite *Markov chain. Then there exists a* unique *vector $\mu \in \mathbb{R}^{|S|}$ (the* invariant distribution*) with the following properties:*

1. $\sum_{s \in S} \mu(s) = 1$;
2. *for all $s \in S$, we have $\sum_{t \to s} \mu(t) \cdot Prob(t \to s) = \mu(s)$.*

Moreover, $\mu(s) > 0$ for all $s \in S$.

Proof. Standard results of the theory of finite Markov chains (see, e.g., [41]) imply that there is a unique *positive* solution to the above system. However, by Lemma 6.2.3, this positive solution is the unique solution of the system in $\mathbb{R}$. □

Proposition 7.3.6 ([41])**.** *Let $M = (S, \to, Prob)$ be a* finite *Markov chain. Let $C_1, \ldots, C_n$ be all BSCCs of M, and let $\mu_1, \ldots, \mu_n$ be their invariant distributions, respectively. For almost every run $w \in Run(s)$ there is $1 \leq k \leq n$ such that w gets trapped within C_k, and moreover, for all $t \in C_k$*

$$\lim_{m \to \infty} \frac{|\{1 \leq i \leq m \mid w(i) = t\}|}{m} = \mu_k(t)$$

The following lemma generalizes Theorem 4.7 of [28], and is proved using similar techniques.

Lemma 7.3.7. *Let $pX \in C$. If f is C-homogeneous, then for almost all $w \in Clean(pX)$*

$$\mathbf{G}(w) = \lim_{k \to \infty} \frac{f(w^k)}{k} = \frac{\sum_{qY \in C} \mu(qY) \cdot E^f(qY)}{\sum_{qY \in C} \mu(qY) \cdot E^{\mathbf{1}}(qY)}$$

where μ is the invariant probability distribution of $\mathbf{X}_\Delta^C$ and $\mathbf{1}$ assigns the reward 1 to each configuration.

Proof. For every $i \geq 1$, we define a function $M_i^f : Run(pX) \to \mathbb{R}^+$ as follows: Given $w \in Clean(pX)$, we put $M_i^f(w) = f((w^{ind_{i+1}(w)-1})_{ind_i(w)})$ the reward accumulated between the i^{th} minimum (inclusive) and $i + 1^{th}$ minimum (not inclusive). Given $w \in Run(pX) \setminus Clean(pX)$, we put $M_i^f(w) = 0$. Each M_i^f is a discrete random variable. Indeed, each M_i^f is constant over $Clean(w^{ind_{i+1}(w)})$, which is measurable by Lemma 4.6.7. It is clear that each M_i^f is discrete as there are only countably many distinct sets of the form $Clean(w^{ind_{i+1}(w)})$.

Given $w \in Clean(pX)$, we have

$$\begin{aligned}\mathbf{G}(w) &= \lim_{k\to\infty} \frac{f(w^k)}{k} = \lim_{k\to\infty} \frac{\sum_{i=1}^k M_i^f(w)}{\sum_{i=1}^k M_i^{\mathbf{1}}(w)} \\ &= \lim_{k\to\infty} \frac{\frac{1}{k}\sum_{i=1}^k M_i^f(w)}{\frac{1}{k}\sum_{i=1}^k M_i^{\mathbf{1}}(w)} = \frac{\lim_{k\to\infty}\frac{1}{k}\sum_{i=1}^k M_i^f(w)}{\lim_{k\to\infty}\frac{1}{k}\sum_{i=1}^k M_i^{\mathbf{1}}(w)}\end{aligned}$$

whenever $\mathbf{G}(w)$ and the last expression are both defined.

Given a run $w \in Clean(pX)$, a state $qY \in C$ and $k \geq 1$, we denote $K[qY,k](w) = \{1 \leq i \leq k \mid \mathbf{X}_\Delta^i(w) = qY\}$. Then for all runs $w \in Good(pX)$ (and hence, by Lemma 4.3.6, for almost all $w \in Clean(pX)$)

$$\begin{aligned}\lim_{k\to\infty}\frac{1}{k}\sum_{i=1}^k M_i^f(w) &= \lim_{k\to\infty}\frac{1}{k}\sum_{qY\in C}\sum_{i\in K[qY,k](w)} M_i^f(w) \\ &= \lim_{k\to\infty}\frac{1}{k}\sum_{qY\in C}|K[qY,k](w)|\cdot\frac{\sum_{i\in K[qY,k](w)} M_i^f(w)}{|K[qY,k](w)|} \\ &= \lim_{k\to\infty}\sum_{qY\in C}\frac{|K[qY,k](w)|}{k}\cdot\frac{\sum_{i\in K[qY,k](w)} M_i^f(w)}{|K[qY,k](w)|} \\ &= \sum_{qY\in C}\left(\lim_{k\to\infty}\frac{|K[qY,k](w)|}{k}\cdot\lim_{k\to\infty}\frac{\sum_{i\in K[qY,k](w)} M_i^f(w)}{|K[qY,k](w)|}\right)\end{aligned}$$

whenever the last expression is defined.

Claim (1)**.** *Given $qY \in C$, for almost all $w \in Clean(pX)$*

$$\lim_{k\to\infty}\frac{|K[qY,k](w)|}{k} = \mu(qY) > 0$$

Proof of the Claim. By Proposition 7.3.6, there is a set $A \subseteq Run[\mathbf{X}_\Delta](pX)$ satisfying $\mathcal{P}(A) = 1$ such that for all $v \in A$ and $qY \in C$ the limit $\lim_{k\to\infty}\frac{|\{1\leq i\leq k|v(i)=qY\}|}{k}$ equals $\mu(qY) > 0$. It follows from Lemma 4.3.6 that $\mathcal{P}_{pX}(fp^{-1}(A)) = \mathcal{P}(A) = 1$. Finally, observe that if $w \in fp^{-1}(v)$ for $v \in A$, then $\lim_{k\to\infty}\frac{|K[qY,k](w)|}{k} = \lim_{k\to\infty}\frac{|\{1\leq i\leq k|v(i)=qY\}|}{k} = \mu(qY)$.

$\Diamond$

Claim (2)**.** *Given $qY \in C$, for almost all $w \in Clean(pX)$*

$$\lim_{k\to\infty}\frac{\sum_{i\in K[qY,k](w)} M_i^f(w)}{|K[qY,k](w)|} = E^f(qY)$$

Proof of the Claim. We define a sequence of functions $N_1, N_2, \ldots$ over $Run(pX)$ as follows: Given $i \geq 1$ and a run $w \in Run(pX)$, we put

$$N_i(w) = \begin{cases} M_k^f(w) & \text{if } \mathbf{X}_\Delta^k(w) = qY,\ |\{0 \leq j \leq k \mid \mathbf{X}_\Delta^j(w) = qY\}| = i; \\ 0 & \text{otherwise.}\end{cases}$$

Intuitively: $N_i(w)$ equals $M_k^f(w)$, where k is the index of i^{th} occurrence of qY in $fp(w)$. If either $w \notin \mathit{Clean}(pX)$ or qY occurs less than i times in $fp(w)$, then $N_k(w) = 0$.

We show that for all $i \geq 0$ the functions N_i are discrete random variables. Let $w \in \mathit{Clean}(pX)$ be such that $N_i(w) = M_k^f(w)$. It follows that for all $w' \in \mathit{Clean}(w^{ind_{k+1}(w)})$ we have $N_i(w') = M_k^f(w')$. However, $M_k^f(w')$ is constant over the measurable set $\mathit{Clean}(w^{ind_{k+1}(w)})$, and hence also N_i is constant over $\mathit{Clean}(w^{ind_{k+1}(w)})$. It follows that N_i is a discrete random variable.

It follows from Lemma 4.6.11 and the fact that f is C-homogeneous, that the sequence of the variables $N_1, N_2, \ldots$ is independent w.r.t. $\mathcal{P}_{pX}$, and that for all $i \geq 1$ and all $x \in \mathbb{R}$, we have $\mathcal{P}_{pX}(N_i = x) = \mathcal{P}_{qY}(M_{qY}^f = x)$. (To apply Lemma 4.6.11 define $\zeta(v) = f(v^{|v|-1})$ for all $v \in \mathit{FPath}(qY)$. Then $M_{qY}^f = \zeta \circ \wp_1$, and the variables $X_1, X_2, \ldots$ become the variables $N_1, N_2, \ldots$, respectively.) Thus, by Theorem 2.1.2, for almost all $w \in \mathit{Clean}(pX)$

$$\begin{aligned}\lim_{k\to\infty} \frac{\sum_{i\in K[qY,k](w)} M_i^f(w)}{|K[qY,k](w)|} &= \lim_{k\to\infty} \frac{\sum_{i=1}^k N_i(w)}{k} = E(N_1 \mid \mathit{Clean}(pX)) \\ &= E(M_{qY}^f \mid \mathit{Clean}(qY)) = E^f(qY)\end{aligned}$$

which proves Claim 2.

$\Diamond$

By Claim (1) and Claim (2), we obtain that for almost all runs $w \in \mathit{Clean}(pX)$

$$\lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} M_i^f(w) = \sum_{qY\in C} \mu(qY) \cdot E^f(qY) \tag{7.1}$$

As a special case, for almost all $w \in \mathit{Clean}(pX)$

$$\lim_{k\to\infty} \frac{1}{k} \sum_{i=1}^{k} M_i^{\mathbf{1}}(w) = \sum_{qY\in C} \mu(qY) \cdot E^{\mathbf{1}}(qY)$$

Our assumptions about Δ (see Section 7.2) and Lemma 7.3.4 imply that $\sum_{qY\in C} \mu(qY) \cdot E^{\mathbf{1}}(qY) < \infty$. Hence, for almost all $w \in \mathit{Clean}(pX)$

$$\lim_{k\to\infty} \frac{\sum_{i=1}^k M_i^f(w)}{\sum_{i=1}^k M_i^{\mathbf{1}}(w)} = \frac{\sum_{qY\in C} \mu(qY) \cdot E^f(qY)}{\sum_{qY\in C} \mu(qY) \cdot E^{\mathbf{1}}(qY)}$$

It remains to show that $\mathbf{G}(w)$ is defined for almost all $w \in \mathit{Clean}(pX)$. Given $k \geq 1$ and $w \in \mathit{Clean}(pX)$ we denote $\ell(k,w)$ the number such that $ind_{\ell(k,w)}(w) \leq k$ and $ind_{\ell(k,w)+1}(w) > k$. It is easy to see that for all $k \geq ind_2(w)$ we have

$$\frac{\sum_{i=1}^{\ell(k,w)-1} M_i^f(w)}{\sum_{i=1}^{\ell(k,w)} M_i^{\mathbf{1}}(w)} \leq \frac{f(w^k)}{\mathbf{1}(w^k)} \leq \frac{\sum_{i=1}^{\ell(k,w)} M_i^f(w)}{\sum_{i=1}^{\ell(k,w)-1} M_i^{\mathbf{1}}(w)} \tag{7.2}$$

Now for almost all $w \in Clean(pX)$ we have

$$\lim_{k\to\infty} \frac{\sum_{i=1}^{\ell(k,w)} M_i^f(w)}{\sum_{i=1}^{\ell(k,w)-1} M_i^{\mathbf{1}}(w)} = \lim_{k\to\infty} \frac{\frac{1}{k}\sum_{i=1}^{k} M_i^f(w)}{\frac{1}{k}\sum_{i=1}^{k-1} M_i^{\mathbf{1}}(w)} = \frac{\sum_{qY\in C} \mu(qY)\cdot E^f(qY)}{\sum_{qY\in C} \mu(qY)\cdot E^{\mathbf{1}}(qY)} \quad (7.3)$$

where the second equation follows from the equation (7.1) because

$$\begin{aligned}\lim_{k\to\infty} \frac{\sum_{i=1}^{k-1} M_i^f(w)}{k} &= \lim_{k\to\infty} \frac{\sum_{i=1}^{k-1} M_i^f(w)}{k-1}\cdot\frac{k-1}{k} \\ &= \lim_{k\to\infty} \frac{\sum_{i=1}^{k-1} M_i^f(w)}{k-1}\cdot \lim_{k\to\infty}\frac{k-1}{k} = \sum_{qY\in C}\mu(qY)\cdot E^f(qY)\end{aligned}$$

Similarly, one can prove that for almost all $w \in Clean(pX)$

$$\lim_{k\to\infty} \frac{\sum_{i=1}^{\ell(k,w)-1} M_i^f(w)}{\sum_{i=1}^{\ell(k,w)} M_i^{\mathbf{1}}(w)} = \lim_{k\to\infty} \frac{\frac{1}{k}\sum_{i=1}^{k-1} M_i^f(w)}{\frac{1}{k}\sum_{i=1}^{k} M_i^{\mathbf{1}}(w)} = \frac{\sum_{qY\in C} \mu(qY)\cdot E^f(qY)}{\sum_{qY\in C} \mu(qY)\cdot E^{\mathbf{1}}(qY)} \quad (7.4)$$

Putting the equations (7.4), (7.3), and (7.2) together we obtain the desired result. □

In what follows we denote $E_C^f = \frac{\sum_{qY\in C} \mu(qY)\cdot E^f(qY)}{\sum_{qY\in C} \mu(qY)\cdot E^{\mathbf{1}}(qY)}$.

Lemma 7.3.8. *The value $E_C^f \in \mathbb{R}_\infty^+$ is effectively expressible in $ExTh(\mathbb{R})$ by a formula of polynomial size, computable in polynomial space.*

Proof. By Lemma 7.3.5, the invariant distribution μ is the unique solution of a system of linear equations whose coefficients are transition probabilities of $\mathbf{X}_\Delta^C$. Lemma 4.3.8 implies that this system of equations (together with formulae expressing its coefficients) is of polynomial size, and is computable in polynomial space. Hence, by Proposition 2.3.3, the components of the invariant distribution are effectively expressible in $ExTh(\mathbb{R})$ by formulae of polynomial size, computable in polynomial space.

Note that all values $E^{\mathbf{1}}(qY)$ are finite due to our assumptions, and that $E_C^f = \infty$ if and only if $E^f(qY) = \infty$ for some $qY \in C$. Then, by Lemma 7.3.4 and Proposition 2.3.3, the value E_C^f is effectively expressible in $ExTh(\mathbb{R})$ by a formula of polynomial size, computable in polynomial space. □

Gain and well-defined reward functions

It follows immediately from Lemma 7.3.7 and Lemma 7.3.8 that whenever f is C-homogeneous, then $\mathbf{G} = E_C^f$ a.s. over $Clean(pX)$, for all $pX \in C$, and moreover, E_C^f is effectively expressible in $ExTh(\mathbb{R})$. As we have mentioned above, the problem with general well-defined reward functions is that they does not have to be C-homogeneous. In what follows we give an effectively checkable condition which identifies a subclass of C-homogeneous well-defined reward functions, and show that if f does not satisfy this condition, then $\mathbf{G} = \infty$ a.s. over $Clean(pX)$, for all $pX \in C$.

Definition 7.3.9. *A transition $qY \hookrightarrow sU$ of $\mathbf{X}_\Delta$ is* active *if there is a path $v \in MPath(qY, sU)$ such that $h(v) > 0$. Moreover, a transition $qY \hookrightarrow sU$ is* strictly increasing *if $qY \to sUV$ in Δ, where $c(V) > 0$.*

The following lemma shows how the presence of the strictly increasing and active transitions in $\mathbf{X}_\Delta$ influences the behavior of Δ on the "long run". It implies that f is C-homogeneous if there are either no strictly increasing transitions or no active transitions in $\mathbf{X}^C_\Delta$. The lemma is formulated in a slightly more general way because we will refer to it several times in the future (in different contexts).

Lemma 7.3.10.

1. *Let $qY \in \mathbf{S}_\Delta$ (not necessarily in C). If there are* no strictly increasing transitions *reachable from qY in $\mathbf{X}_\Delta$, then for all $w \in Good(qY)$ and all $i \geq 1$, we have $c(tail(\min_i(w))) = 0$. In particular, if there are no strictly increasing transitions in $\mathbf{X}^C_\Delta$, then f is C-homogeneous.*

2. *Let $qY \in \mathbf{S}_\Delta$ (not necessarily in C). If there are* no active transitions *reachable from qY in $\mathbf{X}_\Delta$, then for all $w \in Good(qY)$ and all $i \geq 0$, we have $h(w(i)) = 0$ and $f(w(i)) = g(w(i))$. In particular, if there are no active transitions in $\mathbf{X}^C_\Delta$, then f is C-homogeneous.*

3. *If $\mathbf{X}^C_\Delta$ contains* at least one strictly increasing transition, *then for all $pX \in C$ and almost all $w \in Clean(pX)$ holds*

$$\forall \lambda > 0 : \exists i \geq 0 : \forall j \geq i : f(w(j)) \geq h(w(j)) \cdot \lambda$$

4. *If $\mathbf{X}^C_\Delta$ contains* at least one active transition, *then $E^h_C > 0$.*

5. *If $\mathbf{X}^C_\Delta$ contains both* strictly increasing transitions *and* active transitions, *then $\mathbf{G} = \infty$ a.s. over $Clean(pX)$, for all $pX \in C$.*

Proof. 1. Let $w \in Good(qY)$, $i \geq 1$, and let us denote $\min_i(w) = rZ\alpha$. We prove that $c(\alpha) = 0$. To the contrary, assume that $c(\alpha) > 0$ and let us assume that i is the least index with such a property. It is clear that $i \geq 2$, and by the minimality of i we have $\min_{i-1}(w) = r'Z'\beta$ where $c(\beta) = 0$. It follows that $\alpha = Z''\beta$ where $r'Z' \to rZZ''$ and $c(Z'') > 0$ (because $c(\alpha) > 0$ and $c(\beta) = 0$). Moreover, $w \in Good(qY)$, and hence $r'Z' \hookrightarrow rZ$ is a strictly increasing transition reachable from qY, which contradicts our assumption.

2. Let $w \in Good(qY)$ and $i \geq 1$. Let us suppose that $\mathbf{X}^i_\Delta(w) = rZ$ and $\mathbf{X}^{i+1}_\Delta(w) = sU$. Then $rZ \hookrightarrow sU$ is a transition in $\mathbf{X}^C_\Delta$ because $fp(w)$ is a run in $\mathbf{X}_\Delta$. If it was $h(\wp_i(w)) > 0$ (see Definition 4.6.10), then the path $\wp_i(w)$ would witness that $rZ \hookrightarrow sU$ is an active transition in $\mathbf{X}_\Delta$. Hence, $h(\wp_i(w)) = 0$ and $f(\wp_i(w)) = g(\wp_i(w))$, for all $i \geq 0$, which implies the desired result.

3. Let $qY \hookrightarrow sU$ be a strictly increasing transition in $\mathbf{X}_\Delta^C$ due to a transition $qY \to sUZ$, where $c(Z) > 0$. It follows immediately from Corollary 4.6.12 that for almost all runs $w \in \mathit{Clean}(qY)$ there are infinitely many indexes $i \geq 0$ such that $\min_i(w) = qY\alpha$ and $\min_{i+1}(w) = sUZ\alpha$. It follows that almost all runs $w \in \mathit{Clean}(qY)$ have the following property: For each $\lambda > 0$ there is $i \geq 0$ such that for all $j \geq i$ we have $\#_Z(\mathit{tail}(w(j))) \geq \frac{\lambda}{c(Z)}$. Thus if we denote $w(j) = r\alpha$, then

$$f(w(j)) = f(r\alpha) \geq h(r) \cdot c(Z) \cdot \frac{\lambda}{c(Z)} = h(w(j)) \cdot \lambda$$

4. Clearly, there is an active transition $qY \hookrightarrow sU$ in $\mathbf{X}_\Delta^C$, such that for some $u \in \mathit{MPath}(qY, sU)$ we have not only $h(u) > 0$, but also $h(u^{|u|-1}) > 0$. We show that $E^h(qY) > 0$. The rest follows from the definition of E_C^h and the positiveness of the invariant distribution of $\mathbf{X}_\Delta^C$ (see Proposition 7.3.5).

   Let us denote $v = u^{|u|-1}$. By the definition of $\mathbf{X}_\Delta$ we have $[qY\uparrow] > 0$ and $[sU\uparrow] > 0$, and hence by Lemma 4.6.7,

$$\mathcal{P}_{qY}(u * \mathit{Clean}(sU)) \cdot h(v) = \frac{\mathcal{P}(\mathit{Run}(u)) \cdot [sU\uparrow]}{[qY\uparrow]} \cdot h(v) > 0$$

   Moreover, for all $w \in u * \mathit{Clean}(sU)$, we have $M_{qY}^h(w) = h(v)$. By the definition of the expectation,

$$\begin{aligned} E^h(qY) &= E(M_{qY}^h \mid \mathit{Clean}(qY)) \geq \mathcal{P}_{qY}(M_{qY}^h = h(v)) \cdot h(v) \\ &\geq \mathcal{P}_{qY}(u \odot \mathit{Clean}(sU)) \cdot h(v) > 0 \end{aligned}$$

5. Let $pX \in C$. By 3., 4., and Proposition 2.0.2, for almost all $w \in \mathit{Clean}(pX)$ and arbitrary $\lambda > 0$ holds

$$\mathbf{G}(w) = \lim_{k\to\infty} \frac{f(w^k)}{k} \geq \lim_{k\to\infty} \frac{h(w^k) \cdot \lambda}{k} = \lambda \cdot E_C^h > 0$$

   Hence, $\mathbf{G}(w) \geq \lim_{\lambda\to\infty} \lambda \cdot E_C^h = \infty$.

□

We define

$$\mathbf{G}_C^f = \begin{cases} E_C^f & \text{no } \textit{strictly increasing} \text{ or no } \textit{active} \text{ transitions in } \mathbf{X}_\Delta^C; \\ \infty & \text{otherwise.} \end{cases}$$

Corollary 7.3.11. *$\mathbf{G} = \mathbf{G}_C^f$ a.s. over $\mathit{Clean}(pX)$, for all $pX \in C$. Moreover, the value $\mathbf{G}_C^f \in \mathbb{R}_\infty^+$ is effectively expressible in $ExTh(\mathbb{R})$ by a formula of polynomial size, computable in polynomial space.*

Proof. By Lemma 4.3.8, the underlying transition system of $\mathbf{X}_\Delta$ can effectively be computed in polynomial space. The existence of strictly increasing and active transitions in $\mathbf{X}_\Delta^C$ can be decided in polynomial time using standard methods for non-probabilistic PDA (see [29]). The rest follows from Lemma 7.3.10, Lemma 7.3.7, and Lemma 7.3.8. □

7.3.2 Expressibility of Average Reward

We show that the average reward $\mathbf{A}$ is expressible using the gain $\mathbf{G}$. Let us define a simple reward function $t : \mathcal{C}(\Delta) \to \{0, 1\}$ that given $p\alpha$ indicates whether $p\alpha$ is a trigger, i.e.,

$$t(p\alpha) = \begin{cases} 1 & p \in T; \\ 0 & \text{otherwise.} \end{cases}$$

The following lemma connects the gain $\mathbf{G}^t$ with the indicator I_{inf} (we use the notation of Lemma 7.3.1).

Lemma 7.3.12. $\mathbf{G}^t_C > 0$ *if and only if* $I^C_{inf} = 1$.

Proof. Let us fix $pX \in C$. First, let us assume that $\mathbf{G}^t_C > 0$. Then, by Corollary 7.3.11, $\mathbf{G}^t > 0$ a.s. over $Clean(pX)$. It follows that $I_{inf} = 1$ a.s. over $Clean(pX)$, and hence $I^C_{inf} = 1$ by Lemma 7.3.1.

Now let $\mathbf{G}^t_C = 0$. Then, by Lemma 7.3.10 (4. and 2., where we substitute h with t), for all $w \in Good(pX)$ and all $i \geq 0$, we have $t(w(i)) = 0$. It follows that $I_{inf}(w) = 0$ a.s. over $Clean(pX)$, and hence $I^C_{inf} = 0$ by Lemma 7.3.1. □

We denote

$$\mathbf{A}^f_C = \begin{cases} \frac{\mathbf{G}^f_C}{\mathbf{G}^t_C} & \text{if } \mathbf{G}^t_C > 0; \\ \bot & \text{otherwise.} \end{cases}$$

Theorem 7.3.13. $\mathbf{A} = \mathbf{A}^f_C$ *a.s. over* $Clean(pX)$*, for all* $pX \in C$*. Moreover, the value* $\mathbf{A}^f_C \in \mathbb{R}^+_\infty \cup \{\bot\}$ *is effectively expressible in* $ExTh(\mathbb{R})$ *by a formula of polynomial size, computable in polynomial space.*

Proof. If $\mathbf{G}^t_C = 0$, then $I^C_{inf} = 0$ by Lemma 7.3.12. Hence, $\mathbf{A} = \bot = \mathbf{A}^f_C$ a.s. over $Clean(pX)$, for all $pX \in C$, by Lemma 7.3.1 and the definition of $\mathbf{A}$. Moreover, the problem whether $\mathbf{G}^t_C = 0$ is decidable in polynomial space due to Lemma 7.3.12 and Lemma 7.3.1.

Now let us assume that $\mathbf{G}^t_C > 0$, and hence that $I^C_{inf} = 1$ (by Lemma 7.3.12). Let us denote

$$A = \{w \in Clean(pX) \mid \mathbf{G}(w) \neq \bot, \mathbf{G}^t(w) > 0\}$$

It follows from Corollary 7.3.11 that $\mathcal{P}(A \mid Clean(pX)) = 1$. Also, every run $w \in A$ contains infinitely many triggers because $\mathbf{G}^t(w) > 0$. If we denote $\bar{w}$ a suffix of $w \in A$ such that $\bar{w}(0)$ is a trigger, then

$$\frac{\mathbf{G}(w)}{\mathbf{G}^t(w)} = \frac{\mathbf{G}(\bar{w})}{\mathbf{G}^t(\bar{w})} = \frac{\lim_{n\to\infty} \frac{f(\bar{w}^n)}{n}}{\lim_{n\to\infty} \frac{t(\bar{w}^n)}{n}} = \lim_{n\to\infty} \frac{\frac{f(\bar{w}^n)}{n}}{\frac{t(\bar{w}^n)}{n}} = \lim_{n\to\infty} \frac{f(\bar{w}^n)}{t(\bar{w}^n)} = \mathbf{A}(\bar{w}) = \mathbf{A}(w)$$

where the first and the last equation follow from Proposition 2.0.2, the third equation follows from Proposition 2.0.1, and the fifth one follows from the fact that $\mathbf{A}(\bar{w})$ is the limit of a subsequence of the sequence $\frac{f(\bar{w}^0)}{t(\bar{w}^0)}, \frac{f(\bar{w}^1)}{t(\bar{w}^1)}, \ldots$. Hence, $\mathbf{A} = \frac{\mathbf{G}^f_C}{\mathbf{G}^t_C}$. The rest follows from Corollary 7.3.11.

□

7.3.3 Expressibility of Average Deviation and Ratio

Our goal in this section is to show how the average deviation and ratio can be expressed using the average reward. Let us fix $\xi \in \mathbb{R}^+$. In this section we concentrate only on the variables of the form $\mathbf{D}[\xi]$ and $\mathbf{R}[\xi, \infty]$. This restriction is without the loss of generality because $\mathbf{R}[\xi, \lambda]$ is equal to $\mathbf{R}[\xi, \infty] - \mathbf{R}[y, \infty]$ for a suitable y (see Section 7.3.4).

The following lemma shows that if there are some strictly increasing transitions in $\mathbf{X}^C_\Delta$ then, in principle, the reward function f can be substituted with a simple reward function (eliminating thus the strictly increasing transitions).

Lemma 7.3.14. *Let us assume that $I^C_{inf} = 1$ and that there is at least one strictly increasing transition in $\mathbf{X}^C_\Delta$.*

1. *If there is at least one active transition in $\mathbf{X}^C_\Delta$, then $\mathbf{D}[\xi] = \infty$ a.s. over $Clean(pX)$, for all $pX \in C$. If there are no active transitions in $\mathbf{X}^C_\Delta$, then $\mathbf{D}[\xi] = \mathbf{D}^g[\xi]$ a.s. over $Clean(pX)$, for all $pX \in C$.*

2. *$\mathbf{R}[\xi, \infty] = \mathbf{R}^{f'}[\xi, \infty]$ a.s. over $Clean(pX)$, for all $pX \in C$, where we have $f'(q\alpha) = g(q) + h(q) \cdot \frac{\xi}{\min(h)}$ (here $\min(h) = \min\{h(q) \mid h(q) > 0\}$).*

Proof. 1. First, let us assume that there is at least one active transition in $\mathbf{X}^C_\Delta$. For all $pX \in C$ and all $w \in Clean(pX)$, where $I_{inf}(w) = 1$, we have

$$\frac{\sum_{i=1}^k |f(w[i]) - \xi|}{k} \geq \frac{\sum_{i=1}^k (f(w[i]) - \xi)}{k} = \frac{\sum_{i=1}^k f(w[i])}{k} - \xi$$

Hence, for almost all $w \in Clean(pX)$, holds

$$\mathbf{D}[\xi] = \lim_{k\to\infty} \frac{\sum_{i=1}^k |f(w[i]) - \xi|}{k} \geq \lim_{k\to\infty} \frac{\sum_{i=1}^k f(w[i])}{k} - \xi = \mathbf{A}^f_C - \xi = \infty$$

because $\mathbf{A}^f_C = \frac{\mathbf{G}^f_C}{\mathbf{G}^t_C}$ (by Theorem 7.3.13), and $\mathbf{G}^f_C = \infty$ (by definition). If there are no active transitions in $\mathbf{X}^C_\Delta$, then it suffices to apply Lemma 7.3.10.

2. By Lemma 7.3.10, for all $pX \in C$ and almost every run $w \in Clean(pX)$, there is $i \geq 0$ such that for all $j \geq i$ we have $f(w(j)) \geq h(w(j)) \cdot \frac{\xi}{\min(h)}$. Let $w \in Clean(pX)$ be such a run. Now for all k such that the service $w[k]$ starts after the i'th step in w, we have $B(w[k], \xi, \infty) = 1$ if and only if either $h(w[k]) > 0$ or $g(w[k]) \geq \xi$, which holds if and only if $f'(w[k]) \geq \xi$. By Proposition 2.0.2, $\mathbf{R}[\xi, \infty](w) = \mathbf{R}^{f'}[\xi, \infty](w)$.

□

Now we consider the case where there are no strictly increasing transitions in $\mathbf{X}_\Delta$. We show that there are effectively expressible values $\mathbf{D}^f_C[\xi]$ and $\mathbf{R}^f_C[\xi,\infty]$ such that $\mathbf{D}[\xi] = \mathbf{D}^f_C[\xi]$ and $\mathbf{R}[\xi,\infty] = \mathbf{R}^f_C[\xi,\infty]$ a.s. over $Clean(pX)$, for all $pX \in C$.

First, we prove this claim under the following additional assumption: For all $pX \in C$ and all $q\alpha$ reachable from pX in M_Δ, we have that $g(q) > \xi$ whenever $h(q) > 0$. This assumption allows us to decide whether the reward accumulated along the current service (in a given run) already exceeded the threshold ξ by observing only control states of configurations (i.e., ignoring the stack contents). Later we alleviate this assumption (Lemma 7.3.16) by showing that necessary information about the stack contents can be encoded into control states.

Lemma 7.3.15. *Let us assume that for all $pX \in C$ and all $q\alpha$ reachable from pX in M_Δ, we have $g(q) > \xi$ whenever $h(q) > 0$. We also assume that $I^C_{inf} = 1$ and that there are no strictly increasing transitions in $\mathbf{X}^C_\Delta$. Then there are values $\mathbf{D}^f_C[\xi], \mathbf{R}^f_C[\xi,\infty] \in \mathbb{R}^+_\infty \cup \{\bot\}$ such that $\mathbf{D}[\xi] = \mathbf{D}^f_C[\xi]$ and $\mathbf{R}[\xi,\infty] = \mathbf{R}^f_C[\xi,\infty]$ a.s. over $Clean(pX)$, for all $pX \in C$.*

Moreover, the values $\mathbf{D}^f_C[\xi]$ and $\mathbf{R}^f_C[\xi,\infty]$ are effectively expressible in $ExTh(\mathbb{R})$ by formulae of size polynomial in $|\Delta| \cdot 2^{|\xi|+|f|}$, computable in space polynomial in $|\Delta| \cdot 2^{|\xi|+|f|}$.

Proof. Observe that we may safely assume that the value ξ and all values of the functions g, h, and c are from $\mathbb{N}_0 = \{0,1,2,\ldots\}$. Indeed, it is easy to show that for $y > 0$ holds $\mathbf{D}^{y\cdot f}[y\cdot\xi] = y \cdot \mathbf{D}^f[\xi]$ and $\mathbf{R}^{y\cdot f}[y\cdot\xi,\infty] = \mathbf{R}^f[\xi,\infty]$, where $y \cdot f$ is the function defined by $(y\cdot f)(p\alpha) = y \cdot f(p\alpha)$ for all $p\alpha \in \mathcal{C}(\Delta)$. Hence, it suffices to consider $y \cdot f$ and $y \cdot \xi$ instead of f and ξ, respectively, where y is the product of the denominator of ξ and denominators of all values of the functions g, h, and c. Clearly, $|y\cdot f|$ and $|y\cdot\xi|$ (i.e., the size of the binary representation of $y\cdot\xi$) are linear in $|f| + |\xi|$.

We define a new pPDA Δ' (together with new reward functions f_D and f_R) such that the average deviation and ratio along runs of Δ correspond to the average reward $\mathbf{A}$ along runs of Δ' (w.r.t. the reward functions f_D and f_R, respectively).

The idea of the construction is as follows. The pPDA Δ' simulates Δ, and in addition, stores (in its control states) the reward accumulated so far in the current service. More concretely, control states of Δ' are of the form $q[\pi]$ where q ranges over control states of Δ and $\pi \geq 0$ (encoded binary). If a run of Δ' (initiated in $p[0]X$) enters a configuration of the form $q[\pi]\alpha$ (where $\alpha \in \Gamma^*$), then the simulated pPDA Δ is in the configuration $q\alpha$, and π is the reward accumulated so far during the current service (excluding the last configuration). The accumulated reward is counted in π only up to the threshold ξ (then a special symbol $\uparrow$ is stored instead). Observe that due to our assumptions, the value π depends only on the *control states* encountered so far during the current service.

Now it is easy to see how, e.g., $\mathbf{R}[\xi,\infty]$ is reduced to $\mathbf{A}$. Indeed, it suffices to define a new simple reward function $f_R : \mathcal{C}(\Delta') \to \mathbb{R}^+$ which assigns 1 to all

configurations with control states of the form $q[\pi]$ where q is a trigger, and either $\pi = \uparrow$ or $\pi + g(q) \geq \xi$. All other configurations are assigned 0 by f_R. Then f_R assigns 1 to exactly those services of Δ' that correspond to the services of Δ whose accumulated reward (w.r.t. f) is greater than or equal to ξ.

Formally: We define the pPDA $\Delta' = (Q', \Gamma, \delta', Prob')$ such that

$$Q' = \{q[\pi] \mid q \in Q, 0 \leq \pi \leq \xi\} \cup \{q[\uparrow] \mid q \in Q\}$$

and transitions of Δ' are defined as follows (here $0 \leq \pi \leq \xi$):

1. $q[\pi]X \xrightarrow{x} r[\pi + g(q)]\alpha$ if $qX \xrightarrow{x} r\alpha$, $q \notin T$, $\pi + g(q) \leq \xi$;
2. $q[\pi]X \xrightarrow{x} r[\uparrow]\alpha$ if $qX \xrightarrow{x} r\alpha$, $q \notin T$ and $\pi + g(q) > \xi$;
3. $q[\pi]X \xrightarrow{x} r[0]\alpha$ if $qX \xrightarrow{x} r\alpha$ and $q \in T$
4. $q[\uparrow]X \xrightarrow{x} r[\uparrow]\alpha$ if $qX \xrightarrow{x} r\alpha$ and $q \notin T$;
5. $q[\uparrow]X \xrightarrow{x} r[0]\alpha$ if $qX \xrightarrow{x} r\alpha$ and $q \in T$;
6. All transitions of Δ' are defined by the rules 1 – 5.

A configuration $q[\pi]\alpha$ of Δ' is a trigger if and only if $q \in T$. The reward function f induces a well-defined reward function $f' : \mathcal{C}(\Delta') \to \mathbb{R}^+$ such that $f'(q[\pi]\alpha) = f(q\alpha)$ for all $q[\pi]\alpha \in \mathcal{C}(\Delta')$.

We define a function $\Theta : \mathcal{C}(\Delta') \to \mathcal{C}(\Delta)$ as follows: Given $q[\pi]\alpha \in Q' \times \Gamma^*$, we put $\Theta(q[\pi]\alpha) = q\alpha$. The following claim (which assures that Δ' faithfully simulates Δ) is easily proved by inspecting the transition rules of Δ'.

Claim (1). *Θ is a quotient of $M_{\Delta'}$ onto M_Δ.*

Note that Claim (1) together with Proposition 2.2.5 imply that for all $q[\pi]X$ we have $\Theta(Clean(q[\pi]X)) = Clean(qX)$ and $[q[\pi]X\uparrow] = [qX\uparrow]$. It follows that $\Theta(Good(q[\pi]X)) = Good(qX)$, and that $p[0]X$ is a state of $\mathbf{X}_{\Delta'}$ for all $pX \in \mathbf{S}_\Delta$.

In order to define the new reward functions f_D and f_R we need the following technical claim.

Claim (2). *Let $w \in Clean(p[0]X)$ such that $I_{inf}(w) = 1$, let $v = w[i]$ for some $i \geq 1$, let $0 \leq j \leq |v|$, and let us denote $v(j) = q[\pi]\alpha$. If $\pi \neq \uparrow$, then $f'(v^j) = \pi + g(q) + h(q) \cdot c(\alpha)$, where $h(q) = 0$ whenever $\pi + g(q) \leq \xi$. Moreover,*

1. *$f'(v^{j-1}) > \xi$ if and only if $\pi = \uparrow$;*
2. *$f'(v^j) \leq \xi$ if and only if $\pi \neq \uparrow$ and $\pi + g(q) \leq \xi$;*
3. *$f'(v^j) > \xi$ and $f'(v^{j-1}) \leq \xi$ if and only if $\pi \neq \uparrow$ and $\pi + g(q) > \xi$.*

where we put $f'(v^{-1}) = 0$.

Proof of the Claim. Let us denote $v = q_0[\pi_0]\alpha_0, \ldots, q_k[\pi_k]\alpha_k$. It is easy to show by induction on $0 \leq j \leq k$ that if $g(q_0, \ldots, q_{j-1}) \leq \xi$, then $\pi_j = g(q_0, \ldots, q_{j-1})$, else $\pi_j = \uparrow$. Hence, if $\pi_j \neq \uparrow$, then $g(q_0, \ldots, q_{j-1}) \leq \xi$, which implies $h(q_0, \ldots, q_{j-1}) = 0$ by our assumptions. It follows that if $\pi \neq \uparrow$, then $f'(v^j) = g(q_0, \ldots, q_{j-1}) + g(q_j) + h(q_j) \cdot c(\alpha_j) = \pi_j + g(q_j) + h(q_j) \cdot c(\alpha_j)$.

Observe also that if $\pi \neq \uparrow$ and $j \geq 1$, then $f'(v^{j-1}) = g(q_0, \ldots, q_{j-1}) \leq \xi$. It follows that $f'(v^{j-1}) > \xi$ implies $\pi_j = \uparrow$. If $f'(v^j) \leq \xi$, then $g(q_0, \ldots, q_j) \leq \xi$, which implies that $\pi_j \neq \uparrow$ and $\pi_j + g(q_j) = g(q_0, \ldots, q_j) \leq \xi$. Finally, if $f'(v^j) > \xi$ and $f'(v^{j-1}) \leq \xi$, then $\pi_j \neq \uparrow$, and moreover, $\pi_j + g(q_j) > \xi$ because otherwise $h(q_j) > 0$ and $g(q_j) \leq \xi$, which contradicts our assumptions. Now the claim follows from the fact that the right hand sides of 1–3 are in contradiction among each other, and one of the left hand sides of 1–3 is always satisfied.

$\Diamond$

Now let us consider $\mathbf{D}[\xi]$. Let us define the reward function $f_D : \mathcal{C}(\Delta') \to \mathbb{R}^+$ as follows: Given $q[\pi]\alpha \in \mathcal{C}(\Delta')$ we put

$$f_D(q[\pi]\alpha) = \begin{cases} f(q\alpha) & \text{if } \pi = \uparrow; \\ \xi - (\pi + g(q)) & \text{if } \pi + g(q) \leq \xi \text{ and } q \in T; \\ 0 & \text{if } \pi + g(q) \leq \xi \text{ and } q \notin T; \\ (\pi + g(q) - \xi) + h(q) \cdot c(\alpha) & \text{if } \pi + g(q) > \xi. \end{cases}$$

Note that $f_D(q[\pi]\alpha)$ can be written in the form $g'(q[\pi]) + h'(q[\pi]) \cdot c(\alpha)$ for suitable simple reward functions g' and f', which implies that f_D is well-defined. It follows immediately from Claim (2) and the definition of Θ that for all $p[0]X$ and all $w \in \mathit{Clean}(p[0]X)$, we have $\mathbf{A}^{f_D}(w) = \mathbf{D}^{f'}[\xi](w) = \mathbf{D}^{f}[\xi](\Theta(w))$.

Now let us deal with $\mathbf{R}[\xi, \infty]$. We define a simple reward function $f_R : \mathcal{C}(\Delta') \to \{0, 1\}$ as follows: Given $q[\pi]\alpha \in \mathcal{C}(\Delta')$, we put $f_R(q[\pi]\alpha) = 1$ if and only if $q \in T$, and either $\pi = \uparrow$ or $\pi + g(q) \geq \xi$. It follows from Claim (2) that for all $w \in \mathit{Clean}(p[0]X)$ we have $\mathbf{A}^{f_R}(w) = \mathbf{R}^{f'}[\xi, \infty](w) = \mathbf{R}^{f}[\xi, \infty](\Theta(w))$.

In the rest of this proof we show (using Theorem 7.3.13) that there are effectively expressible values $\mathbf{A}^{f_D}_{C'}, \mathbf{A}^{f_R}_{C'} \in \mathbb{R}^+_\infty \cup \{\bot\}$ such that $\mathbf{A}^{f_D} = \mathbf{A}^{f_D}_{C'}$ and $\mathbf{A}^{f_R} = \mathbf{A}^{f_R}_{C'}$ a.s. over $\mathit{Clean}(p[0]X)$, for all $pX \in C$. Then, by Claim (1) and Proposition 2.2.5, also $\mathbf{D}^{f}[\xi] = \mathbf{A}^{f_D}_{C'}$ and $\mathbf{R}^{f}[\xi, \infty] = \mathbf{A}^{f_R}_{C'}$ a.s. over $\mathit{Clean}(pX)$, for all $pX \in C$. Hence, it suffices to put $\mathbf{D}^{f}_{C}[\xi] = \mathbf{A}^{f_D}_{C'}$ and $\mathbf{R}^{f}_{C}[\xi, \infty] = \mathbf{A}^{f_R}_{C'}$.

The key to this assertion is contained in the following claim.

Claim (3)**.** *For all $pX \in C$ there are no strictly increasing transitions w.r.t. f_D reachable from $p[0]X$ in $\mathbf{X}_{\Delta'}$. Moreover, there is a BSCC C' of $\mathbf{X}_{\Delta'}$ such that for all $pX \in C$ the component C' is the* only *BSCC of $\mathbf{X}_{\Delta'}$ reachable from $p[0]X$.*

Proof of the Claim. Let us fix $pX \in C$. If there is a strictly increasing transition $q[\pi]Y \hookrightarrow q'[\pi']Y'$ (due to a transition $q[\pi]Y \to q'[\pi']Y'Z$ where $c(Z) > 0$) reachable from $p[0]X$, then there is a run $w \in \mathit{Good}(p[0]X)$ and $i \geq 1$ such that $\mathbf{X}^i_\Delta(w) = q[\pi]Y$, $\mathbf{X}^{i+1}_\Delta(w) = q'[\pi']Y'$. However, by the definition of Θ and

Claim (1), $\Theta(w) \in Good(pX)$, $\mathbf{X}^i_\Delta(\Theta(w)) = qY$, $\mathbf{X}^{i+1}_\Delta(\Theta(w)) = q'Y'$ and $q \to q'Y'Z$ in Δ, which implies that $qY \hookrightarrow q'Y'$ is a strictly increasing transition reachable from pX in $\mathbf{X}^C_\Delta$, a contradiction. Hence, there are no strictly increasing transitions w.r.t. f_D reachable from $p[0]X$.

Let us denote C' the set of all states $q[\pi]Y$ of $\mathbf{X}_{\Delta'}$ satisfying the following condition: There is a run $w \in Good(p[0]X)$ and $i \geq 2$ such that $\mathbf{X}^i_\Delta(w) = q[\pi]Y$ and $w^{ind_i(w)-1}$ contains a trigger (we say that w and i *witness* that $q[\pi]Y \in C'$).

We show that C' is a BSCC of $\mathbf{X}_{\Delta'}$. First, we show that C' is non-empty. By Claim (1) and Proposition 2.2.5, $\mathcal{P}_{p[0]X}(I_{inf} = 1) = \mathcal{P}_{pX}(I_{inf} = 1) = 1$ because Θ preserves triggers. It follows that almost all runs of $Good(p[0]X)$ contain infinitely many triggers, and hence there is a run $w \in Good(p[0]X)$ and $i \geq 2$ such that $w^{ind_i(w)-1}$ contains a trigger. Thus, $\mathbf{X}^i_\Delta(w) \in C'$ and C' is non-empty. Now let $q[\pi]Y \in C'$ and let us assume that $q[\pi]Y \hookrightarrow q'[\pi']Y'$. Let $w \in Good(p[0]X)$ and $i \geq 2$ witness that $q[\pi]Y \in C'$. There is a run $w' \in Good(q[\pi]Y)$ such that $\mathbf{X}^2_\Delta(w') = q'[\pi']Y'$. If $w'' = w^{ind_i(w)} * w'$, then $\mathbf{X}^i_\Delta(w'') = \mathbf{X}^i_\Delta(w) = q[\pi]Y$, $\mathbf{X}^{i+1}_\Delta(w'') = q'[\pi']Y'$, and $(w'')^{ind_i(w'')-1} = w^{ind_i(w)-1}$ contains a trigger. It follows that $q'[\pi']Y' \in C'$.

Now let us consider $q'[\pi']Y' \in C'$ and $q[\pi]Y \in \mathbf{S}_{\Delta'}$ such that $qY \in C$. We show that there is a path from $q[\pi]Y$ to $q'[\pi']Y'$ in $\mathbf{X}_{\Delta'}$. Observe that there is a path from qY to pX in $\mathbf{X}^C_\Delta$ because C is a BSCC. It follows that there is a run $v \in Good(qY)$ such that $\mathbf{X}^j_\Delta(v) = pX$ for some $j \geq 1$. Let $w \in Good(q[\pi]Y)$ be the unique run such that $\Theta(w) = v$ (see Proposition 2.2.5). It follows from the definition of Θ that $ind_j(w) = ind_j(v)$, and thus $\mathbf{X}^j_\Delta(w) = p[\sigma]X$ for some σ. Hence, there is a path from $q[\pi]Y$ to $p[\sigma]X$.

Let $w' \in Good(p[0]X)$ and $i \geq 2$ witness that $q'[\pi']Y' \in C'$. Let $k \leq ind_i(w')$ be such that $w'(k-1)$ is a trigger. It follows that $w'(k) = r[0]\alpha$ for some $r \in Q$ and $\alpha \in \Gamma^*$. Let $u \in Good(p[\sigma]X)$ be the unique run such that $\Theta(u) = \Theta(w')$. Observe that $u(k) = r[0]\alpha$ by the definition of Δ'. It follows that $u_k = w'_k$ because $\Theta(u_k) = \Theta(w'_k)$ and because Θ maps $Run(r[0]\alpha)$ isomorphically onto $Run(r\alpha)$. Also, observe that $ind_i(u) = ind_i(\Theta(u)) = ind_i(\Theta(w')) = ind_i(w')$. Now because $k \leq ind_i(w')$ we have $u(ind_i(u)) = u(ind_i(w')) = u_k(ind_i(w') - k) = w'_k(ind_i(w') - k) = w'(ind_i(w'))$, and hence $\mathbf{X}^i_\Delta(u) = \mathbf{X}^i_\Delta(w') = q'[\pi']Y'$. It follows that there is a path from $p[\sigma]X$ to $q'[\pi']Y'$ in $\mathbf{X}_{\Delta'}$. Putting this together with the fact that $p[\sigma]X$ is reachable from $q[\pi]Y$ in $\mathbf{X}_{\Delta'}$, we obtain that $q'[\pi']Y'$ is reachable from $q[\pi]Y$ in $\mathbf{X}_{\Delta'}$.

We have proved that for all $q'[\pi']Y' \in C'$ and all $q[\pi]Y \in \mathbf{S}_{\Delta'}$ satisfying $qY \in C$ there is a path from $q[\pi]Y$ to $q'[\pi']Y'$ in $\mathbf{X}_{\Delta'}$. ¿From this we obtain that $C' \in \text{BSCC}_{\Delta'}$ because for all $q[\pi]Y \in C'$ we have $qY \in C$. We also obtain that C' is the only BSCC of $\mathbf{X}_{\Delta'}$ reachable from $q[0]Y$ for each $qY \in C$ because all states reachable from $q[0]Y$ in $\mathbf{X}_{\Delta'}$ are of the form $q'[\pi']Y'$ where $q'Y' \in C$ (this follows from Claim (1) because for all $w \in Good(q[0]Y)$ we have $\Theta(w) \in Good(qY)$). $\Diamond$

Let us finish the proof of the lemma. Let C' be the BSCC from Claim (3). By Theorem 7.3.13, $\mathbf{A}^{f_D} = \mathbf{A}^{f_D}_{C'}$ a.s. over $Clean(q[\pi]Y)$, for all $q[\pi]Y \in C'$. We show that $\mathbf{A}^{f_D} = \mathbf{A}^{f_D}_{C'}$ a.s. over $Clean(p[0]X)$, for all $pX \in C$. Let us denote $\Gamma_0 = \{X \in \Gamma \mid c(X) = 0\}$. It follows from Lemma 7.3.2 that for all $\beta \in \Gamma_0^*$ we have $\mathbf{A}^{f_D} = \mathbf{A}^{f_D}_{C'}$ a.s. over $Run(p[0]X, C', \beta)$ because clearly $(f_D)_\beta = f_D$. By Lemma 7.3.10, we have $\mathcal{P}_{p[0]X}(\biguplus_{\beta \in \Gamma_0^*} Run(p[0]X, C', \beta)) = 1$ because (by Claim (3)) there are no strictly increasing transitions (w.r.t. f_D) reachable from $p[0]X$ in $\mathbf{X}_{\Delta'}$. Hence, $\mathbf{A}^{f_D} = \mathbf{A}^{f_D}_{C'}$ a.s. over $Clean(p[0]X)$. The fact that $\mathbf{A}^{f_R} = \mathbf{A}^{f_R}_{C'}$ a.s. over $Clean(p[0]X)$ is proved similarly.

Complexity analysis: For the complexity analysis we *drop* the assumption that ξ and all values of the functions g, h, and c are from $\mathbb{N}_0$. In the above proof we define $\mathbf{D}^f_C[\xi] = \mathbf{A}^{f_D}_{C'}$ and $\mathbf{R}^f_C[\xi, \infty] = \mathbf{A}^{f_R}_{C'}$. Hence, to compute formulae expressing these values, we apply Theorem 7.3.13 to Δ', C', and the reward functions f_D and f_R. One can easily show that Δ' and the reward functions f_D and f_R are computable in time polynomial in $|\Delta| \cdot 2^{|\xi|+|f|}$ (remember that before applying the above construction, one has to multiply ξ and f with the product of the denominator of ξ and denominators of all values of the functions g, h, and c). By Lemma 4.3.8, the underlying transition system of $\mathbf{X}_{\Delta'}$ (and hence also the BSCC C') can be computed in space polynomial in $|\Delta'|$. The rest follows from Theorem 7.3.13.

□

Now it remains to alleviate the assumption that for all $pX \in C$ and all $q\alpha$ reachable from pX in M_Δ, we have $g(q) > \xi$ whenever $h(q) > 0$. We prove an analogy of Lemma 7.3.15 without this assumption.

Lemma 7.3.16. *Let us assume that $I^C_{inf} = 1$ and that there are no strictly increasing transitions in $\mathbf{X}^C_\Delta$. There are values $\mathbf{D}^f_C[\xi], \mathbf{R}^f_C[\xi, \infty] \in \mathbb{R}^+_\infty \cup \{\bot\}$ such that $\mathbf{D}[\xi] = \mathbf{D}^f_C[\xi]$ and $\mathbf{R}[\xi, \infty] = \mathbf{R}^f_C[\xi, \infty]$ a.s. over $Clean(pX)$, for all $pX \in C$. Moreover, the values $\mathbf{D}^f_C[\xi]$ and $\mathbf{R}^f_C[\xi, \infty]$ are effectively expressible in $ExTh(\mathbb{R})$.*

Proof. Similarly to the proof of the previous lemma, we assume that the value ξ and all values of the functions g, h, and c are from $\mathbb{N}_0 = \{0, 1, 2, \ldots\}$.

We show how control states and stack symbols of Δ can be extended with an additional information in such a way that the resulting pPDA Δ' satisfying all premises of Lemma 7.3.15. Intuitively, the pPDA Δ' simulates Δ and at the same time remembers the value of c over a bottom part of the stack, in its control states (up to the threshold $\xi + \max\{c(X) \mid X \in \Gamma\}$). More concretely, control states of Δ' are of the form $q[\pi]$ where $q \in Q$ and $\pi \geq 0$, and stack symbols are either symbols of Γ, or "marked" symbols $X^\downarrow$ where $X \in \Gamma$. Each configuration reachable from $p[c(X)]X^\downarrow$ is of the form $q[\pi]\beta Y_1^\downarrow \cdots Y_\ell^\downarrow$, where $\beta \in \Gamma^*$, and $\beta = \varepsilon$ if $\pi \leq \xi$. If a run $Run(p[c(X)]X^\downarrow)$ enters a configuration of the form $q[\pi]\beta Y_1^\downarrow \cdots Y_\ell^\downarrow$, then the simulated pPDA Δ is in the configuration $q\beta Y_1 \cdots Y_\ell$ and π equals $c(Y_1 \cdots Y_\ell)$.

We define a new well-defined reward function f' such that for all $q[\pi]\alpha \in \mathcal{C}(\Delta')$ we have $f'(q[\pi]\alpha) = g'(q[\pi]) + h'(q[\pi]) \cdot c'(\alpha)$, where the functions g', h' and c' are defined as follows: $g'(q[\pi]) = g(q) + h(q) \cdot \pi$; $h'(q[\pi])$ equals either 0 or $h(q)$, depending on whether $\pi \leq \xi$ or not, respectively; $c'(X^{\downarrow}) = 0$ and $c'(X) = c(X)$, for all $X \in \Gamma$. It is easy to observe that $f'(q[\pi]\beta Y_1^{\downarrow} \cdots Y_\ell^{\downarrow}) = f(q\beta Y_1 \cdots Y_\ell)$, and that $h'(q[\pi]) > 0$ implies $g'(q[\pi]) > \xi$.

Formally, let us denote $\max(c) = \max\{c(X) \mid X \in \Gamma\}$. We define a pPDA $\Delta' = (Q', \Gamma', \delta', Prob')$ such that $Q' = \{q[\pi] \mid 0 \leq \pi \leq \xi + \max(c)\}$ and $\Gamma' = \Gamma \cup \Gamma^{\downarrow}$ where $\Gamma^{\downarrow} = \{X^{\downarrow} \mid X \in \Gamma\}$. In order to simplify the definition of transitions of Δ' we use the following additional notation:

Given $X_1 \cdots X_k \in \Gamma^+$ and $\pi \in \mathbb{R}$, we denote

$$\rho_\Gamma(X_1 \cdots X_k, \pi) = X_1 \cdots X_{i-1} \cdot X_i^{\downarrow} \cdots X_k^{\downarrow}$$

and

$$\rho_Q(X_1 \cdots X_k, \pi) = c(X_i, \ldots, X_k) + \pi$$

where $1 \leq i \leq k$ is the *least* number such that $0 \leq c(X_i \cdots X_k) + \pi \leq \xi + \max(c)$ (if there is no such i, then we put $\rho_\Gamma(X_1 \cdots X_k, \pi) = X_1 \cdots X_k$, and $\rho_Q(X_1 \cdots X_k, \pi) = \pi$). We also define $\rho_\Gamma(\varepsilon, \pi) = \varepsilon$ and $\rho_Q(\varepsilon, \pi) = \pi$.

Transitions of Δ' are defined as follows:

1. $q[\pi]X \xrightarrow{x} r[\pi]\alpha$ for $X \in \Gamma$ if and only if $pX \xrightarrow{x} r\alpha$ in Δ;

2. $q[\pi]X^{\downarrow} \xrightarrow{x} r[\pi']\alpha$ if and only if $qX \xrightarrow{x} r\beta$, where $\alpha = \rho_\Gamma(\beta, \pi - c(X))$ and $\pi' = \rho_Q(\beta, \pi - c(X))$.

Given $q[\pi]\alpha \in \mathcal{C}(\Delta')$, we define $f'(q[\pi]\alpha) = g'(q[\pi]) + h'(q[\pi]) \cdot c'(\alpha)$, where

- $g'(q[\pi]) = g(q) + h(q) \cdot \pi$;
- for all $X \in \Gamma$, we put $c'(X) = c(X)$ and $c'(X^{\downarrow}) = 0$;
- if $\pi \leq \xi$, then $h'(q[\pi]) = 0$, else $h'(q[\pi]) = h(q)$.

A configuration $q[\pi]\alpha$ of Δ' is a trigger if and only if $q \in T$. Given $X \in \Gamma$, we denote $\theta(X^{\downarrow}) = \theta(X) = X$. The function θ extends to $(\Gamma')^*$ in an obvious way. We define a function $\Theta : \mathcal{C}(\Delta') \to \mathcal{C}(\Delta)$ as follows: Given $q[\pi]\alpha \in \mathcal{C}(\Delta')$, we define $\Theta(q[\pi]\alpha) = q\beta$, where $\beta = \theta(\alpha)$.

Claim (1)**.** *Θ is a quotient of $M_{\Delta'}$ onto M_Δ and for all $q[\pi]\alpha$ reachable from $p[c(X)]X^{\downarrow}$, we have $f'(q[\pi]\alpha) = f(\Theta(q[\pi]\alpha))$.*

Proof of the Claim. The pPDA Δ' is designed in such a way that all configurations reachable from $p[c(X)]X^{\downarrow}$ are of the form $q[\pi]\beta\gamma$ where $\gamma \in (\Gamma^{\downarrow})^*$, $c(\theta(\gamma)) = \pi$, and $\beta \in \Gamma^*$ satisfies $\beta = \varepsilon$ whenever $\pi \leq \xi$. It follows that $f'(q[\pi]\beta\gamma) = f(\Theta(q[\pi]\beta\gamma))$. The fact that Θ is a quotient follows immediately from the definition of transitions of Δ'.

$\Diamond$

By Claim (1), for $q[\pi]\alpha \in \mathcal{C}(\Delta')$ holds $\Theta(\mathit{Clean}(q[\pi]\alpha)) = \mathit{Clean}(\Theta(q[\pi]\alpha))$ and $\mathcal{P}(\mathit{Clean}(q[\pi]\alpha)) = \mathcal{P}(\Theta(\mathit{Clean}(q[\pi]\alpha))) = \mathcal{P}(\mathit{Clean}(\Theta(q[\pi]\alpha)))$, which implies that $\Theta(\mathit{Good}(q[\pi]\alpha)) = \mathit{Good}(\Theta(q[\pi]\alpha))$.

Let us define $C' = \{p[c(X)]X^{\downarrow} \mid pX \in C\}$. We show that $C' \in \mathrm{BSCC}_{\Delta'}$, and that Δ', f', and C' satisfy all premises of Lemma 7.3.15 (Claims (2), (3), (4)).

Claim (2). *Let $p[c(X)]X^{\downarrow} \in C'$. For all configurations $q[\pi]\alpha$ reachable from $p[c(X)]X^{\downarrow}$ in $M_{\Delta'}$, we have $g'(q[\pi]\alpha) > \xi$ whenever $h'(q[\pi]\alpha) > 0$.*

Proof. Immediately from the definition of f' and the fact that all values of h are in $\mathbb{N}_0$. ◊

Claim (3). $I^{C'}_{inf} = 1$.

Proof. Immediately from the assumption that $I^{C}_{inf} = 1$, the definition of triggers, Claim (1) and Proposition 2.2.5. ◊

Claim (4). *C' is a BSCC of $\mathbf{X}_{\Delta'}$. Moreover, there are no strictly increasing transitions w.r.t. f' in $\mathbf{X}^{C'}_{\Delta'}$.*

Proof of the Claim. Let us fix $p[c(X)]X^{\downarrow} \in C'$. First, we prove that for all $w \in \mathit{Good}(p[c(X)]X^{\downarrow})$ and arbitrary $i \geq 1$ we have $\mathbf{X}^i_{\Delta}(w) \in C'$. We showed in the proof of Claim (1) that $\min_i(w) = q[\pi]\beta\gamma$ where $q \in Q$, $\gamma \in (\Gamma^{\downarrow})^*$, $c(\theta(\gamma)) = \pi$, and $\beta \in \Gamma^*$ satisfies $\beta = \varepsilon$ whenever $\pi \leq \xi$. First, let $\beta \neq \varepsilon$. Then $\gamma = \gamma' \cdot Z^{\downarrow} \cdot \gamma''$ where $c(Z) > 0$ because $c(\theta(\gamma)) = \pi > \xi > 0$. However, then $\min_i(\Theta(w)) = \Theta(\min_i(w)) = q\beta\theta(\gamma')Z\theta(\gamma'')$ where $\Theta(w) \in \mathit{Good}(pX)$ and $c(Z) > 0$. Hence, by Lemma 7.3.10, there is a strictly increasing transition in $\mathbf{X}^C_{\Delta}$, a contradiction. Now let us assume that $\beta = \varepsilon$ but $\gamma = Y^{\downarrow}\gamma'$ where $\pi \neq c(Y)$. It follows that there is $Z^{\downarrow}$ in γ' such that $c(Z) > 0$, and similarly as above we prove that there is a strictly increasing transition in $\mathbf{X}^C_{\Delta}$. Finally, for $\mathbf{X}^i_{\Delta}(w) = q[c(Y)]Y^{\downarrow}$ we have $\mathbf{X}^i_{\Delta}(\Theta(w)) = qY$, which implies that qY is reachable from pX in $\mathbf{X}_{\Delta}$, and thus $qY \in C$. It follows that all states reachable from $p[c(X)]X^{\downarrow}$ in $\mathbf{X}_{\Delta'}$ are in C'.

Now let us consider $q[c(Y)]Y^{\downarrow} \in C'$. Let $w \in \mathit{Good}(pX)$ be a run such that $\mathbf{X}^i_{\Delta}(w) = qY$ for some $i \geq 1$. Let $w' \in \mathit{Good}(p[c(X)]X^{\downarrow})$ be the unique run such that $\Theta(w') = w$. It follows that $\mathbf{X}^i_{\Delta}(w') = q[c(Y)]Y^{\downarrow}$ by the above arguments and the definition of Θ. Hence, $q[c(Y)]Y^{\downarrow}$ is reachable from $p[c(X)]X^{\downarrow}$, and thus C' is a BSCC of $\mathbf{X}_{\Delta'}$.

Finally, let $p[c(X)]X^{\downarrow} \hookrightarrow q[c(Y)]Y^{\downarrow}$ be a strictly increasing transition in $\mathbf{X}^{C'}_{\Delta'}$. Then $p[c(X)]X^{\downarrow} \rightarrow q[c(Y)]Y^{\downarrow}\bar{Z}$ where $c(\bar{Z}) > 0$. However, then $\bar{Z} \in \Gamma$ which is not possible due to the definition of Δ', a contradiction. ◊

Now we apply Lemma 7.3.15 to Δ', f' and C', and obtain the values $\mathbf{D}^{f'}_{C'}[\xi]$ and $\mathbf{R}^{f'}_{C'}[\xi, \infty]$ such that $\mathbf{D}^{f'}[\xi] = \mathbf{D}^{f'}_{C'}[\xi]$ and $\mathbf{R}^{f'}[\xi, \infty] = \mathbf{R}^{f'}_{C'}[\xi, \infty]$ a.s. over $\mathit{Clean}(p[0]X^{\downarrow})$, for all $p[0]X^{\downarrow} \in C'$. Then, by Claim (1) and Proposition 2.2.5, we have that $\mathbf{D}^{f}[\xi] = \mathbf{D}^{f'}_{C'}[\xi]$ and $\mathbf{R}^{f}[\xi, \infty] = \mathbf{R}^{f'}_{C'}[\xi, \infty]$ a.s. over $\mathit{Clean}(pX)$, for all $pX \in C$.

□

Theorem 7.3.17. *There are values $\mathbf{D}^f_C[\xi], \mathbf{R}^f_C[\xi,\infty] \in \mathbb{R}^+_\infty \cup \{\bot\}$ such that $\mathbf{D}[\xi] = \mathbf{D}^f_C[\xi]$ and $\mathbf{R}[\xi,\infty] = \mathbf{R}^f_C[\xi,\infty]$ a.s. over $Clean(pX)$, for all $pX \in C$. The values $\mathbf{D}^f_C[\xi]$ and $\mathbf{R}^f_C[\xi,\infty]$ are effectively expressible in $ExTh(\mathbb{R})$. Moreover, if f is simple, then these values are expressible by formulae of size polynomial in $|\Delta| \cdot 2^{|\xi|+|f|}$, computable in space polynomial in $|\Delta| \cdot 2^{|\xi|+|f|}$.*

Proof. First, by Lemma 7.3.1, the problem whether $I^C_{inf} = 1$ is decidable in polynomial space. If $I^C_{inf} = 0$, then all variables of LV are undefined a.s. over $Clean(pX)$, for all $pX \in C$. Let us assume that $I^C_{inf} = 1$. If there is at least one strictly increasing transition in $\mathbf{X}^C_\Delta$, then, by Lemma 7.3.14, one can substitute the reward function f with a simple reward function, and thus eliminate strictly increasing transitions from $\mathbf{X}^C_\Delta$. If there are no strictly increasing transitions in $\mathbf{X}^C_\Delta$, then the result follows from Lemma 7.3.16 and Lemma 7.3.15.

□

7.3.4 Regularity of L_C

In this subsection we finish the proof of Theorem 7.2.1 from Section 7.2. We still work with the pPDA Δ, the well-defined reward function f, the configuration $p_0X_0 \in \mathbf{S}_\Delta$ and the component $C \in \mathrm{BSCC}_\Delta$ such that $\mathcal{P}(Run(p_0X_0, C)) > 0$, fixed in the beginning of Section 7.3.

Let us fix an indicator $I \in \mathcal{I}$ of the form $I[V, \ell, u]$ where $\ell, u \in \mathbb{R}^+_\infty$ and $V \in LV$. We may safely assume that V has one of the following three forms: $\mathbf{A}$, $\mathbf{D}[\xi]$, and $\mathbf{R}[\xi,\infty]$, where $\xi \in \mathbb{R}^+$. Indeed, it is easy to show that $\mathbf{R}[\xi,\lambda]$, where $\lambda < \infty$, is equal to $\mathbf{R}[\xi,\infty] - \mathbf{R}[\lambda + \frac{1}{y},\infty]$, where the number y is the product of the denominator of λ and the denominators of all values of the functions g, h, and c (observe that the size of the binary representation of y is linear in $|f| + |\lambda|$).

By Theorem 7.3.17, for each $\beta \in \Gamma^*$ there is an effectively expressible constant $V^{f_\beta}_C$ such that $V^{f_\beta} = V^{f_\beta}_C$ a.s. over $Clean(pX)$, for all $pX \in C$. By Lemma 7.3.2, $V = V^{f_\beta}_C$ a.s. over $Run(p_0X_0, C, \beta)$, for all $\beta \in \Gamma^*$ such that $\mathcal{P}(p_0X_0, C, \beta) > 0$.

Let us define

$$L_C = \{\beta \in \Gamma^* \mid V^{f_\beta}_C \neq \bot, \ell \leq V^{f_\beta}_C \leq u\}$$

Observe that if f is simple, then either $L_C = \Gamma^*$ or $L_C = \emptyset$ by the definition of f_β.

Lemma 7.3.18. $\mathcal{P}(I = Hit[L_C] \mid Run(p_0X_0, C)) = 1$

Proof. By definition, for all $w \in Run(p_0X_0, C, \beta)$ holds $tail(Entry(w)) = \beta$. Hence, $Hit[L_C]$ is constant over $Run(p_0X_0, C, \beta)$, and $Hit[L_C] = 1$ over $Run(p_0X_0, C, \beta)$ if and only if $\beta \in L_C$. Putting above arguments together, we obtain that $I = 1$ a.s. over $Run(p_0X_0, C, \beta)$ if and only if $\ell \leq V \leq u$ a.s. over $Run(p_0X_0, C, \beta)$ if and only if $\ell \leq V^{f_\beta}_C \leq u$ if and only if $\beta \in L_C$ if and only

if $Hit[L_C] = 1$ over $Run(p_0X_0, C, \beta)$. It follows that $I = Hit[L_C]$ a.s. over $Run(p_0X_0, C, \beta))$. However, then

$$\begin{aligned}
&\mathcal{P}(I = Hit[L_C] \mid Run(p_0X_0, C)) = \\
&= \sum_{\beta \in \Gamma^*} \mathcal{P}(I = Hit[L_C] \mid Run(p_0X_0, C, \beta)) \cdot \mathcal{P}(Run(p_0X_0, C, \beta) \mid Run(p_0X_0, C)) \\
&= \sum_{\beta \in \Gamma^*} \mathcal{P}(Run(p_0X_0, C, \beta) \mid Run(p_0X_0, C)) \;=\; 1
\end{aligned}$$

□

It remains to show that L_C is effectively regular. The following lemma contains the essential argument:

Lemma 7.3.19. *There is an effectively computable $k \geq 0$, such that for all $\beta, \beta' \in \Gamma^*$, satisfying $c(\beta) > k$ and $c(\beta') > k$, we have $\beta \in L_C$ if and only if $\beta' \in L_C$.*

Proof. If $I^C_{inf} = 0$, then $V^{f_\beta}_C = \bot$ for all $\beta \in \Gamma^*$ (by Lemma 7.3.1), and thus $L_C = \emptyset$. If there are no active transitions in $\mathbf{X}^C_\Delta$, then, by Lemma 7.3.10 and Lemma 4.3.6, we have $V^{f_\beta}_C = V^g_C$. Hence, the value $V^{f_\beta}_C$ does not depend on β, and one can put $k = 0$.

Let us assume that $I^C_{inf} = 1$ and that there is at least one active transition in $\mathbf{X}^C_\Delta$. It follows from Lemma 7.3.10 that $\mathbf{G}^h_C = E^h_C > 0$, and hence that $\mathbf{A}^h_C > 0$.

First, let us assume that $V = \mathbf{D}[\xi]$, and let $\beta \in \Gamma^*$. Using similar arguments as in the proof of Lemma 7.3.14, one can show that $\mathbf{D}^{f_\beta}_C[\xi] \geq \mathbf{A}^{f_\beta}_C - \xi$. Moreover, for all $pX \in C$ and almost all $w \in Clean(pX)$, we have

$$\begin{aligned}
\mathbf{A}^{f_\beta}_C &= \mathbf{A}^{f_\beta}(w) \;=\; \lim_{k\to\infty} \frac{\sum_{i=1}^k f_\beta(w[i])}{k} \\
&= \lim_{k\to\infty} \frac{\sum_{i=1}^k f(w[i]) + c(\beta)\cdot h(w[i])}{k} \;=\; \mathbf{A}^f_C + c(\beta)\cdot \mathbf{A}^h_C
\end{aligned}$$

and thus $\mathbf{D}^{f_\beta}_C[\xi] \geq \mathbf{A}^f_C + c(\beta)\cdot \mathbf{A}^h_C - \xi$. Let us assume that $u = \infty$. Then $c(\beta) > \frac{\ell+\xi}{\mathbf{A}^h_C}$ implies $\beta \in L_C$, for all $\beta \in \Gamma^*$. Hence, it suffices to put $k = \frac{\ell+\xi}{\mathbf{A}^h_C}$. Now let $u < \infty$. Then $c(\beta) > \frac{u+\xi}{\mathbf{A}^h_C}$ implies $\beta \notin L_C$, and we can put $k = \frac{u+\xi}{\mathbf{A}^h_C}$. In both cases the number k is effectively computable, because $\mathbf{A}^h_C > 0$ can effectively be approximated (from below) using Theorem 2.3.1.

Now let us assume that $V = \mathbf{R}[\xi, \infty]$. Let us consider $\beta \in \Gamma^*$ such that $c(\beta) > \frac{\xi}{\min(h)}$, where $\min(h) = \min\{h(q) \mid q \in Q, h(q) > 0\}$. Then one can easily show that $\mathbf{R}^{f_\beta}_C[\xi, \infty] = \mathbf{R}^{f'}_C[\xi, \infty]$, where $f'(p\alpha) = g(p) + h(p)\cdot \frac{\xi}{\min(h)}$ for all $p\alpha \in \mathcal{C}(\Delta)$. However, f' is simple, and hence the value $\mathbf{R}^{f_\beta}_C[\xi, \infty]$ does not depend on β. Thus, we can put $k = \frac{\xi}{\min(h)}$. □

Definition 7.3.20. *We denote* $\Gamma_{c>0} = \{X \in \Gamma \mid c(X) > 0\}$. *The* c-span *of a given word* $\beta \in \Gamma^*$ *is the subsequence of symbols of* $\Gamma_{c>0}$ *contained in* β *(i.e., given* $\beta = \beta_1 X_1 \beta_2 \cdots \beta_i X_i \beta_{i+1}$, *where* $X_1 \cdots X_i \in \Gamma^*_{c>0}$ *and* $\beta_1 \cdots \beta_i \in (\Gamma \backslash \Gamma_{c>0})^*$, *the* c*-span of* β *is* $X_1 \cdots X_i$*).*

The following Corollary 7.3.21 finishes the proof of Theorem 7.2.1.

Corollary 7.3.21. *The language* L_C *is effectively regular.*

Proof. Observe that for each $\beta \in \Gamma^*$, the value of $c(\beta)$ depends only on the c-span of β. For a given $\gamma \in \Gamma^*_{c>0}$, let $L(\gamma)$ be the set of all $\beta \in \Gamma^*$ whose c-span is γ. Obviously, each $L(\gamma)$ is a regular language. Moreover, we either have that $L(\gamma) \subseteq L_C$ or $L(\gamma) \cap L_C = \emptyset$. By Lemma 7.3.19, there is an effectively computable constant k' such that the union of all $L(\gamma')$, where the length of γ' is larger than k', either forms a subset of L_C, or is disjoint with L_C. Now it is easy to see that L_C is effectively regular, because the membership to L_C is decidable by Theorem 2.3.1. □

Now we analyze the complexity of deciding whether $L_C = \Gamma^*$ (or equivalently, whether $\ell \leq V^f_C \leq u$), under the assumption that f is simple. If $V = \mathbf{A}$, then the problem whether $\ell \leq V^f_C \leq u$ is in **PSPACE** by Theorem 7.3.13 and Theorem 2.3.1. If $V = \mathbf{D}[\xi]$ or $V = \mathbf{R}[\xi, \infty]$, then this problem is in **EXPSPACE** by Theorem 7.3.17 and Theorem 2.3.1. Finally, observe that in all these cases, the problem whether $L_C = \Gamma^*$ is decidable in *space polynomial* in $|\Delta|$.

7.4 Other Proofs

7.4.1 Proof of Lemma 7.2.2

Throughout this proof we make use of the notation and results introduced in Section 4.5. Let us consider the set of configurations $L' = \{qY\alpha \mid qY \in C, \alpha \in L\}$. It is easy to show that the set of configurations L' is accepted by an effectively computable DFA $\mathcal{A}$ (in the sense of Definition 4.4.1). Let us consider the pPDA $\Delta[\mathcal{A}]$ (see Section 4.5). Let s_0 be the initial state of $\mathcal{A}$.

We denote C' the set of all states of $\mathbf{X}_{\Delta[\mathcal{A}]}$ of the form $q(Y, s)$, where $qY \in C$. Let us denote $D = C' \cap \mathcal{H}[\mathcal{A}]$ and $E = C' \backslash D$. We denote $\mathcal{P}(D \texttt{ before } E)$ the probability of all runs of $Good(p(X, s_0))$, whose footprints reach the set D before the set E.

We claim that $\mathcal{P}(Hit[L]{=}1 \cap Run(pX, C)) = \mathcal{P}(D \texttt{ before } E)$. Indeed, let Θ be the quotient of $M_{\Delta[\mathcal{A}]}$ onto M_Δ defined in Lemma 4.5.2. It is easy to see that Θ preserves minima (i.e., if $w(i)$ is the k^{th} minimum of a run w of $M_{\Delta[\mathcal{A}]}$, then $\Theta(w(i))$ is the k^{th} minimum of $\Theta(w)$). Hence, by the definition of Θ and $\Delta[\mathcal{A}]$, the footprint of $w \in Good(p(X, s_0))$ enters the set D before E if and only if $\Theta(w) \in Run(pX, C)$ and $Hit[L](\Theta(w)){=}1$, and thus the above equality of

probabilities follows from Proposition 2.2.5 and Lemma 4.3.6. We obtain that

$$\mathcal{P}(Hit[L]{=}1 \mid Run(pX, C)) = \frac{\mathcal{P}(D \texttt{ before } E)}{\mathcal{P}(Run(pX, C))}$$

Note that the probability $\mathcal{P}(Run(pX, C))$ is equal to $\mathcal{P}(pX \hookrightarrow^* C) \cdot [pX\uparrow]$ (Lemma 4.3.6), and hence is effectively expressible in $ExTh(\mathbb{R})$ by Lemma 4.3.9 and Corollary 4.1.10.

It remains to show that $\mathcal{P}(D \texttt{ before } E)$ is effectively expressible. However, observe that $\frac{\mathcal{P}(D \texttt{ before } E)}{[p(X,s_0)\uparrow]}$ equals the probability that a run of $\mathbf{X}_{\Delta[\mathcal{A}]}$, initiated in $p(X, s_0)$, reaches the set of states D before E. We show that the later probability is effectively expressible. Let us modify $\mathbf{X}_{\Delta[\mathcal{A}]}$ as follows: substitute all outgoing transitions from states of E with self-loops with the probability 1. It is easy to see that the probability of reaching D from $p(X, s_0)$ in the resulting chain is equal to the probability of reaching D before E from $p(X, s_0)$ in $\mathbf{X}_{\Delta[\mathcal{A}]}$. However, the effective expressibility of the former probability can easily be shown using arguments similar to the proof of Lemma 4.3.9.

□

7.4.2 Proof of Theorem 7.2.5

First, we extend some notions, introduced in Section 4.3.1 for runs, to paths of $FPath(q_0Z_0)$. Let $v = p_0\alpha_0, \ldots, p_k\alpha_k$ be a path of $FPath(q_0Z_0)$. A configuration $p_i\alpha_i$ of v is *minimal* if $|\alpha_i| \leq |\alpha_j|$ for all $i < j \leq k$. The *k-th (developing) minimum* of v, denoted $\min_k(v)$, is the k-th minimal configuration of v (note that the k-th minimum can be undefined). The *index* of the k-th minimum, denoted $ind_k(v)$, is the number i such that $v(i)$ is the k-th minimum of v. To v we associate its *(developing) footprint*: $fp(v) = \mathbf{X}^1_\Delta(v), \ldots, \mathbf{X}^k_\Delta(v)$, where k is the number of minima of v and $\mathbf{X}^i_\Delta(v)$ equals the head of $\min_i(v)$, for each $1 \leq i \leq k$. We define a function $Entry$ which for every $v \in FPath(q_0Z_0)$ returns either $v(ind_j(v))$, where $j \geq 1$ is the least number such that $\mathbf{X}^j_\Delta(v) \in C$ for some $C \in \text{BSCC}_\Delta$, or $\perp$ if there is no such j.

Given $n \geq 0$, we define the indicator $G^n : Run(q_0Z_0) \to \{0, 1\}$ as follows: For $w \in Run(q_0Z_0)$ we put $G^n(w) = 1$ if and only if $Entry(w^n) \neq \perp$ and $tail(Entry(w^n)) \in L_C$ for some $C \in \text{BSCC}_\Delta$. Clearly, the value of $G^n(w)$ is computable from the prefix of w of length n. Let $\zeta > 0$. We show that there effectively exists n such that $\mathcal{P}(G^n \neq I) < \zeta$.

Let us denote $D = \bigcup_{C \in \text{BSCC}_\Delta} C$. Given $n \geq 0$, we denote A^n the set of all runs $w \in Run(q_0Z_0)$ that satisfy the following condition: there is $i \geq 1$ such that $\mathbf{X}^i_\Delta(w) \in D$ and $ind_i(w) \leq n$.

Claim (1)**.** *For all $n \geq 1$ we have $\mathcal{P}(I = G^n \mid A^n) = 1$ whenever $\mathcal{P}(A^n) > 0$.*

Proof of the Claim. Given $C \in \text{BSCC}_\Delta$, we denote $A^n_C = A^n \cap Run(p_0X_0, C)$. By Theorem 7.2.1, we have $\mathcal{P}(I = Hit[L_C] \mid A^n_C) = 1$, for all $C \in \text{BSCC}_\Delta$.

However, it follows immediately from definitions that $Hit[L_C](w) = G^n(w)$ for all $w \in A^n_C$. Hence,

$$\mathcal{P}(I = G^n \mid A^n) = \sum_{C \in \mathrm{BSCC}_\Delta} \mathcal{P}(I = Hit[L_C] \mid A^n_C) \cdot \mathcal{P}(A^n_C \mid A^n) = 1$$

$\Diamond$

Claim (2)**.** *There is effectively computable n such that $\mathcal{P}(A^n) > 1 - \zeta$.*

Proof of the Claim. Given $n \geq 0$ and $qY \in D$, we denote B^n_{qY} the set of all paths $v \in FPath(q_0 Z_0)$ such that $|v| \leq n$, $last(fp(v)) = qY$, and for all $j < |fp(v)|$ holds $fp(v)(j) \notin D$. It follows from Lemma 4.6.9 that each B^n_{qY} is clean-prefix-free and that $A^n = \biguplus_{qY \in D} Clean(B^n_{qY})$.

By Lemma 4.6.7, we have $\mathcal{P}(A^n) = \sum_{qY \in D} [qY\uparrow] \cdot \sum_{v \in B^n_{qY}} \mathcal{P}(Run(v))$. By Theorem 4.1.8, each value $[qY\uparrow] > 0$ can effectively be approximated from below, and hence there is effectively computable $\epsilon(qY)$ such that $\frac{2[qY\uparrow]}{2+\zeta} < \epsilon(qY) \leq [qY\uparrow]$. Note that $[qY\uparrow] - \frac{\zeta}{2} \cdot \epsilon(qY) < \epsilon(qY)$.

Given $n \geq 1$, we denote $\epsilon_n = \sum_{qY \in D} \epsilon(qY) \cdot \sum_{v \in B^n_{qY}} \mathcal{P}(Run(v)) \leq \mathcal{P}(A^n)$. It is easy to see that ϵ_n is effectively computable for arbitrary n. Moreover,

$$\begin{aligned}
\epsilon_n &> \sum_{qY \in D} \left([qY\uparrow] - \frac{\zeta}{2} \cdot \epsilon(qY) \right) \cdot \sum_{v \in B^n_{qY}} \mathcal{P}(Run(v)) \\
&= \mathcal{P}(A^n) - \frac{\zeta}{2} \cdot \left(\sum_{qY \in D} \epsilon(qY) \cdot \sum_{v \in B^n_{qY}} \mathcal{P}(Run(v)) \right) \\
&\geq \mathcal{P}(A^n) - \frac{\zeta}{2}
\end{aligned}$$

where we used the fact that

$$\sum_{qY \in D} \epsilon(qY) \cdot \sum_{v \in B^n_{qY}} \mathcal{P}(Run(v)) \leq \sum_{qY \in D} [qY\uparrow] \cdot \sum_{v \in B^n_{qY}} \mathcal{P}(Run(v)) = \mathcal{P}(A^n) \leq 1$$

Hence, $0 \leq \mathcal{P}(A^n) - \epsilon_n < \frac{\zeta}{2}$. Now observe that $1 = \mathcal{P}(\bigcup_{n=1}^{\infty} A^n) = \lim_{n \to \infty} \mathcal{P}(A^n)$. It follows that there is $n \geq 1$ such that $\mathcal{P}(A^n) > 1 - \frac{\zeta}{2}$, which implies $\epsilon_n > 1 - \zeta$. On the other hand, if $\epsilon_n > 1 - \zeta$, then $\mathcal{P}(A^n) > 1 - \zeta$. Thus it suffices to find some n such that $\epsilon_n > 1 - \zeta$, which can be done effectively. $\Diamond$

Claim (1) and Claim (2) yield

$$\mathcal{P}(I = G^n) \geq \mathcal{P}(I = G^n \mid A^n) \cdot \mathcal{P}(A^n) = \mathcal{P}(A^n) > 1 - \zeta$$

where n is from Claim (2), and hence $\mathcal{P}(I \neq G^n) < \zeta$.

7.4.3 Proof of Theorem 7.2.7

We show that the set of all configurations satisfying $\mathcal{P}^{=1}(I=1)$ is effectively regular (see Definition 4.4.1). In other words, we show that there is an effectively computable DFA $\mathcal{A}$ such that $\mathcal{C}(\mathcal{A}) = \{p\alpha \in \mathcal{C}(\Delta) \mid p\alpha \models \mathcal{P}^{=1}(I=1)\}$. The rest follows from results of Chapter 5.

We rely on results and notation introduced in Section 7.3.4. Given $\alpha \in \Gamma^*$, we denote $sp(\alpha)$ the c-span of α (see Definition 7.3.20). Given $k \geq 0$, we denote $S[k] = \{\alpha \in \Gamma^* \mid |sp(\alpha)| \geq k\}$. As in Section 7.3, given $\beta \in \Gamma^*$ we denote f_β the well-defined reward function defined by $f_\beta(p\alpha) = (g(p)+h(p)\cdot c(\beta))+h(p)\cdot c(\alpha)$. To simplify our notation, given $qY \in \mathbf{S}_\Delta$ and a set of runs A, we denote $\mathcal{P}_{qY}(A) = \mathcal{P}(A \mid Clean(qY))$. Let us denote $\mathcal{I}_{var}$ the set of all indicators of $\mathcal{I}$ of the form $I[V,\ell,u]$, where $V \in LV$, and $\ell, u \in \mathbb{R}_{\pm\infty}$.

Claim (1)**.** *Let $I[V,\ell,u] \in \mathcal{I}_{var}$. There is an effectively computable $k \geq 0$ such that for all $pX \in \mathbf{S}_\Delta$ and all $\alpha, \beta \in S[k]$ holds*

$$\mathcal{P}_{pX}(I[V^{f_\alpha},\ell,u] = I[V^{f_\beta},\ell,u]) = 1$$

Proof. Given $\beta \in \Gamma^*$ and $C \in \mathrm{BSCC}_\Delta$, we denote

$$L_C^\beta = \{\alpha \in \Gamma^* \mid V_C^{f_{\alpha\beta}} \neq \bot, \ell \leq V_C^{f_{\alpha\beta}} \leq u\}$$

By Lemma 7.3.18, for all $pX \in \mathbf{S}_\Delta$ satisfying $\mathcal{P}(pX, C) > 0$ holds

$$\mathcal{P}(I[V^{f_\beta},\ell,u] = Hit[L_C^\beta] \mid Run(pX,C)) = 1 \tag{7.5}$$

because clearly $(f_\beta)_\alpha = f_{\alpha\beta}$. By Lemma 7.3.19, for each $C \in \mathrm{BSCC}_\Delta$ there is an efectively computable $k_C \geq 0$ such that either $L_C^\beta = \Gamma^*$ for all $\beta \in S[k_C]$, or $L_C^\beta = \emptyset$ for all $\beta \in S[k_C]$. Hence, for all $\alpha, \beta \in S[k_C]$ and all $pX \in \mathbf{S}_\Delta$, where $\mathcal{P}(Run(pX,C)) > 0$, we have $I[V^{f_\alpha},\ell,u] = I[V^{f_\beta},\ell,u]$ a.s. over $Run(pX,C)$ due to the equation (7.5). Let us denote $k = \max_{C\in\mathrm{BSCC}_\Delta} k_C$. Then, clearly, $\mathcal{P}_{pX}(I[V^{f_\alpha},\ell,u] = I[V^{f_\beta},\ell,u]) = 1$ for all $\alpha, \beta \in S[k]$ and all $pX \in \mathbf{S}_\Delta$ due to Proposition 4.3.10. ♢

Let $I[V_1,\ell_1,u_1],\ldots,I[V_i,\ell_i,u_i]$ be all indicators of $\mathcal{I}_{var}$ occuring in I (remember that I is a Boolean combination of indicators of $\mathcal{I}_{var} \cup \{I_{inf}\}$). By Claim (1), for each $1 \leq j \leq i$ there is an effectively computable $k_j \geq 0$ such that $\mathcal{P}_{pX}(I[V_j^{f_\alpha},\ell_j,u_j] = I[V_j^{f_\beta},\ell_j,u_j]) = 1$ for all $pX \in \mathbf{S}_\Delta$ and all $\alpha, \beta \in S[k_j]$.

Let us denote $k = \max\{k_1,\ldots,k_i\}$. Given $\beta \in \Gamma^*$ and a run w in M_Δ, we denote $I^{f_\beta}(w)$ the value of the indicator I evaluated using the reward function f_β instead of f. By the above arguments, $\mathcal{P}_{pX}(I^{f_\alpha} = I^{f_\beta}) = 1$ for all $\alpha, \beta \in S[k]$ and all $pX \in \mathbf{S}_\Delta$. Given $\alpha \in \Gamma^*$, we denote

$$\mathcal{H}[\alpha] = \{pX \in \mathbf{S}_\Delta \mid \mathcal{P}(I=1 \mid Clean(pX\alpha)) = 1\} \cup ((Q\times\Gamma)\backslash\mathbf{S}_\Delta)$$

Claim (2). *For all $\alpha \in \Gamma^*$ holds $\mathcal{H}[\alpha] = \mathcal{H}[sp(\alpha)]$, and moreover, for all $\alpha, \beta \in S[k]$ holds $\mathcal{H}[\alpha] = \mathcal{H}[\beta]$. Each set $\mathcal{H}[\alpha]$ is effectively computable.*

Proof. It is easy to verify that for all clean runs w and all $\alpha \in \Gamma^*$, we have $I(w\lfloor\alpha) = I^{f_\alpha}(w) = I^{f_{sp(\alpha)}}(w) = I(w\lfloor sp(\alpha))$. By Lemma 4.6.4, for all $pX \in \mathbf{S}_\Delta$ and all $\alpha \in \Gamma^*$ holds

$$\begin{aligned}\mathcal{P}(I = 1 \mid Clean(pX\alpha)) &= \mathcal{P}_{pX}(I^{f_\alpha} = 1) = \mathcal{P}_{pX}(I^{f_{sp(\alpha)}} = 1) \\ &= \mathcal{P}(I = 1 \mid Clean(pX)\lfloor sp(\alpha))\end{aligned}$$

which implies $\mathcal{H}[\alpha] = \mathcal{H}[sp(\alpha)]$. Moreover, by Claim (1), for all $\alpha, \beta \in S[k]$

$$\begin{aligned}\mathcal{P}(I = 1 \mid Clean(pX\alpha)) &= \mathcal{P}_{pX}(I^{f_\alpha} = 1) = \mathcal{P}_{pX}(I^{f_\beta} = 1) \\ &= \mathcal{P}(I = 1 \mid Clean(pX\beta))\end{aligned}$$

which implies $\mathcal{H}[\alpha] = \mathcal{H}[\beta]$. The effective computability of $\mathcal{H}[\alpha]$ follows from Theorem 7.2.4. ◊

Let us denote $\Gamma_{c>0}^{\leq k} = \{\alpha \in \Gamma_{c>0}^* \mid |\alpha| \leq k\}$. We define the DFA $\mathcal{A}$ to have the alphabet $Q \cup \Gamma$, the set of states $(2^Q \times \Gamma_{c>0}^{\leq k}) \cup \{Acc\}$, the initial state (J, ε) where $J = \{q \in Q \mid \mathcal{P}(I = 1 \mid Run(q\varepsilon)) = 1\}$, the only accepting state Acc, and the transition function γ defined as follows: Given $X \in \Gamma$ and a state (A, β) of $\mathcal{A}$, we put $(A', \beta') = \gamma((A, \beta), X)$ where

1. $A' = \{p \in Q \mid pX \in \mathcal{H}[\beta], \sum_{r \in A}[pXr] + [pX\uparrow] = 1\}$;
2. if $X\beta \in \Gamma_{c>0}^{\leq k}$, then $\beta' = X\beta$, else $\beta' = \beta$.

Given $q \in Q$ and a state (A, β) of $\mathcal{A}$, we define $\gamma((A, \beta), q) = Acc$ if and only if $q \in A$.

It follows from Claim (2), Corollary 4.1.10, and Theorem 2.3.1 that the automaton $\mathcal{A}$ is effectively computable. It remains to prove the following claim:

Claim (3). $\mathcal{C}(\mathcal{A}) = \{p\alpha \in \mathcal{C}(\Delta) \mid p\alpha \models \mathcal{P}^{=1}(I = 1)\}$

Proof. Let $p \in Q$, $\alpha \in \Gamma^*$, and let us assume that $\gamma((J, \varepsilon), \alpha^R) = (A', \beta')$. We show that $p \in A'$ if and only if $p\alpha \models \mathcal{P}^{=1}(I = 1)$. The rest follows from the definition of $\mathcal{A}$.

By induction: If $\alpha = \varepsilon$, then $\gamma((J, \varepsilon), \varepsilon) = (J, \varepsilon)$, and the result follows from the definition of J. Let us assume that $\alpha = X\zeta$, and that $\gamma((J, \varepsilon), \zeta^R) = (A, \beta)$. It is easy to prove that β is a suffix of $sp(\zeta)$, and moreover, if $sp(\zeta) \neq \beta$, then $|\beta| = k$ (i.e., $\beta, \zeta \in S[k]$). By Claim (2), $\mathcal{H}[\beta] = \mathcal{H}[\zeta]$.

Given $p \in Q$, we denote $\Omega_p = \{w \in Run(pX\zeta) \mid I(w) = 1\}$, $\Omega'_p = \{w \in Clean(pX\zeta) \mid I(w) = 1\}$, and given $r \in Q$, we denote $\Omega''_r = \{w \in Run(r\zeta) \mid I(w) = 1\}$. Clearly, $\mathcal{P}(\Omega_p) = 1$ if and only if $p\alpha \models \mathcal{P}^{=1}(I = 1)$. Hence, it suffice to show that $p \in A'$ if and only if $\mathcal{P}(\Omega_p) = 1$. Moreover, for all $pX \in \mathbf{S}_\Delta$, $\mathcal{P}(\Omega'_p) = [pX\uparrow]$ if and only if $pX \in \mathcal{H}[\zeta]$.

By induction, $\mathcal{P}(\Omega''_r) = 1$ if and only if $r \in A$. It is easy to verify that $\Omega_p = \Omega'_p \uplus \biguplus_{r\in Q} FPath(pXr)\lfloor\zeta \odot \Omega''_r$, which implies

$$\mathcal{P}(\Omega_p) = \mathcal{P}(\Omega'_p) + \sum_{r\in Q}[pXr]\cdot\mathcal{P}(\Omega''_r)$$

by Lemma 2.2.3 and Lemma 4.6.2.

First, assume that $p \in A'$. Then $pX \in \mathcal{H}[\beta] = \mathcal{H}[\zeta]$, and thus $\mathcal{P}(\Omega'_p) = [pX\uparrow]$. We also have $[pXs] = 0$ for all $s \in Q\backslash A$ by the definition of $\mathcal{A}$, and $\mathcal{P}(\Omega''_r) = 1$ for all $r \in A$ by the induction hypothesis. Hence, $\mathcal{P}(\Omega_p) = [pX\uparrow]+\sum_{r\in A}[pXr] = 1$.

Now let us assume that $p \notin A'$. Then either $pX \notin \mathcal{H}[\beta] = \mathcal{H}[\zeta]$, or $[pX\uparrow] + \sum_{r\in A}[pXr] < 1$. If $pX \notin \mathcal{H}[\zeta]$, then $\mathcal{P}(\Omega'_p) < [pX\uparrow]$, which implies that $\mathcal{P}(\Omega_p) < 1$. If $[pX\uparrow] + \sum_{r\in A}[pXr] < 1$, then there is $s \in Q\backslash A$ such that $[pXs] > 0$. By induction hypothesis, $\mathcal{P}(\Omega''_s) < 1$, which implies $\mathcal{P}(\Omega_p) < 1$. $\Diamond$

Bibliography

[1] P. Abdulla, N.B. Henda, and R. Mayr. Verifying infinite Markov chains with a finite attractor or the global coarseness property. In *Proceedings of LICS 2005*, pp. 127–136. IEEE, 2005.

[2] P.A. Abdulla, C. Baier, S.P. Iyer, and B. Jonsson. Reasoning about probabilistic channel systems. In *Proceedings of CONCUR 2000*, vol. 1877 of *LNCS*, pp. 320–330. Springer, 2000.

[3] R. Alur, S. Chaudhuri, K. Etessami, and P. Madhusudan. On-the-fly reachability and cycle detection for recursive state machines. In *Proceedings of TACAS 2005*, vol. 3440 of *LNCS*, pp. 61–76. Springer, 2005.

[4] R. Alur, K. Etessami, and M. Yannakakis. Analysis of recursive state machines. In *Proceedings of CAV 2001*, vol. 2102 of *LNCS*, pp. 207–220. Springer, 2001.

[5] R. Alur and P. Madhusudan. Visibly pushdown languages. In *Proceedings of STOC 2004*, pp. 202–211. ACM Press, 2004.

[6] C. Baier. *On the Algorithmic Verification of Probabilistic Systems*. Habilitation, Universität Mannheim, 1998.

[7] 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)*, vol. 1601 of *LNCS*, pp. 34–52. Springer, 1999.

[8] T. Ball and S.K. Rajamani. Bebop: A symbolic model checker for boolean programs. In *SPIN 00: SPIN Workshop*, vol. 1885 of *LNCS*, pp. 113–130. Springer, 2000.

[9] T. Ball and S.K. Rajamani. The SLAM project: debugging system software via static analysis. In *Proceedings of POPL 2002*, pp. 1–3. ACM Press, 2002.

[10] S. Basu, R. Pollack, and M. F. Roy. *Algorithms in real algebraic geometry*. Springer, first edition, 2003.

[11] J.A. Bergstra, A. Ponse, and S.A. Smolka, editors. *Handbook of Process Algebra*. 2001.

[12] N. Bertrand and Ph. Schnoebelen. Model checking lossy channel systems is probably decidable. In *Proceedings of FoSSaCS 2003*, vol. 2620 of *LNCS*, pp. 120–135. Springer, 2003.

[13] P. Billingsley. *Probability and Measure*. New York: John Wiley & Sons, third edition, 1995.

[14] A. Bouajjani, J. Esparza, and O. Maler. Reachability analysis of pushdown automata: application to model checking. In *Proceedings of CONCUR'97*, vol. 1243 of *LNCS*, pp. 135–150. Springer, 1997.

[15] T. Brázdil, J. Esparza, and A. Kučera. Analysis and prediction of the long-run behavior of probabilistic sequential programs with recursion. In *Proceedings of FOCS 2005*, pp. 521–530. IEEE, 2005.

[16] T. Brázdil and A. Kučera. Computing the expected accumulated reward and gain for a subclass of infinite Markov chains. In *Proceedings of FST&TCS 2005*, vol. 3821 of *LNCS*, pp. 372–383. Springer, 2005.

[17] T. Brázdil, A. Kučera, and O. Stražovský. On the decidability of temporal properties of probabilistic pushdown automata. In *Proceedings of STACS'2005*, vol. 3404 of *LNCS*, pp. 145–157. Springer, 2005.

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

[19] E.M. Clark, O. Grumberg, and D.A. Peled. *Model Checking*. The MIT Press, 1999.

[20] E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Trans. Program. Lang. Syst.*, 8(2):244–263, 1986.

[21] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. In *Proceedings of ICALP'90*, vol. 443 of *LNCS*, pp. 336–349. Springer, 1990.

[22] C. Courcoubetis and M. Yannakakis. The complexity of probabilistic verification. *JACM*, 42(4):857–907, 1995.

[23] L. de Alfaro. Temporal logics for the specification of performance and reliability. In *Proceedings of STACS'97*, vol. 1200 of *LNCS*, pp. 165–176. Springer, 1997.

[24] L. de Alfaro. How to specify and verify the long-run average behavior of probabilistic systems. In *Proceedings of LICS'98*, pp. 454–465. IEEE, 1998.

[25] E. A. Emerson and A. P. Sistla. Deciding branching time logic. In *STOC '84: Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pp. 14–24, New York, NY, USA, 1984. ACM Press.

[26] J. Esparza, D. Hansel, P. Rossmanith, and S. Schwoon. Efficient algorithms for model checking pushdown systems. In *Proceedings of CAV 2000*, vol. 1855 of *LNCS*, pp. 232–247. Springer, 2000.

[27] J. Esparza, A. Kučera, and R. Mayr. Model-checking probabilistic pushdown automata. In *Proceedings of LICS 2004*, pp. 12–21. IEEE, 2004.

[28] J. Esparza, A. Kučera, and R. Mayr. Quantitative analysis of probabilistic pushdown automata: Expectations and variances. In *Proceedings of LICS 2005*, pp. 117–126. IEEE, 2005.

[29] J. Esparza, A. Kučera, and S. Schwoon. Model-checking LTL with regular valuations for pushdown systems. In *Proceedings of TACS'2001*, vol. 2215 of *LNCS*, pp. 316–339. Springer, 2001.

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

[31] K. Etessami and M. Yannakakis. Algorithmic verification of recursive probabilistic systems. In *Proceedings of TACAS 2005*, vol. 3440 of *LNCS*, pp. 253–270. Springer, 2005.

[32] K. Etessami and M. Yannakakis. Checking LTL properties of recursive Markov chains. In *Proceedings of 2nd Int. Conf. on Quantitative Evaluation of Systems (QEST'05)*, 2005.

[33] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proceedings of STACS'2005*, vol. 3404 of *LNCS*, pp. 340–352. Springer, 2005.

[34] K. Etessami and M. Yannakakis. Recursive Markov decision processes and recursive stochastic games. In *Proceedings of ICALP 2005*, vol. 3580 of *LNCS*. Springer, 2005.

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

[36] S.P. Iyer and M. Narasimha. Probabilistic lossy channel systems. In *Proceedings of TAPSOFT'97*, vol. 1214 of *LNCS*, pp. 667–681. Springer, 1997.

[37] J. D. Ullman J. E. Hopcroft, R. Motwani. *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, second edition, 2000.

[38] J. G. Kemeny and J. L. Snell. *Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix"*. Springer, first edition, 1983.

[39] J. G. Kemeny, J. L. Snell, A. W. Knapp, and D.S. Griffeath. *Denumerable Markov Chains*. Springer, second edition, 1976.

[40] M.Z. Kwiatkowska. Model checking for probability and time: from theory to practice. In *Proceedings of LICS 2003*, pp. 351–360. IEEE, 2003.

[41] J. R. Norris. *Markov Chains*. Oxford University Press, first edition, 1998.

[42] Ch. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[43] E. L. Post. A variant of a recursively unsolvable problem. *Bulletin of the American Mathematical Society*, 52:264–268, 1946.

[44] A. Rabinovich. Quantitative analysis of probabilistic lossy channel systems. In *Proceedings of ICALP 2003*, vol. 2719 of *LNCS*, pp. 1008–1021. Springer, 2003.

[45] A. Remke, B.R. Haverkort, and L. Cloth. Model checking infinite-state Markov chains. In *Proceedings of TACAS 2005*, vol. 3440 of *LNCS*, pp. 237–252. Springer, 2005.

[46] W. Rudin. *Principles of Mathematical Analysis*. McGraw-Hill, third edition, 1976.

[47] S. Safra. *Complexity of automata on infinite objects*. PhD thesis.

[48] J. Stoer and R. Bulirsch. *Introduction to Numerical Analysis*. Springer, third edition, 2004.

[49] W. Thomas. Automata on infinite objects. *Handbook of TCS*, B:135–192, 1991.

[50] M. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proceedings of FOCS'85*, pp. 327–338. IEEE, 1985.

[51] M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Banff Higher Order Workshop*, pp. 238–266, 1995.

[52] I. Walukiewicz. Model checking CTL properties of pushdown systems. In *Proceedings of FST&TCS'2000*, vol. 1974 of *LNCS*, pp. 127–138. Springer, 2000.

[53] I. Walukiewicz. Pushdown processes: Games and model-checking. *I&C*, 164(2):234–263, 2001.