Reduction Methods for Probabilistic Model Checking

Dissertation

zur Erlangung des akademischen Grades
Doktor rerum naturalium (Dr. rer. nat.)

vorgelegt an der
Technischen Universität Dresden
Fakultät Informatik

eingereicht von

M.Sc. Marcus Thomas Größer
geboren am 21. März 1975 in Bernkastel-Kues

Gutachter:

Prof. Dr. rer. nat. habil. Christel Baier
Technische Universität Dresden

Prof. Dr. Pedro R. D'Argenio
Universidad Nacional de Córdoba

Prof. Dr. Ir. Joost-Pieter Katoen
Rheinisch-Westfälische Technische Hochschule Aachen

Tag der Verteidigung: 28.08.2008

Dresden, im September 2008

Abstract

Model Checking is a *fully automatic* verification method that has undergone a vast development for almost 30 years now. In contrast to simulation and testing, model checking is a verification technique that explores all possible system states exhaustively and can therefore reveal errors that have not been discovered by testing or simulation. It thus is a prominent verification technique for safety-critical systems. However, exploring the entire state space makes model checking very sensitive to the size of the system to be verified.

In this thesis, we address the issue of reduction techniques for probabilistic model checking. Taking probabilities into account in addition to nondeterministic behavior expands the possibilities of modeling certain aspects of the system under consideration. While nondeterministic systems are considered in connection to underspecification, interleaving of several processes and interaction with the specified system from the outside, the probabilities can be exploited to model a certain probability of error or other stochastic behavior both occurring in various real world applications, e.g. randomized algorithms or communication protocols over faulty media. In this thesis we restrict our investigations to models that are specified by Markov decision processes.

On the one hand we study the applicability of partial order reduction methods on Markov decision processes. These allow to construct a submodel of the model to be verified and to model check the (smaller) submodel, yielding a valid answer also for the original model. We investigate Doron Peled's ample set method in a probabilistic setting and point out that the classical conditions on the ample sets are not sufficient when dealing with Markov decision processes. We show a conservative extension of the classical conditions which makes the ample set method work for Markov decision processes with respect to linear-time properties. Here conservative means that the new stronger conditions are equivalent to the classical ones, if they are applied to non-probabilistic (classical) systems. We also show how to extend the classical conditions for branching time properties such that the ample set method works for Markov decision processes with respect to probabilistic branching time properties.

In the context of automata-theoretic model checking another chance to enhance the performance is to generate a "small" automaton for the given specification that one wants to verify for a system. We introduce and investigate the concept of probabilistic ω-automata. It turned out that they do not apply to the model checking of MDPs as their emptiness problem is undecidable. Nevertheless they form an interesting field of research. We introduce probabilistic Büchi automata (PBA) as acceptors for languages of infinite words, where a word is accepted by a PBA if and only if the set of accepting runs for this word has a positive measure. We show that PBA strictly subsume the ω-regular languages and also study the efficiency (with respect to the size) of PBA. We show that PBA are closed under union, intersection and complementation. Moreover we prove that the emptiness problem is undecidable for PBA. This result implies the undecidability of some qualitative ω-regular properties for partially observable Markov decision processes. Furthermore we investigate

PBA under the so-called almost-sure semantics, for which a word is accepted by the PBA if and only if the set of accepting runs for this word has measure one. We show a weaker expressiveness of PBA under the almost-sure semantics and prove that the emptiness problem becomes decidable. In this context we show a more general result, namely that the almost-sure Büchi objective and the positive co-Büchi objective are decidable for partially observable Markov decision processes.

Acknowledgements

First and foremost I want to express my gratitude to my supervisor Christel Baier. I am deeply indebted to her because she has taught me almost everything that I know about computer science. I want to thank her for her excellent guidance, her friendly character, many many discussions and her patience.

I would also like to thank all the people that I worked with during my PhD studies. Special thanks go to Pedro D'Argenio for being a such a nice person and the members of the VOSS-projects.

Another person that I want to mention is my long-time office-mate Frank Ciesinski who is not always as funny as he thinks he is, but who is quite a delight. Thanks go also to Nathalie Bertrand for her very pleasant nature and for playing squash with me.

Greetings to my friends in Bonn, especially to Marcus H., Martin O., Yvonne D. and Stefan S. with whom I spent many amusing and entertaining evenings, nights and holidays. Greetings also to all the people that I went on a climbing trip with, especially Artur G., Christoph W. and my girlfriend Monika.

At last I thank my parents, my brother, my sister and my girlfriend for their support and encouragement.

to

Irmchen

* 24.07.1943 † 22.01.2008

[0]Special thanks go to Emil, Gustav, Manfred, Mathilde, Helmut B. and Tim E.

0 Contents

List of Figures

1 Introduction

Model Checking is a *fully automatic* verification method for both hard- and software systems that has undergone a very promising development for almost 30 years now. In contrast to simulation and testing, model checking is a verification technique that explores all possible system states exhaustively and checks whether the system truly satisfies a given specification. It thus has the potential to reveal errors that have not been discovered by testing or simulation. Exploring the entire state space makes model checking very sensitive to the size of the system to be verified. Unfortunately, the systems under consideration suffer from the so-called *state space explosion*. Consider for example a piece of software code that uses 10 boolean variables and 20 control locations (it may consist of 20 lines). Then the system that evolves from executing this code has $20 \cdot 2^{10}$ possible different states. The same problem arises when modeling hardware circuits, as every register bit adds a factor 2 to the size of the state space. Thus, industrial model checking problems can easily yield systems that exceed the amount of available computer memory.

In this thesis, we focus on techniques that apply to probabilistic model checking and its state space explosion problem. Taking probabilities into account in addition to nondeterministic behavior expands the possibilities of modeling certain aspects of the system under consideration. While nondeterministic systems are considered in connection to underspecification, interleaving of several processes and interaction with the specified system from the outside, the probabilities can be exploited to model a certain probability of error or other stochastic behavior both occurring in various real world applications, e.g. randomized algorithms or communication protocols over faulty media. To model probabilistic systems, Markov chains and Markov decision processes (MDPs) are commonly used. In contrast to Markov chains, nondeterminism and probabilism coexist in MDPs. In the past, many of the specification formalisms and verification techniques that have been established in the non-probabilistic setting have been adapted to reason about quantitative aspects of probabilistic systems, e.g. with the help of process equivalences and model checking against temporal logical specifications. In the context of specifiying probabilistic systems against formulae of Linear Temporal Logic (LTL) [Pnu77] a specification consists of an LTL-formula φ that expresses a certain path property and a probability bound, e.g. "$= 1$", "$\geq p$" or "$< p$" for some $p \in [0, 1]$. Hence, LTL can serve to formulate *qualitative* properties such as "with probability 1 any request will eventually be answered" or *quantitative* properties such as "there is a 98 % chance to reach a goal state" or "the probability that a waiting process is never allowed to enter its critical section is less than 0.005". In the context of specifying branching behavior, the "for all path" and "there exists a path" quantifier of Computation Tree Logic (CTL) [CE81] have been substituted by a probabilistic operator that requires a given probability on a certain set of paths, yielding the logic PCTL (Probabilistic Computation Tree Logic) [Han94, HJ94, BdA95]. Verification algorithms for qualitative or quantitative probabilistic model checking rely on modifications of the model checking techniques for non-probabilistic systems (such as graph algorithms to explore the state space and algorithms that construct an automaton for the given specification/formula) and their combination with numerical methods to solve linear equation systems or linear programming problems [Var85, VW86, CY90, PZ93, HJ94, ASBB95, BdA95, CY95, dA97a, dA97b, dA97a, BS98, BK98, CSS03, BRV04]. Thus, the state space explosion problem is at least as relevant (or even more) than in the non-probabilistic setting. In Figure 1.1 we show a schematic view on the automata-theoretic verification procedure.

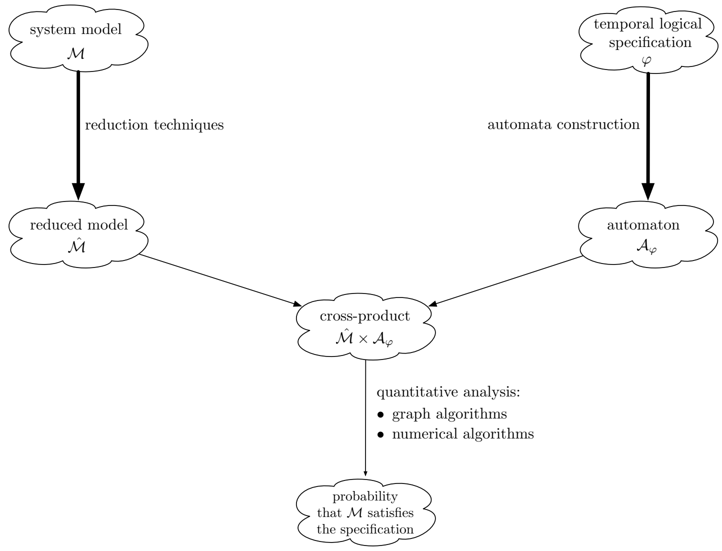


Figure 1.1: Basic scheme of the automata-theoretic model checking procedure

To reason about non-probabilistic systems, a variety of methods have been developed to tackle the state space explosion problem. This includes symbolic model checking as well as various reduction techniques, see e.g. [CGP99] for an overview. The symbolic methods are mainly based on multi-terminal binary decision diagrams and focus on a compact internal representation of the (full) system [BCM$^+$92, HMPS94]. So the goal is not to avoid the state space explosion, but to use a very compact representation of the given model. Many symbolic techniques have been modified and extended such that they apply to probabilistic systems [HMPS94, BCHG$^+$97, BM99, Par02, HKN$^+$03, MP03, KNP04]. For instance, the PCTL-model checkers PRISM [KNP02], ProbVERUS [HGCC99] and RAPTURE [JDL02] are based on a symbolic representation of the system to be analyzed. Also hybrid approaches which combine the compact model representation of symbolic techniques with the good performance of numerical computations of explicit techniques have been developed [KNP04].

Somewhat orthogonal to this approach are the many reduction techniques. Here the goal is to generate a reduced subsystem which is "equivalent" (with respect to the properties to be verified) to the original system. Then model checking is applied to the reduced system, yielding the desired answer not only for the reduced system, but also for the original system. A large class of reduction techniques are bisimulation-minimization techniques

[HT92, BEMC00, PLS00, CS02, KKLW07] that aim to aggregate bisimilar states and to construct an "equivalent" quotient of the original model. The basic concepts of these techniques are also used in (property-driven) abstraction-refinement methods [DJJL01, Hut02, Hut05]. Recently, symmetry reduction techniques have been developed for probabilistic model checking [KNP06b]. These techniques aim at models with non-trivial but interchangeable components and use the inherent internal symmetries to reduce the state space. Also a novel abstraction method for Markov decision processes based on stochastic two-player games has been proposed in [KNP06a]. Another class of reduction techniques are partial order reduction methods which have been thouroughly studied for non-probabilistic models [Pel93, HP94, Val94, God96, GPS96, PPH97, Pel97].

In the context of automata-theoretic model checking, another chance to enhance the performance is to generate a "small" automaton for the given specification. It is well-known that the smallest equivalent nondeterministic Büchi automaton (NBA) for a given LTL formula might be of exponential size (in the size of the formula) and that the determinization of NBA into deterministic Rabin automata (DRA) might also cause an exponential blow-up [Mic88, Tho90]. Moreover, [KV98] gives a class of LTL formulae such that the size of the smallest equivalent deterministic Rabin (Streett or Büchi) automaton is double exponential in the square root of the length of the LTL formula. Various algorithms have been developed to construct a "small" NBA for a given LTL formula using the standard tableau-based approach [GPVW96] or a detour involving an intermediate translation step using alternating co-Büchi automata [GO01]. These NBA can then be transformed into a deterministic ω-automata using an improved transformation [Pit06]. Some of the algorithms have been successfully implemented [KB05] and improved for stutter insensitive properties [KB07]. Also a transformation from LTL to deterministic ω-automata was developed that circumvents the usual tableau construction [Sch97]. A novel approach [MS08] transforms LTL formulae into symbolically represented deterministic automata, avoiding the concept of Safra trees which showed to be not amenable to a symbolic implementation.

In this thesis, we address the issue of reduction techniques for probabilistic model checking to increase the efficiency of the qualitative/quantitative analysis of probabilistic systems against temporal logics. Especially we focus on methods to reduce the size of the system to be analyzed in LTL model checking for MDPs. Figure 1.1 shows the common basic scheme of LTL model checking for MDPs. We focus on two aspects, namely a certain reduction technique to construct a smaller but "equivalent" submodel of the given model and on an efficient automata construction introducing a new class of ω-automata.

- One particular reduction method that was used very successfully in LTL model checking for classical (non-probabilistic) systems is the partial order reduction [Val92, Pel93, Val94, HP94, God96, GPS96, Pel97, Val97, PPH97]. We investigated a special instance of partial order reduction, the so-called ample set method, and extended this method to probabilistic systems. We established results that allow to use the ample set method for LTL model checking for MDPs [BGC04] and also extended these result to the model checking of PCTL [BDG06] for MDPs. This will be presented in chapter 3 of this thesis. The probabilistic partial order reduction has been implemented in the quantitative LTL model checker LiQuor [BCG05, BC06a] which has been developed in our workgroup by Frank Ciesinski. Similarly to the non-probabilistic scenario (e.g.

the model checker SPIN [Hol03]), it shows good reductions in practice.

- The automata-theoretic approach of LTL model checking starts with a specification given as an LTL formula and transforms the formula into an equivalent ω-automaton. Dealing with MDPs, the known methods use a deterministic-in-limit ω-automaton for the verification of qualitative LTL specifications and a deterministic ω-automaton for the verification of quantitative LTL specifications [CY95]. In both cases this yields a double exponential blow-up (in the size of the formula) which cannot be avoided in general as it meets the lower bounds of LTL model checking for MDPs shown in [CY95]. We introduced and studied a new class of ω-automata, namely probabilistic ω-automata, especially probabilistic Büchi automata (PBA) [BG05] hoping that they could be used for qualitative LTL model checking for MDPs (and yield "small" automata). It showed that they have a variety of interesting (and surprising) properties. On the one hand they can indeed be exponentially smaller than nondeterministic Streett automata and nondeterministic Büchi automata. On the other hand, the emptiness problem for PBA is undecidable and so is the model checking problem for MDPs against a PBA [BBG08]. Nevertheless, probabilistic ω-automata (as language acceptors) turned out to be a very challenging and interesting field of research and chapter 4 is dedicated to that topic. As PBA are a special case of POMDPs, the established undecidability results have a relevance for partial information games with ω-regular winning objectives [CDHR06] as well as POMDPs [Son71, Mon82, PT87, Lov91], which are used to model a wide range of applications, such as mobile robot navigation, probabilistic planning task, elevator control, etc. PBA also find an application in randomized monitoring [CSV08].

1.1. Aims and outline of the thesis

The goals of this thesis are twofold. On the one hand we study the concept of partial order reduction in a probabilistic setting and on the other hand we investigate probabilistic Büchi automata as acceptors for languages of infinite words. As both of these topics were new research topics, this thesis presents several original contributions. It also gives a survey of known results. This thesis consists of four chapters, starting with an introduction in chapter 1. Chapter 2 contains preliminaries on Markovian models, ω-automata and temporal logics.

In chapter 3 we examine the concept of partial order reduction in a probabilistic setting. In particular we study how the ample set method can be extended for Markov decision processes.

a) We prove a conservative extension of the ample set method with respect to Markov decision processes and linear-time properties. (Section 3.3)

b) We prove a conservative extension of the ample set method with respect to Markov decision processes and branching time properties. (Section 3.4)

c) We give a short overview of the known connections between different partial order reduction criteria and probabilistic process equivalences on Markov decision processes. (Section 3.5)

In chapter 4 we introduce probabilistic ω-automata as acceptors for languages of infinite words.

a) We study the expressiveness and efficiency of probabilistic Büchi automata (PBA). We show that PBA strictly subsume the ω-regular languages and that the accepted language of a PBA does not only depend on the topological structure of the automaton, but also on the precise transition probabilities. (Section 4.2)

b) We investigate composition operators for PBA and prove that PBA are closed under complementation using an advanced powerset construction. (Section 4.3)

c) We show that the emptiness problem is undecidable for PBA and conclude divers undecidability results for related questions. As PBA are a special case of partially observable Markov decision processes (POMDPs), we conclude the undecidability of qualitative ω-regular properties for POMDPs, which to our knowlegde is a new result. (Subsection 4.4.1)

d) We also consider PBA under a threshold semantics and show that the class of recognizable languages might be (depending on the threshold) a proper superset of the class of languages that are recognizable by PBA with the standard semantics (i.e. the threshold equals zero). (Subsection 4.4.1.4)

e) We examine a different semantics for PBA, the almost-sure semantics and show a weaker expressiveness under this semantics. In particular, with the almost-sure semantics PBA are not closed under complementation, but still dependent on the exact transition probabilities. We prove that the emptiness problem is decidable for PBA with the almost-sure semantics. We do so by showing a more general result, namely that the almost-sure repeated reachability problem for POMDs is decidable. Moreover we prove that the positive co-Büchi objective is decidable for POMDPs. We also deduct that a nonempty almost-sure PBA recognizable language contains a finite-memory word. (Subsection 4.4.2)

2 Preliminaries

2.1. Basic mathematical preliminaries

$\mathbb{N}_{\geq 0}$, resp. $\mathbb{N}_{\geq 1}$ denotes the set of natural numbers greater or equal than 0, resp. 1.

$\overset{\infty}{\exists}\, i$ denotes "there exist infinitely many i $\in \mathbb{N}_{\geq 0}$ ".

By $\dot\cup$ we denote a disjoint union.

Definition 2.1.1. [Words]
Given a finite set Σ,

$$\Sigma^+ = \{s_1 \dots s_n \mid n \in \mathbb{N}_{\geq 1}, s_i \in \Sigma, 1 \leq i \leq n\}$$

denotes the set of finite nonempty words over Σ. $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$, where ϵ indicates the empty word, denotes the set of finite words.

$$\Sigma^\omega = \{s_1 s_2 \dots \ | s_i \in \Sigma, i \in \mathbb{N}_{\geq 1}\}$$

denotes the set of infinite words over Σ. Given a finite nonempty word $\rho = s_1 s_2 \dots s_n$, we denote the last letter s_n by $\mathsf{last}(\rho)$. The length $|\rho|$ of ρ equals n. For an infinite word ω, the length is equal to ∞. Given a word $\omega = s_1 s_2 \dots$ and $i \leq |\omega|$, we denote the first letter s_1 of ω by $\mathsf{first}(\omega)$, the ith letter of ω by ω_i (i.e. $\omega_i = s_i$) and the i-th prefix by $\omega{\uparrow}^i = s_1 s_2 \dots s_i$. Given an infinite word $\omega = s_1 s_2 \dots$ the ith suffix $s_i s_{i+1} \dots$ is denoted by $\omega{\uparrow}_i$.

We often identify any ω-regular language $L \subseteq \Sigma^\omega$ with the ω-regular expressions that describe L. E.g., $(a+b)^* a^\omega$ is identified with the set of infinite words over $\Sigma = \{a, b\}$ that contain only finitely many b's.

Definition 2.1.2. [Probability distribution]
Given an at most countable set S, a probability distribution on S is a function

$$\mu : S \to [0,1] \quad \text{such that} \quad \sum_{s \in S} \mu(s) = 1.$$

Given a probability distribution on S, $\mathsf{supp}(\mu)$ denotes the *support*, i.e. the states s of S with $\mu(s) > 0$. For $s \in S$, μ_s^1 denotes the unique Dirac distribution on S that satisfies $\mu_s^1(s) = 1$. By $\mathsf{Distr}(S)$ we denote the set of all probability distributions on S.

2.2. Preliminaries on various models

In this section we introduce the main models that we will work with throughout this thesis. This are Markovian models like Markov chains and Markov decision processes which will be used as description models for abstract systems and ω-automata which serve as language acceptors for ω-regular languages.

2.2.1. Markovian models

We first introduce discrete Markov chains which are basically directed graphs where the edges are labeled with a probability in $[0,1]$, such that in each state the probabilities of its outgoing edges sum up to one.

Definition 2.2.1. [Discrete Markov chain]
A discrete Markov chain is a tuple

$$\mathcal{M} = (S, \mathsf{p}, \mu),$$

where

- S is an at most countable nonempty set of states,
- $\mathsf{p} : S \times S \to [0,1]$ is a transition probability function such that $\mathsf{p}(s,.)$ is a probability distribution on S for all $s \in S$,
- μ is a probability distribution on S (called the *initial* distribution).

Throughout this thesis we will also use the notation p_{st} instead of $\mathsf{p}(s,t)$ for $s,t \in S$.

Let $T = \{(s,t) \mid \mathsf{p}_{st} > 0, s,t \in S\}$ be the set of transitions with positive probability. We refer to the directed graph (S,T) as the *underlying graph* of $\mathcal{M}$. Note that T is total, so (S,T) has no terminal nodes.

A Markov chain induces a stochastic process on the set S of its states in a natural way. The probability that the process starts in a certain state with step 0 is determined by the starting distribution. Moreover, being in state s in the $(n-1)$th step, the probability that the process is in state t in the nth step is equal to p_{st}. The fact that those probabilities do not depend on the previous steps (*history-independent* or *memoryless*) is called the *Markov property*. For a detailed discussion on Markov chains see e.g. [KSK66].

Definition 2.2.2. [Path and corresponding notation]
An (in)finite path of a Markov chain $\mathcal{M}$ is an (in)finite state sequence $\pi = s_0, s_1, \ldots$.

Given a finite path $\pi = s_0, s_1, \ldots, s_n$, we denote the first state s_0 of π by $\mathsf{first}(\pi)$ and the last state s_n by $\mathsf{last}(\pi)$. The length $|\pi|$ of π equals n. For an infinite path π, the length is equal to ∞. Given a path $\pi = s_0, s_1, \ldots,$ and $i \leq |\pi|$, we denote the ith state of π by π_i (i.e. $\pi_i = s_i$) and the i-th prefix by $\pi\!\uparrow^i = s_0, s_1, \ldots, s_i$. Given a finite or infinite path $\pi = s_0, s_1, \ldots$, an index $i \leq |\pi|$, then the ith suffix $s_i, s_{i+1}, \ldots$ is denoted by $\pi\!\uparrow_i$. We denote by $\mathsf{Path}_{\mathsf{fin}}$ (resp. $\mathsf{Path}_{\mathsf{inf}}$) the set of finite (resp. infinite) paths of a given Markov chain and by $\mathsf{Path}_{\mathsf{fin}}(s)$ (resp. $\mathsf{Path}_{\mathsf{inf}}(s)$) the set of finite (resp. infinite) paths starting in the state s. ϵ denotes the empty path.

Remark 2.2.3. If necessary we will index Path by the corresponding system, e.g. $\mathsf{Path}^{\mathcal{M}_1}_{\mathsf{inf}}$. We will do so also for other objects than Path. Nevertheless we try to omit the indexing, if the reference is clear from the context.

We will often deal with *state-labeled* systems that are equipped with a labeling function as follows.

Definition 2.2.4. [State-labeled system]
Given some kind of transition system (e.g. a Markov chain) with state space S and a set of atomic propositions AP, a labeling function of the given system with respect to the set AP is a function $L : S \to 2^{\mathsf{AP}}$ that labels a state s with those atomic propositions in AP that are supposed to hold in s.

For such systems we define the so called *trace* of a path which is the projection to the state labels.

Definition 2.2.5. [Trace of a path]
Given a state-labeled system and an infinite path $\pi = s_0, s_1, \ldots$ of the system, we define the infinite word

$$\mathsf{trace}(\pi) = L(s_0)L(s_1)\ldots \in (2^{\mathsf{AP}})^{\omega}$$

to be the trace of π. Note that as $\mathsf{trace}(\pi)$ is a word, we start counting by 1, that means $\mathsf{trace}(\pi)_i = L(\pi_{i-1})$.

We now define the probability space that formalizes the stochastic process induced by a Markov chain.

Definition 2.2.6. [Basic cylinder]
Given any kind of transition system $\mathcal{M}$, we define for $\pi \in \mathsf{Path}^{\mathcal{M}}_{\mathsf{fin}}$ the *basic cylinder* of π as

$$\Delta(\pi) = \{\rho \in \mathsf{Path}^{\mathcal{M}}_{\mathsf{inf}} \;:\; \rho\uparrow^{|\pi|} = \pi\}.$$

Definition 2.2.7. [Probability space of a Markov chain]
Given a discrete Markov chain $\mathcal{M} = (S, \mathsf{p}, \mu)$ we define a probability space

$$\Psi = (\Delta, \mathsf{Pr}),$$

such that

- Δ is the σ-algebra generated by the empty set and the set of basic cylinders over $\mathcal{M}$.
- Pr is the uniquely induced probability measure which satisfies the following: $\mathsf{Pr}(\Delta(\epsilon)) = 1$ and for all basic cylinders $\Delta(s_0, s_1, \ldots, s_n)$ over S :

$$\mathsf{Pr}(\Delta(s_0, s_1, \ldots, s_n)) = \mu(s_0) \cdot \mathsf{p}_{s_0 s_1} \cdot \ldots \cdot \mathsf{p}_{s_{n-1} s_n}$$

Given a state $s \in S$, we denote by Pr_s the probability measure that is obtained if $\mathcal{M}$ is equipped with the starting distribution μ_s^1, thus $\mathsf{Pr}_s(\Delta(s)) = 1$.

In contrast to Markov chains, nondeterminism and probabilism coexist in Markov decision processes (MDPs). In an MDP, any state s might have several outgoing action-labeled transitions, each of them is associated with a probability distribution which yields the probabilities for the successor states. As in [Put94, LS91, dA97a] we assume here that for any state s, the outgoing transitions of s have different action labels. This corresponds to the so-called reactive model in the classification of [vGSST90].

Definition 2.2.8. [Markov decision process (MDP)]
A Markov decision process is a tuple

$$\mathcal{M} = (S, \mathsf{Act}, \delta, \mu),$$

where

- S is a finite nonempty set of states,
- Act is a finite nonempty set of actions,
- $\delta : S \times \mathsf{Act} \times S \to [0, 1]$ is a transition probability function such that for all $s \in S$ and $\alpha \in \mathsf{Act}$, either $\delta(s, \alpha, .)$ is a probability distribution on S or $\delta(s, \alpha, .)$ is the null-function (i.e. $\delta(s, \alpha, t) = 0$ for all $t \in S$),
- μ is a probability distribution on S (called the *initial* distribution).

$\mathsf{Act}(s) = \{\alpha \in \mathsf{Act} \mid \exists t \in S : \delta(s, \alpha, t) > 0\}$ denotes the set of actions that are *enabled* in state s. We require for each state $s \in S$, that $\mathsf{Act}(s)$ in nonempty. If $\mathsf{Act}(s) = \mathsf{Act}$ for all states $s \in S$, we call the MDP *total*.

The intuitive operational behavior of an MDP is as follows. If s is the current state then first one of the actions $\alpha \in \mathsf{Act}(s)$ is chosen nondeterministically. Afterwards action α is executed leading to state t with probability $\delta(s, \alpha, t)$. By $\delta(s, \alpha) = \{t \mid \delta(s, \alpha, t) > 0\}$ we denote the set of α-successors of s. Given a state set $S' \subseteq S$, then $\delta(S', \alpha) = \cup_{s \in S'} \delta(s, \alpha)$ denotes the set of α-successors of S'. Moreover, given an action sequence $\alpha_1 \ldots \alpha_{i+1}$, we define $\delta(s, \alpha_1 \ldots \alpha_i \alpha_{i+1}) = \delta(\delta(s, \alpha_1 \ldots \alpha_i), \alpha_{i+1})$.

Action α is called a *probabilistic* action if it has a random effect, i.e. if there is at least one state s where α is enabled and that has two or more α-successors. Otherwise α is called non-probabilistic.

Definition 2.2.9. [Path and corresponding notation]
An infinite path of an MDP is an infinite sequence $\pi = s_0, \alpha_1, s_1, \alpha_2, \ldots \in (S \times \mathsf{Act})^\omega$ such that $\alpha_i \in \mathsf{Act}(s_{i-1})$ for $i \in \mathbb{N}_{\geq 1}$. We write paths in the form

$$\pi = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$$

$\mathsf{first}(\pi) = s_0$ denotes the starting state of π and $\pi\!\uparrow^i = s_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_i} s_i$ its i-th prefix. Finite paths are finite prefixes of infinite paths that end in a state. We use the notations $\mathsf{first}(\pi)$, resp. $\mathsf{last}(\pi)$ for the first, resp. last state of of a finite path π and $|\pi|$ for the length (number of actions). $\pi_i = s_i$ denotes the $(i+1)$st state of π and $\mathsf{Act}_i(\pi)$ denotes the ith action on π. $\mathsf{Path}_{\mathsf{fin}}(s)$ (resp. $\mathsf{Path}_{\mathsf{inf}}(s)$) denotes the set of all finite (resp. infinite) paths of $\mathcal{M}$ with starting state s. $\mathsf{Path}_{\mathsf{fin}}$ (resp. $\mathsf{Path}_{\mathsf{inf}}$) stands for the set of all finite (resp. infinite) paths in $\mathcal{M}$.

Given a path $\pi = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \ldots$ (finite or infinite), we denote by $\mathsf{Act}(\pi) = \{\alpha \mid \exists i \leq |\pi| : \alpha = \alpha_i\}$ the set of actions that occur along the path π and by $\overrightarrow{\pi_{\mathsf{Act}}} = \alpha_1\alpha_2 \ldots \in \mathsf{Act}^* \cup \mathsf{Act}^\omega$ the action sequence of π.

If $\pi = s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \ldots$ is an infinite path, then $\mathsf{Lim}(\pi)$ denotes the pair (T, A) where $T = \mathsf{inf}(\pi)$ is the set of states in π that are visited infinitely often and $A : T \to 2^{\mathsf{Act}}$ is the function that assigns to any state $t \in T$ the set $A(t)$ of actions $\alpha \in \mathsf{Act}$ such that $(s_i = t) \wedge (\alpha_{i+1} = \alpha)$ for infinitely many indices i.

A scheduler denotes an instance that resolves the nondeterminism in the states, and thus, yields a Markov chain and a probability measure on the paths. Intuitively, a scheduler takes as input the "history" of a computation (formalized by a finite path π) and chooses the next action (resp. a distribution on actions).

Definition 2.2.10. [Scheduler]
Given a Markov decision process $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$, a history dependent randomized scheduler is a function

$$\mathcal{U} : \mathsf{Path}_{\mathsf{fin}} \to \mathsf{Distr}(\mathsf{Act}),$$

such that $\mathsf{supp}(\mathcal{U}(\pi)) \subseteq \mathsf{Act}(\mathsf{last}(\pi))$ for all $\pi \in \mathsf{Path}_{\mathsf{fin}}$.

A scheduler $\mathcal{U}$ is called deterministic, if $\mathcal{U}(\pi)$ is a Dirac distribution for all $\pi \in \mathsf{Path}_{\mathsf{fin}}$. $\mathcal{U}$ is called memoryless, if $\mathcal{U}(\pi) = \mathcal{U}(\mathsf{last}(\pi))$ for all $\pi \in \mathsf{Path}_{\mathsf{fin}}$. $\mathsf{Sched}_{\mathsf{HR}}$ (resp. $\mathsf{Sched}_{\mathsf{HD}}$) denotes the set of history dependent, randomized (resp. deterministic) schedulers and $\mathsf{Sched}_{\mathsf{MR}}$ (resp. $\mathsf{Sched}_{\mathsf{MD}}$) denotes the set of memoryless randomized (resp. deterministic) schedulers.

We call a (finite or infinite) path $s_0 \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \ldots$ a $\mathcal{U}$-path, if $\mathcal{U}(s_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_i} s_i)(\alpha_{i+1}) > 0$ for all $0 \leq i < |\pi|$.

Remark 2.2.11. If all actions in Act are non-probabilistic and the initial distribution is a Dirac distribution, then considering only deterministic schedulers, our notion of an MDP reduces to an ordinary transition system with at most one outgoing α-transition per state and action α and exactly one initial state.

Given an MDP $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$ and a scheduler $\mathcal{U}$ for $\mathcal{M}$, the behavior of $\mathcal{M}$ under $\mathcal{U}$ can be formalized by an infinite-state Markov chain $\mathcal{M}_{\mathcal{U}} = (\mathsf{Path}^{\mathcal{M}}_{\mathsf{fin}}, \mathsf{p}, \mu)$, where

$$\mathsf{p}(\pi,\ \pi') = \mathcal{U}(\pi)(\alpha) \cdot \delta(\mathsf{last}(\pi), \alpha, \mathsf{last}(\pi')),$$

for $\pi, \pi' \in \mathsf{Path}^{\mathcal{M}}_{\mathsf{fin}}$ with $|\pi'| = |\pi| + 1$, $\pi'\!\uparrow^{|\pi|} = \pi$ and α is the last action on the path π', i.e.

$$\pi \xrightarrow{\alpha} \mathsf{last}(\pi') \ = \ \pi'.$$

As the states of $\mathcal{M}_{\mathcal{U}}$ are finite paths of $\mathcal{M}$, this notation is somewhat inconvenient. Let $\Omega = (\mathsf{Path}_{\mathsf{inf}}^{\mathcal{M}_{\mathcal{U}}}, \Delta^{\mathcal{M}_{\mathcal{U}}})$ and $\Omega' = (\mathsf{Path}_{\mathsf{inf}}^{\mathcal{M}}, \Delta^{\mathcal{M}})$, where $\Delta^{\mathcal{M}_{\mathcal{U}}}$ is the σ-algebra generated by the empty set and the set of basic cylinders over $\mathcal{M}_{\mathcal{U}}$ and $\Delta^{\mathcal{M}}$ is the σ-algebra generated by the empty set and the set of basic cylinders over $\mathcal{M}$. We define

$$\mathsf{f} : \mathsf{Path}_{\mathsf{inf}}^{\mathcal{M}_{\mathcal{U}}} \to \mathsf{Path}_{\mathsf{inf}}^{\mathcal{M}},$$

as $\mathsf{f}(\pi_0 \xrightarrow{\alpha_1} \pi_1 \xrightarrow{\alpha_2} \ldots) = \mathsf{last}(\pi_0) \xrightarrow{\alpha_1} \mathsf{last}(\pi_1) \xrightarrow{\alpha_2} \ldots$ (note that the π_i's are finite paths of $\mathcal{M}$). Then f is a measurable function and we define the the following probability measure on $\Delta^{\mathcal{M}}$.

$$\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(A') = \mathsf{Pr}^{\mathcal{M}_{\mathcal{U}}}(\mathsf{f}^{-1}(A')), \text{ for } A' \in \Delta^{\mathcal{M}}.$$

Then given a scheduler $\mathcal{U}$ for $\mathcal{M}$, the probability measure $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}$ formalizes the behavior of $\mathcal{M}$ under $\mathcal{U}$, where we have the convenience to talk about measures of sets of infinite paths of $\mathcal{M}$. As for Markov chains, given a state $s \in S$, we denote by $\mathsf{Pr}_s^{\mathcal{M},\mathcal{U}}$ the probability measure that is obtained if $\mathcal{M}$ is equipped with the starting distribution μ_s^1. For more information on measure theory see e.g. [Bau78].

We will also fix the following notation for convenience. Given an MDP $\mathcal{M}$, a scheduler $\mathcal{U}$ and a path property E, we will write

$$\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(E) := \mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\{\pi \in \mathsf{Path}_{\mathsf{inf}}^{\mathcal{M}} \mid \pi \text{ satisfies } E\})$$

for the probability that the property E holds in $\mathcal{M}$ under the scheduler $\mathcal{U}$.

Throughout this thesis, we shall use the concepts of de Alfaro's end components [dA97a, dA98], which can be seen as the MDP counterpart to terminal strongly connected components in Markov chains. Intuitively, an end component of an MDP is a nonempty strongly connected subMDP, that means an end component consists of a nonempty state set $T \subseteq S$ and a nonempty action set $A(t)$ for each state $t \in T$ such that, once T is entered and only actions in $A(t)$ are chosen, the set T will not be left and any state of T can be reached from any other state in T.

Definition 2.2.12. [End components, cf. [dA97a, dA98]]
Let $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$ be an MDP. An end component of $\mathcal{M}$ is a pair (T, A) where $\emptyset \neq T \subseteq S$ and $A : T \to 2^{\mathsf{Act}}$ is a function such that

- $\emptyset \neq A(s) \subseteq \mathsf{Act}(s)$ for all states $s \in T$,
- $\sum_{t \in T} \delta(s, \alpha, t) = 1$ for all states $s \in T$ and actions $\alpha \in A(s)$,
- the underlying digraph $(T, \to_A)$ of (T, A) is strongly connected.

Here, $\to_A$ denotes the edge-relation induced by A, that is $s \to_A t$ if and only if $\delta(s, \alpha, t) > 0$ for some action $\alpha \in A(s)$.

Given an MDP $\mathcal{M}$ and a scheduler $\mathcal{U}$ it holds that in the process induced by $\mathcal{U}$, almost all path of $\mathcal{M}$ (following $\mathcal{U}$) "end" in an end component, that is their limit $\mathsf{Lim}(.)$ forms an end component. For the following lemma see [dA97a, dA98].

Lemma 2.2.13 (Almost-sure end component). For any MDP $\mathcal{M}$ and scheduler $\mathcal{U}$,

$$\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\{\pi \in \mathsf{Path}_{\mathsf{inf}}^{\mathcal{M}} \mid \mathsf{Lim}(\pi) \text{ is an end component }\}) = 1.$$

We shall establish some results on partially observable Markov decision processes [Son71, Mon82, PT87, Lov91] in chapter 4. Therefore we give the definition here.

Definition 2.2.14. [Partially observable Markov decision process (POMDP)]
A finite partially observable Markov decision process is a pair

$$(\mathcal{M}, \sim),$$

where

- $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$ is a Markov decision process,
- $\sim \subseteq S \times S$ is an equivalence relation such that for all $s \sim t \in S$, $\mathsf{Act}(s) = \mathsf{Act}(t)$.

Given a POMDP $(\mathcal{M}, \sim)$, an observation-based scheduler $\mathcal{U}$ is a scheduler for $\mathcal{M}$ that is consistent with $\sim$, i.e. $\mathcal{U}(s_0 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_n} s_n) = \mathcal{U}(t_0 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_n} t_n)$ if $s_i \sim t_i$ for $0 \leq i \leq n$. $\mathsf{Sched}^{(\mathcal{M},\sim)}$ denotes the set of observation-based schedulers.

2.2.2. ω-automata

Similar to finite automata that serve to accept languages of finite words, there exists the concept of nondeterministic ω-automata with which an accepted language of infinite words is associated. As in the finite case the automaton works as follows. Reading a certain input letter in a given state, the automaton nondeterministically moves to a successor state. As the input word is infinite the automaton produces a set of infinite runs for the input word. In order to accept the input word, the automaton has to produce at least one "accepting" run, where in contrast to the finite case the acceptance condition takes the infinite behavior into account. For more information on ω-automata see e.g. [Tho90, GTW02].

Definition 2.2.15. [Nondeterministic ω-automata]
A nondeterministic ω-automaton is a tuple

$$\mathcal{A} = (Q, \Sigma, \delta, Q_0, \mathsf{Acc}),$$

where

- Q is a finite nonempty set of states,
- Σ is a finite nonempty input alphabet,
- $\delta : Q \times \Sigma \to 2^Q$ is a transition function,
- $Q_0 \subseteq Q$ is a nonempty set of initial states and
- Acc is an *acceptance condition*.

We call an automaton *deterministic* if $|Q_0| = 1$ and $|\delta(p, a)| \leq 1$ for all $p \in Q$ and $a \in \Sigma$. If $|\delta(p, a)| \geq 1$ for all $p \in Q$ and $a \in \Sigma$, we call the automaton *total*.

Within this thesis, we consider the following acceptance conditions.

- Büchi acceptance condition: $\mathsf{Acc} \subseteq Q$ (we then write F instead of Acc)
- Rabin or Streett acceptance condition: $\mathsf{Acc} = \{(H_1, K_1), \ldots, (H_n, K_n)\}$, $H_i, K_i \subseteq Q, 1 \leq i \leq n$

Let $T \subseteq Q$ be a subset of states. Given a Büchi acceptance condition F, the set T is called *accepting*, if $T \cap F \neq \emptyset$. Given a Rabin acceptance condition, T is called *accepting*, if there exists $1 \leq i \leq n$ such that $T \cap H_i = \emptyset$ and $T \cap K_i \neq \emptyset$. Given a Streett acceptance condition, T is called *accepting*, if for all $1 \leq i \leq n$ it holds that $T \cap H_i \neq \emptyset$ or $T \cap K_i = \emptyset$. Thus, Rabin and Streett acceptance are complementary to each other .

Remark 2.2.16. Note that a given Büchi acceptance condition F can be expressed by the equivalent Rabin condition $\{(\emptyset, F)\}$ and also by the equivalent Streett condition $\{(F, Q)\}$.

A nondeterministic (resp. deterministic) ω-automaton with a Büchi, resp. Rabin, resp. Streett acceptance condition is called nondeterministic (resp. deterministic) Büchi (NBA (resp. DBA)), resp. Rabin (NRA (resp. DRA)), resp. Streett (NSA (resp. DSA)) automaton.

ω-automata serve as language acceptors for languages of infinite words over the input alphabet. A *run* for an infinite word $\omega = a_1 a_2 \ldots$ is an infinite state sequence $\pi = p_0, p_1, \ldots$ such that $p_0 \in Q_0$ and $p_i \in \delta(p_{i-1}, a_i), i \in \mathbb{N}_{\geq 1}$. $\mathsf{inf}(\pi) = \{p \in Q \mid \overset{\infty}{\exists} i \in \mathbb{N}_{\geq 0} \text{ s.th. } \pi_i = p\}$ denotes the set of states that occur infinitely often in π. An infinite run π is called *accepting*, if $\mathsf{inf}(\pi)$ is accepting with respect to the acceptance condition. We will sometimes refer to finite runs, which mean finite state sequences $p_0, p_1, \ldots p_n$ such that $p_0 \in Q_0$, $p_i \in \delta(p_{i-1}, a_i), 1 \leq i \leq n$ and $\delta(p_n, a_{n+1}) = \emptyset$. That is, the automaton cannot consume the input letter a_{n+1} in state p_n and *rejects*.

The *accepted language* of a nondeterministic ω-automaton $\mathcal{A}$ is defined as

$$\mathcal{L}(\mathcal{A}) = \{\omega \in \Sigma^\omega \mid \exists \text{ accepting run for } \omega \text{ in } \mathcal{A}\}.$$

Given an automata type, e.g. NBA, we denote by e.g. $\mathbb{L}(\mathsf{NBA})$ the class of languages definable by this type of automata. It is well known that [Tho90, GTW02]

$$\mathbb{L}(\mathsf{DBA}) \subsetneq \mathbb{L}(\mathsf{NBA}) = \mathbb{L}(\mathsf{DRA}) = \mathbb{L}(\mathsf{NRA}) = \mathbb{L}(\mathsf{DSA}) = \mathbb{L}(\mathsf{NSA}) = \mathbb{L}(\omega\text{-}\mathsf{reg}),$$

where $\mathbb{L}(\omega\text{-}\mathsf{reg})$ denotes the class of ω-regular languages.

2.3. Preliminaries on Temporal Logics

At last we will shortly introduce two prominent temporal logics, namely Linear Temporal Logic (LTL) and Probabilistic Computation Tree Logic (PCTL).

2.3.1. Linear-time properties and Linear Temporal Logic (LTL)

In this thesis we will be dealing with systems where each state is labeled with a subset of a set of atomic propositions AP. Thus each infinite path of such a system produces a trace which is an infinite word over the alphabet 2^{AP}. A *linear-time property* (LT property) is just a selection of such possible traces, that is an LT property is a language of infinite words over 2^{AP}.

Definition 2.3.1. [Linear-time property]
A linear-time property over a given set AP of atomic propositions is a subset of $(2^{\mathsf{AP}})^{\omega}$.

A path π of a system is said to satisfy a given LT property E, if and only if $\mathtt{trace}(\pi) \in \mathsf{E}$.

Note that given a set of atomic propositions AP and a nondeterministic ω-automaton $\mathcal{A}$ over the alphabet 2^{AP}, then $\mathcal{L}(\mathcal{A})$ is an (ω-regular) LT property.

Another important formalism to specify linear-time properties is the logic LTL [Pnu77] whose syntax is given in the following definition.

Definition 2.3.2. [Syntax of LTL]
Given a set AP of atomic propositions, LTL formulae over the set AP are formed according to the following abstract grammar.

$$\varphi ::= \mathsf{true} \quad \Big| \quad a \quad \Big| \quad \neg\varphi \quad \Big| \quad \varphi_1 \wedge \varphi_2 \quad \Big| \quad \mathcal{X}\varphi \quad \Big| \quad \varphi_1 \mathcal{U} \varphi_2,$$

where $a \in \mathsf{AP}$. LTL denotes the set of LTL formulae over a given set AP.

Definition 2.3.3. [Semantics of LTL (interpretation over infinite words)]
Let φ be an LTL formula over the set of atomic propositions AP. We define the language of φ

$$\mathcal{L}(\varphi) = \{\omega \in (2^{\mathsf{AP}})^{\omega} \quad | \quad \omega \models \varphi\},$$

where $\models \subseteq (2^{\mathsf{AP}})^{\omega} \times \mathsf{LTL}$ is the smallest relation satisfying the properties in Figure 2.1.

Thus, an LTL formula over AP defines an LT property over AP.

$$\begin{array}{cccll}
\omega & \models & \mathsf{true} & & \\
\omega & \models & a & \text{iff} & a \in \omega_1 \\
\omega & \models & \neg\varphi & \text{iff} & \omega \not\models \varphi \\
\omega & \models & \varphi_1 \wedge \varphi_2 & \text{iff} & \omega \models \varphi_1 \wedge \omega \models \varphi_2 \\
\omega & \models & \mathcal{X}\varphi & \text{iff} & \omega\uparrow_2 \models \varphi \\
\omega & \models & \varphi_1 \mathcal{U} \varphi_2 & \text{iff} & \exists\, j \in \mathbb{N}_{\geq 1}.\big[\omega\uparrow_j \models \varphi_2 \ \wedge \ \omega\uparrow_i \models \varphi_1, 1 \leq i < j\big]
\end{array}$$

Figure 2.1: LTL semantics for infinite words over 2^{AP}

Other boolean connectives such as disjunction $\vee$, implication $\rightarrow$, etc. can be derived as usual, e.g. $\varphi_1 \vee \varphi_2 := \neg(\neg\varphi_1 \wedge \neg\varphi_2)$. We will use the following common notations to denote the path property

$$\begin{array}{lrcl} \text{“eventually } \varphi \text{”}: & \Diamond\varphi & := & \mathsf{true}\,\mathcal{U}\varphi \\ \text{“always } \varphi \text{”}: & \Box\varphi & := & \neg(\mathsf{true}\,\mathcal{U}\neg\varphi) \end{array}$$

We then denote the property “infinitely often φ” by $\Box\Diamond\varphi$ and the property “eventually always φ” by $\Diamond\Box\varphi$.

LTL formulae can be used to express Büchi/Rabin and Streett acceptance for ω-automata. Given a Büchi acceptance condition F, let $\mathsf{AP} = \{\mathsf{final}\}$ and exactly the accepting states $s \in F$ are labeled with final. A run π of the automaton is accepting, if and only if $\mathsf{trace}(\pi) \models \Box\Diamond\mathsf{final}$. A Rabin acceptance condition $\mathsf{Acc} = \{(H_1, K_1), \ldots, (H_n, K_n)\}$ can be expressed by

$$\bigvee_{1 \leq i \leq n} \big[(\Diamond\Box\neg H_i \,\wedge\, \Box\Diamond K_i)\big],$$

whereas a Streett acceptance can be expressed by

$$\bigwedge_{1 \leq i \leq n} \big[(\Box\Diamond K_i \,\rightarrow\, \Box\Diamond H_i)\big].$$

Here (and in the rest of the thesis) we sometimes identify state sets with a labeling, e.g. exactly the states in H_i are labeled with H_i.

2.3.2. Probabilistic Computation Tree Logic (PCTL)

For specifying branching behavior, we will use the logic PCTL (Probabilistic Computation Tree Logic) [Han94, HJ94, BdA95]. PCTL is a probabilistic modification of Computation Tree Logic (CTL) [CE81], where the “for all path” and “there exists a path” quantifier have been substituted by a probabilistic operator that requires a given probability on a certain set of paths. While in CTL the formula $\forall\varphi$ indicates that the path formula φ must hold for all paths, a PCTL formula $[\varphi]_{>0.98}$ requires the set of paths that satisfy φ to have a probability measure of more than 0.98. The syntax of PCTL is given in the following definition.

Definition 2.3.4. [Syntax of PCTL]
Given a set AP of atomic propositions, PCTL formulae over the set AP are formed according to the following abstract grammar, where Φ denotes a PCTL *state formula* and φ denotes a PCTL *path formula*.

$$\begin{array}{lcl} \Phi & ::= & \mathsf{true} \;\big|\; a \;\big|\; \neg\Phi \;\big|\; \Phi_1 \wedge \Phi_2 \;\big|\; [\varphi]_{\bowtie p} \\ \varphi & ::= & \mathcal{X}\Phi \;\big|\; \Phi_1\mathcal{U}\Phi_2, \end{array}$$

where $a \in \mathsf{AP}$, $\bowtie \in \{<, \leq, >, \geq\}$ is a comparison operator and $p \in [0, 1]$ is a probability bound.

Definition 2.3.5. [Semantics of PCTL]
Given a set AP of atomic propositions and a state-labeled MDP $\mathcal{M}$ where $\mathsf{Labels} = 2^{\mathsf{AP}}$, the semantics of PCTL with respect to a given scheduler class Sched is defined as the smallest relation $\models$ satisfying the properties in Figure 2.2.

$$
\begin{array}{llll}
s & \models & \mathsf{true} & \\
s & \models & a & \text{iff} \quad a \in L(s) \\
s & \models & \neg\Phi & \text{iff} \quad s \not\models \Phi \\
s & \models & \Phi_1 \wedge \Phi_2 & \text{iff} \quad s \models \Phi_1 \wedge s \models \Phi_2 \\
s & \models & [\varphi]_{\bowtie p} & \text{iff} \quad \mathsf{Pr}_s^{\mathcal{M},\mathcal{U}}(\{\pi \in \mathsf{Path}_{\mathsf{inf}}^{\mathcal{M}} \mid \pi \models \varphi\}) \bowtie p \quad \forall \mathcal{U} \in \mathsf{Sched} \\
\pi & \models & \mathcal{X}\Phi & \text{iff} \quad \pi_1 \models \Phi \\
\pi & \models & \Phi_1 \mathcal{U} \Phi_2 & \text{iff} \quad \exists j \in \mathbb{N}_{\geq 0}.\big[\pi_j \models \Phi_2 \ \wedge \ \pi_i \models \Phi_1, 0 \leq i < j\big]
\end{array}
$$

Figure 2.2: PCTL semantics with respect to a given MDP $\mathcal{M}$ and a scheduler class Sched

Note that the set $\{\pi \in \mathsf{Path}_{\mathsf{inf}}^{\mathcal{M}} \mid \pi \models \varphi\}$ in the 5th item of Figure 2.2 is measurable [Var85, CY95].

3 | Probabilistic Partial Order Reduction

In this chapter we will investigate probabilistic partial order reduction. Partial order reduction is a prominent reduction technique that has been thouroughly examined in the setting of non-probabilistic systems [Pel93, HP94, Val94, God96, GPS96, PPH97, Pel97]. Usually the starting point is a description of an asynchronous parallel system by a representation of the subsystems that run in parallel, e.g. as in the (non-probabilistic) model checker SPIN [Hol03] which uses a guarded command language as input language. The rough idea behind partial order reduction is to construct a reduced state graph by abolishing redundancies in the transition system that origin from the interleaving of independent activities that are executed in parallel. For independent actions α and β, the interleaving semantics represents their parallel execution by the nondeterministic choice between the action sequences $\alpha\beta$ and $\beta\alpha$. As $\alpha\beta$ and $\beta\alpha$ have the same effect to the control and program variables, and thus lead to the same state, the investigation of one order ($\alpha\beta$ or $\beta\alpha$) as a representative for both suffices under certain side conditions. More general, instead of constructing the full system $\mathcal{T}$, the goal is to generate an "equivalent" sub-system $\mathcal{T}_{\mathsf{red}}$ of the full transition system $\mathcal{T}$. Here "equivalence" is considered with respect to the type of property to be verified. Thus, for a path property, we would require that any path π in $\mathcal{T}$ is "represented" in $\mathcal{T}_{\mathsf{red}}$ by an "equivalent" path $\hat{\pi}$. Of course, the algorithmic construction and analysis of $\mathcal{T}_{\mathsf{red}}$ should be more efficient than model checking the full system $\mathcal{T}$.

We give a small example to illustrate these ideas. Consider two processes $\mathcal{P}_1$ and $\mathcal{P}_2$ where $\mathcal{P}_1$ increments a variable x (action α) twice and $\mathcal{P}_2$ increments a variable y (action β) twice. Assume that we are only interested in the value of the variable y, that is each state is labeled with its y value. Then action α does not change the labeling, but action β does. We then get the following picture of the parallel execution of the two processes, where the shade of a state node represents its y value (the greater y is, the darker the node is). Now assume that

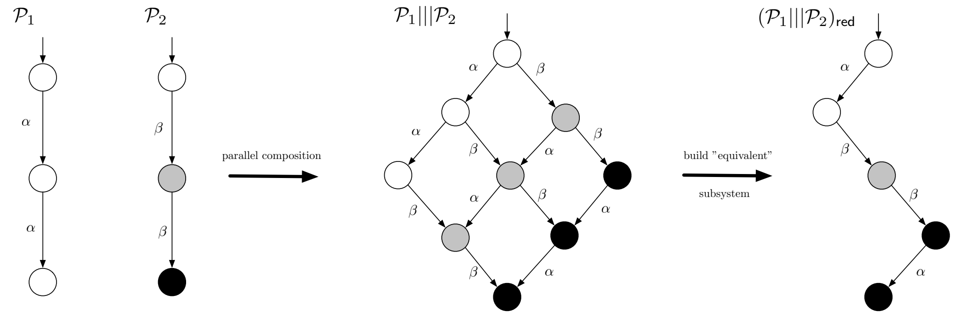


Figure 3.1: The idea of partial order reduction

we want to check whether the property

"The value of y never decreases."

holds on any path. For the system $\mathcal{P}_1|||\mathcal{P}_2$ of the parallel execution of $\mathcal{P}_1$ and $\mathcal{P}_2$ in Figure 3.1 this means

"The shades of the nodes never get lighter."

along any path.

Obviously each path of the system satisfies this property. Now this property has a remarkable feature. In order to decide whether a path satisfies the property or not, it is only relevant what changes of the labeling occur along the path, but not how often a certain labeling is repeated before it changes. The property is so-called *stutter invariant*. It cannot distinguish between two paths that follow the same pattern of changes in the labeling (but may differ in the number of repetitions of a certain labeling). Such two paths are called *stutter equivalent*. Now consider the reduced system $(\mathcal{P}_1|||\mathcal{P}_2)_{\mathsf{red}}$ in Figure 3.1. As any path of $\mathcal{P}_1|||\mathcal{P}_2$ has a stutter equivalent path in $(\mathcal{P}_1|||\mathcal{P}_2)_{\mathsf{red}}$ and the property under consideration cannot distinguish between such paths, it is sufficient to check whether all paths of the reduced system satisfy the property. If all paths of the reduced system satisfy the property, so do all paths of the original system (and vice versa as the reduced system is a subset of the original system). Thus the reduced system is "equivalent" to the original system with respect to the property.

The goal of partial order reduction is to give criteria with which an "equivalent" reduced system can be generated. These criteria heavily depend on the class of property that one wants to preserve (e.g. linear-time properties, branching time properties). In the early 1990s several partial order reduction techniques have been developed for non-probabilistic systems [Val92, Pel93, Val94, HP94, God96, GPS96, Pel97, Val97, PPH97].

In this chapter we concentrate on one instance of partial order reduction techniques, the so-called *ample set method* which was developed by Doron Peled (see e.g. [Pel93, Pel97]) and generalize this method to the probabilistic setting.

The chapter is structured as follows. First we will introduce the necessary preliminaries and then explain the ample set method in the setting of non-probabilistic systems with respect to linear-time properties. This provides to the reader who is not familiar with partial order reduction a good insight into its functioning. We will then examine how the ample set method can be generalized to MDPs with respect to various kinds of properties.

Throughout the whole chapter we will deal with state-labeled systems.

3.1. Preliminaries for the ample set method

We start with the most basic definition, namely that of the reduced system. Given a nondeterministic transition system $\mathcal{M}$, either non-probabilistic (Kripke-structure) or probabilistic (MDP), the rough idea of the ample set method is to assign to any reachable state s of $\mathcal{M}$ an action-set $\mathsf{ample}(s) \subseteq \mathsf{Act}(s)$ and to construct a reduced system $\hat{\mathcal{M}}$ that results by using the action-sets $\mathsf{ample}(s)$ instead of $\mathsf{Act}(s)$. That is, starting from the initial states of $\mathcal{M}$, one builds up $\hat{\mathcal{M}}$ by only applying ample transitions. The reduced system should be equivalent to the original system in the desired sense

$$\hat{\mathcal{M}} \;\equiv\; \mathcal{M},$$

e.g. simulation equivalent or bisimulation equivalent, etc. Depending on the desired equivalence the defined ample-sets have to fulfill certain conditions to ensure the equivalence.

Definition 3.1.1. [Reduced system]
Given an MDP $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$ and given a function $\mathsf{ample} : S \to 2^{\mathsf{Act}}$ with $\emptyset \neq \mathsf{ample}(s) \subseteq \mathsf{Act}(s)$ for all states s, the state space of the reduced MDP

$$\hat{\mathcal{M}} = (\hat{S}, \mathsf{Act}, \hat{\delta}, \hat{\mu})$$

induced by ample is the smallest set $\hat{S} \subseteq S$ that contains the states s, such that $\mu(s) > 0$ and any state t where $\delta(s, \alpha, t) > 0$ for some $s \in \hat{S}$ and $\alpha \in \mathsf{ample}(s)$. The transition probability function of $\hat{\mathcal{M}}$ is given by

$$\hat{\delta}(s, \alpha, t) = \delta(s, \alpha, t)$$

for $\alpha \in \mathsf{ample}(s)$ and 0 otherwise. The initial distribution of $\hat{\mathcal{M}}$ is that of $\mathcal{M}$, i.e. $\hat{\mu}(s) = \mu(s)$, for $s \in \hat{S}$.

We call a state *s fully expanded* if $\mathsf{ample}(s) = \mathsf{Act}(s)$.

As already mentioned in the introduction of this chapter, partial order reduction tries to give criteria how to generate an "equivalent" subsystem of the given system, by abolishing redundancies that origin in the interleaving of independent activities that are executed in parallel. These equivalences typically identify those paths whose traces (i.e. words obtained from the paths by projection on the state labels) agree up to stuttering. In this context stuttering refers to the repetition of the same state-labels.

Definition 3.1.2. [Stutter equivalence for words]
Two infinite words ω_1 and ω_2 over the alphabet Σ are called *stutter equivalent*,

$$\omega_1 \equiv_{st} \omega_2$$

if and only if there is an infinite word $a_1, a_2, \ldots$ over the alphabet Σ such that

$$\omega_1 = a_1^{k_1}, a_2^{k_2}, \ldots \quad \text{and} \quad \omega_2 = a_1^{n_1}, a_2^{n_2}, \ldots,$$

where $k_i, n_i \in \mathbb{N}_{\geq 1}$. Two infinite paths π_1 and π_2 in a state-labeled MDP are called stutter equivalent, denoted $\pi_1 \equiv_{st} \pi_2$, if and only if their traces $\mathsf{trace}(\pi_1)$ and $\mathsf{trace}(\pi_2)$ over 2^{AP} are stutter equivalent.

We call a linear-time property over AP stutter invariant, if it cannot distinguish between stutter equivalent paths.

Definition 3.1.3. [Stutter invariant LT properties]
A linear-time property E over AP is called stutter invariant if it holds for all stutter equivalent words $\omega_1, \omega_2 \in (2^{\mathsf{AP}})^\omega$ that

$$\omega_1 \in \mathsf{E} \quad \text{if and only if} \quad \omega_2 \in \mathsf{E}.$$

For the partial order reduction we shall moreover need the concept of *stutter actions*, i.e. actions that have no effect on the state-labels, no matter in which state they are taken.

Definition 3.1.4. [Stutter action]
Given a state-labeled MDP $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$, we call action $\alpha \in \mathsf{Act}$ a stutter action if and only if for all states $s, t \in S$ it holds that:

$$\delta(s, \alpha, t) > 0 \text{ implies } L(s) = L(t).$$

We refer to $s \xrightarrow{\beta} t$ as a non-probabilistic stutter step if $\beta \in \mathsf{Act}(s)$ is a non-probabilistic stutter action and t is the unique β-successor of s.

The main ingredient of any partial order reduction technique in the probabilistic or non-probabilistic setting is an adequate notion for the independence of actions. The rough idea is a formalization of actions belonging to different processes that are executed in parallel and do not affect each other, e.g. as they only refer to local variables and do not require any kind of synchronization. In non-probabilistic systems independence of two actions α and β means that for any state s where both α and β are enabled the execution of α does not affect the enabledness of β (i.e. the α-successor of s has an outgoing β-transition), and vice versa, and in addition the action sequences $\alpha\beta$ and $\beta\alpha$ lead to the same state. In the probabilistic setting the additional requirement that $\alpha\beta$ and $\beta\alpha$ have the same probabilistic effect is made.

Definition 3.1.5. [Independence of actions, cf. [BGC04, DN04]]
Two actions α, β with $\alpha \neq \beta$ are called independent in an MDP $\mathcal{M}$ if and only if for all states $s \in S$ with $\{\alpha, \beta\} \subseteq \mathsf{Act}(s)$:

(1) $\delta(s, \alpha, t) > 0$ implies $\beta \in \mathsf{Act}(t)$,

(2) $\delta(s, \beta, u) > 0$ implies $\alpha \in \mathsf{Act}(u)$

(3) for all states $v \in S$:

$$\sum_{t \in S} \delta(s, \alpha, t) \cdot \delta(t, \beta, v) \;=\; \sum_{u \in S} \delta(s, \beta, u) \cdot \delta(u, \alpha, v)$$

Two different actions α and β are called dependent if and only if α and β are not independent. If $A \subseteq \mathsf{Act}$ and $\alpha \in \mathsf{Act} \setminus A$ then α is called independent from A if and only if for all actions $\beta \in A$, α and β are independent. Otherwise α is called dependent on A.

The formal definition for the independence of actions α and β in a composed transition system (which captures the semantics of the parallel composition of some processes that run in parallel) relies on recovering the interleaving diamonds.

Applying the above definition to non-probabilistic actions α and β (i.e. where $\delta(s, \alpha, t)$, $\delta(s, \beta, t) \in \{0, 1\}$ for all states s, t) yields the standard definition of independence of actions in ordinary transition systems.

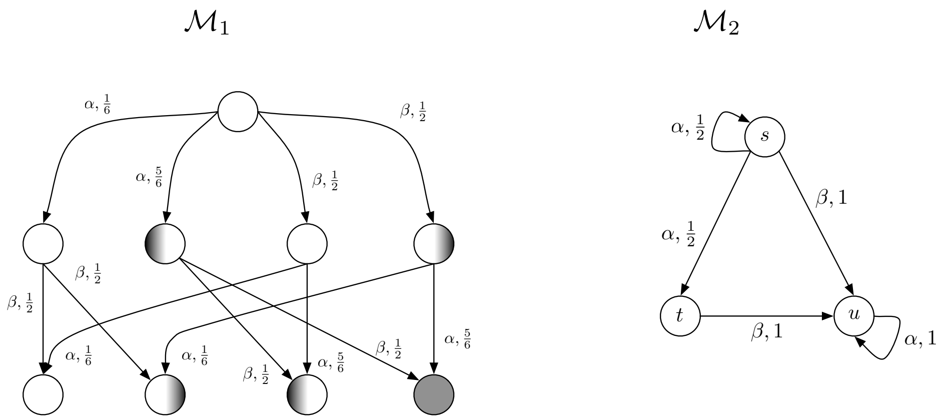


Figure 3.2: Examples of independent actions

Example 3.1.6 (Independent actions)**.** Figure 3.2 shows a fragment of an MDP $\mathcal{M}_1$ representing the parallel execution of independent actions α and β. For example, α might stand for the outcome of the experiment of tossing a "one" with a dice, while β stands for tossing a fair coin. In general, whenever α and β represent stochastic experiments that are independent in the classical sense then α and β viewed as actions of an MDP are independent. However, there are also other situations where two actions can be independent that do not have a fixed probabilistic branching pattern. For instance, actions α and β in the MDP $\mathcal{M}_2$ in Figure 3.2 are independent. To see this, first notice that only in state s both α and β are enabled. The α-successors t, s of s have a β-transition to state u, while the β-successor u has an α-transition to itself. The probabilistic effect under the action sequences $\alpha\beta$ and $\beta\alpha$ is the same as in either case state u is reached with probability 1.

3.2. The ample set method for Kripke structures and linear-time properties

In this section we will deal with non-probabilistic transition systems (Kripke structures), which will be viewed as an MDP with $\delta(s, \alpha, t) \in \{0, 1\}$ for all states s, t and actions α. Moreover, the initial distribution is assumed to be a Dirac distribution, that is, there is a unique initial state.

We will now briefly recall the conditions on the ample sets that Doron Peled proposed to ensure stutter equivalence for a given non-probabilistic system $\mathcal{T}$ and its reduced system $\hat{\mathcal{T}}$ [Pel97]. Later, we will discuss to which extent those conditions are adequate for MDPs. What does stutter equivalence between non-probabilistic transition systems mean? It means, that given a path from one of the systems, the other system must be able to produce a stutter equivalent path.

Definition 3.2.1. [Stutter equivalence for Kripke structures]
Given two non-probabilistic systems $\mathcal{T}_i = (S_i, \mathsf{Act}_i, \delta_i, \mu_i),\ i = 1, 2$ we call $\mathcal{T}_1$ and $\mathcal{T}_2$

stutter equivalent,

$$\mathcal{T}_1 \ \equiv_{st} \ \mathcal{T}_2,$$

if and only if for each initial path π_1 of $\mathcal{T}_1$ there exists an initial path π_2 of $\mathcal{T}_2$ such that $\mathsf{trace}(\pi_2) \equiv_{st} \mathsf{trace}(\pi_1)$ and vice versa.

Before getting to the ample set method we shortly illustrate its impact on LTL model checking. Consider a stutter invariant linear-time property E and two stutter equivalent non-probabilistic systems $\mathcal{T}_1$ and $\mathcal{T}_2$. Then

$$\mathcal{T}_1 \models \mathsf{E} \quad \text{if and only if} \quad \mathcal{T}_2 \models \mathsf{E},$$

where $\mathcal{T} \models \mathsf{E}$ means that for all initial paths π of $\mathcal{T}$, $\mathsf{trace}(\pi) \in \mathsf{E}$. In fact, assume $\mathcal{T}_1 \models \mathsf{E}$ and let π_2 be an arbitrary initial path of $\mathcal{T}_2$. As the two systems are stutter equivalent, there is an initial path p_1 in $\mathcal{T}_1$ that is stutter equivalent to π_2, i.e. $\pi_1 \equiv_{st} \pi_2$. As $\mathcal{T}_1 \models \mathsf{E}$, $\pi_1 \in \mathsf{E}$ and therefore $\pi_2 \in \mathsf{E}$ as E is a stutter invariant property. This shows the claim as the roles of $\mathcal{T}_1$ and $\mathcal{T}_2$ can be swapped.

A particular type of stutter invariant linear-time properties are specifications which are described by a formula of the Next Step free fragment of Linear Temporal Logic ($\mathrm{LTL}_{\backslash \mathcal{X}}$) (see [Lam83]). A formula of that fragment is an LTL formula that does not use the Next step operator $\mathcal{X}(.)$ Thus given an $\mathrm{LTL}_{\backslash \mathcal{X}}$ formula φ and two stutter equivalent systems $\mathcal{T}_1$ and $\mathcal{T}_2$ it holds that

$$\mathcal{T}_1 \models \varphi \quad \text{if and only if} \quad \mathcal{T}_2 \models \varphi,$$

where $\mathcal{T} \models \varphi$ means $\mathcal{T} \models \mathcal{L}(\varphi)$.

We will now explain criteria on the ample sets that guarantee stutter equivalence between a given non-probabilistic system $\mathcal{T}$ and its reduced system $\hat{\mathcal{T}}$ (and therefore preserve stutter invariant linear-time properties). Basically the conditions shown in Figure 3.3 have been proposed in [Pel97] where the following result has been shown.

(A0) **(Nonemptiness-condition)** For all states $s \in S$, $\emptyset \neq \mathsf{ample}(s) \subseteq \mathsf{Act}(s)$.

(A1) **(Stutter-condition)** If $s \in \hat{S}$ and $\mathsf{ample}(s) \neq \mathsf{Act}(s)$ then all actions $\alpha \in \mathsf{ample}(s)$ are stutter actions.

(A2) **(Dependence-condition)** For each path $\pi = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \xrightarrow{\gamma} \ldots$ in $\mathcal{T}$ where $s \in \hat{S}$ and γ is dependent on $\mathsf{ample}(s)$ there exists an index $i \in \{1, \ldots, n\}$ such that $\alpha_i \in \mathsf{ample}(s)$.

(A3) **(Cycle-condition)** For each cycle $s_0 \to s_1 \to \ldots \to s_n = s_0$ in $\hat{\mathcal{T}}$ it holds that: $\alpha \in \bigcap_{i=0}^{n-1} \mathsf{Act}(s_i)$ implies $\alpha \in \bigcup_{i=0}^{n-1} \mathsf{ample}(s_i)$.

Figure 3.3: Conditions for the ample-sets

Theorem 3.2.2 (Ample set method - linear-time properties, cf. [Pel97]).
Given a state-labeled non-probabilistic system $\mathcal{T} = (S, \mathsf{Act}, \delta, \mu)$ as well as a function $\mathsf{ample} : S \to 2^{\mathsf{Act}}$ that satisfies the conditions (A0)-(A3) in Figure 3.3, it holds that

$$\mathcal{T} \ \equiv_{st} \ \hat{\mathcal{T}}.$$

Proof. We will give a rough sketch of the proof.

> **(A0)** Condition (A0) ensures that the reduced system is a sub-MDP of the original one and has no terminal states (as the original one).

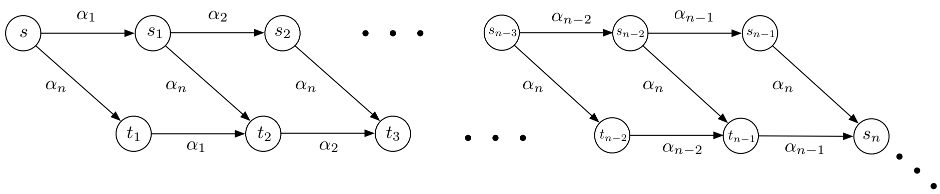
Thus each path of $\hat{\mathcal{T}}$ is also a path of $\mathcal{T}$. It remains to show that for any path π of $\mathcal{T}$ there is a stutter equivalent path $\hat{\pi}$ of $\hat{\mathcal{T}}$. Let

$$\pi = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} s_2 \xrightarrow{\alpha_3} \dots$$

be a path of $\mathcal{T}$. We show how to construct a stutter equivalent path π_1 starting in s such that the first action of π_1 is an ample action of s. If $\alpha_1 \in \mathsf{ample}(s)$, let π_1 be π. If $\alpha_1 \notin \mathsf{ample}(s)$, let n be the smallest number such that $\alpha_n \in \mathsf{ample}(s)$. If none of the α_i is an ample-action of s, let n be ∞. Assume that n is finite.

> **(A1)** As s is not fully expanded and $\alpha_n \in \mathsf{ample}(s)$, (A1) ensures that α_n is a stutter action.
>
> **(A2)** As α_n is the first ample-action of s that occurs along π, (A2) ensures that α_n is independent from $\{\alpha_1, \dots, \alpha_{n-1}\}$.

That leads to the following picture.



Note that $L(s_i) = L(t_{i+1}), i = 0, \dots, n-1$ as α_n is a stutter action (where $s_0 = s$). Hence we may replace the original action sequence $\alpha_1 \dots \alpha_{n-1}\alpha_n$ by the action sequence $\alpha_n\alpha_1 \dots \alpha_{n-1}$ to obtain a path

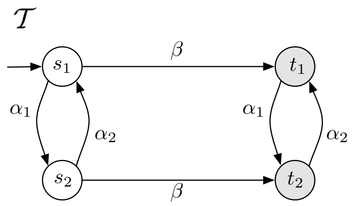
$$\pi_1 = s \xrightarrow{\alpha_n} t_1 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_{n-2}} t_{n-1} \xrightarrow{\alpha_n - 1} s_n \xrightarrow{\alpha_{n+1}} s_{n+1} \xrightarrow{\alpha_{n+2}} \dots$$

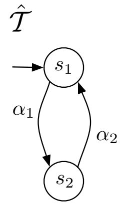
which is stutter equivalent to π.

If $n = \infty$, by similar arguments one can replace π by a stutter equivalent path π_1 with the same starting state $\mathsf{first}(\pi)$ and the action sequence $\beta\alpha_1\alpha_2 \dots$ where β is an arbitrary action in $\mathsf{ample}(\mathsf{first}(\pi))$.

In either case we obtain a path π_1 that starts with a transition in the reduced system $\hat{\mathcal{T}}$. We now may apply the same technique to the path π_1 (more precisely, to the suffix of π_1 that starts in the second state) to obtain a stutter equivalent path π_2 whose first two transitions are transitions in $\hat{\mathcal{T}}$. We continue in this way until the path π of $\mathcal{T}$ is "transformed" into a path $\hat{\pi}$ in $\hat{\mathcal{T}}$. Although conditions (A0)-(A2) are sufficient to guarantee the stutter equivalence of π and the paths $\pi_1, \pi_2, \dots$, the cycle condition **(A3)** is needed to ensure the stutter equivalence of π and $\hat{\pi}$. Without the cycle-condition we might postpone some action of π forever and then some suffix of the path $\hat{\pi}$ consists only of stutter actions. As the cycle-condition requires that for each cycle in $\hat{\mathcal{T}}$, it holds that if α is enabled in $\mathcal{M}$ in each of the states of the cycle, then α is also enabled in $\hat{\mathcal{M}}$ in at least one of the states of the cycle, this ensures that the actions of π are eventually taken in $\hat{\pi}$. □

We finish this section with an example that shows the necessity of the cycle-condition (A3). Consider the following transtion system $\mathcal{T}$ and its reduced system $\hat{\mathcal{T}}$ that emanates from $\mathcal{T}$ when we consider the ample sets as follows. $\mathsf{ample}(s_i) = \{\alpha_i\}$ and all other states (the grey states) are fully expanded. This satisfies conditions (A0)-(A2), but not the cycle condition.





The state labeling is given by the shades of the states, so α_1 and α_2 are stutter actions. Then $\mathcal{T}$ and $\hat{\mathcal{T}}$ are obviously not stutter equivalent, as a grey state can be reached in $\mathcal{T}$, but not in $\hat{\mathcal{T}}$. The problem is as described above. Consider the path π in $\mathcal{T}$ that follows the action sequence $\beta(\alpha_1\alpha_2)^\omega$. Then π_1 follows $\alpha_1\beta(\alpha_2\alpha_1)^\omega$. In general π_i follows $(\alpha_1\alpha_2)^{\frac{i}{2}}\beta(\alpha_1\alpha_2)^\omega$ if i is even and $(\alpha_1\alpha_2)^{\frac{i-1}{2}}\alpha_1\beta(\alpha_2\alpha_1)^\omega$ if i is odd. So the π_i's are stutter equivalent to π but the limit path $\hat{\pi}$ postpones action β forever and is therefore not stutter equivalent to π.

3.3. The ample set method for MDPs and linear-time properties

In this section we show how the ample set method can be extended to MDPs, that means how the ample set conditions (A0)-(A3) can be extended such that a given MDP $\mathcal{M}$ and its reduced MDP $\hat{\mathcal{M}}$ are stutter equivalent. These results have been published in [BGC04]. First we give the definition of stutter equivalence of two Markov decision processes.

Definition 3.3.1. [Stutter equivalence for MDPs]
Given two state-labeled Markov decision processes $\mathcal{M}_i = (S_i, \mathsf{Act}_i, \delta_i, \mu_i),\ i = 1, 2$ we call $\mathcal{M}_1$ and $\mathcal{M}_2$ stutter equivalent,

$$\mathcal{M}_1 \ \equiv_{st} \ \mathcal{M}_2,$$

if and only if for each scheduler $\mathcal{U}_1$ of $\mathcal{M}_1$ there exists a scheduler $\mathcal{U}_2$ of $\mathcal{M}_2$ such that,

$$\mathsf{Pr}^{\mathcal{M}_1,\mathcal{U}_1}(\mathsf{E}) \ = \ \mathsf{Pr}^{\mathcal{M}_2,\mathcal{U}_2}(\mathsf{E})$$

for each stutter invariant measurable linear-time property $\mathsf{E} \subseteq (2^{\mathsf{AP}})^\omega$, and vice versa.

Here, a stutter invariant measurable linear-time property means a language L that is an element of the σ-algebra generated by the empty set and the trace-cylinders

$$\Delta(\ell_1^+, \ldots, \ell_n^+) \ = \ \{\omega \in (2^{\mathsf{AP}})^\omega \mid \exists k_1, \ldots k_n \geq 1 \text{ s.th. } \ell_1^{k_1} \ldots \ell_n^{k_n} \text{ is a prefix of } \omega\},$$

where $\ell_i \neq \ell_{i+1}, i = 1, \ldots, n-1$ and the ℓ_i's are subsets of 2^{AP}.

Again, before treating the ample set method we shortly explain its impact on probabilistic LTL model checking. In probabilistic LTL model checking, a specification consists of an LTL formula, equipped with a probability bound and a comparison operator e.g. $(\Diamond a, \leq \frac{1}{3})$. An MDP $\mathcal{M}$ is said to satisfy a specification $(\varphi, \bowtie p)$, i.e. $\mathcal{M} \models (\varphi, \bowtie p)$, if and only if

$$\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\varphi) \ \bowtie \ p$$

for all schedulers $\mathcal{U}$ of $\mathcal{M}$.

LTL$_{\setminus\mathcal{X}}$-formulae induce stutter invariant measurable languages [Var85]. Thus, if $\mathcal{M}_1$ and $\mathcal{M}_2$ are stutter equivalent, we obtain for every probabilistic LTL$_{\setminus\mathcal{X}}$ specification $(\varphi, \bowtie p)$ that

$$\mathcal{M}_1 \models (\varphi, \bowtie p) \ \text{ if and only if } \ \mathcal{M}_2 \models (\varphi, \bowtie p).$$

Hence two stutter equivalent MDPs $\mathcal{M}_1$ and $\mathcal{M}_2$ are fully equivalent for probabilistic LTL$_{\setminus\mathcal{X}}$ specifications. This even holds for arbitrary specifications consisting of a stutter invariant measurable ω-language equipped with a probability bound and a comparison operator. As each ω-regular language over 2^{AP} is measurable (see e.g. [Var85, CY95]), the above holds for stutter invariant languages that are generated by a nondeterministic ω-automaton.

That means that given a probabilistic LTL$_{\setminus\mathcal{X}}$ specification, it suffices to model check $\hat{\mathcal{M}}$ instead of $\mathcal{M}$ if we can guarantee the stutter equivalence between a given MDP and the reduced system. As $\hat{\mathcal{M}}$ is in general smaller than $\mathcal{M}$ this yields a possible speedup of the analysis. Of course the algorithmic construction of appropriate ample sets together with the construction and the analysis of $\hat{\mathcal{M}}$ should be more efficient than model checking the full system $\mathcal{M}$. As we will see in the following, the partial order reduction criteria for the probabilistic setting are rather strong and might often lead to a minor savings of states. Nevertheless even a reduction that cannot shrink the state space of an MDP but only the number of transitions can increase the efficiency of the probabilistic model checking procedure. The latter relies on solving linear programs where the number of linear inequalities for any state s is given by the number of outgoing transitions from s. Thus removing some transitions via efficient reduction algorithms that e.g. operate on syntactic descriptions of the processes, simplifies the linear program to be solved, and can therefore yield a speed-up of the analysis.

3.3.1. The old conditions are not sufficient

In section 3.2 we introduced conditions (A0)-(A3) on the ample sets of a non-probabilistic system $\mathcal{T}$ that guarantee stutter equivalence between $\mathcal{T}$ and the reduced system $\hat{\mathcal{T}}$. We will now examine why these conditions are not sufficient in the probabilistic case, i.e. we give a counterexample of an MDP $\mathcal{M}$ and ample sets that satisfy (A0)-(A3) such that $\mathcal{M} \not\equiv_{st} \hat{\mathcal{M}}$.

Remember how the ample set method worked in the non-probabilistic case. Given a path π of $\mathcal{T}$ with the underlying action sequence $\alpha_1, \alpha_2, \ldots$ where $\alpha_1 \notin \mathsf{ample}(\mathsf{first}(\pi))$, the main idea is to permute the first ample action of $\mathsf{first}(\pi)$ that occurs along that path to the beginning of the action sequence. We gain a new stutter equivalent path π_1. If no ample action of $\mathsf{first}(\pi)$ occurs along the path, then we prepend some ample action to the action sequence $\alpha_1, \alpha_2, \ldots$ and gain a stutter equivalent path π_1. Conditions (A1) and (A2) guarantee the existence of the path π_1. Now the first action of π_1 is in the reduced system so we repeat this procedure ad infinitum (using $\pi_i\!\uparrow_i$). We thus gain a path $\hat{\pi}$ in $\hat{\mathcal{T}}$ and the cycle condition (A3) ensures that $\pi \equiv_{st} \hat{\pi}$.

But when dealing with MDPs, a problem arises. A scheduler for a given MDP $\mathcal{M}$ might schedule a non-ample action of the starting state. As this action can be probabilistic there might be several successors. For each of those the scheduler is able to schedule different ample actions of the starting state. Which of those should be choosen to be permuted to the front? In fact, a scenario as above must be forbidden. That is why in the probabilistic setting, we will need an additional branching condition to make the ample set method work. The example in Figure 3.4 shows that conditions (A0)-(A3) cannot ensure the stutter equivalence of a given MDP $\mathcal{M}$ and its reduced MDP $\hat{\mathcal{M}}$. Consider the MDP $\mathcal{M}$ in Figure 3.4 (α is

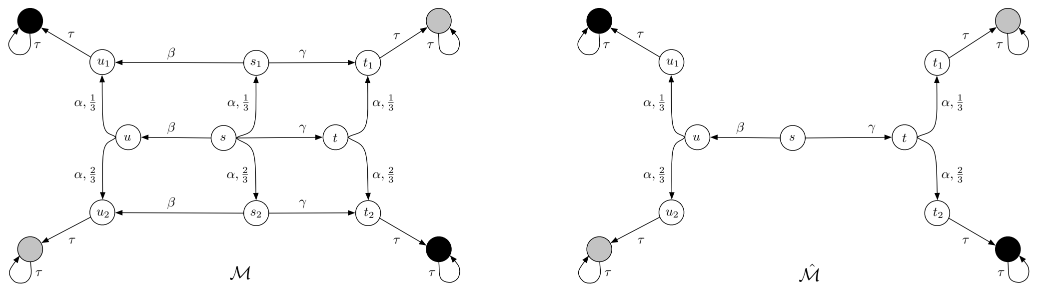


Figure 3.4: Conditions (A0)-(A3) do not establish stutter equivalence for MDPs

the only probabilistic action of $\mathcal{M}$) and the ample sets $\mathsf{ample}(s) = \{\beta, \gamma\}$ and $\mathsf{ample}(s') = \mathsf{Act}(s')$ for all other states s'. It is easy to see that conditions (A0)-(A3) are satisfied (we assume that the labeling is given by ● and ◯). Then $\hat{\mathcal{M}}$ in Figure 3.4 shows the reduced MDP of $\mathcal{M}$ with respect to the chosen ample sets, but

$$\hat{\mathcal{M}} \quad \not\equiv_{st} \quad \mathcal{M},$$

as the maximum probability of eventually reaching ● in $\mathcal{M}$ is 1 (by choosing first α and then β in state s_1 and γ in state s_2). It is instead $\frac{2}{3}$ in $\hat{\mathcal{M}}$.

The problem that arises is the following. The scheduler of $\mathcal{M}$ schedules a probabilistic non-ample action of the starting state s. Depending on the outcome (moving to state s_1 or s_2), the scheduler chooses different ample actions (of s). Thus choosing α first postpones the real nondeterministic decision between the ample actions β and γ. The reduced system $\hat{\mathcal{M}}$ is not able to mimic such a behavior as it has to decide for a particular ample action of s (more precisely a distribution over the ample actions of s) in its first step (before the outcome of α is known). This decision is fixed from then on. It is exactly this behavior that one has to forbid to gain stutter equivalence between the given system $\mathcal{M}$ and its reduced sytem. That means that if the system can branch probabilistically with non-ample actions (with respect to the starting state) then there should be only one ample action of the starting state. Thus we propose an additional branching-condition (A4) that ensures exactly this.

3.3.2. A sufficient extension of the old ample set conditions

(A0)	**(Nonemptiness-condition)** For all states $s \in S$, $\emptyset \neq \mathsf{ample}(s) \subseteq \mathsf{Act}(s)$.
(A1)	**(Stutter-condition)** If $s \in \hat{S}$ and $\mathsf{ample}(s) \neq \mathsf{Act}(s)$ then all actions $\alpha \in \mathsf{ample}(s)$ are stutter actions.
(A2)	**(Dependence-condition)** For each path $\pi = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \xrightarrow{\gamma} \ldots$ in $\mathcal{M}$ where $s \in \hat{S}$ and γ is dependent on $\mathsf{ample}(s)$ there exists an index $i \in \{1, \ldots, n\}$ such that $\alpha_i \in \mathsf{ample}(s)$.
(A3)	**(End component condition)** For each end component (T, A) in $\hat{\mathcal{M}}$ it holds that: $\alpha \in \bigcap_{t \in T} A(t)$ implies $\alpha \in \bigcup_{t \in T} \mathsf{ample}(t)$.
(A4)	**(Branching-condition)** If $\pi = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \xrightarrow{\alpha} \ldots$ is a path in $\mathcal{M}$ where $s \in \hat{S}, \alpha_1, \ldots, \alpha_n, \alpha \notin \mathsf{ample}(s)$ and α is probabilistic, then $\lvert\mathsf{ample}(s)\rvert = 1$.

Figure 3.5: Conditions for the ample-sets of MDPs

For the sake of completeness we stated conditions (A0)-(A2) again and please note that we weakened the cycle-condition to an end component condition (A3).

Remark 3.3.2 (Conservative extension of the ample set method)**.**

- If applied to non-probabilistic systems, the end component condition (A3) is equivalent to the cycle-condition (A3). However the end component condition also allows for certain cycles violating the cycle-condition. For instance, for the MDP $\mathcal{M}_2$ in Figure 3.2, page 23, the end component condition allows to choose $\mathsf{ample}(s) = \{\alpha\}$ (provided that α is a stutter-action), as state s is not contained in an end component. However, this choice of $\mathsf{ample}(s)$ violates the cycle condition as the state s not only has an α self-loop, but also an outgoing β transition.
- One should notice that condition (A4) is irrelevant for non-probabilistic systems.

Thus the extended ample set method falls back to the original one, if applied to non-probabilistic systems.

It remains to show

Theorem 3.3.3 (Ample set method for MDPs - linear-time properties).
Given a state-labeled Markov decision process $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$ as well as a function $\mathsf{ample} : S \to 2^{\mathsf{Act}}$ that satisfies the conditions (A0)-(A4) in Figure 3.5, it holds that

$$\mathcal{M} \ \equiv_{st} \ \hat{\mathcal{M}}.$$

Proof. Let an MDP $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$ and a function $\mathsf{ample} : S \to \mathsf{Act}$ that satisfies the conditions (A0)-(A4) in Figure 3.5 be given. We have to show that for each scheduler $\mathcal{U}$ of $\mathcal{M}$ there exists a scheduler $\hat{\mathcal{U}}$ of $\hat{\mathcal{M}}$ such that,

$$\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\mathsf{E}) \ = \ \mathsf{Pr}^{\hat{\mathcal{M}},\hat{\mathcal{U}}}(\mathsf{E}) \tag{+}$$

for each stutter invariant measurable linear-time property $\mathsf{E} \subseteq (2^{\mathsf{AP}})^{\omega}$, and vice versa.

As $\hat{\mathcal{M}}$ is a sub-MDP of $\mathcal{M}$ it is obvious that for any scheduler $\hat{\mathcal{U}}$ of $\hat{\mathcal{M}}$ there is a scheduler $\mathcal{U}$ of $\mathcal{M}$ such that (+) holds. We just may set $\mathcal{U} = \hat{\mathcal{U}}$.

Now let a scheduler $\mathcal{U}$ of $\mathcal{M}$ be given. We will later show how to construct an infinite sequence of schedulers $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \ldots$ of $\mathcal{M}$, such that condition (+) holds for $\mathcal{U}$ and $\mathcal{U}_i$ (for each state s) and $\mathcal{U}_i(\pi) \in \mathsf{Distr}(\mathsf{ample}(\mathsf{last}(\pi)))$ for all finite $\mathcal{U}_i$ paths π of length $< i$. That is

$$\mathsf{Pr}_s^{\mathcal{M},\mathcal{U}}(\mathsf{E}) = \mathsf{Pr}_s^{\mathcal{M},\mathcal{U}_i}(\mathsf{E}) \tag{++}$$

for each stutter invariant measurable linear-time property E and each state s and for each $\mathcal{U}_i$-path π the i-th prefix of π is a path in $\hat{\mathcal{M}}$. Moreover $\mathcal{U}_{i+1}$ mimicks $\mathcal{U}_i$ on its first $i-1$ steps, i.e.

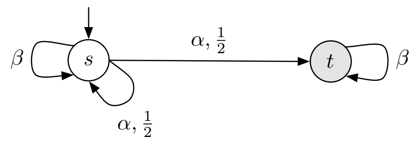
$$\mathcal{U}_{i+1}(\pi) = \mathcal{U}_i(\pi)$$

if $|\pi| \leq i-1$. Finally, a scheduler $\hat{\mathcal{U}}$ for $\hat{\mathcal{M}}$ is derived from the schedulers $\mathcal{U}_i$ as follows. We define

$$\hat{\mathcal{U}}(\hat{\pi}) \ = \ \mathcal{U}_{i+1}(\hat{\pi}),$$

if $\hat{\pi}$ is a finite $\hat{\mathcal{U}}$-path of length i.

The remaining argumentation is similar to the non-probabilistic case. We cannot immediately conclude that $\mathcal{U}$ and $\hat{\mathcal{U}}$ yield the same probabilities for the trace-cylinders because the generated $\hat{\mathcal{U}}$-paths might "delay" a certain action of a $\mathcal{U}$-path ad infinity as in the following example.



The state labeling is given by the shades of the states, thus, β is a stutter action, while α is not. For $\mathsf{ample}(s) = \{\beta\}$ and scheduler $\mathcal{U}$ where $\mathcal{U}(\pi) = \alpha$ for all paths π with $\mathsf{last}(\pi) = s$, the construction explained in subsection 3.3.2.1 yields

$$\mathcal{U}_i(\underbrace{s \xrightarrow{\beta} s \ldots \xrightarrow{\beta} s}_{\text{length } j}) = \begin{cases} \beta & : \text{ for } j \leq i-1, \\ \alpha & : \text{ for } j = i. \end{cases}$$

Thus, scheduler $\hat{\mathcal{U}}$ always schedules β in the state s. In fact, $\hat{\mathcal{U}}$ is the only scheduler for $\hat{\mathcal{M}}$ as $\hat{\mathcal{M}}$ consists only of state s with the β-loop. Under $\mathcal{U}$ and each of the schedulers $\mathcal{U}_i$, we obtain probability 1 to reach the grey state t, while the probability to reach state t under $\hat{\mathcal{U}}$ is 0. However, in this example, conditions (A0), (A1), (A2) and (A4) hold, but the end component $(s, \{\beta\})$ violates the end component condition (A3).

We now show that the end component condition (A3) ensures that any action of a $\mathcal{U}$-path will be "consumed" by $\hat{\mathcal{U}}$ almost surely. To simplify the notations we consider here only the case where $\mathcal{U}(s) = \alpha_1 \notin \mathsf{ample}(s)$ and show that

$$\mathsf{Pr}_s^{\hat{\mathcal{M}},\hat{\mathcal{U}}}(\{\hat{\pi} \in \mathsf{Paths}_{\mathsf{inf}}^{\hat{\mathcal{M}}}(s) \mid \alpha_1 \in \mathsf{Act}(\hat{\pi})\}) = 1.$$

Let us assume that the sum is strictly less than 1. Then,

$$\mathsf{Pr}_s^{\hat{\mathcal{M}},\hat{\mathcal{U}}}\big(\{\hat{\pi} \in \mathsf{Paths}_{\mathsf{inf}}^{\hat{\mathcal{M}}}(s) : \alpha_1 \notin \mathsf{Act}(\hat{\pi})\}\big) > 0 \qquad \text{(+++)}$$

All infinite $\hat{\mathcal{U}}$-paths $\hat{\pi}$ with $\mathsf{first}(\hat{\pi}) = s = s_0$ and $\alpha_1 \notin \mathsf{Act}(\hat{\pi})$ have the form

$$\hat{\pi} \;=\; s \xrightarrow{\beta_1} s_1 \xrightarrow{\beta_2} s_2 \xrightarrow{\beta_3} \ldots$$

where $\alpha_1 \in \mathsf{Act}(s_i) \setminus \mathsf{ample}(s_i)$ for all indices i. This follows from condition (A2) which guarantees that α_1 is independent from the β_i's, and hence, enabled in all states s_i. Moreover, $\mathcal{U}_i$ (and $\hat{\mathcal{U}}$) would have chosen α_1 for $\hat{\pi}\!\uparrow^i$ if $\alpha_1 \in \mathsf{ample}(s_i)$. Because of (+++) and Lemma 2.2.13, page 13 we may choose such a $\hat{\mathcal{U}}$-path $\hat{\pi}$ where $\mathsf{Lim}(\hat{\pi})$ is an end component. Recall that $\mathsf{Lim}(\hat{\pi})$ consists of all states that occur infinitely often in $\hat{\pi}$ and their actions that are chosen infinitely often in $\hat{\pi}$. Thus, it holds that $\alpha_1 \in \mathsf{Act}(t) \setminus \mathsf{ample}(t)$ for all states t of an end component in $\hat{\mathcal{M}}$. But this contradicts the end component condition (A3). This observation together with the fact that $\mathcal{U}$ and the intermediate schedulers $\mathcal{U}_i$ yield the same probabilities for all stutter invariant measurable linear-time properties allows us to conclude that for all $\bar{\ell} = (\ell_1, \ldots, \ell_n) \in (2^{\mathsf{AP}})^*$ and all action sequences $\bar{\alpha} = \alpha_1 \ldots \alpha_n$,

$$\mathsf{Pr}_s^{\mathcal{M},\mathcal{U}}(\Delta(\bar{\ell}^+, \bar{\alpha})) \;=\; \mathsf{Pr}_s^{\hat{\mathcal{M}},\hat{\mathcal{U}}}(\Delta(\bar{\ell}^+, \bar{\alpha})).$$

Here, $\Delta(\bar{\ell}^+, \bar{\alpha})$ denotes the set of all finite paths π of minimal length that induce a trace of the form $\ell_1^+ \ldots \ell_n^+$ and that have an action sequence which results from $\bar{\alpha}$ by exchanging the order of independent actions and possibly adding stutter actions. From this, we derive (+) for all stutter invariant measurable linear-time properties E which concludes the proof. $\square$

3.3.2.1. Constructing the schedulers $\mathcal{U}_i$

We now explain how the desired schedulers $\mathcal{U}_i, i = 1, 2, \ldots$ are constructed. Recall that starting with a scheduler $\mathcal{U}$ of $\mathcal{M}$, we want to construct an infinite sequence of schedulers $\mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \ldots$ of $\mathcal{M}$, such that for each i the constrain (++)

$$\mathsf{Pr}_s^{\mathcal{M},\mathcal{U}}(\mathsf{E}) = \mathsf{Pr}_s^{\mathcal{M},\mathcal{U}_i}(\mathsf{E})$$

holds for each stutter invariant measurable linear-time property E and each state s and for each $\mathcal{U}_i$-path π the i-th prefix of π is a path in $\hat{\mathcal{M}}$. Using standard arguments of measure theory, to prove (++) for all stutter invariant measurable linear-time properties, it suffices to establish condition (++) for all trace-cylinders $\Delta(\ell_1^+, \ldots, \ell_k^+)$.

Assume that an MDP $\mathcal{M}$, a scheduler $\mathcal{U}$ and ample sets satisfying conditions (A0)-(A4) are given. We fix a state $s \in \hat{\mathcal{M}}$ and present the definition of $\mathcal{U}_1(\pi)$ for finite paths π starting in s. But first we fix some notation.

Notation 3.3.4. Recall that if π is a finite or infinite path then $\overrightarrow{\pi_{\mathsf{Act}}}$ denotes the action sequence of π and $\mathsf{Act}(\pi)$ denotes the set of actions occurring in π.

Given a state s we denote by $\mathsf{na(s)}$ the actions of s that are not in the ample set of s, i.e. $\mathsf{na(s)} = \mathsf{Act}(s) \setminus \mathsf{ample}(s)$.

Given a regular language $A \subseteq \mathsf{Act}^*$, a finite path ρ and a scheduler $\mathcal{U}$, we denote by

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{A}) := \mathsf{Pr}_s^{\mathcal{U}}\big(\{\pi \in \mathsf{Path}_{\mathsf{inf}}(s) \mid \pi\!\uparrow^{|\rho|} = \rho \wedge \exists \vec{a} \in A \text{ s.th. } (\overrightarrow{\pi_{\mathsf{Act}}})\!\uparrow^{|\rho|+|\vec{a}|} = \overrightarrow{\rho_{\mathsf{Act}}} \circ \vec{a}\}\big)$$

the probability that $\mathcal{U}$ produces a path that has ρ as a prefix and follows afterwards some action sequence in A. Note that the set of such paths is measurable. Thus for example, $\mathsf{Pr}_s^{\mathcal{U}}(s \xrightarrow{\alpha_1 \mathsf{na(s)}^* \beta})$ denotes the probability that $\mathcal{U}$ produces a path that starts in s, executes α_1, a possibly empty sequence of non-ample actions of s and then β.

By

$$\begin{aligned}\mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{A} t) \;\; := \;\; & \mathsf{Pr}_s^{\mathcal{U}}\big(\{\pi \in \mathsf{Path}_{\mathsf{inf}}(s) \mid \\ & \pi\!\uparrow^{|\rho|} = \rho \,\wedge\, \exists \vec{a} \in A \text{ s.th. } (\overrightarrow{\pi_{\mathsf{Act}}})\!\uparrow^{|\rho|+|\vec{a}|} = \overrightarrow{\rho_{\mathsf{Act}}} \circ \vec{a} \wedge \pi_{|\rho|+|\vec{a}|} = t\}\big)\end{aligned}$$

we denote the probability of the set of such paths that in addition visit the state t after executing the particular action sequence of A.

We denote by

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^\omega}) := \mathsf{Pr}_s^{\mathcal{U}}\big(\{\pi \in \mathsf{Path}_{\mathsf{inf}}(s) \mid \pi\!\uparrow^{|\rho|} = \rho \,\wedge\, \mathsf{Act}(\pi\!\uparrow_{|\rho|}) \cap \mathsf{ample}(s) = \emptyset\}\big)$$

the probability that $\mathcal{U}$ produces a path that has ρ as a prefix and afterwards never executes an action in $\mathsf{ample}(s)$.

Notation 3.3.5. For finite paths starting in s, where $\mathsf{ample}(s) \neq \mathsf{Act}(s)$, we define $\sim$ to be the finest equivalence such that

(1)
$$\pi \sim \sigma \;\text{ and }\; \pi \sim \pi'$$

for all finite paths π, π' and σ of the form

$$\begin{array}{rcllllll} \pi & = & s \xrightarrow{\beta} & t_0 \xrightarrow{\alpha_1} & \ldots & \xrightarrow{\alpha_{n-1}} t_{n-1} & \xrightarrow{\alpha_n} t \\ \pi' & = & s \xrightarrow{\beta} & u_0 \xrightarrow{\alpha_1} & \ldots & \xrightarrow{\alpha_{n-1}} u_{n-1} & \xrightarrow{\alpha_n} t \\ \sigma & = & s \xrightarrow{\alpha_1} & s_1 \xrightarrow{\alpha_2} & \ldots & \xrightarrow{\alpha_n} s_n & \xrightarrow{\beta} t \end{array}$$

where $\alpha_1, \ldots, \alpha_n \notin \mathsf{ample}(s)$ are independent from $\beta \in \mathsf{ample}(s)$ and there exist state-labels $\ell_0, \ldots, \ell_n \subseteq \mathsf{AP}$ such that

$$\mathsf{trace}(\pi) = \mathsf{trace}(\pi') = \ell_0, \ell_0, \ell_1, \ldots, \ell_n$$

and $\mathsf{trace}(\sigma) = \ell_0, \ell_1, \ldots, \ell_n, \ell_n$. Note that β is a stutter action as $\mathsf{ample}(s) \neq \mathsf{Act}(s)$. Moreover

(2)

$$\pi \circ \rho \;\sim\; \pi' \circ \rho \;\sim\; \sigma \circ \rho$$

for all π, π', σ as in (1) and all finite paths ρ starting in t.

$[\pi]$ denotes the $\sim$-equivalence class of π.

We now give the definition of $\mathcal{U}_1$. There are three different possibilities.

Case [1] s is fully expanded, i.e. $\mathsf{ample}(s) = \mathsf{Act}(s)$.

We define $\mathcal{U}_1(\pi_1) := \mathcal{U}(\pi_1)$ for all $\pi_1 \in \mathsf{Path}_{\mathsf{fin}}(s)$. Then (++) obviously holds for each stutter invariant linear-time property E.

Case [2] $|\mathsf{ample}(s)| = 1$

Assume $\mathsf{ample}(s) = \{\beta\}$. Obviously we have to define

$$\mathcal{U}_1(s)(\beta) = 1.$$

Let $\pi \in \mathsf{Path}_{\mathsf{fin}}(s)$ be a path of length greater than zero. We define $\mathcal{U}_1(\pi)$ as shown in Figure 3.6. Some explanations are in order. If π is a $\mathcal{U}_1$-path as in Notation 3.3.5 then the

$\alpha \neq \beta$

$$\mathcal{U}_1(\pi)(\alpha) \;=\; \frac{1}{\sum\limits_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi')} \cdot \Big(\sum_{\sigma \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) \cdot \mathcal{U}(\sigma)(\alpha) + \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \,\in\, [\pi] \\ \mathsf{ample}(s) \,\cap\, \mathsf{Act}(\rho) \,=\, \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho) \cdot \mathcal{U}(\rho)(\alpha) \cdot \delta\big(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi)\big) \Big)$$

$$\mathcal{U}_1(\pi)(\beta) \;=\; \frac{\sum\limits_{\sigma \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) \cdot \mathcal{U}(\sigma)(\beta)}{\sum\limits_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi')}$$

Recall that $s = \mathsf{first}(\pi)$.

Figure 3.6: Definition of the scheduler $\mathcal{U}_1(\pi)$, if $|\pi| \geq 1$ ($\mathsf{ample}(s) = \{\beta\}$)

$\mathcal{U}$-paths $\sigma \in [\pi]$ have the form

$$\sigma = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_i} s_i \xrightarrow{\beta} v_i \xrightarrow{\alpha_{i+1}} \ldots \xrightarrow{\alpha_n} v_n,$$

where $0 \leq i \leq n$, and $\pi\!\uparrow_{i+1} = v_i \xrightarrow{\alpha_{i+1}} \ldots \xrightarrow{\alpha_n} v_n$. Moreover, $L(s_k) = L(\pi_{k+1}), k = 1, \ldots, i$. So these are the paths, where β has already been executed.

The paths ρ in the right sum in Figure 3.6 have the form $\rho = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n$ where $\alpha_1, \ldots, \alpha_n \notin \mathsf{ample}(s)$ and

$$\rho \xrightarrow{\beta} \mathsf{last}(\pi) \;=\; s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \xrightarrow{\beta} \mathsf{last}(\pi) \;\in [\pi].$$

These are the paths where β has not yet been executed. Note that there might be other paths $\tilde{\pi}$ that follow the action sequence $\beta, \alpha_1, \ldots, \alpha_n$ and are stutter equivalent to the path π, but which end in a different state than π, i.e. $\mathsf{last}(\tilde{\pi}) \neq \mathsf{last}(\pi) = t$. Thus $\tilde{\pi} \notin [\pi]$ and $[\tilde{\pi}] \neq [\pi]$. But then $\rho \xrightarrow{\beta} \mathsf{last}(\tilde{\pi})$ is in $[\tilde{\pi}]$. Thus the path ρ accounts for $[\pi]$ as well as $[\tilde{\pi}]$. The factor $\delta(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi))$ brings in just the fraction that corresponds to $[\pi]$.

If the given $\mathcal{U}_1$-path π has the form

$$\pi = s \xrightarrow{\beta} t_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} t_n \xrightarrow{\gamma_1} t_{n+1} \ldots \xrightarrow{\gamma_m} t_{n+m},$$

where $\alpha_1, \ldots, \alpha_n \notin \mathsf{ample}(s)$ are independent from β, $\gamma_1 \in \mathsf{ample}(s)$ or γ_1 and β are dependent, then the sum on the right in Figure 3.6 equals zero as no such path ρ exists.

We first show that $\mathcal{U}_1$ is indeed a HR-scheduler, i.e. we have to prove that

(i) $\mathcal{U}_1(\pi)(\alpha) > 0$ implies $\alpha \in \mathsf{Act}(\mathsf{last}(\pi))$ and

(ii) $\sum\limits_{\gamma \in \mathsf{Act}(s)} \mathcal{U}_1(\pi)(\gamma) = 1.$

Both (i) and (ii) are obvious if π has length zero (i.e. $\pi = s$).

Let π be a path of length greater than zero, i.e. $|\pi| \geq 1$. Then (i) is an easy verification. Indeed, the definition of $\mathcal{U}_1(\pi)(\alpha)$ in Figure 3.6 shows that $\mathcal{U}_1(\pi)(\alpha) > 0$ implies either

- $\mathcal{U}(\sigma)(\alpha) > 0$ for some path $\sigma \in [\pi]$, so $\alpha \in \mathsf{Act}(\mathsf{last}(\sigma)) = \mathsf{Act}(\mathsf{last}(\pi))$ or
- $\mathcal{U}(\rho)(\alpha) > 0$ for some path ρ, such that $\mathsf{Act}(\rho) \cap \mathsf{ample}(s) = \emptyset$, $\alpha \notin \mathsf{ample}(s)$ and $\delta(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi)) > 0$. Thus $\alpha \in \mathsf{Act}(\mathsf{last}(\rho))$, $\beta \in \mathsf{Act}(\mathsf{last}(\rho))$ and condition (A2) ensures that α is independent to β and therefore $\alpha \in \mathsf{Act}(\mathsf{last}(\pi))$.

(ii) is an immediate consequence of the following observation.

Lemma 3.3.6. Let π be a path of length greater than zero. Then

$$\begin{aligned} \sum_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \;&=\; \sum_{\sigma \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) && \textbf{(I)} \\ &+ \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho) \cdot (1 - \mathcal{U}(\rho)(\beta)) \cdot \delta(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi)) \end{aligned}$$

Proof. see subsection 3.3.2.2 □

Case [3] $1 < |\mathsf{ample}(s)| < |\mathsf{Act}(s)|$

Note that in this case condition (A4) ensures that for each path $s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n$ in $\mathcal{M}$ with $\alpha_i \notin \mathsf{ample}(s), 1 \leq i \leq n$ it holds that α_i is non-probabilistic, $1 \leq i \leq n$. Thus as long as no ample action occurs on a path, all actions yield unique successors. Despite this, it is possible that different ample actions of s are scheduled by $\mathcal{U}$, as $\mathcal{U}$ can choose non-ample actions probabilistically and therefore yield an execution tree. But this does not raise a problem, as $\mathcal{U}_1$ can simulate this. The main idea is that $\mathcal{U}_1$ executes in s an ample action β with the probability that β occurs as the first ample action under the scheduler $\mathcal{U}$. It then redistributes the appropriate probability mass to the particular paths. As there might be $\mathcal{U}$-paths that never execute an ample action of s, we fix a designated ample action $\beta_s \in \mathsf{ample}(s)$, that will be prepended to those paths.

We define the scheduler $\mathcal{U}_1$ for the path of length zero.

$$\mathcal{U}_1(s)(\beta) \;=\; \mathsf{Pr}^{\mathcal{U}}_s(s \xrightarrow{\mathsf{na(s)}^*\beta})$$

for $\beta_s \neq \beta \in \mathsf{ample}(s)$ and

$$\mathcal{U}_1(s)(\beta_s) \;=\; \mathsf{Pr}^{\mathcal{U}}_s(s \xrightarrow{\mathsf{na(s)}^*\beta_s}) + \mathsf{Pr}^{\mathcal{U}}_s(s \xrightarrow{\mathsf{na(s)}^\omega}).$$

Let $\pi \in \mathsf{Path}_{\mathsf{fin}}(s)$ be a path of length greater than zero, i.e. $|\pi| \geq 1$. We define $\mathcal{U}_1(\pi)$ as shown in Figure 3.7, where $\beta \in \mathsf{ample}(s)$ is the first action that occurs on π. Some

$\alpha \notin \mathsf{ample}(s)$ and $\beta \neq \beta_s$:

$$\mathcal{U}_1(\pi)(\alpha) \;=\; \frac{1}{\sum\limits_{\pi' \in [\pi]} \mathsf{Pr}^{\mathcal{U}_1}_s(\pi')} \cdot \Bigg(\sum_{\sigma \in [\pi]} \mathsf{Pr}^{\mathcal{U}}_s(\sigma) \cdot \mathcal{U}(\sigma)(\alpha) + \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha \mathsf{na(s)}^* \beta}) \cdot \delta\big(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi)\big) \Bigg)$$

$\gamma \in \mathsf{ample}(s)$:

$$\mathcal{U}_1(\pi)(\gamma) \;=\; \frac{\sum\limits_{\sigma \in [\pi]} \mathsf{Pr}^{\mathcal{U}}_s(\sigma) \cdot \mathcal{U}(\sigma)(\gamma)}{\sum\limits_{\pi' \in [\pi]} \mathsf{Pr}^{\mathcal{U}_1}_s(\pi')}$$

$s = \mathsf{first}(\pi)$ and $\beta \in \mathsf{ample}(s)$ is the first action on π.

Figure 3.7: Definition of the scheduler $\mathcal{U}_1(\pi)$, if $|\pi| \geq 1$ ($1 < |\mathsf{ample}(s)| < |\mathsf{Act}(s)|$)

explanations are in order. Note that the sum in the right summand ranges over at most one path. Indeed, if ρ is a path in the right sum in Figure 3.7, then it has the form $\rho = s \xrightarrow{\alpha_1}$

$s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n$ where $\alpha_1, \ldots, \alpha_n \notin \mathsf{ample}(s)$ and

$$\rho \xrightarrow{\beta} \mathsf{last}(\pi) \;=\; s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \xrightarrow{\beta} \mathsf{last}(\pi) \;\in [\pi].$$

Since the α_i's are not in $\mathsf{ample}(s)$, condition (A4) implies that the α_i's are non-probabilistic, so ρ is uniquely defined. If the given $\mathcal{U}_1$-path π has the form

$$\pi = s \xrightarrow{\beta} t_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_n} t_n \xrightarrow{\gamma_1} t_{n+1} \ldots \xrightarrow{\gamma_m} t_{n+m},$$

where $\alpha_1, \ldots, \alpha_n \notin \mathsf{ample}(s)$ are independent from β, $\gamma_1 \in \mathsf{ample}(s)$ or γ_1 and β are dependent, then the sum on the right in Figure 3.7 equals zero as no such path ρ exists.

The paths $\sigma \in [\pi]$ are the paths where β has already been executed as the first $\mathsf{ample}(s)$ action on σ. The path ρ in the second summand is the path where no ample action of s has yet been executed (the sum ranges over at most one path). Note that there might be other paths $\tilde{\pi}$ that follow the action sequence $\beta, \alpha_1, \ldots, \alpha_n$ and are stutter equivalent to the path π, but which end in a different state than π, i.e. $\mathsf{last}(\tilde{\pi}) \neq \mathsf{last}(\pi) = t$. Thus $\tilde{\pi} \notin [\pi]$ and $[\tilde{\pi}] \neq [\pi]$. But then $\rho \xrightarrow{\beta} \mathsf{last}(\tilde{\pi})$ is in $[\tilde{\pi}]$. Thus the path ρ accounts for $[\pi]$ as well as $[\tilde{\pi}]$. The factor $\delta(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi))$ brings in just the fraction that corresponds to $[\pi]$. The factor

$$\mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha \mathsf{na(s)}^* \beta}) \;=\; \mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha}) \cdot \frac{\mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha \mathsf{na(s)}^* \beta})}{\mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha})}$$

brings in only the fraction of the probability that $\mathcal{U}$ executes ρ and schedules α afterwards, where the first action of $\mathsf{ample}(s)$ that appears, will be β.

If the first action on π is the designated ample action β_s, then we define $\mathcal{U}_1(\pi)(\alpha)$ for $\alpha \notin \mathsf{ample}(s)$ as

$$\begin{aligned}
\mathcal{U}_1(\pi)(\alpha) \;=\; & \frac{1}{\sum\limits_{\pi' \in [\pi]} \mathsf{Pr}^{\mathcal{U}_1}_s(\pi')} \cdot \Big(\sum_{\sigma \in [\pi]} \mathsf{Pr}^{\mathcal{U}}_s(\sigma) \cdot \mathcal{U}(\sigma)(\alpha) \;+ \\
& \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta_s} \mathsf{last}(\pi) \,\in\, [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha \mathsf{na(s)}^* \beta_s}) \cdot \delta\big(\mathsf{last}(\rho), \beta_s, \mathsf{last}(\pi)\big) \;+ \\
& \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta_s} \mathsf{last}(\pi) \,\in\, [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha \mathsf{na(s)}^\omega}) \cdot \delta\big(\mathsf{last}(\rho), \beta_s, \mathsf{last}(\pi)\big) \Big)
\end{aligned}$$

The factor

$$\mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha \mathsf{na(s)}^\omega}) \;=\; \mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha}) \cdot \frac{\mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha \mathsf{na(s)}^\omega})}{\mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\alpha})}$$

in the last summand brings in only the fraction of the probability that $\mathcal{U}$ executes ρ and schedules α afterwards, where no action of $\mathsf{ample}(s)$ will ever occur.

Again we have to show that $\mathcal{U}_1$ is indeed a HR-scheduler, i.e. we have to prove that

(i) $\mathcal{U}_1(\pi)(\alpha) > 0$ implies $\alpha \in \mathsf{Act}(\mathsf{last}(\pi))$ and

(ii) $\sum\limits_{\gamma \in \mathsf{Act}(s)} \mathcal{U}_1(\pi)(\gamma) = 1.$

Both (i) and (ii) are obvious if π has length zero (i.e. $\pi = s$).

Let π be a path of length greater than zero, i.e. $|\pi| \geq 1$. Then (i) is an easy verification as in case [2]. (ii) can be derived from the following observation.

Lemma 3.3.7. Let π be a path such that $s = \mathsf{first}(\pi)$ and $\beta \in \mathsf{ample}(s)$ is the first action on π (and $\beta \neq \beta_s$). Then

$$\begin{aligned} \sum_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \;&=\; \sum_{\sigma \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) \qquad\qquad \textbf{(I')} \\ &+ \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+\beta}) \cdot \delta(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi)) \end{aligned}$$

If $\beta = \beta_s$ then the right hand side of equation (I') contains the additional summand

$$\sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta_s} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^\omega}) \cdot \delta(\mathsf{last}(\rho), \beta_s, \mathsf{last}(\pi))$$

Proof. see subsection 3.3.2.3 $\square$

Note that equation (I) can be written in the same style as equation (I') as

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho) \cdot (1 - \mathcal{U}(\rho)(\beta)) \;=\; \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}}) \;=\; \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+\beta}) + \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^\omega}),$$

if $\mathsf{ample}(s) = \{\beta\}$.

We now show that $\mathcal{U}$ and $\mathcal{U}_1$ yield the same probabilities for all stutter invariant measurable languages.

Notation 3.3.8. If π is a finite $\mathcal{U}_1$-path of length ≥ 1 starting in s (thus, the first action in π is in $\mathsf{ample}(s)$) then $[\pi]^+$ denotes the union of all equivalence classes $[\pi']$ where π' is a finite path with $\overrightarrow{\pi_{\mathsf{Act}}} = \overrightarrow{\pi'_{\mathsf{Act}}}$ and $\mathsf{trace}(\pi) = \mathsf{trace}(\pi')$.

Note that the paths in $[\pi]^+$ are pairwise stutter equivalent. From (I'), resp. (I) we derive

$$\sum_{\pi' \in [\pi]^+} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \;=\; \sum_{\sigma \in [\pi]^+} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) \;+ \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi]^+ \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+\beta}) \qquad \textbf{(II)}$$

Again as for equation (I'), if $\beta = \beta_s$ then the right hand side of equation (II) contains the additional summand

$$\sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta_s} \mathsf{last}(\pi) \in [\pi]^+ \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^\omega})$$

Notation 3.3.9. Given a state s, an action $\beta \neq \beta_s, \beta \in \mathsf{ample}(s)$ and the sequences $\bar{\ell} = \ell_0, \ell_1, \ldots, \ell_n \in (2^{\mathsf{AP}})^*$ and $\bar{\alpha} = \alpha_1, \ldots, \alpha_n \in \mathsf{Act}^*$, we define the path-set $\mathsf{PCyl}^s_\beta(\bar{\ell}, \bar{\alpha})$ consisting of (1) all infinite paths

$$\varsigma = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_{i-1}} s_{i-1} \xrightarrow{\beta} t_{i-1} \xrightarrow{\alpha_i} t_i \xrightarrow{\alpha_{i+1}} \ldots \xrightarrow{\alpha_n} t_n \longrightarrow \ldots$$

where $i \in \{1, \ldots, n+1\}$, $\{\alpha_1, \ldots, \alpha_{i-1}\} \cap \mathsf{ample}(s) = \emptyset$ and $\mathsf{trace}(\varsigma)$ has the prefix $\ell_0, \ell_1, \ldots, \ell_{i-1}, \ell_{i-1}, \ell_i, \ell_{i+1}, \ldots, \ell_n$ and (2) all infinite paths

$$\varsigma = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \xrightarrow{\alpha} \ldots$$

where $\{\alpha_1, \ldots, \alpha_n, \alpha\} \cap \mathsf{ample}(s) = \emptyset$, $\overrightarrow{\varsigma_{\mathsf{Act}}} \in \mathsf{na(s)}^*\beta\mathsf{Act}^\omega$ and $\mathsf{trace}(\varsigma)$ has the prefix $\ell_0, \ell_1, \ldots, \ell_n$.

$\mathsf{PCyl}^s_{\beta_s}(\bar{\ell}, \bar{\alpha})$ additionally contains all infinite paths

$$\varsigma = s \xrightarrow{\alpha_1} s_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} s_n \ldots$$

where no $\mathsf{ample}(s)$ action appears on ς, i.e. $\overrightarrow{\varsigma_{\mathsf{Act}}} \in \mathsf{na(s)}^\omega$ and $\mathsf{trace}(\varsigma)$ has the prefix $\ell_0, \ell_1, \ldots, \ell_n$.

If π is a $\mathcal{U}_1$-path with $\mathsf{first}(\pi) = s$, $\overrightarrow{\pi_{\mathsf{Act}}} = \beta\alpha_1 \ldots \alpha_n$ and $\mathsf{trace}(\pi) = \ell_0, \ell_0, \ell_1, \ldots, \ell_n$ then $\varsigma \in \mathsf{PCyl}^s_\beta(\bar{\ell}, \bar{\alpha})$ if and only if either ς has a prefix in $[\pi]^+$ or $\overrightarrow{\varsigma_{\mathsf{Act}}} = \alpha_1 \ldots \alpha_n\alpha \ldots \in \mathsf{na(s)}^*\beta\mathsf{Act}^\omega$ (or $\in \mathsf{na(s)}^\omega$ if $\beta = \beta_s$), where $\{\alpha_1, \ldots, \alpha_n, \alpha\} \cap \mathsf{ample}(s) = \emptyset$ and $\ell_0, \ell_1, \ldots, \ell_n$ is a prefix of $\mathsf{trace}(\varsigma)$. The latter case is only possible if $\alpha_1, \ldots, \alpha_n$ (and α) are independent from β. This observation yields with $s = \mathsf{first}(\pi)$ and β the first action on π.

$$\mathsf{Pr}^{\mathcal{U}_1}_s(\mathsf{PCyl}^s_\beta(\bar{\ell}, \bar{\alpha})) = \sum_{\pi' \in [\pi]^+} \mathsf{Pr}^{\mathcal{U}_1}_s(\pi') \qquad \textbf{(III)}$$

$$\mathsf{Pr}^{\mathcal{U}}_s(\mathsf{PCyl}^s_\beta(\bar{\ell}, \bar{\alpha})) = \sum_{\sigma \in [\pi]^+} \mathsf{Pr}^{\mathcal{U}}_s(\sigma) + \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi]^+ \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\mathsf{na(s)}^+\beta}) \qquad \textbf{(IV)}$$

If $\beta = \beta_s$ then the right hand side of equation (IV) contains the additional summand

$$\sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta_s} \mathsf{last}(\pi) \in [\pi]^+ \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}^{\mathcal{U}}_s(\rho \xrightarrow{\mathsf{na(s)}^\omega})$$

Combining (II), (III) and (IV) yields:

$$\mathsf{Pr}^{\mathcal{U}_1}_s(\mathsf{PCyl}^s_\beta(\bar{\ell}, \bar{\alpha})) = \mathsf{Pr}^{\mathcal{U}}_s(\mathsf{PCyl}^s_\beta(\bar{\ell}, \bar{\alpha})) \qquad \textbf{(V)}$$

We now abstract away from the action sequences and the ample action β and define $\mathsf{PCyl}^s(\bar{\ell})$ as the union of all path-sets $\mathsf{PCyl}^s_\beta(\bar{\ell}, \bar{\alpha})$. As the path-sets $\mathsf{PCyl}^s_\beta(\bar{\ell}, \bar{\alpha})$ are pairwise disjoint, we conclude from (V) that

$$\begin{aligned}\mathsf{Pr}_s^{\mathcal{U}_1}(\mathsf{PCyl}^s(\bar{\ell})) &= \sum_{\substack{\bar{\alpha} \\ \beta \,\in\, \mathsf{ample}(s)}} \mathsf{Pr}_s^{\mathcal{U}_1}(\mathsf{PCyl}^s_\beta(\bar{\ell},\bar{\alpha})) \\ &= \sum_{\substack{\bar{\alpha} \\ \beta \,\in\, \mathsf{ample}(s)}} \mathsf{Pr}_s^{\mathcal{U}}(\mathsf{PCyl}^s_\beta(\bar{\ell},\bar{\alpha})) \quad = \quad \mathsf{Pr}_s^{\mathcal{U}}(\mathsf{PCyl}^s(\bar{\ell})). \qquad \textbf{(VI)}\end{aligned}$$

If we are given a trace-cylinder $\Delta(\ell_1^+,\ldots,\ell_n^+) \subseteq (2^{\mathsf{AP}})^\omega$ (where $\ell_i \neq \ell_{i+1}$, $i=1,\ldots,n-1$) then the induced path-cylinder

$$\mathsf{PCyl}^s(\ell_1^+,\ldots,\ell_n^+) = \{\varsigma \in \mathsf{Path}_{\mathsf{inf}}(s) : \mathsf{trace}(\varsigma) \in \Delta(\ell_1^+,\ldots,\ell_n^+)\}$$

agrees with

$$\bigcup_{k_1,\ldots,k_n \geq 1} \left(\bigcup_{\substack{\ell \in 2^{\mathsf{AP}} \\ \ell \neq \ell_n}} \mathsf{PCyl}^s(\ell_1^{k_1},\ldots,\ell_n^{k_n},\ell) \cup \bigcap_{k \geq 1} \mathsf{PCyl}^s(\ell_1^{k_1},\ldots,\ell_n^{k_n},\ell_n^k) \right)$$

The paths-sets $\mathsf{PCyl}^s(\ell_1^{k_1},\ldots,\ell_n^{k_n},\ell)$ and $\bigcap_{k\geq 1}\mathsf{PCyl}^s(\ell_1^{k_1},\ldots,\ell_n^{k_n},\ell_n^k)$ are pairwise disjoint. Thus, we obtain from (VI) that

$$\mathsf{Pr}_s^{\mathcal{U}_1}(\Delta(\ell_1^+,\ldots,\ell_n^+)) = \mathsf{Pr}_s^{\mathcal{U}}(\Delta(\ell_1^+,\ldots,\ell_n^+)).$$

Hence

$$\mathsf{Pr}_s^{\mathcal{U}_1}(\mathsf{E}) \;=\; \mathsf{Pr}_s^{\mathcal{U}}(\mathsf{E})$$

for any stutter invariant measurable linear-time property E.

Inductive definition of the scheduler sequence. In a similar fashion we can now define a sequence of schedulers $\mathcal{U}_2, \mathcal{U}_3, \ldots$, where $\mathcal{U}_{i+1}$ is constructed from $\mathcal{U}_i$, mimicking its first $i-1$ steps, i.e.

$$\mathcal{U}_{i+1}(\pi) \;=\; \mathcal{U}_i(\pi)$$

if $|\pi| \leq i-1$. For the paths of length $\geq i$ we apply the technique described in the construction of $\mathcal{U}_1$ from $\mathcal{U}$. This yields that $\mathcal{U}_i(\pi) \in \mathsf{Distr}(\mathsf{ample}(\mathsf{last}(\pi)))$ for each path π of length $< i$, that is the i-th prefix of any $\mathcal{U}_i$-path is a path in $\hat{\mathcal{M}}$.

Using induction on i, we obtain for each $i \in \mathbb{N}_{\geq 1}$

$$\mathsf{Pr}_s^{\mathcal{U}_i}(\mathsf{E}) \;=\; \mathsf{Pr}_s^{\mathcal{U}}(\mathsf{E})$$

for all stutter invariant measurable linear-time property E, which is what we wanted to show.

3.3.2.2. Proof of Lemma 3.3.6

Remember that case [2] holds, so $|\mathsf{ample}(s)| = 1$. In particular, $\mathsf{ample}(s) = \{\beta\}$. We need to show that

$$\begin{aligned}\sum_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') &= \sum_{\sigma \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) \\ &+ \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho) \cdot (1 - \mathcal{U}(\rho)(\beta)) \cdot \delta(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi))\end{aligned} \qquad \textbf{(I)}$$

holds. First we fix some notation. Let $\pi_1 = s \xrightarrow{\beta} t_0 \xrightarrow{\alpha_1} \dots \xrightarrow{\alpha_{n-1}} t_{n-1} \xrightarrow{\alpha_n} t_n$, where $\alpha_1, \dots, \alpha_n \notin \mathsf{ample}(s)$ are independent from β and $\pi_2 = v_0 \xrightarrow{\gamma_1} v_1 \xrightarrow{\gamma_2} \dots \xrightarrow{\gamma_{m-1}} v_{m-1} \xrightarrow{\gamma_m} v_m$ where γ_1 is dependent on $\mathsf{ample}(s)$ and $v_0 = t_n$. Let π be $\pi_1 \circ \pi_2$ and $\ell_i = \{s : L(s) = L(t_i)\}$, $0 \le i \le n$. We observe that if $m \ge 1$, then for $\pi_1' \in [\pi_1]$ the following holds.

$$[\pi_1' \circ \pi_2{\uparrow}^{-1}] = [\pi_1 \circ \pi_2{\uparrow}^{-1}] \qquad (*)$$

$$[\pi_1 \circ \pi_2{\uparrow}^{-1}] = \{\pi_1'' \circ \pi_2{\uparrow}^{-1} \mid \pi_1'' \in [\pi_1]\}, \qquad (**)$$

where for a given path σ of length ≥ 1, $\sigma{\uparrow}^{-1}$ denotes the prefix $\sigma{\uparrow}^{|\sigma|-1}$.

We split the proof of equation (I) in two parts, one dealing with $m \ge 1$ and one dealing with $m = 0$.

Case $m \ge 1$:
Therefore $\pi = \pi_1 \circ \pi_2$, where $|\pi_2| \ge 1$. It thus follows that

$$\sum_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \quad =$$

$$\sum_{\pi_1' \in [\pi_1]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi_1' \circ \pi_2) \quad =$$

$$\sum_{\pi_1' \in [\pi_1]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi_1' \circ \pi_2{\uparrow}^{-1}) \cdot \mathcal{U}_1(\pi_1' \circ \pi_2{\uparrow}^{-1})(\gamma_m) \cdot \delta(v_{m-1}, \gamma_m, v_m) \quad =$$

$$\sum_{\pi_1' \in [\pi_1]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi_1' \circ \pi_2{\uparrow}^{-1}) \cdot \delta(v_{m-1}, \gamma_m, v_m) \cdot \frac{\left[\sum_{\sigma' \in [\pi_1' \circ \pi_2 \uparrow^{-1}]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\gamma_m) + 0\right]}{\sum_{\pi'' \in [\pi_1' \circ \pi_2 \uparrow^{-1}]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \quad \overset{(*)}{=}$$

$$\sum_{\pi_1' \in [\pi_1]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi_1' \circ \pi_2{\uparrow}^{-1}) \cdot \delta(v_{m-1}, \gamma_m, v_m) \cdot \frac{\sum_{\sigma' \in [\pi_1 \circ \pi_2 \uparrow^{-1}]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\gamma_m)}{\sum_{\pi'' \in [\pi_1 \circ \pi_2 \uparrow^{-1}]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \quad \overset{(**)}{=}$$

$$\sum_{\sigma' \in [\pi_1 \circ \pi_2 \uparrow^{-1}]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\gamma_m) \cdot \delta(v_{m-1}, \gamma_m, v_m) \cdot \frac{\sum\limits_{\pi_1' \in [\pi_1]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi_1' \circ \pi_2 \uparrow^{-1})}{\sum\limits_{\pi_1'' \in [\pi_1]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi_1'' \circ \pi_2 \uparrow^{-1})} =$$

$$\sum_{\sigma' \in [\pi_1 \circ \pi_2 \uparrow^{-1}]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma' \xrightarrow{\gamma_m} v_m) =$$

$$\sum_{\sigma \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma)$$

which is what we wanted to show, since the second sum on the right hand side of equation (I) equals zero in the case that $m \geq 1$.

Case $m = 0$: (thus $\pi_2 = t_n$)
So $\pi = \pi_1 = s \xrightarrow{\beta} t_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} t_{n-1} \xrightarrow{\alpha_n} t_n$, where $\alpha_1, \ldots, \alpha_n \notin \mathsf{ample}(s)$ are independent from β. We fix some notation. Let

$$X_n = \{x \mid \exists \pi' \in [\pi] \text{ s.th. } x = \pi_n'\}.$$

Note that $L(x) = L(t_{n-1})$ for all $x \in X_n$ (by definition of $\sim$ and thus $[\pi]$). To each $x \in X_n$ we define a distinguished path π_x such that $\mathsf{last}(\pi_x) = x$ and $\pi_x \xrightarrow{\alpha_n} t_n \in [\pi]$. Then

$$[\pi] = \dot{\bigcup_{x \in X_n}} \dot{\bigcup_{\pi' \in [\pi_x]}} \{\pi' \xrightarrow{\alpha_n} t_n\} \ \dot{\bigcup} \ \{\pi' \in [\pi] \mid \overrightarrow{\pi'_{\mathsf{Act}}} = \alpha_1 \ldots \alpha_n \beta\}. \qquad (***)$$

It follows that

$$\sum_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') =$$

$$\sum_{\substack{x \in X_n \\ \pi' \in [\pi_x]}} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi' \xrightarrow{\alpha_n} t_n) + \sum_{\substack{\pi'' \in [\pi]: \\ \overrightarrow{\pi''_{\mathsf{Act}}} = \alpha_1 \ldots \alpha_n \beta}} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'') =$$

$$\sum_{\substack{x \in X_n \\ \pi' \in [\pi_x]}} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \mathcal{U}_1(\pi')(\alpha_n) \cdot \delta(x, \alpha_n, t_n) + 0 =$$

$$\sum_{\substack{x \in X_n \\ \pi' \in [\pi_x]}} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \delta(x, \alpha_n, t_n) \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot \Big[\sum_{\sigma' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\alpha_n) +$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n-1} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} y_{n-1}}_{\sigma''}) \cdot \mathcal{U}(\sigma'')(\alpha_n) \cdot \delta(y_{n-1}, \beta, x)\Big] =$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot \sum_{\sigma' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\alpha_n) \cdot \delta(x, \alpha_n, t_n) \Big] \quad +$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n-1} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} y_{n-1}}_{\sigma''}) \cdot \mathcal{U}(\sigma'')(\alpha_n) \cdot \delta(y_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \Big]$$

For the sake of readability we will compute the two summands separately. Note that (i) $[\pi'] = [\pi_x]$ for $\pi' \in [\pi_x]$. The first summand gives

$$\text{(a)} \quad =$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot \sum_{\sigma' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\alpha_n) \cdot \delta(x, \alpha_n, t_n) \Big] \quad \stackrel{\text{(i)}}{=}$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot \sum_{\sigma' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\alpha_n) \cdot \delta(x, \alpha_n, t_n) \Big] \quad =$$

$$\sum_{x \in X_n} \Big[\sum_{\sigma' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\alpha_n) \cdot \delta(x, \alpha_n, t_n) \cdot \frac{\sum\limits_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi')}{\sum\limits_{\pi'' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \Big] \quad =$$

$$\sum_{x \in X_n} \sum_{\sigma' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma' \xrightarrow{\alpha_n} t_n)$$

Recall that (ii) β is a stutter action and (iii) α_n and β are independent. The second summand leads to

$$\text{(b)} \quad =$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n-1} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} y_{n-1}}_{\sigma''}) \cdot \mathcal{U}(\sigma'')(\alpha_n) \cdot \delta(y_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \Big] \quad \stackrel{\text{(i)}}{=}$$

$$\sum_{x \in X_n} \Big[\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n-1} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} y_{n-1}}_{\sigma''}) \cdot \mathcal{U}(\sigma'')(\alpha_n) \cdot \delta(y_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \cdot$$

$$\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \Big] \quad =$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n-1} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} y_{n-1}}_{\sigma''}) \cdot \mathcal{U}(\sigma'')(\alpha_n) \cdot \sum_{x \in X_n} \delta(y_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \stackrel{\text{(ii)}}{=}$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n-1} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} y_{n-1}}_{\sigma''}) \cdot \mathcal{U}(\sigma'')(\alpha_n) \cdot \sum_{x} \delta(y_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \stackrel{\text{(iii)}}{=}$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n-1} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} y_{n-1}}_{\sigma''}) \cdot \mathcal{U}(\sigma'')(\alpha_n) \cdot \sum_{x} \delta(y_{n-1}, \alpha_n, x) \cdot \delta(x, \beta, t_n) \stackrel{\text{(ii)}}{=}$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} y_{n-1}}_{\sigma''}) \cdot \mathcal{U}(\sigma'')(\alpha_n) \cdot \delta(y_{n-1}, \alpha_n, y_n) \cdot \delta(y_n, \beta, t_n) =$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n} \mathsf{Pr}_s^{\mathcal{U}}(s \xrightarrow{\alpha_1} y_1 \ldots \xrightarrow{\alpha_{n-1}} y_{n-1} \xrightarrow{\alpha_n} y_n) \cdot \delta(y_n, \beta, t_n) =$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} y_1 \ldots \xrightarrow{\alpha_{n-1}} y_{n-1} \xrightarrow{\alpha_n} y_n}_{=\sigma'}) \cdot \delta(y_n, \beta, t_n) \cdot$$

$$\big((1 - \mathcal{U}(\sigma')(\beta)) + \mathcal{U}(\sigma')(\beta)\big) =$$

$$\sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho) \cdot (1 - \mathcal{U}(\rho)(\beta)) \cdot \delta(\mathsf{last}(\rho), \beta, t_n) +$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n} \mathsf{Pr}_s^{\mathcal{U}}(\underbrace{s \xrightarrow{\alpha_1} y_1 \ldots \xrightarrow{\alpha_{n-1}} y_{n-1} \xrightarrow{\alpha_n} y_n}_{=\sigma'}) \cdot \mathcal{U}(\sigma')(\beta) \cdot \delta(y_n, \beta, t_n) =$$

$$\sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho) \cdot (1 - \mathcal{U}(\rho)(\beta)) \cdot \delta(\mathsf{last}(\rho), \beta, t_n) +$$

$$\sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n} \mathsf{Pr}_s^{\mathcal{U}}(s \xrightarrow{\alpha_1} y_1 \ldots \xrightarrow{\alpha_{n-1}} y_{n-1} \xrightarrow{\alpha_n} y_n \xrightarrow{\beta} t_n)$$

This concludes to

$$\sum_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \;=\; \text{(a)} \;+\; \text{(b)} =$$

$$\sum_{x \in X_n} \sum_{\sigma' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma' \xrightarrow{\alpha_n} t_n) + \sum_{\substack{i=1 \\ y_i \in \ell_i}}^{n} \mathsf{Pr}_s^{\mathcal{U}}(s \xrightarrow{\alpha_1} y_1 \ldots \xrightarrow{\alpha_{n-1}} y_{n-1} \xrightarrow{\alpha_n} y_n \xrightarrow{\beta} t_n) +$$

$$\sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho) \cdot (1 - \mathcal{U}(\rho)(\beta)) \cdot \delta(\mathsf{last}(\rho), \beta, t_n) \stackrel{(***)}{=}$$

$$\sum_{\sigma\in[\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) + \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho) \cdot (1 - \mathcal{U}(\rho)(\beta)) \cdot \delta(\mathsf{last}(\rho), \beta, t_n)$$

which is equation (I) and completes the proof.

3.3.2.3. Proof of Lemma 3.3.7

Remember that case [3] holds, so $1 < |\mathsf{ample}(s)| < \mathsf{Act}(s)$. Let π be a path such that $s = \mathsf{first}(\pi)$ and $\beta \in \mathsf{ample}(s)$ is the first action on π (and $\beta \neq \beta_s$). Then we need to show equation (I').

$$\begin{aligned}\sum_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \;&=\; \sum_{\sigma \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) && \textbf{(I')}\\ &+ \sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+\beta}) \cdot \delta(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi))\end{aligned}$$

We use the same notation as in the proof of Lemma 3.3.6 in subsection 3.3.2.2.

Let $\pi_1 = s \xrightarrow{\beta} t_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} t_{n-1} \xrightarrow{\alpha_n} t_n$, where $\alpha_1, \ldots, \alpha_n \notin \mathsf{ample}(s)$ are independent from β and $\pi_2 = v_0 \xrightarrow{\gamma_1} v_1 \xrightarrow{\gamma_2} \ldots \xrightarrow{\gamma_{m-1}} v_{m-1} \xrightarrow{\gamma_m} v_m$ where γ_1 is dependent on $\mathsf{ample}(s)$ and $v_0 = t_n$. Let π be $\pi_1 \circ \pi_2$ and $\ell_i = \{s : L(s) = L(t_i)\}$, $0 \leq i \leq n$.

We split the proof of (I') in two parts, one dealing with $m \geq 1$ and one dealing with $m = 0$.

Case $m \geq 1$: This follows the same lines as in the proof of Lemma 3.3.6.

Case $m = 0$: (thus $\pi_2 = t_n$)
So $\pi = \pi_1 = s \xrightarrow{\beta} t_0 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{n-1}} t_{n-1} \xrightarrow{\alpha_n} t_n$, where $\alpha_1, \ldots, \alpha_n \notin \mathsf{ample}(s)$ are independent from β. Note that the sum in the second summand of the definition of $\mathcal{U}_1(\pi'\!\uparrow^{-1})(\alpha_n)$ ranges over the single path $\rho' = s \xrightarrow{\alpha_1} s_1 \ldots \xrightarrow{\alpha_{n-1}} s_{n-1}$ for $\pi' \in [\pi]$. Indeed the s_i's are uniquely defined as the α_i's are non-probabilistic actions (since we are in case [3]). Let s_n be the uniquely defined state such that $\delta(s_{n-1}, \alpha_n, s_n) = 1$. We then compute

$$\begin{aligned}&\sum_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') &&=\\ &\sum_{\substack{x \in X_n \\ \pi' \in [\pi_x]}} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi' \xrightarrow{\alpha_n} t_n) \;+\; \sum_{\substack{\pi'' \in [\pi]: \\ \overrightarrow{\pi''_{\mathsf{Act}}} = \alpha_1 \ldots \alpha_n \beta}} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'') &&=\\ &\sum_{\substack{x \in X_n \\ \pi' \in [\pi_x]}} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \mathcal{U}_1(\pi')(\alpha_n) \cdot \delta(x, \alpha_n, t_n) \;+\; 0 &&=\\ &\sum_{\substack{x \in X_n \\ \pi' \in [\pi_x]}} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \delta(x, \alpha_n, t_n) \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot \Big[\sum_{\sigma' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\alpha_n) \;+ \\ &\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \delta(s_{n-1}, \beta, x)\Big] &&=\end{aligned}$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot \sum_{\sigma' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\alpha_n) \cdot \delta(x, \alpha_n, t_n) \Big] \quad +$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot$$

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \delta(s_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \Big]$$

For the sake of readability we compute the two summands separately. As in the proof of Lemma 3.3.6, the first summand amounts to

$$\text{(a)} \quad =$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot \sum_{\sigma' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}}(\sigma') \cdot \mathcal{U}(\sigma')(\alpha_n) \cdot \delta(x, \alpha_n, t_n) \Big] \quad =$$

$$\sum_{x \in X_n} \sum_{\sigma' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma' \xrightarrow{\alpha_n} t_n)$$

The second summand leads to

$$\text{(b)} \quad =$$

$$\sum_{x \in X_n} \Big[\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi']} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \cdot$$

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \delta(s_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \Big] \quad \stackrel{\text{(i)}}{=}$$

$$\sum_{x \in X_n} \Big[\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \delta(s_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \cdot$$

$$\sum_{\pi' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \cdot \frac{1}{\sum\limits_{\pi'' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi'')} \Big] \quad =$$

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \sum_{x \in X_n} \delta(s_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \quad \stackrel{\text{(ii)}}{=}$$

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \sum_{x} \delta(s_{n-1}, \beta, x) \cdot \delta(x, \alpha_n, t_n) \quad \stackrel{\text{(iii)}}{=}$$

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \sum_{x} \delta(s_{n-1}, \alpha_n, x) \cdot \delta(x, \beta, t_n) \quad \stackrel{\text{(iv)}}{=}$$

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \delta(s_n, \beta, t_n)$$

Recall that (iv) s_n is the uniquely defined state such that $\delta(s_{n-1}, \alpha_n, s_n) = 1$.

As α_n is non-probabilistic it holds that

$$\begin{array}{lcll} \mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) & = & \mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n} s_n \xrightarrow{\mathsf{na(s)}^* \beta}) & = \\ \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^* \beta}) & = & \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\beta}) + \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+ \beta}), & \end{array}$$

where $\rho = \rho' \xrightarrow{\alpha_n} s_n$. So we continue the above computation which leads to

$$\begin{array}{lr} \text{(b)} & = \\ \mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^* \beta}) \cdot \delta(s_n, \beta, t_n) & = \\ \big(\mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\beta}) + \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+ \beta})\big) \cdot \delta(s_n, \beta, t_n) & = \\ \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\beta}) \cdot \delta(s_n, \beta, t_n) \ + \ \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+ \beta}) \cdot \delta(s_n, \beta, t_n) & \end{array}$$

This concludes to

$$\begin{array}{lr} \sum\limits_{\pi' \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}_1}(\pi') \ = \ \text{(a)} \ + \ \text{(b)} & = \\ \sum\limits_{x \in X_n} \sum\limits_{\sigma' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma' \xrightarrow{\alpha_n} t_n) \ + \ \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\beta}) \cdot \delta(s_n, \beta, t_n) & + \\ \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+ \beta}) \cdot \delta(s_n, \beta, t_n) & = \\ \sum\limits_{x \in X_n} \sum\limits_{\sigma' \in [\pi_x]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma' \xrightarrow{\alpha_n} t_n) \ + \ \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\beta} t_n) & + \\ \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+ \beta}) \cdot \delta(s_n, \beta, t_n) & \stackrel{(***)}{=} \\ \sum\limits_{\sigma \in [\pi]} \mathsf{Pr}_s^{\mathcal{U}}(\sigma) & + \\ \sum\limits_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^+ \beta}) \cdot \delta(\mathsf{last}(\rho), \beta, \mathsf{last}(\pi)) & \end{array}$$

which is equation (I') and completes the proof.

Remark 3.3.10. ($[\beta = \beta_s]$) In the case that $\beta = \beta_s$, similar computations apply. Note that as α_n is non-probabilistic it holds that

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^\omega}) \ = \ \mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n} s_n \xrightarrow{\mathsf{na(s)}^\omega}) \ = \ \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^\omega}),$$

where $\rho = \rho' \xrightarrow{\alpha_n} s_n$. So if the first action on π is β_s, then the additional summand

$$\sum_{\substack{\rho \text{ s.t. } \rho \xrightarrow{\beta_s} \mathsf{last}(\pi) \in [\pi] \\ \mathsf{ample}(s) \cap \mathsf{Act}(\rho) = \emptyset}} \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\alpha \mathsf{na(s)}^\omega}) \cdot \delta\big(\mathsf{last}(\rho), \beta_s, \mathsf{last}(\pi)\big)$$

in the definition of $\mathcal{U}_1(\pi)(\alpha)$ for $\alpha \notin \mathsf{ample}(s)$ leads to the additional summand

$$\mathsf{Pr}_s^{\mathcal{U}}(\rho' \xrightarrow{\alpha_n \mathsf{na(s)}^\omega}) \cdot \delta(s_n, \beta_s, t_n) \;=\; \mathsf{Pr}_s^{\mathcal{U}}(\rho \xrightarrow{\mathsf{na(s)}^\omega}) \cdot \delta(s_n, \beta_s, t_n)$$

in the computations above and therefore to the additional summand of equation (I') in Lemma 3.3.7.

3.4. The ample set method for MDPs and branching time properties

While conditions (A0)-(A4) ensure the equivalence of a given MDP $\mathcal{M}$ and its reduced sub-MDP $\hat{\mathcal{M}}$ in the context of stutter invariant linear-time specifications, they are not sufficient for branching time specifications. This is rather not surprising as the conditions (A0)-(A4) fall back to the original conditions (A0)-(A3), if applied to non-probabilistic systems. In the non-probabilistic setting, these conditions ensure equivalence of a given system $\mathcal{T}$ and its reduced system $\hat{\mathcal{T}}$ in the linear-time setting, but not in the branching time setting (see [GKPP95]). In [GKPP95] the authors proposed the branching condition

(A4.1) (Branching condition)
$|\mathsf{ample}(s)| = 1$ or $\mathsf{ample}(s) = \mathsf{Act}(s)$ for any state $s \in \hat{S}$.

and showed that the additional condition (A4.1) ensures that $\mathcal{T}$ and $\hat{\mathcal{T}}$ are visible bisimilar (which implies equivalence of $\mathcal{T}$ and $\hat{\mathcal{T}}$ with respect to CTL$_{\setminus\mathcal{X}}$ formulae).

Remark 3.4.1. Note that condition (A4.1) is stronger than the branching condition (A4) that we used in the previous section.

The reason why conditions (A0)-(A3) do not ensure CTL$_{\setminus\mathcal{X}}$ equivalence between a non-probabilistic transition system $\mathcal{T}$ and its reduced system $\hat{\mathcal{T}}$ can be seen in the following example in Figure 3.8 which is taken from [GKPP95, Pel97]. In the transition system $\mathcal{T}$ the state labeling is indicated by the shades of the states. As α is independent to β and γ, conditions (A0)-(A3) are fulfilled when choosing the ample-set $\{\beta, \gamma\}$ in the initial state s_0 which leads to the reduced system $\hat{\mathcal{T}}$ (note that all other reachable states of $\hat{\mathcal{T}}$ are trivially fully expanded). But then, the CTL$_{\setminus\mathcal{X}}$-formula

$$\forall\Big(\Box\big(\,{\color{gray}\bullet} \to (\forall\Diamond \bullet \vee \forall\Diamond \circ)\,\big)\Big)$$

holds for $\hat{\mathcal{T}}$, but not for $\mathcal{T}$. The reason for this is that the state t_1 is no longer reachable in $\hat{\mathcal{T}}$. As in this state the nondeterministic choice between β and γ still exists, there is no

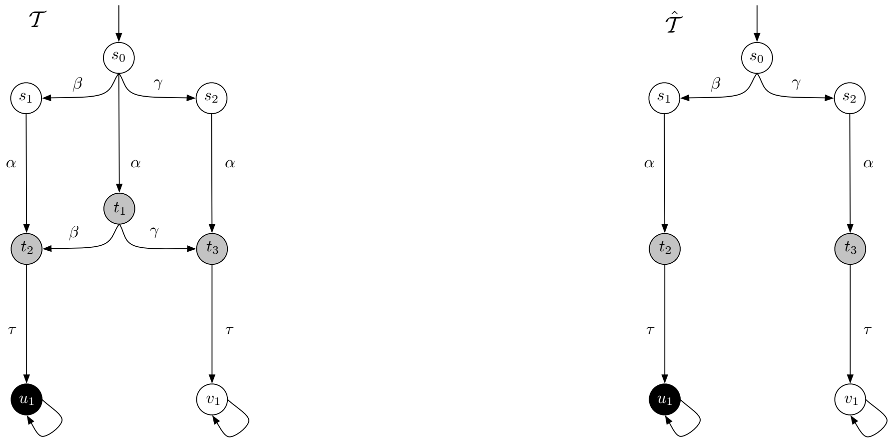


Figure 3.8: Conditions (A0)-(A3) are not sufficient for CTL$_{\setminus\mathcal{X}}$

state in $\hat{\mathcal{T}}$ that is visible bisimilar to t_1 (note that bisimilar states must have the same state labeling).

So the problem that occurs is that a non-fully expanded state has several successors in the reduced system that are not CTL$_{\setminus\mathcal{X}}$-equivalent, because the choice between those successor can be delayed in the original system by taking non-ample actions. But this might lead to some states in the original system (like t_1 in the example above) that do not have a bisimilar state in the reduced system.

In the non-probabilistic setting this problem is solved by the branching condition (A4.1), that requires that the ample-set of a non-fully expanded state consists of a single action, thus yielding only one possible successor in the reduced system. However this does not work in the probabilistic setting as this single ample action can branch probabilistically and therefore yields several successors in the reduced system.

3.4.1. The old conditions are not sufficient

Example 3.4.2 (Conditions (A0)-(A3) and (A4.1) are not sufficient for PCTL$_{\setminus\mathcal{X}}$)**.** We can easily modify the above example to show that condition (A4.1) is not strong enough to ensure the equivalence of a given MDP $\mathcal{M}$ and its reduced sub-MDP $\hat{\mathcal{M}}$ with respect to PCTL$_{\setminus\mathcal{X}}$ formulae. The counterexample given in Figure 3.9 illustrates this. Again the state labeling is indicated by the shades of the states, so β is a stutter action. Actions α and β are independent. Conditions (A0)-(A3) and (A4.1) are fulfilled when choosing the singleton ample-set $\{\beta\}$ in the initial state s_0 which leads to the reduced MDP $\hat{\mathcal{M}}$ in Figure 3.9 (note that all other reachable states of $\hat{\mathcal{M}}$ are trivially fully expanded). But then, the PCTL$_{\setminus\mathcal{X}}$-formula

$$\Big[\Box\big(\,{\color{gray}\bullet} \rightarrow ([\Diamond \bullet]_{=1} \vee [\Diamond \bigcirc]_{=1})\,\big)\Big]_{=1}$$

holds for $\hat{\mathcal{M}}$, but not for $\mathcal{M}$.

The explanation for this is the same as earlier in the non-probabilistic case. Although β

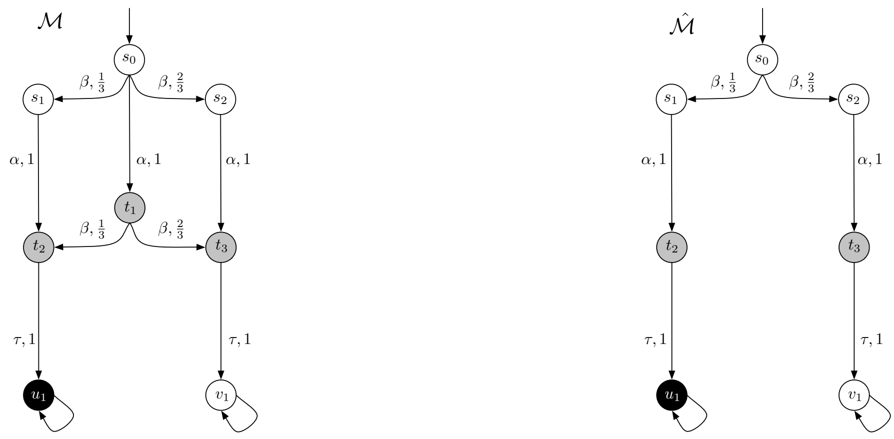


Figure 3.9: Conditions (A0)-(A3) and (A4.1) are not sufficient for PCTL$_{\backslash\mathcal{X}}$

is the only ample action of the state s_0, it branches probabilistically and yields several successors that are not PCTL$_{\backslash\mathcal{X}}$-equivalent. This branching is delayed when executing the non-ample action α in s_0, yielding t_1 as the successor. But then t_1 has no corresponding state in the reduced MDP $\hat{\mathcal{M}}$.

Remark 3.4.3. D'Argenio and Niebert noted in [DN04] that "the interplay between nondeterminism and probabilism is comparable to that of existential and universal quantification." They thus followed the approach by Gerth et al [GKPP95] and proposed condition (A4.1) to ensure that the interplay between nondeterministic and probabilistic choices is preserved: either the unique nondeterministic choice is safe (hence $|\mathsf{ample}(s)| = 1$) or all branching is preserved ($\mathsf{ample}(s) = \mathsf{Act}(s)$). We have already seen in example 3.4.2 that this does not preserve PCTL$_{\backslash\mathcal{X}}$, but [DN04] established a (probabilistic weak) complete forward simulation equivalence between a given MDP $\mathcal{M}$ and the reduced MDP $\hat{\mathcal{M}}$. In their simulation equivalence a state of $\mathcal{M}$ is not simulated by one state of $\hat{\mathcal{M}}$, but by a probability distribution over the states of $\hat{\mathcal{M}}$. Thus, in example 3.4.2 the state t_1 of $\mathcal{M}$ is simulated by $\frac{1}{3} \cdot t_2 + \frac{2}{3} \cdot t_3$ in $\hat{\mathcal{M}}$.

The reader should notice that [DN04] required a stronger underlying structure for the given MDP. They assumed each action to have a fixed probabilistic branching pattern. Moreover they assumed the stronger cycle condition (A3) and not the end component condition.

3.4.2. Sufficient conditions for preserving branching time properties

We have seen that in order to preserve branching time properties, it is crucial to ensure that a non-fully expanded state yields a unique successor in the reduced system. We therefore propose a stronger branching condition

(A4.2) (Branching condition)
If $\mathsf{ample}(s) \neq \mathsf{Act}(s)$ then $\mathsf{ample}(s)$ is a singleton consisting of a non-probabilistic action.

Remark 3.4.4 (Conservative extension of the ample set method)**.** It should be noticed that condition (A4.2) is stronger than (A4.1) and collapses to (A4.1) for ordinary transition systems. Thus, the conditions (A0)-(A3), (A4.2) that we suggest for a reduction that preserves probabilistic branching time properties yield a conservative adaption of the conditions (A0)-(A3), (A4.1) suggested by Gerth et al. [GKPP95, Pel97] for non-probabilistic systems and CTL$_{\setminus\mathcal{X}}$-properties.

Remark 3.4.5 (Cycle versus end component condition)**.** Please notice, that in combination with (A4.2) the end component condition (A3) collapses to the original cycle condition. This follows from the fact that for any cycle in $\hat{\mathcal{M}}$ where none of its states is fully expanded, the ample-sets of all its states are singletons consisting of a non-probabilistic action. So the cycle under consideration is an end component in $\hat{\mathcal{M}}$ and thus, the end component condition implies the cycle condition.

In the remainder of this subsection we will show the correctness of our approach which is stated in the following theorem.

Theorem 3.4.6 (Ample set method for MDPs - branching time properties).
If (A0)-(A3) and (A4.2) are fulfilled then $\mathcal{M}$ and $\hat{\mathcal{M}}$ satisfy the same $\mathrm{PCTL}^*_{\setminus\mathcal{X}}$ state formulae.

We use a proof technique similar to those of [GKPP95, Pel97] where (A0)-(A3) and (A4.1) are shown to be sufficient for CTL$^*_{\setminus\mathcal{X}}$ properties and non-probabilistic transition systems. However, we have here the additional difficulty to reason about probabilistic behaviors and will therefore need the concept of weight functions.

Definition 3.4.7. [Weight functions, cf. [JL91]]
Let S, S' be finite sets and $\mathcal{R} \subseteq S \times S'$. If μ and μ' are probability distributions on S and S' respectively then a weight function for (μ, μ') with respect to $\mathcal{R}$ denotes a function $w : S \times S' \to [0, 1]$ such that

- $w(s, s') > 0$ implies $(s, s') \in \mathcal{R}$,
- $\sum_{s' \in S'} w(s, s') = \mu(s)$ for all $s \in S$ and $\sum_{s \in S} w(s, s') = \mu'(s')$ for all $s' \in S'$.

We write $\mu \sqsubseteq_{\mathcal{R}} \mu'$ to denote the existence of a weight function for (μ, μ') with respect to $\mathcal{R}$ and refer to $\sqsubseteq_{\mathcal{R}}$ as the lifting of $\mathcal{R}$ to distributions.

In the sequel, we will use the following observation which is e.g. shown in [Bai98, Des99].

Proposition 3.4.8 (Transitivity of $\sqsubseteq_{\mathcal{R}}$)**.** If $\mathcal{R}$ is a binary, transitive relation on a set S and μ, μ', μ'' are distributions on S such that $\mu \sqsubseteq_{\mathcal{R}} \mu'$ and $\mu' \sqsubseteq_{\mathcal{R}} \mu''$, then $\mu \sqsubseteq_{\mathcal{R}} \mu''$.

The following definition can be viewed as a probabilistic variant of the so-called *visible bisimulation* that has been introduced in [GKPP95, Pel97].

Definition 3.4.9. [Probabilistic visible bisimulation (pvb)]
Let $\mathcal{M} = (S, \mathsf{Act}, \delta, \mu)$ and $\mathcal{M}' = (S', \mathsf{Act}', \delta', \mu')$ be two state-labeled MDPs with the same set of atomic propositions AP and the labeling functions L and L'. Let $\mathcal{R} \subseteq S \times S'$ be a binary relation. Then $\mathcal{R}$ is called a probabilistic visible simulation if $\mu \sqsubseteq_{\mathcal{R}} \mu'$ and for any pair (s, s') in $\mathcal{R}$ the following three conditions are fulfilled.

(1) $L(s) = L'(s')$

(2) For any action $\alpha \in \mathsf{Act}(s)$ at least one of the following two conditions holds:

(2.1) α is a non-probabilistic stutter action such that $(t, s') \in \mathcal{R}$ for the unique α-successor t of s.

(2.2) There is a finite path of the form $s' = s'_0 \xrightarrow{\beta_0} s'_1 \xrightarrow{\beta_1} \ldots \xrightarrow{\beta_{n-1}} s'_n$ in $\mathcal{M}'$ (with $n \geq 0$) such that

- $\beta_0, \ldots, \beta_{n-1}$ are non-probabilistic stutter actions,
- $(s, s'_i) \in \mathcal{R}$ for $1 \leq i \leq n$,
- $\alpha \in \mathsf{Act}'(s'_n)$ and $\delta(s, \alpha, \cdot) \sqsubseteq_{\mathcal{R}} \delta'(s'_n, \alpha, \cdot)$.

(3) If there is an infinite path of the form $s = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} s_3 \xrightarrow{\alpha_3} \ldots$ in $\mathcal{M}$ consisting of non-probabilistic stutter actions $\alpha_0, \alpha_1, \alpha_2, \ldots$ and such that $(s_i, s') \in \mathcal{R}$, $i = 0, 1, 2, \ldots$ then there is a finite path of the form

$$s' = s'_0 \xrightarrow{\gamma_0} s'_1 \xrightarrow{\gamma_1} \ldots \xrightarrow{\gamma_{j-1}} s'_j \xrightarrow{\gamma_j} s'_{j+1}$$

in $\mathcal{M}'$ such that $(s, s'_i) \in \mathcal{R}$, $i = 0, 1, \ldots, j$ and an index k such that $(s_k, s'_{j+1}) \in \mathcal{R}$, and $\gamma_0, \gamma_1, \ldots, \gamma_{j-1}, \gamma_j$ are non-probabilistic stutter actions.

$\mathcal{R}$ is called a probabilistic visible bisimulation for $(\mathcal{M}, \mathcal{M}')$ if $\mathcal{R}$ is a probabilistic visible simulation for $(\mathcal{M}, \mathcal{M}')$ and $\mathcal{R}^{-1}$ is a probabilistic visible simulation for $(\mathcal{M}', \mathcal{M})$. We write $\mathcal{M} \approx_{\text{pvb}} \mathcal{M}'$ if and only if there exists a probabilistic visible bisimulation for $(\mathcal{M}, \mathcal{M}')$.

Our goal is to show that $\mathcal{M} \approx_{\text{pvb}} \hat{\mathcal{M}}$ where $\mathcal{M}$ denotes the "full" MDP and $\hat{\mathcal{M}}$ the reduced MDP that results from ample-sets satisfying (A0)-(A3) and (A4.2). The following proposition completes then our argumentation.

Proposition 3.4.10 (Soundness of pvb for $\mathrm{PCTL}^*_{\backslash\mathcal{X}}$). Let $\mathcal{M}$ and $\mathcal{M}'$ be two MDPs as in Definition 3.4.9 such that $\mathcal{M} \approx_{\text{pvb}} \mathcal{M}'$. Then, $\mathcal{M}$ and $\mathcal{M}'$ satisfy the same $\mathrm{PCTL}^*_{\backslash\mathcal{X}}$ state formulae.

Proof. (Sketch). One proof obligation relies on proving that the coarsest probabilistic visible bisimulation $\mathcal{R}$ is a divergence-sensitive probabilistic branching bisimulation and the latter is sound for $\mathrm{PCTL}^*_{\backslash\mathcal{X}}$ [SL95, Seg95].[1]

Another different proof obligation is to provide a direct proof for the claim and to show by structural induction on the syntax of $\mathrm{PCTL}^*_{\backslash\mathcal{X}}$ state/path formulae that whenever $\mathcal{R}$ is a probabilistic visible bisimulation then

- for all $\mathrm{PCTL}^*_{\backslash\mathcal{X}}$ state formulae Φ and $(s, s') \in \mathcal{R}$: $s \models \Phi$ iff $s' \models \Phi$

[1]There are some minor differences between our approach and those in [SL95, Seg95], e.g. they use an action-labeled setting and prove the preservation result under the assumption of probabilistic convergence (rather than considering a divergence-sensitive variant of probabilistic branching bisimulation). However, the main argumentation for the preservation result for a notion of divergence-sensitive probabilistic branching bisimulation will be the same.

- for all $\mathrm{PCTL}^*_{\setminus \mathcal{X}}$ path formulae φ and $(\pi, \pi') \in \mathcal{R}_{\text{path}}$: $\pi \models \varphi$ iff $\pi' \models \varphi$

Here, $\mathcal{R}_{\text{path}}$ denotes the "lifting" of $\mathcal{R}$ to paths (which has to be defined in an appropriate way). We skip the details of this proof obligation too as it relies on standard arguments provided e.g. in [SL95, Seg95] (and also [DGJP02] for an MDP-like model where probabilistic and nondeterministic states alternate). $\square$

In the sequel, we assume ample sets that satisfy (A0)-(A3) and (A4.2). Our goal is now to establish a probabilistic visible bisimulation that relates $\mathcal{M}$ and $\hat{\mathcal{M}}$.

Definition 3.4.11. [Forming path, relation $\leadsto$]
Let $\mathcal{M}$ be an MDP as before and let $s, s' \in S$. A forming path from s to s' is a finite path π of the form

$$s = s_0 \xrightarrow{\beta_0} s_1 \xrightarrow{\beta_1} \ldots \xrightarrow{\beta_{n-1}} s_n = s' \qquad (*)$$

where $\beta_0, \ldots, \beta_{n-1}$ are non-probabilistic stutter actions and, for $i = 0, 1, \ldots, n-1$, the singleton action-set $\{\beta_i\}$ satisfies the dependence condition (A2) for state s_i. That is, for any finite path $s_i \xrightarrow{\alpha_0} t_1 \xrightarrow{\alpha_1} \ldots \xrightarrow{\alpha_{m-1}} t_m \xrightarrow{\gamma} \ldots$ where γ is dependent on β_i there exists $j \in \{0, 1, \ldots, m-1\}$ such that $\alpha_j = \beta_i$. We write $s \leadsto s'$ if and only if there exists a forming path from s to s'. $\sqsubseteq_{\leadsto}$ denotes the lifting of $\leadsto$ to distributions on S via weight functions as in Definition 3.4.7.

As the formal definition of forming paths only refers to non-probabilistic actions and agrees exactly with the definition of forming paths in the non-probabilistic setting [GKPP95, Pel97], the following properties that were established for non-probabilistic systems also hold for MDPs. First, we observe that the relation $\leadsto$ is transitive and reflexive (even though, in general, non-symmetric). Second, if π is a forming path from s to s' of length n as in $(*)$ in Definition 3.4.11 then $s_i \leadsto s_j$ for $0 \le i \le j \le n$. In addition, forming paths enjoy the property that they can be replicated after an independent operation is performed. In a probabilistic setting this can be depicted as in Fig. 3.10 and is formally stated in the next proposition.

Proposition 3.4.12 (Properties of forming paths)**.** Let s, s' be two states in $\mathcal{M}$ such that $s \leadsto s'$ and let $\alpha \in \mathsf{Act}(s)$.

(a) If there is a forming path from s to s' in which α does not occur then $\alpha \in \mathsf{Act}(s')$ and $\delta(s, \alpha, \cdot) \sqsubseteq_{\leadsto} \delta(s', \alpha, \cdot)$.

(b) If α is a non-probabilistic stutter action with $s \xrightarrow{\alpha} t$ and $t \not\leadsto s'$ then $\alpha \in \mathsf{Act}(s')$ and $s' \xrightarrow{\alpha} t'$ where $t \leadsto t'$. In particular, we also have $\delta(s, \alpha, \cdot) \sqsubseteq_{\leadsto} \delta(s', \alpha, \cdot)$.

Proof.

(a) We prove part (a) using induction on the length n of a forming path from s to s' where α does not occur. The basis of induction $n = 0$ is obvious as we then have $s = s'$. In the induction step $n-1 \Longrightarrow n (n \ge 1)$ we assume that

$$s = s_0 \xrightarrow{\beta_0} s_1 \xrightarrow{\beta_1} \ldots \xrightarrow{\beta_{n-2}} s_{n-1} \xrightarrow{\beta_{n-1}} s_n = s'$$

is a forming path from s to s' such that $\alpha \notin \{\beta_0, \ldots, \beta_{n-1}\}$. By induction hypothesis we have $\alpha \in \mathsf{Act}(s_{n-1})$ and

$$\delta(s, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(s_{n-1}, \alpha, \cdot) \tag{+}$$

As the dependence condition (A2) holds for state s_{n-1} and the singleton action-set $\{\beta_{n-1}\}$, actions α and β_{n-1} are independent. Therefore the probabilistic effect of the action sequences $\alpha\beta_{n-1}$ and $\beta_{n-1}\alpha$ in state s_{n-1} are the same, which gives

$$\sum_{t \in S} \delta(s_{n-1}, \alpha, t) \cdot \delta(t, \beta_{n-1}, u) \;=\; \sum_{t \in S} \delta(s_{n-1}, \beta_{n-1}, t) \cdot \delta(t, \alpha, u)$$

for any state $u \in S$. Observe that β_{n-1} is a non-probabilistic stutter action and that $s_n = s'$ is the unique β_{n-1}-successor of s_{n-1}. Thus the above equation simplifies to

$$\sum_{\substack{t \in \mathsf{supp}(\delta(s_{n-1},\alpha,\cdot)) \\ \beta_{n-1}(t) = u}} \delta(s_{n-1}, \alpha, t) = \delta(s', \alpha, u)$$

for any state $u \in S$, where $\beta_{n-1}(t)$ is the unique β_{n-1}-successor of t if $\beta_{n-1} \in \mathsf{Act}(t)$, otherwise $\beta_{n-1}(t)$ is undefined (this is illustrated in Figure 3.10 for $n = 2$). We may now choose the weights

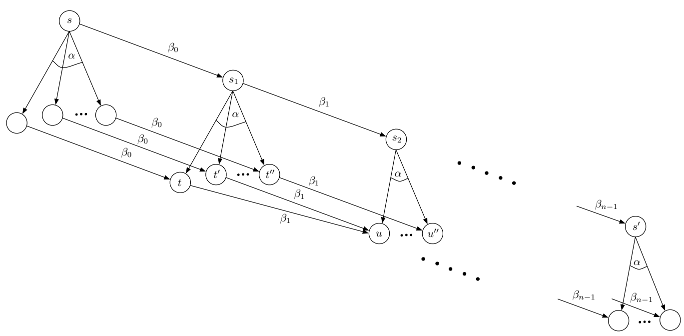


Figure 3.10: Illustration of Proposition 3.4.12, part (a)

$$w(t, \beta_{n-1}(t)) = \delta(s_{n-1}, \alpha, t) \tag{++}$$

for $t \in \mathsf{supp}(\delta(s_{n-1}, \alpha, \cdot))$ and $w(\cdot, \cdot) = 0$ in all remaining cases.

Since α and β_{n-1} are independent and (A2) holds for state s_{n-1} and the singleton action-set $\{\beta_{n-1}\}$, condition (A2) also holds for any α-successor t of s_{n-1} and the singleton action-set $\{\beta_{n-1}\}$. Thus $t \xrightarrow{\beta_{n-1}} \beta_{n-1}(t)$ is a forming path and therefore $t \rightsquigarrow \beta_{n-1}(t)$ for any α-successor t. Hence, with (++)

$$\delta(s_{n-1}, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(s_n, \alpha, \cdot).$$

Using the induction hypothesis (+) and the transitivity of $\sqsubseteq_{\rightsquigarrow}$ (see Proposition 3.4.8) we get $\delta(s, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(s_n, \alpha, \cdot)$.

(b) Let α be a non-probabilistic stutter action such that $s \xrightarrow{\alpha} t$ and $t \not\rightsquigarrow s'$. We have to show that $\alpha \in \mathsf{Act}(s')$ and that $t \rightsquigarrow \alpha(s')$ where given a non-probabilistic action α, $\alpha(s)$ denotes the unique α-successor of s if $\alpha \in \mathsf{Act}(s)$ and is undefined otherwise. We claim that α does not occur on any forming path from s to s'. But then part (a) applies (as $s \rightsquigarrow s'$) and we obtain that $\alpha \in \mathsf{Act}(s')$ and $\delta(s, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(s', \alpha, \cdot)$. As α is a non-probabilistic action, $\delta(s, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(s', \alpha, \cdot)$ implies that $\alpha(s) \rightsquigarrow \alpha(s')$, which shows the claim.

It remains to show that α does not occur on any forming path from s to s'. Indeed assume a forming path

$$s = s_0 \xrightarrow{\beta_0} s_1 \xrightarrow{\beta_1} \dots \xrightarrow{\beta_{n-1}} s_n = s'$$

on which α occurs. Let i be the minimal index such that $\alpha = \beta_i$. Then starting from $\alpha(s)$, the action sequence $\beta_0, \dots, \beta_{i-1}, \beta_{i+1}, \dots, \beta_{n-1}$ produces a forming path from $\alpha(s)$ to s' (see Figure 3.11). This contradicts the assumption that $\alpha(s) \not\rightsquigarrow s'$. Note

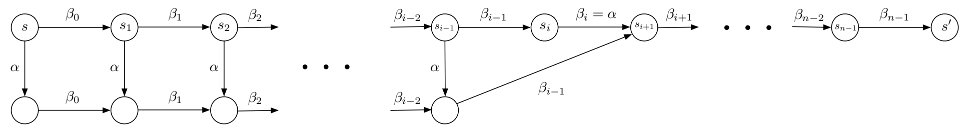


Figure 3.11: Illustration of Proposition 3.4.12, part (b)

that the singleton action-set $\{\beta_j\}$ satisfies the dependence condition (A2) for the state s_j and that $\alpha \neq \beta_j \in \mathsf{Act}(s_j)$ is independent to $\beta_j, j < i$. But then the singleton action-set $\{\beta_j\}$ also satisfies the dependence condition (A2) for the state $\alpha(s_j)$.

□

Note that part (a) of Proposition 3.4.12 applies to all actions $\alpha \in \mathsf{Act}(s)$ which are probabilistic or which are non-stutter actions. But there might also be non-probabilistic stutter actions α enabled in s that do not occur on at least one forming path from s to the given state s'.

Definition 3.4.13. [Relation $\mathcal{R}$]
Recall that ample sets are given that satisfy conditions (A0)-(A3) and (A4.2). As usual, S denotes the state space of $\mathcal{M}$ and $\hat{S} \subseteq S$ the state space of $\hat{\mathcal{M}}$. We define the relation $\mathcal{R}$ by

$$\mathcal{R} = \big\{(s,t) \in S \times \hat{S} : s \rightsquigarrow t \text{ in } \mathcal{M}\big\}.$$

In the following, a forming path in $\hat{\mathcal{M}}$ means a forming path $\hat{s}_0 \xrightarrow{\beta_0} \hat{s}_1 \xrightarrow{\beta_1} \dots \xrightarrow{\beta_{n-1}} \hat{s}_n$ as in Definition 3.4.11 where $\hat{s}_0, \hat{s}_1, \dots, \hat{s}_n \in \hat{S}$ and $\beta_i \in \mathsf{ample}(\hat{s}_i)$, $i = 0, 1, \dots, n-1$.

Proposition 3.4.14 (Forming paths in the reduced MDP)**.** Let $\hat{s}$ be a state in $\hat{\mathcal{M}}$.

(a) If $\hat{\pi}$ is a forming path in $\hat{\mathcal{M}}$ starting in state $\hat{s}$ and $(s, \hat{s}) \in \mathcal{R}$ then $(s, \hat{u}) \in \mathcal{R}$ for all states $\hat{u}$ in $\hat{\pi}$.

(b) If $\alpha \in \mathsf{Act}(\hat{s})$ then there exists a forming path $\hat{\pi}$ in $\hat{\mathcal{M}}$ from $\hat{s}$ to some state $\hat{u}$ such that $\alpha \in \mathsf{ample}(\hat{u})$ and $\delta(\hat{s}, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{u}, \alpha, \cdot)$.

Proof.

(a) This follows immediately from the transitivity of $\rightsquigarrow$.

(b) Part (b) can be derived from Proposition 3.4.12 as follows. Note that any finite path in $\hat{\mathcal{M}}$ where none of its states is fully expanded is a forming path (because of conditions (A1), (A2) and (A4.2)).

Thus, if there is a path in $\hat{\mathcal{M}}$ from $\hat{s}$ to some fully expanded state $\hat{t}$ then there is also a forming path $\hat{\pi}$ in $\hat{\mathcal{M}}$ from $\hat{s}$ to a fully expanded state $\hat{u}$ (let $\hat{u}$ be the first fully expanded state that appears on the path from $\hat{s}$ to $\hat{t}$). Let $\alpha \in \mathsf{Act}(\hat{s})$. If α does not occur in $\hat{\pi}$ then part (a) of Proposition 3.4.12 yields

$$\alpha \in \mathsf{Act}(\hat{u}) = \mathsf{ample}(\hat{u}) \text{ and } \delta(\hat{s}, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{u}, \alpha, \cdot).$$

If α appears in $\hat{\pi}$ then we consider the longest prefix $\hat{\sigma}$ of $\hat{\pi}$ where α does not occur. Let $\hat{v} = \mathsf{last}(\hat{\sigma})$. Then $\hat{\pi}$ has the form

$$\underbrace{\hat{s} \to \ldots \to \hat{v}}_{=\hat{\sigma}} \xrightarrow{\alpha} \ldots \to \hat{u}.$$

In particular, $\alpha \in \mathsf{ample}(\hat{v})$. Again, part (a) of Proposition 3.4.12 yields $\delta(\hat{s}, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{v}, \alpha, \cdot)$.

Now assume that there is no path in $\hat{\mathcal{M}}$ from $\hat{s}$ to some fully expanded state $\hat{t}$. This means that any finite path in $\hat{\mathcal{M}}$ that starts in $\hat{s}$ is a forming path. Let $\hat{\pi}$ be a path in $\hat{\mathcal{M}}$ that starts in $\hat{s}$ and has length $|\hat{S} + 1|$. Then $\hat{\pi}$ contains a cycle. If α appears in $\hat{\pi}$ then (as above) we consider the longest prefix $\hat{\sigma}$ of $\hat{\pi}$ where α does not occur. If $\hat{v} = \mathsf{last}(\hat{\sigma})$, then $\alpha \in \mathsf{ample}(\hat{v})$ and part (a) of Proposition 3.4.12 yields $\delta(\hat{s}, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{v}, \alpha, \cdot)$. If α does not occur in $\hat{\pi}$ then $\alpha \in \mathsf{Act}(\hat{t})$ and $\delta(\hat{s}, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{t}, \alpha, \cdot)$ for each state $\hat{t}$ on $\hat{\pi}$ (by part (a) of 3.4.12). Since $\hat{\pi}$ contains a cycle, the cycle condition (A3) ensure the existence of a state $\hat{u}$ on $\hat{\pi}$ such that $\alpha \in \mathsf{ample}(\hat{u})$ which shows the claim.

□

We can now prove that $\mathcal{M}$ and $\hat{\mathcal{M}}$ are probabilistic visible bisimilar.

Theorem 3.4.15 (**$\mathcal{M}$ and $\hat{\mathcal{M}}$ are probabilistic visible bisimilar).**
$\mathcal{R}$ is a probabilistic visible bisimulation for $(\mathcal{M}, \hat{\mathcal{M}})$.

Proof. Clearly, $\mu \sqsubseteq_{\mathcal{R}} \mu'$ and $\mu' \sqsubseteq_{\mathcal{R}^{-1}} \mu$. We show that for any $(s, \hat{s}) \in \mathcal{R}$ conditions (1)-(3) in Definition 3.4.9 hold, and conversely, that (1)-(3) are fulfilled for the "inverse" pair $(\hat{s}, s) \in \mathcal{R}^{-1}$.

(1) It is obvious that $L(s) = L(\hat{s})$ as all actions on a forming path are stutter actions. Thus, all states on a forming path have the same labeling.

(2) We first show that condition (2) in Definition 3.4.9 holds for $(s, \hat{s}) \in \mathcal{R}$. Let $\alpha \in \mathsf{Act}(s)$.

If α is a non-probabilistic stutter action and $s \xrightarrow{\alpha} t$ where $t \rightsquigarrow \hat{s}$ (and thus, $(t, \hat{s}) \in \mathcal{R}$) then we are in the situation of condition (2.1) in Definition 3.4.9.

Let us now assume that α is probabilistic or a non-stutter action or $s \xrightarrow{\alpha} t$ is a non-probabilistic stutter step where $t \not\rightsquigarrow \hat{s}$. If α is probabilistic or a non-stutter action then it cannot appear on any forming path from s to $\hat{s}$. Thus we may apply part (a) of Proposition 3.4.12. If $s \xrightarrow{\alpha} t$ is a non-probabilistic stutter step where $t \not\rightsquigarrow \hat{s}$, we may apply part (b) of Proposition 3.4.12. In either case this yields

$$\alpha \in \mathsf{Act}(\hat{s}) \text{ and } \delta(s, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{s}, \alpha, \cdot). \qquad (+)$$

As $\hat{s}$ is a state in the reduced MDP $\hat{\mathcal{M}}$, part (b) of Proposition 3.4.14 yields the existence of a forming path from $\hat{s}$ in the reduced MDP $\hat{\mathcal{M}}$ to some state $\hat{u}$ where $\alpha \in \mathsf{ample}(\hat{u})$ and $\delta(\hat{s}, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{u}, \alpha, \cdot)$. Hence, by (+) and the transitivity of $\sqsubseteq_{\rightsquigarrow}$ (see Proposition 3.4.8) we obtain:

$$\alpha \in \mathsf{Act}(\hat{u}) \text{ and } \delta(s, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{u}, \alpha, \cdot). \qquad (++)$$

We may compose the forming path in $\mathcal{M}$ from s to $\hat{s}$ (which exists as $s \rightsquigarrow \hat{s}$) with the forming path $\hat{\pi}$ from $\hat{s}$ to $\hat{u}$ in $\hat{\mathcal{M}}$ and thus obtain $s \rightsquigarrow \hat{u}$. As $\hat{u} \in \hat{S}$, we get $(s, \hat{u}) \in \mathcal{R}$. By part (a) of Proposition 3.4.14 we get $(s, \hat{v}) \in \mathcal{R}$ for all states $\hat{v}$ in $\hat{\pi}$. Thus, (++) yields that we are in the situation of condition (2.2) in Definition 3.4.9.

Let us now consider the inverse pair $(\hat{s}, s) \in \mathcal{R}^{-1} \subseteq \hat{S} \times S$ and an action $\alpha \in \mathsf{ample}(\hat{s})$. As $(\hat{s}, s) \in \mathcal{R}^{-1}$ we have $s \rightsquigarrow \hat{s}$. Thus there is a forming path

$$s \xrightarrow{\beta_0} s_1 \xrightarrow{\beta_1} s_2 \ldots \xrightarrow{\beta_{n-2}} s_{n-1} \xrightarrow{\beta_{n-1}} \hat{s}$$

from s to $\hat{s}$. But then $s_i \xrightarrow{\beta_i} s_{i+1} \ldots \xrightarrow{\beta_{n-2}} s_{n-1} \xrightarrow{\beta_{n-1}} \hat{s}$ is a forming path for all $i = 0, 1, \ldots, n$ (where $s = s_0$ and $\hat{s} = s_n$). Therefore $(\hat{s}, s_i) \in \mathcal{R}^{-1}$ for $i = 0, 1, \ldots, n$. Using the trivial fact that $\delta(\hat{s}, \alpha, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{s}, \alpha, \cdot)$ we are in the case of condition (2.2) in Definition 3.4.9

(3) Let us assume that there is an infinite path of the form $s = s_0 \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \ldots$ in $\mathcal{M}$ consisting of non-probabilistic stutter actions $\alpha_0, \alpha_1, \alpha_2, \ldots$ and such that $(s_i, \hat{s}) \in \mathcal{R}$, $i = 0, 1, 2, \ldots$. We distinguish two cases.

Case (i): $\hat{s}$ is not fully expanded. Then by (A4.2) $\mathsf{ample}(\hat{s}) = \{\gamma\}$ for some non-probabilistic stutter action γ. Let $\hat{s}'$ be the unique successor of γ in $\hat{s}$. As the singleton action set $\{\gamma\}$ satisfies condition (A2) in state $\hat{s}$ (since $\mathsf{ample}(\hat{s}) = \{\gamma\}$), it follows that $\hat{s} \xrightarrow{\gamma} \hat{s}'$ is a forming path. Thus, $\hat{s} \rightsquigarrow \hat{s}'$ and as $s \rightsquigarrow \hat{s}$ and $\hat{s}' \in \hat{S}$, we obtain $(s, \hat{s}') \in \mathcal{R}$. We already showed that condition 2) of Definition 3.4.9 holds for $\mathcal{R}$. As $\alpha_0 \in \mathsf{Act}(s)$, condition 2) applied to $(s, \hat{s}') \in \mathcal{R}$ yields either

(2.1) α_0 is a non-probabilistic stutter action such that $(s_1, \hat{s}') \in \mathcal{R}$, but then the path $\hat{s} \xrightarrow{\gamma} \hat{s}'$ shows that condition 3) holds (with k=1). Or it yields that

(2.2) there is a finite path of the form $\hat{s}' = \hat{s}_0 \xrightarrow{\gamma_0} \hat{s}_1 \xrightarrow{\gamma_1} \ldots \xrightarrow{\gamma_{n-1}} \hat{s}_n$ in $\hat{\mathcal{M}}$ such that

- $\gamma_0, \ldots, \gamma_{n-1}$ are non-probabilistic stutter actions,
- $(s, \hat{s}_i) \in \mathcal{R}$ for $0 \leq i \leq n$,
- $\alpha_0 \in \mathsf{ample}(\hat{s}_n)$ and $\delta(s, \alpha_0, \cdot) \sqsubseteq_{\rightsquigarrow} \hat{\delta}(\hat{s}_n, \alpha_0, \cdot)$.

Observe, that $\mathsf{ample}(\hat{s}_i) = \{\gamma_i\}, i = 0, \ldots, n-1$ and thus $\hat{s}' = \hat{s}_0 \xrightarrow{\gamma_0} \hat{s}_1 \xrightarrow{\gamma_1} \ldots \xrightarrow{\gamma_{n-1}} \hat{s}_n$ is a forming path. Therefore $s \rightsquigarrow \hat{s}_i$ and $(s, \hat{s}_i) \in \mathcal{R}, i = 0, \ldots, n$.

Then the path

$$\hat{s} \xrightarrow{\gamma} \hat{s}' \xrightarrow{\gamma_0} \hat{s}_1 \xrightarrow{\gamma_1} \hat{s}_2 \ldots \xrightarrow{\gamma_{n-1}} \hat{s}_n \xrightarrow{\alpha_0} \alpha_0(\hat{s_n})$$

shows condition 3) (with k=1). Note that $s_1 = \alpha_0(s_0) \rightsquigarrow \alpha_0(\hat{s}_n)$ since $\delta(s, \alpha_0, \cdot) \sqsubseteq_{\rightsquigarrow} \hat{\delta}(\hat{s}_n, \alpha_0, \cdot)$ and α_0 is a non-probabilistic action.

Case (ii): $\hat{s}$ is fully expanded. We claim that there is an index $j \geq 0$, such that α_j does not appear on some forming path from s_j to $\hat{s}$. This can be seen as follows. Let

$$s \xrightarrow{\beta_0} t_1 \xrightarrow{\beta_1} t_1 \ldots \xrightarrow{\beta_{n-1}} t_{n-1} = \hat{s}$$

be a forming path from s to $\hat{s}$. Assume that α_0 appears on this path, say $\alpha_0 \notin \{\beta_0, \ldots, \beta_{i-1}\}$ and $\alpha_0 = \beta_i, i \leq n-1$. Then the path from $s_1 = \alpha_0(s_0)$ to $\hat{s}$ that follows the action sequence $\beta_0, \beta_1, \ldots, \beta_{i-1}, \beta_{i+1}, \beta_{i+2}, \ldots, \beta_{n-1}$ is a forming path. See Figure 3.11 on page 55. Thus there is a forming path from s_1 to $\hat{s}$ that is shorter than the forming path $s \xrightarrow{\beta_0} t_0 \xrightarrow{\beta_1} t_1 \ldots \xrightarrow{\beta_{n-1}} t_{n-1} = \hat{s}$. If α_1 appears on this forming path from s_1 to $\hat{s}$ we can construct an even shorter forming path from s_2 to $\hat{s}$. As we started with a forming path of length $n-1$, this shows that there is an index j such that α_j does not appear on some forming path from s_j to $\hat{s}$. By Proposition 3.4.12, part (a), we obtain that $\alpha_j \in \mathsf{Act}(\hat{s}) = \mathsf{ample}(\hat{s})$ and that $\delta(s_j, \alpha_j, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{s}, \alpha_j, \cdot)$. As α_j is a non-probabilistic action, $\delta(s_j, \alpha_j, \cdot) \sqsubseteq_{\rightsquigarrow} \delta(\hat{s}, \alpha_j, \cdot)$ implies that $s_{j+1} = \alpha_j(s_j) \rightsquigarrow \alpha_j(\hat{s})$. Since $\alpha_j \in \mathsf{ample}(\hat{s})$, the path $\hat{s} \xrightarrow{\alpha_j} \alpha_j(\hat{s})$ shows condition 3) (with k=j+1).

Let us now consider the inverse pair $(\hat{s}, s) \in \mathcal{R}^{-1} \subseteq \hat{S} \times S$ and assume that there is an infinite path

$$\hat{s} = \hat{s}_0 \xrightarrow{\alpha_0} \hat{s}_1 \xrightarrow{\alpha_1} \hat{s}_2 \xrightarrow{\alpha_2} \ldots$$

in $\hat{\mathcal{M}}$ such that the α_i's are non-probabilistic stutter actions and $(\hat{s}_i, s) \in \mathcal{R}^{-1}, i = 0, 1, \ldots$. Thus there is a forming path from s to $\hat{s}_1$. Let $s \xrightarrow{\beta} s_1$ be the first step on such a forming path. Then $s_1 \rightsquigarrow \hat{s}_1 \in \hat{S}$, so $(\hat{s}_1, s_1) \in \mathcal{R}^{-1}$. Therefore the path $s \xrightarrow{\beta} s_1$ shows condition 3) (with k=1).

□

Theorem 3.4.15 together with Proposition 3.4.10 completes the proof of our main result stated in Theorem 3.4.6.

3.5. Partial order reduction versus process equivalences

In this section we give a brief overview of the connections between different partial order reduction criteria and probabilistic process equivalences. With suitable notions of stutter equivalence, simulation and bisimulation equivalence we obtain:

(a) If conditions (A0)-(A3) and (A4) hold, then $\mathcal{M}$ and $\hat{\mathcal{M}}$ are stutter equivalent (see section 3.3), but in general $\hat{\mathcal{M}}$ does not simulate $\mathcal{M}$.

(b) If conditions (A0)-(A3) and (A4.1) hold, then $\mathcal{M}$ and $\hat{\mathcal{M}}$ are simulation equivalent (see [DN04] and Remark 3.4.3, page 50), but in general not bisimilar.

(c) If conditions (A0)-(A3) and (A4.2) hold, then $\mathcal{M}$ and $\hat{\mathcal{M}}$ are bisimilar (see section 3.4).

The underlying notion of stutter equivalence essentially agrees with trace distribution equivalence [Seg95] (reformulated for our model and state labels rather than action labels). The underlying simulation relation has been formally defined in [DN04] and is a variant of probabilistic forward simulation as introduced by Segala [Seg95]. This kind of simulation allows a state s to be simulated by a distribution over states (rather than a single state). The underlying notion of bisimulation is probabilistic visible bisimulation (as defined in Definition 3.4.9) and could also be replaced with a divergence-sensitive, state-based variant of probabilistic branching bisimulation defined in the style of [SL95].

We will now give examples for the statements in (a) and (b). The example in Figure 3.8 shows a reduction satisfying (A0)-(A3) and (A4) where $\hat{\mathcal{M}} = \hat{\mathcal{T}}$ does not simulate $\mathcal{M} = \mathcal{T}$ (as stated in (a)). Here $\mathcal{M}$ and $\hat{\mathcal{M}}$ do not contain probabilistic actions, and hence can be viewed as ordinary transition systems. The intuitive argument why $\hat{\mathcal{M}}$ does not simulate $\mathcal{M}$ is that there is no possibility to mimic the nondeterministic choice of $\mathcal{M}$ in state t_1 via a probabilistic choice over the states t_2 and t_3 in $\hat{\mathcal{M}}$. Note that a scheduler for $\mathcal{M}$ in state t_1 might choose β and γ (and thus combine the states t_2 and t_3) with arbitrary probabilities while a probabilistic forward simulation would require a *fixed* probability distribution over the states t_2 and t_3 in $\hat{\mathcal{M}}$ to mimic the possible behaviors of t_1 in $\mathcal{M}$, which is not possible.

The example in Figure 3.9 shows a reduction satisfying (A0)-(A3) and (A4.1) where $\mathcal{M}$ and $\hat{\mathcal{M}}$ are not bisimilar (as stated in (b)), since there is *no state* in $\hat{\mathcal{M}}$ that corresponds to state t_1 in $\mathcal{M}$. But $\mathcal{M}$ and $\hat{\mathcal{M}}$ are simulation equivalent as the state t_1 in $\mathcal{M}$ is simulated by the probability distribution $\frac{1}{3} \cdot t_2 + \frac{2}{3} \cdot t_3$ in $\hat{\mathcal{M}}$.

Please note that for the simulation equivalence as in (b) [DN04] required a stronger underlying structure for the given MDP. They assumed each action to have a fixed probabilistic branching pattern. Moreover they assumed the stronger cycle condition (A3) and not the end component condition. Using the end component condition (A3) would require a notion of probabilistic forward simulation that allows for (certain) infinite computations to simulate a single transition, while for the cycle condition (A3) a simpler version of simulation suffices where any transition of the simulated process has to be matched by a finite computation tree of the simulating process. The approach of [DN04] works with the cycle condition (A3) and a formalization of finite computation trees by means of SOS-rules. Yet, to deal with the end component condition (A3) and possibly infinite computation trees a further rule that captures the semantics of infinite behaviors could be added.

3.6. Conclusion

We extended a certain partial order reduction approach, namely the ample set method to Markov decision processes. We showed ample set conditions such that

a) a given MDP and its reduced MDP are stutter equivalent.

b) a given MDP and its reduced MDP are probabilistic visible bisimilar.

The main ingredient for a) was to introduce a branching condition. The need for such a branching condition can intuitively be justified by concluding that the experiments

"first toss a coin, then decide between action α and β"
and "first decide between action α and β, then toss a coin"

are different. For b), the branching condition had to be strengthened. The extension of the classical conditions is conservative in both cases, which means that if the new stronger conditions are applied to an ordinary (non-probabilistic) transition system, then they are equivalent to the classical ample set conditions. The proof techniques for the soundness (for both a) and b)) are similar to the ones used in the non-probabilistic setting, but are in detail much more complicated as the probabilism comes into play, especially for a), where we have to reason about schedulers instead of single paths. b) allowed for a "simpler" proof since a "local reasoning" by means of bisimulation is possible.

The presented results make the ample set method applicable to probabilistic model checking and therefore allow to only analyze a fragment of the state space rather than the full state space. We followed here the approach of the ample sets, but we expect that also the concepts of persistent [God96] or stubborn [Val90, Val94] sets could be adapted for the probabilistic case. In [GNB+06], the results presented in this chapter have been extended such that the ample set method also applies to probabilistic reward models in combination with a modification of PCTL that allows to reason about the rewards. Concerning quantitative LTL model checking with fairness, we expect that the established results can be extended such that the ample set method is applicable.

The probabilistic partial order reduction has been implemented in the quantitative LTL model checker LiQuor [BCG05, BC06a] which has been developed in our workgroup by Frank Ciesinski. The implementation uses similar heuristics as the (non-probabilistic) LTL model checking tool SPIN [Hol03]. Below we state some experimental results that we obtained using LiQuor, where our focus is on the benefits of the partial order reduction. We investigated the "Dining Cryptographers", a randomized gossiping protocol and a randomized Leader Election protocol. The second column in the table below indicates the number of processes (e.g. cryptographers) involved.

		$\mathcal{M}$		$\hat{\mathcal{M}}$	
Scenario	#	# of states	# of transitions	# of states	# of transitions
Dining Crypt.	3	9865	36544	6748	20916
Dining Crypt.	4	96753	415687	81817	306236
Dining Crypt.	5	$1.56 \cdot 10^6$	$6.32 \cdot 10^6$	746556	$1.54 \cdot 10^6$
gossiping protocol	3	4015	5298	403	488
gossiping protocol	4	488902	661307	4424	5380
gossiping protocol	5	n/a	n/a	74485	90998
Leader Election	4	53621	156072	21063	78072
Leader Election	5	896231	$3.2 \cdot 10^6$	299670	$1.3 \cdot 10^6$
Leader Election	6	$1.1 \cdot 10^7$	$6.2 \cdot 10^7$	$4.1 \cdot 10^6$	$1.4 \cdot 10^7$

Unfortunately, an on-the-fly technique that combines the construction of the reduced MDP with the verification algorithm, as realized in SPIN, seems to be difficult for a quantitative analysis, as the latter requires solving linear programs rather than performing a cycle-search. Although the construction of the reduced MDP can be performed using the same techniques as for non-probabilistic systems, the question arises whether special algorithms can be developed that generate better (i.e. smaller) ample sets using the topological characteristics of end components (rather than cycles). For detailed implementation issues and more experimental results see the forthcoming PhD thesis of Frank Ciesinski.

4 Probabilistic ω-Automata

We introduce probabilistic variants of ω-automata that serve as acceptors for languages over infinite words. The essential idea is to equip nondeterministic ω-automata with probabilistic distributions that resolve the nondeterministic choices and to define the acceptance of an infinite input word by the requirement that the set of accepting runs has a positive probability measure. Although probabilistic finite automata (PFA) have attracted many researchers, see e.g. [Rab63, Paz66, Fre81, MHC03, DS90, BC03], we are not aware of any paper that deals with probabilistic language acceptors for infinite words and defines the acceptance criterion for an infinite word by means of the probability measure of the accepting runs. There is a wide range of publications where probabilistic automata are used for modeling randomized behaviors, see e.g. [vGSST90, Seg95, Bea02]. These approachs are opposed to our setting as we aim at a probabilistic formalism to represent (non-probabilistic) languages over infinite words. As far as we know, the paper [Rei98] by Reisz is the only approach where probabilistic automata have been used for the recognition of infinite words. However, in the approach of Reisz, an infinite word is accepted if and only if there exists an infinite accepting run for this word with positive probability. With this acceptance criterion, the concept of probabilistic Büchi automata is rather close to nondeterministic Büchi automata that are deterministic in limit [CY95] and, in our opinion, less interesting than probabilistic language acceptors where the probability measure of the set of accepting runs serves as the acceptance criterion.

Potential application areas of probabilistic ω-automata might be any research topic where ω-regular languages are of importance, such as the verification of reactive systems [Var96] or reasoning with biological processes [FOS03]. Given the wide range of applications of PFA (e.g., for speech recognition [RST96], Arthur-Merlin games [BM88], planning questions in Markov decision processes [MHC99, BT00], or prediction of climatic parameters [MLMdCT02]), probabilistic ω-automata might be of interest also for other areas. In addition, probabilistic ω-automata could serve as basis for "quantum ω-automata" (in analogy to quantum finite automata [KW97, AF98] which can be regarded as an extension of PFA) or they might be useful in combination with costs as in [DK03, DP04] where ω-variants of weighted automata are studied. The established results on PBA are also relevant for partial information games with ω-regular winning objectives [CDHR06] as well as POMDPs [Son71, Mon82, PT87, Lov91], which are used to model a wide range of applications. PBA also find an application in randomized monitoring [CSV08].

In this chapter we will investigate the concept of probabilistic ω-automata. This includes several aspects as the expressiveness and the efficiency to represent ω-languages as well as composition operators and the emptiness problem. We will also study an alternative semantics, the so-called almost-sure semantics.

4.1. Introduction to probabilistic ω-automata

In this section we introduce probabilistic ω-automata which can be viewed as nondeterministic ω-automata where the nondeterminism is resolved by a probabilistic choice. That is, for any state p and letter $a \in \Sigma$ either p does not have any a-successor or there is a probability distribution for the a-successors of p. We start our discussion with an acceptance condition that requires that the probability measure of the accepting runs of an input word is positive in order to accept the word.

Probabilistic ω-automata have also been defined in [Rei98, Rei99], but we use a different syntax and semantics as introduced in [BG05]. In the approach of Reisz, an infinite word is accepted if and only if there exists an infinite accepting run for this word with positive probability.

Throughout this chapter, PFA, NFA and DFA denote probabilistic, nondeterministic and deterministic finite automata, which serve as acceptors for languages over *finite* words.

4.1.1. Definition of probabilistic ω-automata

Definition 4.1.1. [Probabilistic ω-automata]
A probabilistic ω-automaton is a tuple

$$\mathcal{P} = (Q, \Sigma, \delta, \mu_0, \mathsf{Acc}),$$

where

- Q is a finite nonempty set of states,
- Σ is a finite nonempty input alphabet,
- $\delta : Q \times \Sigma \times Q \to [0, 1]$ is a transition probability function such that for all $p \in Q$ and $a \in \Sigma$, either $\delta(p, a, .)$ is a probability distribution on Q or $\delta(p, a, .)$ is the null-function (i.e. $\delta(p, a, q) = 0$ for all $q \in Q$),
- μ_0 is a probability distribution on Q (called the *initial* distribution) and
- Acc is an *acceptance condition*.

Apparently, a probabilistic ω-automaton $\mathcal{P}$ is an MDP equipped with an acceptance condition. We will use the notation $\mathcal{P}$ also to denote only the underlying MDP of the automaton. We call the automaton *total* if the underlying MDP is total. The operational behavior of $\mathcal{P}$ for a given input word $\omega = a_1 a_2 \ldots \in \Sigma^\omega$ is as follows. The automaton chooses at random an initial state p_0 according to the initial distribution μ_0. After having consumed the first i input symbols $a_1, \ldots, a_i$, $\mathcal{P}$ in state p_i moves with probability $\delta(p_i, a_{i+1}, p)$ to state p and tries to read the next input symbol a_{i+2} in state p. If there is no outgoing a_{i+1}-transition from the current state p_i, i.e. if $a_{i+1} \notin \mathsf{Act}(p_i)$, then $\mathcal{P}$ rejects. As for nondeterministic automata, a resulting infinite state-sequence $p_0, p_1, \ldots$ is called a run for ω in $\mathcal{P}$. This behavior interprets the input word as a scheduler. Given an input word ω we define the scheduler $\mathcal{U}(\omega)$ such that $\mathcal{U}(\omega)(p_0, \dots, p_n) = \omega_{n+1}$, that is in step i, the scheduler chooses the letter ω_i as the next action. Then the operational behavior of $\mathcal{P}$ reading the input word ω, is formalized by the Markov chain $\mathcal{P}_{\mathcal{U}(\omega)}$. In contrast to nondeterministic automata, where an input word ω is accepted by the automaton, if there exists an accepting run for ω, we require for probabilistic automata that the set of accepting runs for ω has a positive measure.

The *accepted language* of a probabilistic ω-automaton $\mathcal{P}$ is thus defined as

$$\mathcal{L}(\mathcal{P}) = \Big\{\omega \in \Sigma^\omega \mid \ \mathsf{Pr}^{\mathcal{P},\mathcal{U}(\omega)}(\{\pi \in \mathsf{Path}^{\mathcal{P}}_{\mathsf{inf}} \mid \mathsf{inf}(\pi) \text{ is accepting}\}) > 0\Big\}.$$

By the results of [Var85, CY95] the set of accepting runs for ω is measurable when dealing with Büchi-, Rabin- or Streett-acceptance.

Throughout this thesis we will identify an input word ω with its associated scheduler $\mathcal{U}(\omega)$, thus we will write $\mathsf{Pr}^{\mathcal{U},\omega}(.)$ instead of $\mathsf{Pr}^{\mathcal{U},\mathcal{U}(\omega)}(.)$. For the sake of convenience we also fix the following notation for the acceptance probability of a word ω and a given probabilistic ω-automaton $\mathcal{P}$.

$$\mathsf{Pr}^{\mathcal{P}}(\omega) := \mathsf{Pr}^{\mathcal{P},\mathcal{U}(\omega)}(\{\pi \in \mathsf{Path}^{\mathcal{P}}_{\mathsf{inf}} \mid \mathsf{inf}(\pi) \text{ is accepting}\})$$

Note that equipped with the equivalence relation $\sim = Q \times Q$, $\{\mathcal{U}(\omega) \mid \omega \in \Sigma^\omega\} = \mathsf{Sched}^{(\mathcal{P},\sim)}_{\mathsf{HD}}$. Thus when restricting to deterministic observation-based schedulers, a probabilistic automaton can be seen as a POMDP. Notice that w.l.o.g. we can assume a probabilistic automaton to be total.

Let a probabilistic ω-automaton $\mathcal{P}$ and an input word ω be given. Given an end component (T, A) we denote by

$$\mathsf{Pr}^{\mathcal{P},\omega}((T, A)) \ := \ \mathsf{Pr}^{\mathcal{P},\omega}\{\pi \in \mathsf{Path}^{\mathcal{P}}_{\mathsf{inf}} \mid \mathsf{Lim}(\pi) = (T, A)\}$$

the probability of all paths π such that $\mathsf{Lim}(\pi) = (T, A)$. We call the end component (T, A) *accepting* if the state set T is accepting (w.r.t. the acceptance condition of $\mathcal{P}$).

Recall Lemma 2.2.13 stating that given an MDP and a scheduler, almost all runs form an end-component in their limit. Thus the acceptance probability $\mathsf{Pr}(\omega)$ agrees with the probability measure of the set of runs π for ω such that $\mathsf{Lim}(\pi)$ is an accepting end component. As $\mathcal{P}$ has only finitely many end components this yields the following lemma.

Lemma 4.1.2 (AEC-Lemma). For any probabilistic ω-automaton $\mathcal{P}$ and any input word ω, it holds that $\omega \in \mathcal{L}(\mathcal{P})$ if and only if $\mathsf{Pr}^{\mathcal{P},\omega}((T, A)) > 0$ for some accepting end component (T, A).

Notation 4.1.3. We will sometimes abuse the notion of the transition probability function using it as in the case of non-probabilistic automata. This means that we will write $\delta(s, a)$ instead of $\mathsf{supp}(\delta(s, a, .))$, thus $\delta(s, a) = \{t \mid \delta(s, a, t) > 0\}$. Generalizing the input letter to a finite word ρ we define $\delta(s, \rho) = \{t \mid \exists u \ : \ u \in \delta(s, \rho\uparrow^{|\rho|-1})$ and $t \in \delta(u, \mathsf{last}(\rho))\}$. At last we define $\delta(S, \rho) = \bigcup_{s \in S} \delta(s, \rho)$ for a set of states $S \subseteq Q$.

4.1.2. Examples of probabilistic Büchi automata

We will now start out with a few examples for probabilistic Büchi automata (PBA). Given a PBA, intuitively, $\mathsf{Pr}(\omega)$ denotes the probability for the event "infinitely often F" under the scheduling policy induced by ω, that is $\mathsf{Pr}^{\mathcal{P}}(\omega) = \mathsf{Pr}^{\mathcal{P},\omega}(\Box\Diamond F)$. In the pictures of PBA, we use boxes to denote the accepting states and circles for the non-accepting states. We might simply write a as label for a transition from p to q if $\delta(p, a, q) = 1$. Label a, x with $x \in]0, 1[$ for a transition from p to q denotes that $\delta(p, a, q) = x$. An initial state will be indicated through an incoming edge, labeled with the initial probability of the state. Again, if the probability is 1, we might omit it.

Note that the accepted language of a PBA is included in the language that is accepted by the naive associated NBA, i.e. the NBA that stems from the given PBA by ignoring the probabilities.

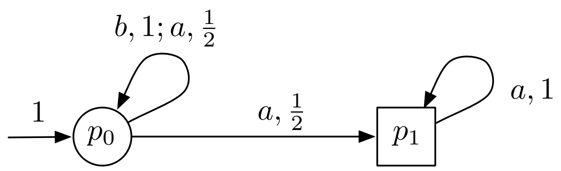


Figure 4.1: PBA for the language $(a + b)^*a^\omega$

Example 4.1.4 (PBA)**.** Figure 4.1 shows a PBA that accepts the language $(a+b)^*a^\omega$. To see this, we first notice that only the words in $(a + b)^*a^\omega$ have an accepting run, because the a-labeled self-loop in the accepting state p_1 is the only outgoing transition of state p_1. On the other hand, $\mathsf{Pr}(a^\omega) = 1$ (as the non-accepting run $p_0, p_0, p_0, \ldots$ has probability 0 while all other runs for a^ω are accepting). For any word $\omega \in (a + b)^n ba^\omega$, the acceptance probability is at least $(\frac{1}{2})^n$ as the the set of accepting runs for ω is equal to the set $\Delta((p_0)^{n+2}) \setminus \{p_0^\omega\}$ and given any word $\omega \in (a + b)^n ba^\omega$, it holds that $\mathsf{Pr}(\omega)(\Delta((p_0)^{n+2}) \setminus \{p_0^\omega\}) \geq (\frac{1}{2})^n$.

Clearly, any DBA can be viewed as a PBA, where $\delta_{\mathsf{PBA}}(p, a, q) = 1$ if $\delta_{\mathsf{DBA}}(p, a) = \{q\}$ and $\mu_0^{\mathsf{PBA}} = \mu_{q_0}^1$. On the other hand it is well known that the language $(a+b)^*a^\omega$ cannot be described by a DBA, thus the above example shows that PBA are strictly more expressive than DBA. It is worth mentioning that the qualitative criteria "accepting runs have positive probability" is different from the acceptance criteria "there is an accepting run" in the context of languages of *infinite* words, while they agree for probabilistic automata viewed as acceptors for *finite* words. In fact, the naive transformation from PBA to NBA which relies on ignoring the probabilities, in general fails to yield an equivalent NBA. Consider for example the automaton $\mathcal{P}_1$ on the left of Figure 4.2. Its underlying NBA (that we obtain by ignoring the probabilities) accepts the language $\big((ac)^*ab\big)^\omega$ whereas the automaton $\mathcal{P}_1$ accepts the language $(ab+ac)^*(ab)^\omega$. The intuitive argument why any word ω in $(ab+ac)^\omega$ with infinitely many c's is rejected relies on the observation that almost all runs for ω are finite and end in state p_1 (where the next input symbol is c and cannot be consumed in state p_1). [1]

[1]The formulation "almost all runs have property x" means that the probability measure of the runs where property x does not hold is 0.

Another example is the PBA $\mathcal{P}_2$ on the right of Figure 4.2. It accepts the empty language as any infinite word in $(ab + ac)^\omega$ has exactly one accepting run in $\mathcal{P}_2$, but its probability is 0. However, the underlying NBA accepts the language $(ab + ac)^\omega$.

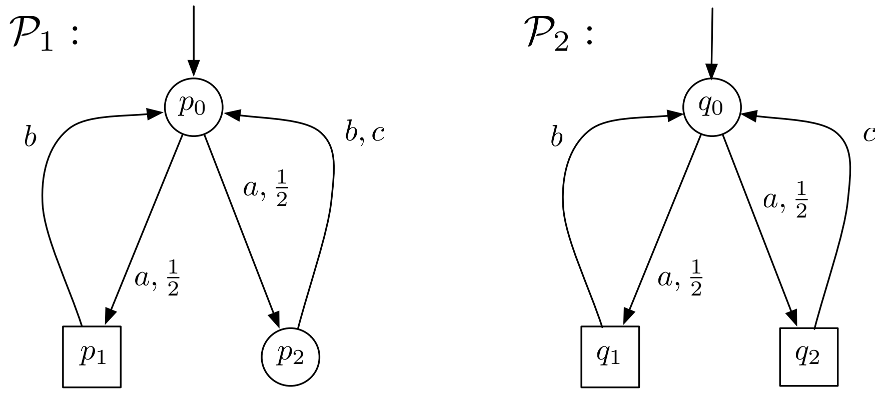


Figure 4.2: PBA for $(ab + ac)^*(ab)^\omega$ and for $\emptyset$

4.2. A closer look

Having introduced a few simple examples of PBA in the last section, we will now examine PBA a little closer.

4.2.1. Expressiveness of PBA

At first we establish the result stating that the class of languages that can be accepted by a PBA strictly contains the class of ω-regular languages.

Theorem 4.2.1 (PBA are strictly more expressive than ω-regular languages).

$$\mathbb{L}(\omega\text{- reg}) \subsetneq \mathbb{L}(\mathsf{PBA})$$

The proof of Theorem 4.2.1 is split into two parts. In Lemma 4.2.2 we show that for any NBA $\mathcal{A}$ there exists a PBA $\mathcal{P}$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{A})$. Then, in Lemma 4.2.3 we provide an example of a PBA for which the accepted language is not ω-regular.

Following [CY95] we call an NBA $\mathcal{A}$ *deterministic in limit* if $|\delta(p, a)| \leq 1$ for any state p that is reachable from an accepting state $q \in F$ and any symbol $a \in \Sigma$. If we regard an NBA $\mathcal{A}$ that is deterministic in limit as a PBA $\mathcal{P}$ (with arbitrary probability distributions to resolve the nondeterministic choices) then $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{P})$. [CY95] provided a transformation from a given NBA $\mathcal{A}$ into an equivalent NBA that is deterministic in limit and whose size is (single) exponential in $|\mathcal{A}|$. This yields the proof of the following lemma.

Lemma 4.2.2 (From NBA to PBA). For any NBA $\mathcal{A}$ there is a PBA $\mathcal{P}$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{A})$ and $|\mathcal{P}| = \mathcal{O}(\exp(|\mathcal{A}|))$.

It remains to provide an example for a PBA that recognizes a language that is not ω-regular.

Lemma 4.2.3 (Example for a PBA that accepts a non ω-regular language). The PBA depicted in Figure 4.3 accepts a language that is not ω-regular.

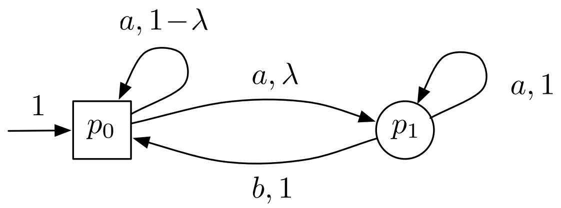


Figure 4.3: PBA $\mathcal{P}_\lambda$ $(0 < \lambda < 1)$ that accepts a non-ω-regular language

Proof. The PBA $\mathcal{P}_\lambda$ depicted in Fig. 4.3 recognizes the language $\mathcal{L}(\mathcal{P}_\lambda) = L_\lambda$ where

$$L_\lambda \;=\; \big\{a^{k_1}ba^{k_2}ba^{k_3}b\ldots : k_0, k_1, \ldots \in \mathbb{N}_{\geq 1} \text{ such that } \prod_{i=1}^{\infty}(1-(1-\lambda)^{k_i}) > 0\big\}.$$

Note that $\mathcal{L}(\mathcal{P}_\lambda) \subseteq (a^+b)^\omega$ as every accepting run for an infinite word ω that has only finitely many b's has to stay in state p_0 from some point on. But such runs have probability 0. Let $\omega = a^{k_1}ba^{k_2}\ldots \in (a^+b)^\omega$. Starting in p_0 and reading the first k_1 letters a, the automaton reaches state p_0 with probability $(1-\lambda)^{k_1}$ and thus state p_1 with probability $1-(1-\lambda)^{k_1}$. Reading the first b the automaton thus rejects with probability $(1-\lambda)^{k_1}$ and carries on to read the input word with probability $1-(1-\lambda)^{k_1}$. This shows that the probability not to reject while reading the word ω is $\prod_{i=0}^{\infty}\big(1-(1-\lambda)^{k_i}\big)$ and moreover this agrees with the probability to visit infinitely often the final state p_0. Therefore $\mathcal{L}(\mathcal{P}_\lambda) = L_\lambda$.

We now show that $\mathcal{L}(\mathcal{P}_\lambda)$ is not ω-regular. Otherwise, there exists an NBA $\mathcal{A}$ with $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{P}_\lambda)$. Then $\mathcal{A}$ has a reachable accepting cycle

$$q_0 \xrightarrow{a_1} \ldots \xrightarrow{a_n} q_n = q_0.$$

(An accepting cycle means a cycle in $\mathcal{A}$ that contains at least one accepting state.) Then, there is at least one index $1 \leq j \leq n$ such that $a_j = b$. W.l.o.g. (cyclic permutation) assume $j = n$. Hence, $a_1a_2\ldots a_n$ is a word of the form $a^{j_1}ba^{j_2}b\ldots a^{j_k}b$. But then, $\mathcal{A}$ accepts a word of the form

$$(a^+b)^*\left(a^{j_1}ba^{j_2}b\ldots a^{j_k}b\right)^\omega,$$

which contradicts the assumption $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{P}_\lambda)$ as no such word is accepted by $\mathcal{P}_\lambda$. This shows that $\mathcal{P}_\lambda$ accepts a non-ω-regular language. □

Remark 4.2.4 (Non-regular convergence conditions)**.** The non-regular convergence condition for the words accepted by the PBA $\mathcal{P}_\lambda$ in Figure 4.3 can be explained by the observation that there are finite input words such that $\mathcal{P}_\lambda$ rejects with arbitrary small probability while reading those. More precisely, when $\mathcal{P}_\lambda$ tries to read a finite word a^kb in state p_0 then $\mathcal{P}_\lambda$

fails to consume the last letter b (i.e. rejects) with probability $(1-\lambda)^k$. If k tends to infinity, the rejecting probability $(1-\lambda)^k$ tends to 0.

Similarly, there are PBA and infinite input words that have accepting runs in the underlying nondeterministic automaton, while the probabilities for the run fragments connecting two accepting states tend to zero.

For example, we regard the PBA $\widetilde{\mathcal{P}}_\lambda$ shown in Figure 4.4. It accepts the following language

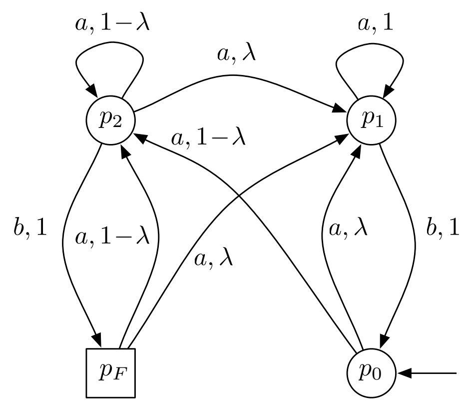


Figure 4.4: PBA $\widetilde{\mathcal{P}}_\lambda$ $(0 < \lambda < 1)$ that accepts a non-ω-regular language

$$\widetilde{L}_\lambda = \big\{a^{k_1}ba^{k_2}ba^{k_3}b\ldots \mid k_1, k_2, k_3 \ldots \in \mathbb{N}_{\geq 1} \text{ s.t. } \prod_{i=1}^{\infty}(1-(1-\lambda)^{k_i}) = 0\big\}.$$

$\widetilde{L}_\lambda$ is thus roughly the complement of the language accepted by the PBA $\mathcal{P}_\lambda$ shown in Figure 4.3. More precisely, it holds that $\widetilde{L}_\lambda = (a^+b)^\omega \setminus \mathcal{L}(\mathcal{P}_\lambda)$. Hence, $\widetilde{\mathcal{P}}_\lambda$ combined with a PBA for $(a+b)^*a^\omega$, $b(a+b)^\omega$ and $(a+b)^*bb(a+b)^\omega$ yields a PBA that recognizes the complement of $\mathcal{L}(\mathcal{P}_\lambda)$. This yields immediately that $\widetilde{\mathcal{P}}_\lambda$ accepts a non-ω-regular language.

We now check that $\widetilde{\mathcal{P}}_\lambda$ is indeed $\widetilde{L}$. Starting in p_0 (or p_F), $(1-\lambda)^{k_i}$ is the probability to be in p_2 after reading the word a^{k_i}. Hence, $1-(1-\lambda)^{k_i}$ represents the probability to be in state p_1 after the input word a^{k_i}. As a consequence $\prod_i\big(1-(1-\lambda)^{k_i}\big)$ is the probability to avoid forever the final state p_F. The probability to visit p_F after reading the word $a^{k_1}ba^{k_2}b\cdots a^{k_{N-1}}b$ and to avoid p_F from then on is therefore

$$(1-\lambda)^{k_{N-1}} \cdot \prod_{i\geq N}\big(1-(1-\lambda)^{k_i}\big)$$

with the convention $k_0 = 0$. Hence, given an input word $\omega = a^{k_1}ba^{k_2}ba^{k_3}b\ldots$, the probability to avoid q_F from some point on is

$$\sum_{N\geq 1}\Big((1-\lambda)^{k_{N-1}}\prod_{i\geq N}\big(1-(1-\lambda)^{k_i}\big)\Big).$$

Thus, $\mathsf{Pr}^{\widetilde{\mathcal{P}}_\lambda}(\omega) = 1 - \sum_{N\geq 1}\Big((1-\lambda)^{k_{N-1}}\prod_{i\geq N}\big(1-(1-\lambda)^{k_i}\big)\Big)$. To prove that $\mathcal{L}(\widetilde{\mathcal{P}}_\lambda)$ is as indicated above, we need to show that:

$$1 - \sum_{N\geq 1}\Big((1-\lambda)^{k_{N-1}}\prod_{i\geq N}\big(1-(1-\lambda)^{k_i}\big)\Big) > 0 \iff \prod_{i\geq 1}\big(1-(1-\lambda)^{k_i}\big) = 0.$$

$\Leftarrow$: First assume that $\prod_{i\geq 1}\Big(1-(1-\lambda)^{k_i}\Big) = 0$. It then holds for all $N\in\mathbb{N}$ that $\prod_{i\geq N}\Big(1-(1-\lambda)^{k_i}\Big) = 0$, and thus $\sum_{N\geq 1}\Big((1-\lambda)^{k_{N-1}}\prod_{i\geq N}\big(1-(1-\lambda)^{k_i}\big)\Big) = 0$. Note that this implies that $\mathsf{Pr}_{\widetilde{\mathcal{P}}_\lambda}(\omega) = 1$, if $\prod_{i\geq 1}\Big(1-(1-\lambda)^{k_i}\Big) = 0$.

$\Rightarrow$: The second implication is more involved. Assume that $\prod_{i\geq 1}\Big(1-(1-\lambda)^{k_i}\Big) > 0$. We have to show that $\sum_{N\geq 1}\Big((1-\lambda)^{k_{N-1}}\prod_{i\geq N}\big(1-(1-\lambda)^{k_i}\big)\Big) = 1$. With $\theta_i = 1-(1-\lambda)^{k_i}$ we obtain:

$$\begin{aligned}
\sum_{N\geq 1}\Big((1-\lambda)^{k_{N-1}}\prod_{i\geq N}\big(1-(1-\lambda)^{k_i}\big)\Big) &= \sum_{N}\Big((1-\theta_{N-1})\prod_{i\geq N}\theta_i\Big)\\
&= \sum_{N}\Big(\prod_{i\geq N}\theta_i - \theta_{N-1}\prod_{i\geq N}\theta_i\Big)\\
&= \sum_{N}\Big(\prod_{i\geq N}\theta_i - \prod_{i\geq N-1}\theta_i\Big)\\
&= \lim_{N\to\infty}\prod_{i\geq N}\theta_i \qquad \text{since } \theta_0 = 0
\end{aligned}$$

To conclude, we have to show that $\lim_{N\to\infty}\prod_{i\geq N}\theta_i = 1$, using the assumption $c := \prod_{i=1}^{\infty}\theta_i > 0$. But this is obvious since

$$0 \neq c \;=\; \prod_{i=1}^{N}\theta_i\cdot\prod_{i=N+1}^{\infty}\theta_i.$$

As the left factor converges to c if N tends to infinity, the right factor has to converge to 1. This completes the proof that

$$\mathcal{L}(\widetilde{\mathcal{P}}_\lambda) = \big\{a^{k_1}ba^{k_2}ba^{k_3}b\cdots \mid (k_i)_{i=1}^{\infty}\in(\mathbb{N}_{\geq 1})^{\mathbb{N}_{\geq 1}} \text{ s.t. } \prod_{i=1}^{\infty}\big(1-(1-\lambda)^{k_i}\big) = 0\big\}.$$

Remark 4.2.5. The above proof shows that each word $\omega\in\mathcal{L}(\widetilde{\mathcal{P}}_\lambda)$ is accepted by $\widetilde{\mathcal{P}}_\lambda$ with probability 1. That means that

$$\mathsf{Pr}^{\widetilde{\mathcal{P}}_\lambda}(\omega)\in\{0,1\}$$

for all $\omega\in\{a,b\}^\omega$. Thus, the automaton $\widetilde{\mathcal{P}}_\lambda$ is quite remarkable as it not only accepts a non-ω-regular language, but also has the property that each word is accepted with probability 0 or 1.

We showed that PBA are strictly more expressive than ω-regular languages. Note that [BG05] identified a subclass of PBA that corresponds to the class of ω-regular languages. For this purpose [BG05] introduced so-called uniform PBA. The uniformity condition is motivated by the observation made in Remark 4.2.4 and serves to rule out PBA with "non-regular converging behaviors", as it is the case for the PBA $\mathcal{P}_\lambda$ and $\widetilde{\mathcal{P}}_\lambda$ of Figure 4.3 and 4.4.

4.2.2. The precise probabilities matter

Another peculiarity of $\widetilde{\mathcal{P}}_\lambda$ as well as the automaton $\mathcal{P}_\lambda$ is that their accepted languages depend on the precise transition probabilities.

Theorem 4.2.6 (The precise probabilities matter).

$$\text{For } 0 < \lambda < \tfrac{1}{2} < \eta < 1, \quad \mathcal{L}(\mathcal{P}_\lambda) \neq \mathcal{L}(\mathcal{P}_\eta).$$

Recall that $\mathcal{L}(\mathcal{P}_\lambda) = \{a^{k_1}ba^{k_2}b\cdots \mid \prod_{i\geq 1}(1-(1-\lambda)^{k_i}) > 0\}$. Theorem 4.2.6 is an immediate consequence of the following lemma (for $n = 2$).

Lemma 4.2.7. For each $n \in \mathbb{N}_{\geq 2}$ there exists a sequence $(k_i)_{i\geq 1}$ such that

$$\textstyle\prod_{i\geq 1}(1-\lambda^{k_i}) > 0 \text{ if and only if } \lambda < \frac{1}{n}.$$

Proof. Given $n \in \mathbb{N}_{\geq 2}$, we define the sequence $(k_i)_{i\geq 1}$ in the following way: the first n elements are set to 1, then the n^2 following elements are set to 2, the n^3 next elements are set to 3, etc. The sequence $(k_i)_{i\geq 1}$ is non-decreasing, and defined by plateaux of increasing values and exponentially increasing length. We show that $\prod_i(1-\lambda^{k_i})$ is positive if and only if $\lambda < \frac{1}{n}$. To see this, we consider the series $\sum_i \log(1-x^{k_i})$ which converges if and only if $\prod_i(1-x^{k_i})$ is positive. Now, $\sum_i \log(1-x^{k_i}) = \sum_i n^i \log(1-x^i)$ by definition of the sequence $(k_i)_{i\geq 1}$, and the latter series behaves as $-\sum_i n^i x^i$ (i.e. either both converge, or both diverge) since $\log(1-\varepsilon) \sim_{\varepsilon\mapsto 0} -\varepsilon$. Hence $\sum_i n^i \log(1-x^i) < \infty$ if and only if $x < \frac{1}{n}$, and $\prod_i(1-\lambda^{k_i}) > 0$ if and only if $\lambda < \frac{1}{n}$ which proofs the claim. □

Thus, given two PBA with the same underlying NBA, their accepted languages might differ.

4.2.3. Efficiency of PBA

We saw in Lemma 4.2.2 on page 67 that for each NBA there exists an equivalent PBA of size exponential in the size of the NBA. We now study the efficiency of PBA in more detail and show that for some languages, PBA can be exponentially better than nondeterministic ω-automata.

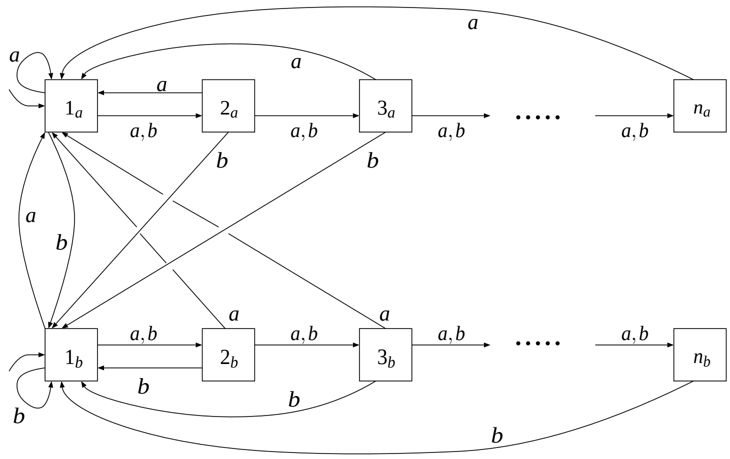


Figure 4.5: PBA for L_n as in the proof of Lemma 4.2.8

4.2.3.0. Example 1

Lemma 4.2.8 (PBA can be exponentially smaller than NSA). There exist ω-regular languages $L_n \subseteq \{a,b\}^\omega$ which are accepted by a PBA with $2n$ states, while any NSA for L_n has at least $\frac{2^n}{n}$ states.

Proof. The language $L_n = \{xy^\omega : x \in \{a,b\}^*, y \in \{a,b\}^n\}$ is accepted by the PBA $\mathcal{P}$ shown in Figure 4.5. Note that all states of $\mathcal{P}$ are accepting and that all states except n_a have a b-transition to the state 1_b and all states except n_b have an a-transition to the state 1_a. We assume uniform distibutions, that is all edges except the a-edge in state n_a and the b-edge in state n_b are taken with probability $\frac{1}{2}$.

Let $\omega = a_1 a_2 \ldots \notin L_n$. Then there are infinitely many indices i such that $a_i = a \wedge a_{i+n} = b$. Since every state except the state n_b has an a-transition to state 1_a, the stochastic process induced by $\mathcal{P}$ and the input word ω will almost surely be infinitely often in state 1_a with the letter b coming up in n steps. But each time (with probability $\frac{1}{2^{n-1}}$) the process will have moved to the state n_a while reading the upcoming $n-1$ letters, thus rejecting upon reading the b. Thus, almost surely the process will reject infinitely often with probability $\frac{1}{2^{n-1}}$ which shows that almost all runs are rejecting. Thus $\mathsf{Pr}^{\mathcal{P},\omega} = 0$ and $\omega \notin \mathcal{L}(\mathcal{P})$. Therefore $\mathcal{L}(\mathcal{P}) \subseteq L_n$.

On the other hand, given a word $\omega \in L_n$, we can write ω as xy^ω with $x \in \{a,b\}^*$ and $y \in \{a,b\}^n$. Then, $\hat{\pi} = 1_{c_1}, \ldots, 1_{c_k}, 1_{d_1}$ is a run for xd_1, where $x = c_1 c_2 \ldots c_k$ and $y = d_1 d_2 \ldots d_n$. (The c's and d's are symbols in $\{a,b\}$). The probability for this run is strictly greater than zero. Since from that state 1_{d_1} on, the process will never reject while reading the remaining suffix of ω and since every infinite run is accepting, this shows that ω will be accepted with a probability greater than zero. This yields $L_n \subseteq \mathcal{L}(\mathcal{P})$.

It remains to show that any NSA for L_n has at least $\frac{2^n}{n}$ states. Let $\mathcal{A}$ be an NSA with $\mathcal{L}(\mathcal{A}) = L_n$. Let $x = c_1 \ldots c_n$, $y = d_1 \ldots d_n \in \{a,b\}^n$ such that

$$c_1 \dots c_n \neq d_i \dots d_n d_1 \dots d_{i-1} \text{ for all } i = 1, \dots, n \tag{+}$$

Then, any two accepting cycles for $(c_1 \dots c_n)^\omega$ and $(d_1 \dots d_n)^\omega$ are disjoint. Otherwise, $\mathcal{A}$ would accept a word of the form $(a+b)^*(c_1 \dots c_j d_i \dots d_n d_1 \dots d_{i-1} c_{j+1} \dots c_n)^\omega$. But such a word is not in L_n because of (+). Thus, $\mathcal{A}$ has at least $\frac{2^n}{n}$ disjoint acceptance cycles, which proves the claim. □

4.2.3.0. Example 2

Another example that illustrates the efficiency of PBA is the language L_n consisting of all infinite words $\omega = a_1 a_2 \dots \in \{a,b,c\}^\omega$ such that, for all $0 \leq i < n$, if $a_{kn+i} = a$ for infinitely many k then $a_{kn+i} = b$ for infinitely many k, and vice versa. L_n is accepted by a PBA of polynomial size, while the size of any NBA that accepts L_n is at least exponential (but there exist small equivalent NSA).

Lemma 4.2.9 (PBA can be exponentially smaller than NBA). Let L_n be the language consisting of all infinite words $\omega = a_1 a_2 a_3 \dots \in \{a,b,c\}^\omega$ such that for all $0 \leq i < n$:

$$\overset{\infty}{\exists}\, k \text{ s.th. } a_{kn+i} = a \text{ if and only if } \overset{\infty}{\exists}\, k \text{ s.th. } a_{kn+i} = b. \tag{++}$$

Then, L_n is accepted by a PBA with $\mathcal{O}(n^2)$ states, while any NBA that accepts L_n has $\Omega(2^n)$ states.

Proof. Safra and Vardi [SV89] proved that any NBA that accepts L_n has $\Omega(2^n)$ states. But there exists an NSA that accepts L_n and consists of $\mathcal{O}(n)$ states.

It remains to show that there is a PBA of quadratic size that accepts L_n. For any word $\omega = a_1 a_2 \dots \in L_n$, we refer to the suffix $a_{rn+1} a_{rn+2} \dots$ such that

(1) for all $0 \leq i < n$ either $a_{kn+i} = c$ for all $k \geq r$ or there are infinitely many k, ℓ with $a_{kn+i} = a$ and $a_{\ell n+i} = b$ and

(2) r is minimal w.r.t. (1)

as the *legal suffix* of ω. A PBA $\mathcal{P}$ with $\mathcal{O}(n^2)$ states that accepts L_n is depicted in Figure 4.6 (we assume uniform distributions). All following calculations with indices $i, j \in \{0, 1, \dots, n-1\}$ are modulo n, i.e. we simply write $i+1$ instead of $(i+1) \mod n$.

The automaton $\mathcal{P}$ consists of

- a subautomaton $\mathcal{P}_{init}$ which serves to wait until the legal suffix of ω starts. It consists of a cycle of accepting states $0, 1, \dots, n-1$ that are passed in this order.
- subautomata $\mathcal{P}_{(i,a)}$ and $\mathcal{P}_{(i,b)}$ that are entered from $\mathcal{P}_{init}$ when reading the letter a, resp. b, in state i. The automata $\mathcal{P}_{(i,a)}$ and $\mathcal{P}_{(i,b)}$ consist of a cycle of nonaccepting states $0, 1, \dots, n-1$ that are passed in this order. They are entered in state $i+1$ (coming from state i of $\mathcal{P}_{init}$ upon reading the letter a, resp b) and can be left via an accepting state only when reading the letter b, resp a, in state i.

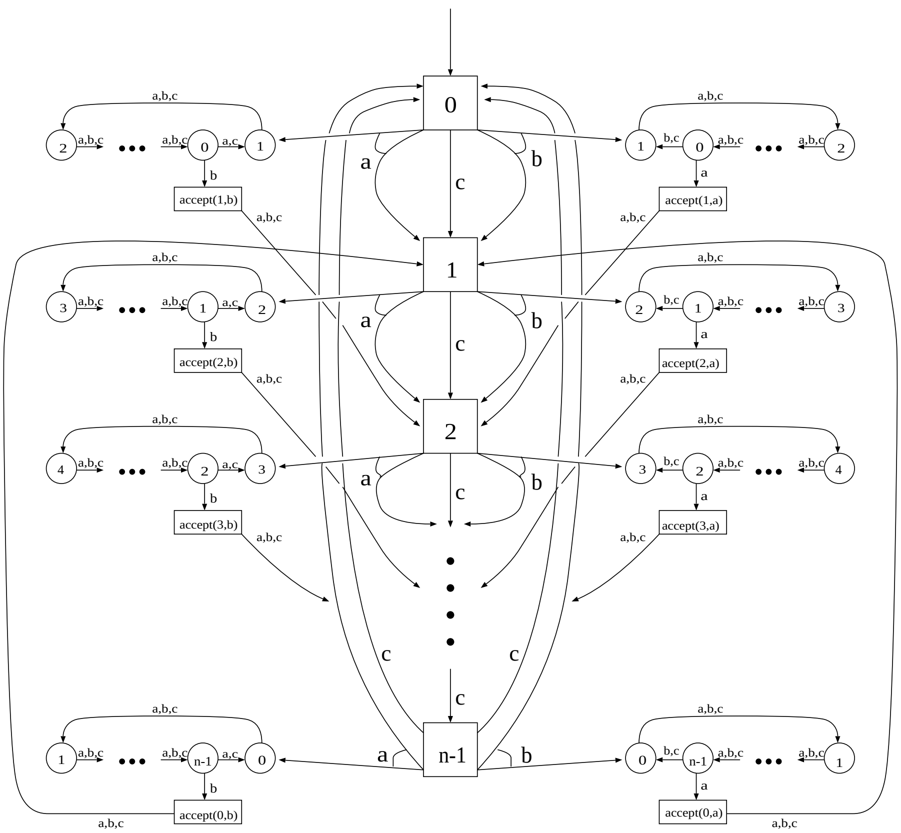


Figure 4.6: PBA with $\mathcal{O}(n^2)$ states, while any equivalent NBA has $\Omega(2^n)$ states

Note that for all words $\omega \in \{a, b, c\}^\omega$ all runs are infinite and almost all runs leave the subautomaton $\mathcal{P}_{init}$ if ω contains infinitely many a's or b's. The automaton rejects if it enters $\mathcal{P}_{(i,a)}$ or $\mathcal{P}_{(i,b)}$, but there is no following position $kn+i$ with $a_{kn+i} = b$ or $a_{kn+i} = a$, respectively.

Let $\omega = a_1 a_2 \ldots \notin L_n$. Without loss of generality, there is some $r \geq 0$ and some $i \in \{0, 1, \ldots, n-1\}$ such that $a_{kn+i} = a$ for infinitely many k, but $a_{kn+i} \neq b$ for all $k \geq r$. Assume for simplicity that for all $j \neq i$, condition (++) is fulfilled. We now consider the stochastic process induced by $\mathcal{P}$ and ω. As there are infinitely many such k's, the process will almost surely enter $\mathcal{P}_{(i,a)}$ but never leave it. Hence, almost all runs for ω are rejecting which yields $\omega \notin \mathcal{L}(\mathcal{P})$. If (++) is violated for several indices $i \in \{0, \ldots, n-1\}$, then the process will almost surely end up in several $\mathcal{P}_{(i,a)}$ and $\mathcal{P}_{(i,b)}$ and never leave those. Hence, almost all runs for ω are rejecting which yields $\omega \notin \mathcal{L}(\mathcal{P})$.

Vice versa, let $\omega = a_1 a_2 \ldots \in L_n$ and $a_{rn+1}, a_{rn+2}, \ldots$ be the legal suffix of ω. Then, all runs for ω that stay in $\mathcal{P}_{init}$ for the first rn input symbols (the prefix $a_1 \ldots a_{rn}$ of ω) will infinitely often be in $\mathcal{P}_{init}$ and are therefore accepting. Hence, $\Pr(\omega) > 0$ and $\omega \in \mathcal{L}(\mathcal{P})$. $\square$

4.3. Composition operators for PBA

After the general discussion on PBA in the last section, we will now present composition operators for PBA that realize union, intersection and complementation. For union and intersection we may roughly apply the same techniques as for NBA. More interesting is the complementation operator which relies on a transformation of the given PBA $\mathcal{P}$ into an equivalent PRA $\mathcal{P}_R$ that accepts each word with probability 0 or 1. This PRA can easily be turned into a PSA for the complement language, which will at last be transformed into an equivalent PBA.

4.3.1. Union and Intersection

Let $\mathcal{P}_1 = (Q_1, \Sigma, \delta_1, \mu_0^1, F_1)$ and $\mathcal{P}_2 = (Q_2, \Sigma, \delta_2, \mu_0^2, F_2)$ be two PBA over the same alphabet.

Union: A PBA $\mathcal{P}$ that accepts the language $\mathcal{L}(\mathcal{P}_1) \cup \mathcal{L}(\mathcal{P}_2)$ can be obtained by taking the disjoint union of the state spaces Q_1 and Q_2, equipped with the transitions in $\mathcal{P}_1$ and $\mathcal{P}_2$. The initial distribution μ_0 in $\mathcal{P}$ assigns probability $\frac{1}{2}\mu_0^i(q)$ to any state $q \in Q_i$ $(i = 1, 2)$. The set of accepting states of $\mathcal{P}$ is $F_1 \cup F_2$. Intuitively, the automaton $\mathcal{P}$ simulates $\mathcal{P}_1$ and $\mathcal{P}_2$, each with probability $\frac{1}{2}$. It thus holds that

$$\mathsf{Pr}^{\mathcal{P}}(\omega) \ = \ \tfrac{1}{2}\mathsf{Pr}^{\mathcal{P}_1}(\omega) \ + \ \tfrac{1}{2}\mathsf{Pr}^{\mathcal{P}_2}(\omega)$$

and therefore $\mathcal{L}(\mathcal{P}) = \mathcal{L}(\mathcal{P}_1) \cup \mathcal{L}(\mathcal{P}_2)$.

Intersection: For the intersection operator, we use the same trick as for NBA and generate a generalized PBA (GPBA) $\mathcal{P}_1 \bowtie \mathcal{P}_2$ that results by the parallel composition of $\mathcal{P}_1$ and $\mathcal{P}_2$ and that accepts the intersection language $\mathcal{L}(\mathcal{P}_1) \cap \mathcal{L}(\mathcal{P}_2)$. A generalized Büchi automaton is equipped with several acceptance sets and in order to be an accepting run, a run has to visit each of the acceptance sets infinitely often. The generalized PBA $\mathcal{P}_1 \bowtie \mathcal{P}_2$ can then be turned into an equivalent PBA (as for NBA). The formal definition of the generalized PBA $\mathcal{P}_1 \bowtie \mathcal{P}_2$ is fairly standard and relies on the idea that $\mathcal{P}_1$ and $\mathcal{P}_2$ run in parallel. Formally, the state space of $\mathcal{P}_1 \bowtie \mathcal{P}_2$ is the cartesian product $Q_1 \times Q_2$. The transition probabilities in the product are given by

$$\delta((p_1, p_2), a, (q_1, q_2)) \ = \ \delta_1(p_1, a, q_1) \cdot \delta_2(p_2, a, q_2).$$

The initial distribution in $\mathcal{P}_1 \bowtie \mathcal{P}_2$ is defined by $\mu_0((p_1, p_2)) = \mu_0^1(p_1) \cdot \mu_0^2(p_2)$. $\mathcal{P}_1 \bowtie \mathcal{P}_2$ has two acceptance sets, namely $Q_1 \times F_2$ and $F_1 \times Q_2$ that both have to be visited infinitely often in order to accept. Then

$$\mathsf{Pr}_{\mathsf{GPBA}}^{\mathcal{P}_1 \bowtie \mathcal{P}_2}(\omega) \ = \ \mathsf{Pr}^{\mathcal{P}_1 \bowtie \mathcal{P}_2, \omega}(\{\pi \mid \pi \models \Box\Diamond Q_1 \times F_2 \wedge \Box\Diamond F_1 \times Q_2\}) \ = \ \mathsf{Pr}^{\mathcal{P}_1}(\omega) \cdot \mathsf{Pr}^{\mathcal{P}_2}(\omega).$$

Hence,the accepted language of the generalized PBA $\mathcal{P}_1 \bowtie \mathcal{P}_2$ is $\mathcal{L}(\mathcal{P}_1) \cap \mathcal{L}(\mathcal{P}_2)$.

We now explain the transformation how a GPBA $\mathcal{P} = (Q, \Sigma, \delta, \mu_0, F_1, \ldots, F_n), n \geq 1$ with the acceptance condition $\bigwedge_{1 \leq i \leq n} \Box\Diamond F_i$ can be transformed into an equivalent PBA

$\mathcal{P}' = (Q', \Sigma, \delta', \mu'_0, F')$. This follows the same idea used for the transformation of GNBA to NBA. $\mathcal{P}'$ consists of n copies of $\mathcal{P}$, where the transitions from a state in the acceptance set F_i of the ith copy lead to the appropriate states in the $(i+1)$st copy. The other transitions stay in their own copy. The automaton $\mathcal{P}'$ starts in the 1st copy of $\mathcal{P}$ and the acceptance condition of $\mathcal{P}'$ is F_1 of the 1st copy. As the automaton switches the copy whenever a state of F_i is visited in the ith copy, this ensures that whenever F_1 is visited infinitely often in the first copy, then all other acceptance sets F_j are visited infinitely often too. Formally, $Q' = Q \times \{1, \dots, n\}$, $\mu'_0(\langle q, 1\rangle) = \mu_0(q)$ and $F' = F_1 \times \{1\}$. The transition function δ' is defined as follows. Let $q, p \in Q$, $a \in \Sigma$ and $i \in \{1, \dots, n\}$.

$$\begin{array}{lrcl} q \in F_i: & \delta'(\langle q, i\rangle, a, \langle p, i+1\rangle) & = & \delta(q, a, p) \\ q \notin F_i: & \delta'(\langle q, i\rangle, a, \langle p, i\rangle) & = & \delta(q, a, p) \end{array}$$

Here the indices are computed modulo n, that is $n+1 = 1$. All other transition probabilities are zero. As each run (for a given input word) in $\mathcal{P}$ has exactly one lifting to a run in $\mathcal{P}'$ with equal stepwise probabilities, the same reasoning as in the case for nondeterministic automata shows $\mathsf{Pr}^{\mathcal{P}}_{\mathsf{GPBA}}(\omega) = \mathsf{Pr}^{\mathcal{P},\omega}(\{\pi \mid \pi \models \bigwedge_{1 \le i \le n} \Box\Diamond F_i\}) = \mathsf{Pr}^{\mathcal{P}'}(\omega)$ for any input word ω.

4.3.2. Complementation of PBA

The question whether the class of languages recognizable by PBA is closed under complementation was left open in [BG05]. We show here that for each PBA $\mathcal{P}$ there exists a PBA that accepts the complement of $\mathcal{L}(\mathcal{P})$ [BBG08].

Theorem 4.3.1 ($\mathbb{L}(\mathsf{PBA})$ **is closed under complementation).**
For each PBA $\mathcal{P}$ there exists a PBA $\mathcal{P}'$ of size $\mathcal{O}(\exp(|\mathcal{P}|))$ such that $\mathcal{L}(\mathcal{P}') = \Sigma^\omega \setminus \mathcal{L}(\mathcal{P})$. Moreover, $\mathcal{P}'$ can be effectively constructed from $\mathcal{P}$.

Proof. The idea for the complementation of a given PBA $\mathcal{P}$ is to provide the following series of transformations

$$\begin{array}{rcl} \text{PBA } \mathcal{P} & \overset{(1)}{\Longrightarrow} & \text{0/1-PRA } \mathcal{P}_R \text{ with } \mathcal{L}(\mathcal{P}_R) = \mathcal{L}(\mathcal{P}) \\ & \overset{(2)}{\Longrightarrow} & \text{0/1-PSA } \mathcal{P}_S \text{ with } \mathcal{L}(\mathcal{P}_S) = \Sigma^\omega \setminus \mathcal{L}(\mathcal{P}_R) \\ & \overset{(3)}{\Longrightarrow} & \text{PBA } \mathcal{P}' \text{ with } \mathcal{L}(\mathcal{P}') = \mathcal{L}(\mathcal{P}_S) \end{array}$$

where 0/1-PRA denotes a PRA with $\mathsf{Pr}^{\mathcal{P}_R}(\omega) \in \{0, 1\}$ for each word $\omega \in \Sigma^\omega$. We will show the transformation of step (1) in Theorem 4.3.2 below, explain step (2) right now and refer to Theorem 4.3.4 for step (3).

Let $\mathcal{P} = (Q, \Sigma, \delta, \mu_0, F)$ be a PBA. The goal is to provide a PBA $\mathcal{P}'$ that accepts the language

$$\overline{\mathcal{L}(\mathcal{P})} = \Sigma^\omega \setminus \mathcal{L}(\mathcal{P}).$$

To do so, we apply Theorem 4.3.2 and construct an equivalent probabilistic Rabin automaton $\mathcal{P}_R$ such that for each infinite word, all runs are infinite and the probability of the accepting runs in $\mathcal{P}_R$ is either 0 or 1. Hence

$$\mathcal{L}(\mathcal{P}) \;=\; \mathcal{L}(\mathcal{P}_R) \;=\; \big\{\omega \in \Sigma^\omega : \mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega) > 0\big\} \;=\; \big\{\omega \in \Sigma^\omega : \mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega) = 1\big\}.$$

For the sake of clarity we may index Pr or $\mathcal{L}$ by *Rabin, Streett* or *Büchi* to stress that the automaton is of such a kind. We now use the duality between Rabin and Streett automata and switch from the Rabin acceptance condition $\bigvee_{1\leq j\leq n}(\Diamond\Box\neg H_j \wedge \Box\Diamond K_j)$ to the Streett acceptance condition $\bigwedge_{1\leq j\leq n}(\Box\Diamond K_j \;\rightarrow\; \Box\Diamond H_j)$. Let $\mathcal{P}_S$ be the probabilistic Streett automaton that agrees with $\mathcal{P}_R$. As $\mathcal{P}_S$ interprets the acceptance condition dually to $\mathcal{P}_R$, a run in $\mathcal{P}_S$ is accepting if and only if it is not accepting in $\mathcal{P}_R$. Thus, for each infinite word ω it holds that

$$\mathsf{Pr}^{\mathcal{P}_S}_{\mathit{Streett}}(\omega) \;=\; 1 - \mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega).$$

Hence

$$\begin{aligned}\mathcal{L}_{\mathit{Streett}}(\mathcal{P}_S) &= \{\omega \mid \mathsf{Pr}^{\mathcal{P}_S}_{\mathit{Streett}}(\omega) > 0\}\\ &= \{\omega \mid \mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega) < 1\}\\ &= \{\omega \mid \mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega) = 0\}\\ &= \overline{\mathcal{L}_{\mathit{Rabin}}(\mathcal{P}_R)} \;=\; \overline{\mathcal{L}_{\mathit{Büchi}}(\mathcal{P})}.\end{aligned}$$

We then can apply Theorem 4.3.4 to transform the PSA $\mathcal{P}_S$ into an equivalent PBA $\mathcal{P}'$, which yields

$$\mathcal{L}(\mathcal{P}') \;=\; \mathcal{L}(\mathcal{P}_S) \;=\; \overline{\mathcal{L}(\mathcal{P}_R)} \;=\; \overline{\mathcal{L}(\mathcal{P})}.$$

Let n be the number of states in the original PBA $\mathcal{P}$. The construction presented in the proof of Theorem 4.3.2 yields that the number of states in $\mathcal{P}_R$ (and $\mathcal{P}_S$) is bounded by $2^{\mathcal{O}(n\log n)}$, while the number of acceptance pairs in $\mathcal{P}_R$ and $\mathcal{P}_S$ is n. Therefore, the size of the PBA $\mathcal{P}'$ generated from $\mathcal{P}_S$ by Theorem 4.3.4 is bounded by $n^2 \cdot 2^{\mathcal{O}(n\log n)}$, thus $\mathcal{P}'$ is at most exponentially larger than $\mathcal{P}$. □

In the previous proof, the most interesting part is step (1), which has some similarities with Safra's determinization algorithm for NBA and also relies on some kind of powerset construction. However, we argue that the probabilistic setting is slighty simpler. Instead of organizing the potential accepting runs in Safra trees, we may deal with up to n independent *sample runs* (where n is the number of states in $\mathcal{P}$) that are representative for all potential accepting runs. The idea is to represent the current states of the sample runs by tuples $\langle p_1, \ldots, p_k\rangle$ of pairwise distinct states in $\mathcal{P}$. Whenever two sample runs meet at some point, say the next states p_1' and p_2' in the first two sample runs agree, then they are merged, which requires a shift operation for the other sample runs and yields a tuple of the form $\langle p_1', p_3', \ldots, p_k', \ldots, q, \ldots\rangle$ where p_i' is a successor of p_i in the i-th sample run. Additionally, new sample runs are generated in case the original PBA $\mathcal{P}$ can be in an accepting state $q \notin \{p_1', \ldots, p_k'\}$. The Rabin condition serves to express the condition that at least one of the sample runs enters the set F of accepting states in $\mathcal{P}$ infinitely often and is a proper run in $\mathcal{P}$ (i.e. is affected by the shift operations only finitely many times). Intuitively, the automaton $\mathcal{P}_R$ "simulates" $\mathcal{P}$ and moreover each time $\mathcal{P}$ could be in an accepting state, $\mathcal{P}_R$ starts a new sample run (if necessary). Let $\omega \in \mathcal{L}(\mathcal{P})$, thus $\mathsf{Pr}^{\mathcal{P}}(\omega) > 0$ and with positive probability $\mathcal{P}$ can be in an accepting state infinitely often. But then $\mathcal{P}_R$ almost surely either already is in a corresponding sample run or starts a new sample run infinitely often and from there on accepts the remaining suffix with positive probability $> c$ for some $c > 0$ (as it "simulates" $\mathcal{P}$ and $\Pr^{\mathcal{P}}(\omega) > 0$). This yields that the automaton $\mathcal{P}_R$ accepts ω with probability 1. We will formalize this idea in the proof of

Theorem 4.3.2 (From PBA to 0/1-PRA).
For each PBA $\mathcal{P}$ there exists a PRA $\mathcal{P}_R$ such that $\mathcal{L}(\mathcal{P}) = \mathcal{L}_{Rabin}(\mathcal{P}_R)$ and such that for each infinite word ω it holds that $\Pr^{\mathcal{P}_R}(\omega) \in \{0, 1\}$.

Proof. Let $\mathcal{P} = (Q, \delta, \mu_0, F)$ be the given PBA. Without loss of generality, we may suppose that $\mathcal{P}$ is total. The idea for the definition of $\mathcal{P}_R$ is to deal with states of the form $\langle p_1, \ldots, p_k, R\rangle$ where $p_1, \ldots, p_k$ are pairwise distinct states that represent the current states of "independent" runs for the given input word. The acceptance condition of $\mathcal{P}_R$ will then require that at least one of these runs in $\mathcal{P}$ is accepting. The last component R is a subset of Q, representing the set of all potential states in which the original automaton $\mathcal{P}$ could be. It will be obtained by the standard powerset construction for finite automata.

To organize the independent runs in a finite-state automaton (rather than an infinite tree) we abstract away from multiple occurrences of some state and merge runs that meet at some point.This causes some technical difficulties because $\mathcal{P}_R$ has to recover fictive sample runs that enter F infinitely often by combining fragments of infinitely many runs. For this reason, we attach a bit $\xi_j \in \{0, 1\}$ for each of the states p_j which indicates whether the last step results from a proper transition in $\mathcal{P}$ (in which case $\xi_j = 0$) or p_j is the first state of a newly generated run (in which case $\xi_j = 1$). These bits will be used in the definition of the Rabin acceptance condition of $\mathcal{P}_R$ which requires that for some j, the j-th run visits F infinitely often and in some infinite suffix, the attached bits are 0.

We will structure our states in $\mathcal{P}_R$ in such a way that we first list the states that result from a proper transition in $\mathcal{P}$ (having the attached bit 0) and then we list the states that are newly generated (because the automaton $\mathcal{P}$ could be in an accepting state). Those have the attached bit 1. Thus, for each state $\langle p_1, \xi_1, \ldots, p_n, \xi_n, R\rangle$, it holds that

$$\text{whenever } \xi_i = 1 \text{ then } \xi_j = 1 \text{ for } i < j \leq n. \qquad (+)$$

Since several sample runs could be in the same next state (with the attached bit 0), we may need to merge them. Therefore we define a normalization operator ν that takes as input k states $p_1, \ldots, p_k$ in $\mathcal{P}$ augmented with bits $\xi_1, \ldots, \xi_k$, possibly with multiple occurrences of some states, and returns a normalized tuple where each state in $\{p_1, \ldots, p_k\}$ appears exactly once, with an appropriate bit. Formally, given a $2k$-tuple $\langle p_1, \xi_1, \ldots, p_k, \xi_k\rangle \in (Q \times \{0, 1\})^k$ where $k \geq 1$ and the ξ_i's satisfy (+) we now define $\nu(\langle p_1, \xi_1, \ldots, p_k, \xi_k\rangle)$ to be the unique tuple $\langle p_{i_1}, \xi'_{i_1}, \ldots, p_{i_\ell}, \xi'_{i_\ell}\rangle$ where $i_1, \ldots, i_\ell \in \{1, \ldots, k\}$ are indices such that

- $i_1 < i_2 < \ldots < i_\ell$ and $\{p_1, \ldots, p_k\} = \{p_{i_1}, \ldots, p_{i_\ell}\}$,
- $p_{i_1}, \ldots, p_{i_\ell}$ are pairwise distinct and $p_{i_h} \notin \{p_1, \ldots, p_{i_h - 1}\}$ for $1 \leq h \leq \ell$,
- $\xi'_{i_h} = 1$ if $h < i_h$ and $\xi'_{i_h} = \xi_{i_h}$ if $h = i_h$.

Note that $\xi'_1, \ldots, \xi'_\ell$ satisfy (+). For instance, $\nu(\langle p, 0, q, 1, p, 1 \rangle) = \nu(\langle p, 0, p, 0, q, 0 \rangle) = \langle p, 0, q, 1 \rangle$. The idea is to identify all tuples $\langle p_1, \xi_1, \ldots, p_k, \xi_k \rangle$ and $\langle q_1, \zeta_1, \ldots, q_j, \zeta_j \rangle$ such that

$$\nu(\langle p_1, \xi_1, \ldots, p_k, \xi_k \rangle) = \nu(\langle q_1, \zeta_1, \ldots, q_j, \zeta_j \rangle).$$

The reason why the normalization operator ν requires $\xi_{i_h} = 1$ if $h < i_h$ is that the bit 1 serves as a separation symbol in the state sequence induced by the $(2h-1)$-st component of the states in a run in $\mathcal{P}_R$. Given a run $\overline{\pi}$ in $\mathcal{P}_R$ such that for infinitely many states in $\overline{\pi}$ the bit in the $2h$-th component is 1, then the state sequence obtained by the $(2h-1)$-st components of the states in $\overline{\pi}$ results from the concatenation of fragments of infinitely many runs in $\mathcal{P}$. Hence, it does not necessarily represent a run in $\mathcal{P}$. This will be important for the acceptance condition in $\mathcal{P}_R$.

We now present the precise definition of the PRA $\mathcal{P}_R$. The state space of the PRA $\mathcal{P}_R$ is

$$\overline{Q} = \bigcup_{1 \le k \le n} \overline{Q}_k$$

where $n = |Q|$ and $\overline{Q}_k$ is the set of all tuples $\langle p_1, \xi_1, \ldots, p_k, \xi_k, R \rangle \in (Q \times \{0,1\})^k \times 2^Q$ such that $p_i \neq p_j$ for $1 \le i < j \le k$ and that $\xi_1, \ldots, \xi_k$ satisfy (+). Let us fix the notation $\overline{Q}_{\ge j} = \bigcup_{j \le k \le n} \overline{Q}_k$ to denote the set of states of $\mathcal{P}_R$ that represent at least j sample runs. Similarly $\overline{Q}_{<j} = \bigcup_{1 \le k < j} \overline{Q}_k$ denotes the set of states of $\mathcal{P}_R$ that represent less than j sample runs. Intuitively, when reading letter a in state $\overline{q} = \langle q_1, \xi_1, \ldots, q_k, \xi_k, R \rangle$ in $\mathcal{P}_R$ then the possible successors are the tuples

$$\overline{p} = \langle p_1, \zeta_1, \ldots, p_k, \zeta_k, p_{k+1}, \zeta_{k+1}, \ldots, p_m, \zeta_m, S \rangle$$

where

(i) $p_i \in \delta(q_i, a)$ for $1 \le i \le k$,

(ii) $p_{k+1}, \ldots, p_m$ are pairwise distinct states in $\mathcal{P}$ such that

$$\{p_{k+1}, \ldots, p_m\} = (\delta(R, a) \cap F) \setminus \{p_1, \ldots, p_k\},$$

(iii) $\zeta_1 = \ldots = \zeta_k = 0$ and $\zeta_{k+1} = \ldots = \zeta_m = 1$,

(iv) $S = \delta(R, a)$.

These tuples $\overline{p}$ might be not contained in $\overline{Q}$, but they will be turned into states of $\mathcal{P}_R$ by applying the ν-operator. The intuitive meaning of condition (i) is the independence of the transitions $q_i \xrightarrow{a} p_i$, $i = 1, \ldots, k$, that serve to mimick $\mathcal{P}$'s behavior by sample runs. Condition (ii) can be understood as the creation of new sample runs that are potential accepting runs in $\mathcal{P}$. We attach the bit 0 to the first k components to denote that the last step of the sample runs $1, \ldots, k$ was a proper transition in $\mathcal{P}$, while the attached bit 1 for runs $k+1, \ldots, m$ indicate that new runs have been generated (condition (iii)). The last condition (iv) states that the last component is obtained with the standard powerset construction. The probability to obtain the tuple $\overline{p}$ (note that $\overline{p} \notin \overline{Q}$ is possible as there might be multiple occurrences of states with the attached bit 0) from state $\overline{q} \in \overline{Q}$ by reading letter a is given by

$$\Delta(\overline{q}, a, \overline{p}) \;=\; \prod_{1 \le i \le k} \delta(q_i, a, p_i),$$

provided that the above conditions (i), (ii), (iii) and (iv) hold. For all other tuples we set $\Delta(\overline{q}, a, \overline{p}) = 0$.

For given states $\overline{q} \in \overline{Q}$ and $\overline{q}' \in \overline{Q}$ in $\mathcal{P}_R$, the transition probability $\delta_{\mathcal{P}_R}(\overline{q}, a, \overline{q}')$ in $\mathcal{P}_R$ is obtained by summing up the values $\Delta(\overline{q}, a, \overline{p})$ where $\overline{p}$ ranges over all tuples that are represented by state $\overline{q}'$ in $\mathcal{P}_R$ and satisfy conditions (i), (ii), (iii) and (iv). Formally, given a state

$$\overline{q}' \;=\; \langle q_1', \xi_1', \ldots, q_\ell', \xi_\ell', R' \rangle \in \overline{Q},$$

let $[\![\overline{q}']\!]$ be the set of all tuples $\overline{p} = \langle p_1, \zeta_1, \ldots, p_m, \zeta_m, S\rangle$ such that

$$\nu(\langle p_1, \zeta_1, \ldots, p_m, \zeta_m\rangle) \;=\; \langle q_1', \xi_1', \ldots, q_\ell', \xi_\ell'\rangle \quad \text{and} \quad R' = S.$$

The transition probabilities in $\mathcal{P}_R$ are defined by:

$$\delta_{\mathcal{P}_R}(\overline{q}, a, \overline{q}') \;=\; \sum_{\overline{p} \in [\![\overline{q}']\!]} \Delta(\overline{q}, a, \overline{p}).$$

The acceptance condition of the probabilistic Rabin Automaton $\mathcal{P}_R$ consists of n acceptance pairs $(H_1, K_1), \ldots, (H_n, K_n)$. Intuitively, the j-th pair (H_j, K_j) formalizes the condition stating that the state sequence obtained by the $(2j-1)$-st components of a given run $\overline{\pi}$ in $\mathcal{P}_R$ stands for an accepting run in $\mathcal{P}$. This requires that F is visited infinitely often and that from some moment on the attached bit at position $2j$ is 0. Intuitively, these conditions assert that the state sequence in Q obtained by the $(2j-1)$-st components of the states in $\overline{\pi}$ contains an infinite suffix which is the suffix of an accepting run in $\mathcal{P}$. Formally, the set $K_j \subseteq \overline{Q}$ consists of all states

$$\langle p_1, \xi_1, \ldots, p_j, \xi_j, \ldots, p_k, \xi_k, R\rangle \in \overline{Q}_{\ge j} \;\text{ such that }\; p_j \in F.$$

The set $H_j \subseteq \overline{Q}$ consists of all states

$$\langle p_1, \xi_1, \ldots, p_j, \xi_j, \ldots, p_k, \xi_k, R\rangle \in \overline{Q}_{\ge j} \;\text{ such that }\; \xi_j = 1.$$

The initial distribution in $\mathcal{P}_R$ is given by

$$\overline{\mu}_0(\langle p, 0, Q_{\mathsf{init}}\rangle) \;=\; \mu_0(p)$$

where Q_{init} is the set of initial states in $\mathcal{P}$, i.e. $Q_{\mathsf{init}} = \{q \in Q : \mu_0(q) > 0\}$.

Given an infinite word $\omega = a_1 a_2 a_3 \ldots \in \Sigma^\omega$, we show the equivalence of the following three statements.

(1) $\mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega) > 0$

(2) $\mathsf{Pr}^{\mathcal{P}}_{\mathit{Büchi}}(\omega) > 0$

(3) $\mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega) = 1$

The equivalence of the statements (1), (2) and (3) yields

$$\mathcal{L}_{\mathit{Rabin}}(\mathcal{P}_R) \ = \ \mathcal{L}_{\mathit{Büchi}}(\mathcal{P}) \ \text{ and } \ \mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega) \in \{0,1\} \text{ for all } \omega \in \Sigma^\omega,$$

which is what we wanted to prove.

(3) $\Rightarrow$ (1): This implication is obvious.

(1) $\Rightarrow$ (2): We now show that (1) implies (2). Suppose that $\mathsf{Pr}^{\mathcal{P}_R}_{\mathit{Rabin}}(\omega) > 0$. Then, there is some $j \in \{1, \ldots, n\}$ such that

$$\mathsf{Pr}^{\mathcal{P}_R}\big\{\overline{\pi} : \overline{\pi} \text{ is a run for } \omega \text{ in } \mathcal{P}_R \text{ such that } \overline{\pi} \models \Diamond\Box\neg H_j \ \wedge \ \Box\Diamond K_j \big\} \ > \ 0.$$

As the set of runs that satisfy $\Diamond\Box\neg H_j$ is the disjoint union of the sets of runs satisfying $\Diamond^{=k-1} H_j \ \wedge \ \Box^{\geq k}\neg H_j, k = 0, 1, 2, \ldots$, there exists $m \in \mathbb{N}_{\geq 0}$ such that

$$\mathsf{Pr}^{\mathcal{P}_R}\big\{\overline{\pi} : \overline{\pi} \text{ is a run for } \omega \text{ such that } \overline{\pi} \models \Box^{\geq m}\neg H_j \ \wedge \ \Box\Diamond K_j \big\} \ > \ 0.$$

Here, $\overline{\pi} \models \Box^{\geq k}\neg H_j$ if and only if $\overline{\pi}^\ell \notin H_j$, for $\ell \geq k$. As the set $\overline{Q} \setminus H_j$ is finite, there exists a state $\overline{r} \notin H_j$, such that

$$\mathsf{Pr}^{\mathcal{P}_R}\big\{\overline{\pi} : \overline{\pi} \text{ is a run for } \omega \text{ such that } \overline{\pi} \models \Diamond^{=m}\overline{r} \ \wedge \ \Box^{\geq m}\neg H_j \ \wedge \ \Box\Diamond K_j \big\} \ > \ 0,$$

where $\overline{\pi} \models \Diamond^{=m}\overline{r}$ if and only if $\overline{\pi}^m = \overline{r}$.

It follows from the transition relation of $\mathcal{P}_R$ that whenever there is a transition from a state $\overline{q} \in \overline{Q}_i$ to a state $\overline{p} \in Q_j$, where $i < j$, then the jth bit in $\overline{p}$ is set to 1. Thus, $\overline{Q}_{\geq j}$ can only be entered from $\overline{Q}_{<j}$ via a state in H_j and therefore a run that satisfies $\Box^{\geq m}\neg H_j \ \wedge \ \Box\Diamond\overline{Q}_{\geq j}$ satisfies $\Box^{\geq m-1}\overline{Q}_{\geq j}$.

As $K_j \subseteq \overline{Q}_{\geq j}$, the condition $\Diamond^{=m}\overline{r} \ \wedge \ \Box^{\geq m}\neg H_j \ \wedge \ \Box\Diamond K_j$ can only hold for runs

$$\overline{\pi} \ = \ \overline{q}_0, \overline{q}_1, \overline{q}_2, \ldots$$

in $\mathcal{P}_R$ that have an infinite suffix $\overline{q}_m, \overline{q}_{m+1}, \overline{q}_{m+2}, \ldots$ consisting of states $\overline{q}_i = \langle p_{1,i}, \xi_{1,i}, \ldots, p_{j,i}, \xi_{j,i}, \ldots, R_i \rangle$ in $\overline{Q}_{\geq j}$ where $\xi_{j,i} = 0$ for all $i \geq m$. Moreover $\overline{q}_m = \overline{r}$ and there are infinitely many indices i such that $p_{j,i} \in F$.

But then the projection to the $(2j-1)$-st components in $\overline{q}_m, \overline{q}_{m+1}, \overline{q}_{m+2}, \ldots$ yields an infinite suffix $p_{j,m}, p_{j,m+1}, p_{j,m+2}, \ldots$ of an accepting run for ω in $\mathcal{P}$. Furthermore, state $r_j = p_{j,m}$ is reachable from an initial state $q_0 \in Q_{\mathsf{init}}$ via a run for the prefix $a_1 \ldots a_m$ of ω, where r_j denotes the $(2j-1)$st component of $\overline{r}$. Thus,

$$\mathsf{Pr}^{\mathcal{P}}(q_0 \xrightarrow{a_1 \ldots a_m} r_j) \ > \ 0$$

in $\mathcal{P}$. Hence

$$\begin{aligned}\mathsf{Pr}^{\mathcal{P}}_{\mathit{Büchi}}(\omega) \ \geq \ \mathsf{Pr}^{\mathcal{P}}(q_0 \xrightarrow{a_1 \ldots a_m} r_j) \cdot \mathsf{Pr}^{\mathcal{P}_R}\big\{ & \pi \text{ is a run for } \omega \text{ in } \mathcal{P}_R \text{ s.t.} \\ & \pi \models \Diamond^{=m}\overline{r} \ \wedge \ \Box^{\geq m}\neg H_j \ \wedge \ \Box\Diamond K_j \big\}.\end{aligned}$$

Hence, $\mathsf{Pr}^{\mathcal{P}}_{\mathit{Büchi}}(\omega) > 0$, and therefore, $\omega \in \mathcal{L}(\mathcal{P})$.

(2) $\Rightarrow$ (3): We now prove that (2) implies (3). That is, we suppose that $\theta = \mathsf{Pr}^{\mathcal{P}}_{\textit{Büchi}}(\omega) > 0$ and aim to show that $\mathsf{Pr}^{\mathcal{P}_R}_{\textit{Rabin}}(\omega) = 1$. We pick some state $p \in F$ such that

$$\mathsf{Pr}^{\mathcal{P}}\big\{\pi : \pi \text{ is an accepting run for } \omega \text{ such that } \pi \models \Box\Diamond p\,\big\} \;>\; 0.$$

Let $R_i = \delta(Q_{\mathsf{init}}, a_1 \ldots a_i)$ for $i \geq 0$. Then, $p \in R_i \cap F$ for infinitely many $i \in \mathbb{N}_{\geq 1}$. For each such index i, let

$$\theta_i \;=\; \mathsf{Pr}^{\mathcal{P}}_{p}\big\{\pi : \pi \text{ is a run for } a_{i+1}a_{i+2}a_{i+3}\ldots \text{ starting in } p \text{ such that } \pi \models \Box\Diamond p\,\big\}$$

Note that θ_i can be written as a sum

$$\theta_i \;=\; \sum_{j=i+1}^{\infty} \varsigma[i,j] \cdot \theta_j$$

where $\varsigma[i,j]$ denotes the probability of the set of runs $q_i, q_{i+1}, \ldots, q_j$ for the finite subword $a_{i+1} \ldots a_j$ of ω with $q_i = q_j = p$ and $p \notin \{q_{i+1}, \ldots, q_{j-1}\}$. As

$$0 \leq \varsigma[i,j] \leq 1 \quad \text{and} \quad \sum_{j>i} \varsigma[i,j] \leq 1,$$

for each $i \in \mathbb{N}_{\geq 1}$ there exists some $j > i$ with $\theta_i \leq \theta_j$. Hence, there exists an infinite sequence $i_1 < i_2 < i_3 < \ldots$ of natural numbers such that $p \in R_{i_h} \cap F$ for all $h \geq 1$ and

$$0 < \theta = \theta_{i_1} \leq \theta_{i_2} \leq \theta_{i_3} \leq \ldots$$

We now regard the stochastic process induced by $\mathcal{P}_R$ and the input word ω. Let $I = \{i_1, i_2, i_3, \ldots\}$. For each index $i \in I$, the process enters a state

$$\overline{p}_i \;=\; \langle p_{1,i}, \xi_{1,i}, \ldots, p_{k,i}, \xi_{k,i}, R_i\rangle \quad \text{where } p \in R_i \cap F \subseteq \{p_{1,i}, \ldots, p_{k,i}\}.$$

Say $p = p_{j,i}$. With probability θ_i, the state sequence obtained by scanning the suffix $a_{i+1}a_{i+2}a_{i+3}\ldots$ of ω from $p = p_{j,i}$ is a run $p_i, p_{i+1}, p_{i+2}, \ldots$ in $\mathcal{P}$ that visits p infinitely often. Thus, with probability at least θ_i, the stochastic process induced by $\mathcal{P}_R$ and ω will generate from position i on a run $\overline{p}_i, \overline{p}_{i+1}, \overline{p}_{i+2}, \ldots$ where after at most $j-1$ shifts via the ν-operator an infinite suffix $p_i, p_{i+1}, p_{i+2}, \ldots$ (with $p_i = p$) of an accepting run in $\mathcal{P}$ will be generated in the $(2\ell-1)$-st component for some $\ell \leq j$. This holds for each index $i \in I$. Hence, the probability for $\mathcal{P}_R$ to generate an accepting run for ω is at least

$$\begin{aligned}
\sum_{h=1}^{\infty} \prod_{1 \leq k < h} (1-\theta_{i_k}) \cdot \theta_{i_h} \;&=\; \lim_{N\to\infty} \sum_{h=1}^{N} \prod_{1 \leq k < h} (1-\theta_{i_k}) \cdot \theta_{i_h} \\
&=\; \lim_{N\to\infty} \Big(1 - \prod_{1 \leq k \leq N} (1-\theta_{i_k})\Big) \\
&\geq\; \lim_{N\to\infty} \big(1 - (1-\theta)^N\big) \;=\; 1 \,-\, 0 \;=\; 1
\end{aligned}$$

This yields $\mathsf{Pr}^{\mathcal{P}_R}_{\textit{Rabin}}(\omega) = 1$ and shows the Theorem.

$\square$

To finish the proof of Theorem 4.3.1, we still have to show step (3) (the transformation from a PSA to an equivalent PBA).

Remark 4.3.3. In order to obtain a complementation with a single exponential blow-up (as stated in Theorem 4.3.1), the transformation in step (3) must avoid an exponential blow-up. We will give here a transformation from an arbitrary PSA to an equivalent PBA that only causes a polynomial blow-up. This is a remarkable result as in the nondeterministic case the switch from Streett to Büchi acceptance may cause an unavoidable exponential blow-up [SV89]. Note that in the following Theorem we do not exploit the special structure of the 0/1-PSA $\mathcal{P}_S$ that results from step (2).

Theorem 4.3.4 (Polynomial transformation from PSA to PBA).
For any PSA $\mathcal{P}_S$ there exists a PBA $\mathcal{P}$ with $\mathcal{L}_{\mathit{Streett}}(\mathcal{P}_S) = \mathcal{L}(\mathcal{P})$ and $|\mathcal{P}| = \mathcal{O}(m^2|\mathcal{P}_S|)$, where m is the number of acceptance-pairs in $\mathcal{P}_S$.

Proof. Let $\mathcal{P}_S = (Q_S, \Sigma, \delta_S, \mu_S^0, \{(H_1, K_1), \ldots, (H_m, K_m)\})$ be a PSA. For simplicity, we may assume that $H_i \cap K_i = \emptyset$ as otherwise K_i could be replaced with $K_i \setminus H_i$. Intuitively, the PBA $\mathcal{P}$ arises from $\mathcal{P}_S$ by making several copies of $\mathcal{P}_S$: a subautomaton $\mathcal{P}_{init}$ in which the process starts, a subautomaton $\mathcal{P}_{accept}$ which has to be visited infinitely often and which is reachable with positive probability via any outgoing transition from the states in $\mathcal{P}_{init}$, and subautomata $\mathcal{P}_i$ and $\mathcal{P}_{i,j}$ for $i, j \in \{1, \ldots, m\}$, $i \neq j$, that are reached from $\mathcal{P}_{accept}$ whenever a state in K_i is visited in $\mathcal{P}_{accept}$. Subautomaton $\mathcal{P}_i$ can only be left via transitions from a H_i-state in $\mathcal{P}_i$ from which we move back to $\mathcal{P}_{accept}$. Subautomaton $\mathcal{P}_{i,j}$ can behave as $\mathcal{P}_i$, but in addition it also serves to take care about the Streett-acceptance pair (H_j, K_j). When we reach a K_j-state in $\mathcal{P}_{i,j}$ then we randomly choose to stay in $\mathcal{P}_{i,j}$ or to move to $\mathcal{P}_j$ or one of the subautomata $\mathcal{P}_{j,k}$. Formally, the PBA $\mathcal{P} = (Q, \Sigma, \delta, \mu_0, F)$ is defined as follows. The state space is

$$Q = Q_{init} \cup Q_{accept} \cup \bigcup_{1 \leq i \leq m} Q_i \cup \bigcup_{\substack{1 \leq i,j \leq m \\ i \neq j}} Q_{i,j}$$

where $Q_* = \{\langle q, * \rangle : q \in Q_S\}$. The set of accepting states is $F = Q_{accept}$. The initial distribution is given by $\mu_0(\langle q, \mathit{init} \rangle) = \mu_S^0(q)$ and $\mu_0(\langle q, * \rangle) = 0$ for all other states $\langle q, * \rangle \in Q$.

The transition probabilities in $\mathcal{P}$ are shown in Figure 4.7 where $q, p \in Q_S$. Here, i, j, k range over all indices in $\{1, \ldots, m\}$ with $i \neq j$ and $j \neq k$ (but possibly $i = k$). In the sequel, we refer to the fragment of the Q_*-states as the $\mathcal{P}_*$-subautomaton.

Why do we need the $\mathcal{P}_{i,j}$ subautomata? The problem is that without the $\mathcal{P}_{i,j}$ subautomata, the Büchi automata could visit infinitely many K_2 states while being in $\mathcal{P}_1$ and thus would not take care of also visiting infinitely many H_2 states, but might nevertheless accept as the following example shows. Assume the above construction without the $\mathcal{P}_{i,j}$ subautomata. For simplicity assume that the acceptance condition consists of two pairs $\{(H_1, K_1), (H_2, K_2)\}$ such that $K_1 \setminus K_2 \neq \emptyset$. Let $k_1 \in K_1 \setminus K_2, k_2 \in K_2, h_1 \in H_1$. W.l.o.g assume $k_2 \notin H_1$. Then the nonaccepting (possible) path

$$k_1, k_2, h_1, k_1, k_2, h_1, \ldots$$

$$
\begin{array}{lll}
\delta(\langle q, init\rangle, a, \langle p, init\rangle) & = & \frac{1}{2}\cdot\delta_S(q,p) \\
\delta(\langle q, init\rangle, a, \langle p, accept\rangle) & = & \frac{1}{2}\cdot\delta_S(q,p) \\
\\
\delta(\langle q, accept\rangle, a, \langle p, accept\rangle) & = & \delta_S(q,p) \qquad \text{if } q\notin K_1\cup\ldots\cup K_m \\
\delta(\langle q, accept\rangle, a, \langle p, i\rangle) & = & \frac{1}{|\{\ell|q\in K_\ell\}|\cdot m}\cdot\delta_S(q,p) \quad \text{if } q\in K_i \\
\delta(\langle q, accept\rangle, a, \langle p, i, j\rangle) & = & \frac{1}{|\{\ell|q\in K_\ell\}|\cdot m}\cdot\delta_S(q,p) \quad \text{if } q\in K_i \\
\\
\delta(\langle q, i\rangle, a, \langle p, accept\rangle) & = & \delta_S(q,p) \qquad \text{if } q\in H_i \\
\delta(\langle q, i\rangle, a, \langle p, i\rangle) & = & \delta_S(q,p) \qquad \text{if } q\notin H_i \\
\\
\delta(\langle q, i, j\rangle, a, \langle p, accept\rangle) & = & \delta_S(q,p)\cdot\begin{cases}0, q\notin H_i\cup K_j\\ 0, q\in K_j\setminus H_i\\ 1, q\in H_i\setminus K_j\\ \frac{1}{m+2}, q\in H_i\cap K_j\end{cases} \\
\\
\delta(\langle q, i, j\rangle, a, \langle p, i, j\rangle) & = & \delta_S(q,p)\cdot\begin{cases}1, q\notin H_i\cup K_j\\ \frac{1}{m+1}, q\in K_j\setminus H_i\\ 0, q\in H_i\setminus K_j\\ \frac{1}{m+2}, q\in H_i\cap K_j\end{cases} \\
\\
\delta(\langle q, i, j\rangle, a, \langle p, j\rangle) & = & \delta_S(q,p)\cdot\begin{cases}0, q\notin H_i\cup K_j\\ \frac{1}{m+1}, q\in K_j\setminus H_i\\ 0, q\in H_i\setminus K_j\\ \frac{1}{m+2}, q\in H_i\cap K_j\end{cases} \\
\\
\delta(\langle q, i, j\rangle, a, \langle p, j, k\rangle) & = & \delta_S(q,p)\cdot\begin{cases}0, q\notin H_i\cup K_j\\ \frac{1}{m+1}, q\in K_j\setminus H_i\\ 0, q\in H_i\setminus K_j\\ \frac{1}{m+2}, q\in H_i\cap K_j\end{cases}
\end{array}
$$

Figure 4.7: Transition probabilities of the PBA constructed in the proof of Theorem 4.3.4 (where $i, j, k \in \{1, \ldots, m\}$ s.th. $i \neq j$ and $j \neq k$ (but possibly $i = k$))

in the Streett automaton would be lifted with positive probability to accepting paths of the form

$$\ldots, \langle k_1, accept\rangle, \langle k_2, 1\rangle, \langle h_1, 1\rangle, \langle k_1, accept\rangle, \langle k_2, 1\rangle, \langle h_1, 1\rangle, \ldots$$

in the Büchi automaton. To avoid this, we need the $\mathcal{P}_{i,j}$ subautomata.

And why do we need the $\mathcal{P}_i$ subautomata? Why are the $\mathcal{P}_{i,j}$ subautomata not sufficient? The problem is that without the $\mathcal{P}_i$ subautomata, the Büchi automata could visit infinitely many K_1 states while being in $\mathcal{P}_{accept}$ moving to $\mathcal{P}_{1,2}$. If the automaton also visits infinitely many K_2 states but no H_1 states, the Büchi automaton will almost surely leave $\mathcal{P}_{1,2}$ and move to $\mathcal{P}_{2,1}$ which it can leave to $\mathcal{P}_{accept}$ if it visits infinitely many H_2 states. Thus it could accept also it might not satisfy the Streett condition $\{(H_1, K_1)\}$. This is shown in the following example. Assume the above construction without the $\mathcal{P}_i$ subautomata. Assume that the acceptance condition consists of two pairs $\{(H_1, K_1), (H_2, K_2)\}$ such that $K_1 \setminus K_2 \neq \emptyset$, $K_2 \setminus H_1 \neq \emptyset$ and $H_2 \setminus K_1 \neq \emptyset$. Let $k_1 \in K_1 \setminus K_2, k_2 \in K_2 \setminus H_1, h_2 \in H_2 \setminus K_1$. Then the nonaccepting (possible) path

$$k_1, k_2, h_2, k_1, k_2, h_2, \ldots$$

in the Streett automaton would be lifted with positive probability to accepting paths of the form

$$\ldots, \langle k_1, accept\rangle, \langle k_2, 1, 2\rangle, \langle h_2, 2, 1\rangle, \langle k_1, accept\rangle, \langle k_2, 1, 2\rangle, \langle h_2, 2, 1\rangle, \ldots$$

in the Büchi automaton. To avoid this, we need the $\mathcal{P}_i$ subautomata.

In the following we will denote by $\mathsf{Acc}^{\mathcal{P}_S}_{\mathit{Streett}} = \wedge_{1\le j\le m}(\Box\Diamond K_j \Rightarrow \Box\Diamond H_j)$ the Streett acceptance condition of $\mathcal{P}_S$ and by $\mathsf{Acc}^{\mathcal{P}_S}_{\mathit{Rabin}} = \vee_{1\le j\le m}(\Diamond\Box\neg H_j \wedge \Box\Diamond K_j)$ the acceptance condition that we gain from the acceptance pairs of $\mathcal{P}_S$ by interpreting them as a Rabin acceptance condition. We now show that $\mathcal{L}_{\mathit{Büchi}}(\mathcal{P}) = \mathcal{L}_{\mathit{Streett}}(\mathcal{P}_S)$

$\subseteq$: Let $\omega \notin \mathcal{L}_{\mathit{Streett}}(\mathcal{P}_S)$, thus $\mathsf{Pr}^{\mathcal{P}_S,\omega}(\{\pi \mid \pi \models \mathsf{Acc}^{\mathcal{P}_S}_{\mathit{Streett}}\}) = 0$ and hence it holds that $\mathsf{Pr}^{\mathcal{P}_S,\omega}(\{\pi \mid \pi \models \mathsf{Acc}^{\mathcal{P}_S}_{\mathit{Rabin}}\}) = 1$. Consider a run π of $\mathcal{P}_S$ that satisfies the Rabin condition $\vee_{1\le j\le m}(\Diamond\Box\neg H_j \wedge \Box\Diamond K_j)$, thus there exist an index j such that $\pi \models \Diamond\Box\neg H_j \wedge \Box\Diamond K_j$. Consider the liftings of π in the constructed Büchi automaton $\mathcal{P}$. (By a lifting of π we mean any run in $\mathcal{P}$ for ω whose projection to the Q_S-components agrees with π.) As the above construction ensures that whenever a K_j-state is visited in $\mathcal{P}_{accept}$ or $\mathcal{P}_{i,j}$ for some $i \neq j$ then with equal positive probability one of the subautomaton $\mathcal{P}_j$ or $\mathcal{P}_{j,k}$ is entered. Hence, if infinitely often a K_j-state is visited and the process does not stay forever in one of the subautomata $\mathcal{P}_i$ (for some $i \neq j$) or $\mathcal{P}_{k,\ell}$ (where it cannot accept), then almost surely $\mathcal{P}_j$ is entered. But $\mathcal{P}_j$ can only be left via a H_j-state. As $\pi \models \Diamond\Box\neg H_j$ this ensures that almost all liftings of π will eventually stay in one of the subautomata $\mathcal{P}_i$ or $\mathcal{P}_{i,k}$, hence they will almost surely not be accepting. This shows that $\mathsf{Pr}^{\mathcal{P}_S,\omega}(\{\pi \mid \pi \models \mathsf{Acc}^{\mathcal{P}_S}_{\mathit{Rabin}}\}) = 1 \Rightarrow \mathsf{Pr}^{\mathcal{P}}_{\mathit{Büchi}}(\omega) = 0$ and $\omega \notin \mathcal{L}_{\mathit{Streett}}(\mathcal{P}_S) \Rightarrow \omega \notin \mathcal{L}_{\mathit{Büchi}}(\mathcal{P})$.

$\supseteq$: Let $\omega \in \mathcal{L}_{\mathit{Streett}}(\mathcal{P}_S)$, thus $\mathsf{Pr}^{\mathcal{P}_S,\omega}(\{\pi \mid \pi \models \wedge_{1\le i\le m}(\Box\Diamond K_i \Rightarrow \Box\Diamond H_i)\}) > 0$. As

$$\{\pi \mid \pi \models \wedge_{1\le i\le m}(\Box\Diamond K_i \Rightarrow \Box\Diamond H_i)\} = \dot{\bigcup_{J\subseteq\{1,\ldots,m\}}} \Big\{\pi \mid \pi \models \mathsf{Acc}^{\mathcal{P}_S}_{\mathit{Streett}} \wedge \bigwedge_{j\in J} \Box\Diamond K_j \wedge \bigwedge_{j\notin J} \Diamond\Box\neg K_j\Big\}$$

there exists $J \subseteq \{1, \ldots, m\}$ such that

$$\mathsf{Pr}^{\mathcal{P}_S,\omega}(\{\pi \mid \pi \models \mathsf{Acc}^{\mathcal{P}_s}_{\mathit{Streett}} \wedge \wedge_{j\in J}\Box\Diamond K_j \wedge \wedge_{j\notin J}\Diamond\Box\neg K_j\}) > 0.$$

As $\Diamond\Box\neg K_j$ is the disjoint union of $\Diamond^{=\ell-1}K_j \wedge \Box^{\ge\ell}\neg K_j, \ell = 0, 1, 2, \ldots$, and as $\{j \mid j \notin J\}$ is finite there exists $r \in \mathbb{N}_{\ge 0}$ such that

$$\mathsf{Pr}^{\mathcal{P}_S,\omega}(\{\pi \mid \pi \models \mathsf{Acc}^{\mathcal{P}_s}_{\mathit{Streett}} \wedge \wedge_{j\in J}\Box\Diamond K_j \wedge \wedge_{j\notin J}\Box^{>r}\neg K_j\}) > 0.$$

Let $\pi_S \in \{\pi \mid \pi \models \mathsf{Acc}^{\mathcal{P}_s}_{\mathit{Streett}} \wedge \wedge_{j\in J}\Box\Diamond K_j \wedge \wedge_{j\notin J}\Box^{>r}\neg K_j\}$ be a run in $\mathcal{P}_S$. Then almost all liftings of π_S to runs for ω in $\mathcal{P}$ that stay in $\mathcal{P}_{init}$ for the first r input symbols and eventually enter $\mathcal{P}_{accept}$ are accepting. (By a lifting of π_S we mean any run in $\mathcal{P}$ for ω whose projection to the Q_S-components agrees with π_S.) This yields

$$\mathsf{Pr}^{\mathcal{P}}_{\mathit{Büchi}}(\omega) \ge \frac{1}{2^{r+1}} \cdot \mathsf{Pr}^{\mathcal{P}_S,\omega}(\{\pi \mid \pi \models \mathsf{Acc}^{\mathcal{P}_s}_{\mathit{Streett}} \wedge \wedge_{j\in J}\Box\Diamond K_j \wedge \wedge_{j\notin J}\Box^{>r}\neg K_j\}) > 0$$

and $\omega \in \mathcal{L}_{\mathit{Büchi}}(\mathcal{P})$.

□

This completes the proof of Theorem 4.3.1 which states that the class of PBA-definable languages is closed under complementation. The main ingredients for the proof are a transformation from PBA to equivalent 0/1 PRA (Theorem 4.3.2) and a polynomial transformation from PSA to PBA (Theorem 4.3.4). The complexity of the latter transformation is worth noting, as in the nondeterministic case the switch from Streett to Büchi acceptance can cause an exponential blow-up [SV89]. To conclude this section, we want to examine the construction from a given PBA to an equivalent 0/1 PRA for a simple example.

4.3.2.1. Example for the transformation from PBA to 0/1 PRA

We consider again the automaton $\mathcal{P}_\lambda$ from Figure 4.3, page 68. This automaton has two remarking properties, namely it accepts a non-ω-regular language and its accepted language depends on the precise transition probabilities. In Figure 4.8 we depict the automaton $\mathcal{P}$ which basically is $\mathcal{P}_{\frac{3}{4}}$, but $\mathcal{P}$ is also total. Given a word $\omega = a^{k_1}ba^{k_2}ba^{k_3}b\ldots \in \mathcal{L}(\mathcal{P})$,

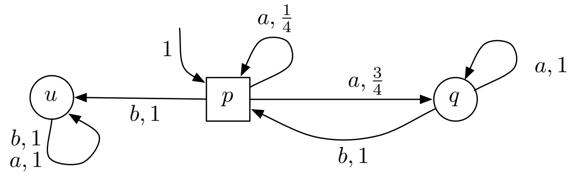


Figure 4.8: Example for the transformation from PBA to 0/1 PRA

it holds that $\mathsf{Pr}^{\mathcal{P}}(\omega) = \prod_{i=1}^{\infty}(1-(\frac{1}{4})^{k_i}) > 0$. Applying the transformation described in the proof of Theorem 4.3.2 yields the 0/1 PRA $\mathcal{P}_R$ depicted in Figure 4.9. For the sake of readability we denote the states in $\mathcal{P}_R$ (which consist of a sequence of states of $\mathcal{P}$, each with an associated bit, and a subset of the states of $\mathcal{P}$) by e.g. $\langle u_0, p_1, \{p,u\}\rangle$ instead of $\langle u, 0, p, 1, \{p,u\}\rangle$. That is, the associated bit is subscripted. The crucial parts of the automaton $\mathcal{P}_R$ are two very similar subautomata that we denote by $\mathcal{P}_{\mathsf{left}}$ and $\mathcal{P}_{\mathsf{right}}$. $\mathcal{P}_{\mathsf{left}}$ consists of the states in the left dashed rectangular box and the left dotted parallelogram and $\mathcal{P}_{\mathsf{right}}$ consists of the states in the right dashed rectangular box and the right dotted parallelogram. Both $\mathcal{P}_{\mathsf{left}}$ and $\mathcal{P}_{\mathsf{right}}$ simulate $\mathcal{P}$, the two states in the corresponding dashed box simulate the state q of $\mathcal{P}$ and the two states in the corresponding dotted parallelogram simulate the state p. Note that the automaton $\mathcal{P}$ basically rejects if it reads the letter b in state p. This is simulated in $\mathcal{P}_R$ as follows. If two consecutive b's are read then $\mathcal{P}_R$ moves with the first b from the dashed box to the lower state of the dotted parallelogram. With the second b, it moves to state $\langle u_0, \{u\}\rangle$ from which it can never accept. If a word $a^k b$ is read, then $\mathcal{P}$'s behavior is simulated, but instead of rejecting with probability $(\frac{1}{4})^k$, $\mathcal{P}_R$ moves to the state $\langle u_0, p_1, \{p,u\}\rangle$ from where the process of $\mathcal{P}$ is simulated in the subautomaton $\mathcal{P}_{\mathsf{left}}$ for the remaining suffix of the input word. Note that for an input word $a^{k_j}ba^{k_{j+1}}b\ldots$, the probability that the occurring b's are not read in the parallelogram but in the box is $\prod_{i=j}^{\infty}(1-(\frac{1}{4})^{k_i}) > \prod_{i=1}^{\infty}(1-(\frac{1}{4})^{k_i}) > 0$ if $a^{k_1}ba^{k_2}b\ldots \in \mathcal{L}(\mathcal{P})$. Thus with

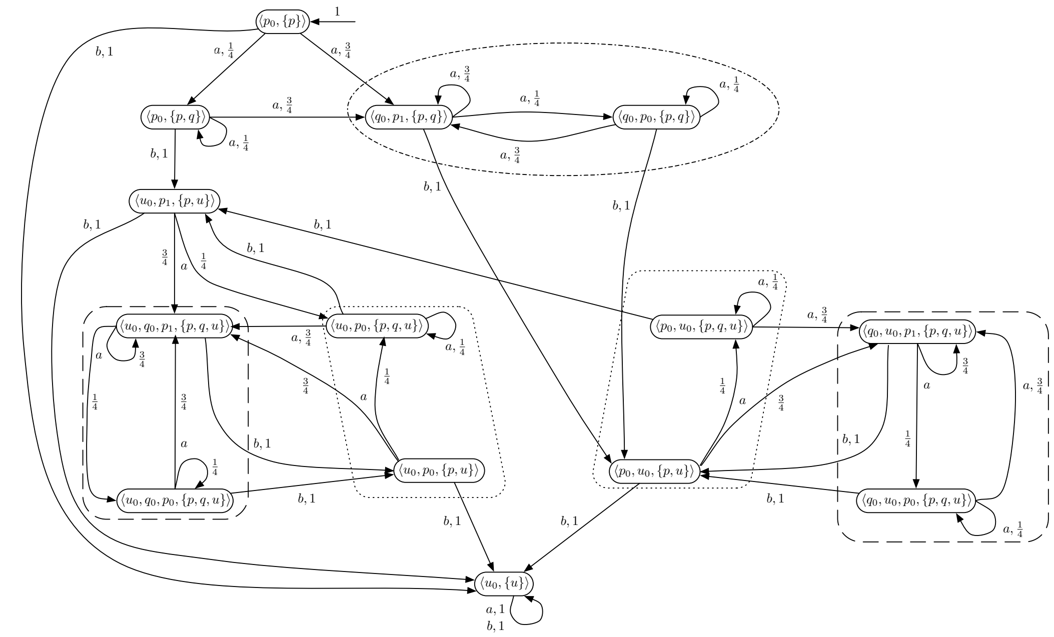


Figure 4.9: The resulting 0/1 PRA $\mathcal{P}_R$

positive probability (bounded from below), the process stays in one of the subautomata $\mathcal{P}_{\mathsf{left}}$ or $\mathcal{P}_{\mathsf{right}}$ where it accepts (in the second component for the automaton $\mathcal{P}_{\mathsf{left}}$ and in the first component for the automaton $\mathcal{P}_{\mathsf{right}}$). This ensures that it accept the words in $\mathcal{L}(\mathcal{P})$ with probability 1.

Note that a word with only finitely many b's will not be accepted as almost all runs are not accepting. If the automaton enters a dashed box (after reading the last b), it will almost surely reject, as it will visit both states in the box almost surely. But such a run does not satisfy the Rabin acceptance condition as there is only an accepting state of $\mathcal{P}$ (namely p) in the third component, but one of the states in the box has the p in the third component associated with the bit 1. If the input word contains no b the same reasoning applies to the two states in the oval.

This examples concludes the discussion of the complementation and we will have a look at decidability issues in the next section.

4.4. Decidability Questions

For finite probabilistic automata (PFA) it has been shown that the emptiness problem is undecidable. Recall that a PFA $\mathcal{P}_{\mathsf{fin}}$ is equipped with a threshold $0 < \lambda < 1$ and that the accepted language $\mathcal{L}(\mathcal{P}_{\mathsf{fin}}, \lambda)$ consists of all input words for which the set of runs that end in an accepting state has a probability greater than λ. From this it easily follows that the emptiness problem for PBA under such a threshold semantics (see Subsection 4.4.1.4) is undecidable. This can be seen as a given PFA $\mathcal{P}_{\mathsf{fin}}$ over the alphabet Σ can be transformed into a PBA $\mathcal{P}$ such that for each finite word $\rho \in \Sigma^*$ it holds that $\mathsf{Pr}^{\mathcal{P}_{\mathsf{fin}}}(\rho) = \mathsf{Pr}^{\mathcal{P}}(\rho c^\omega)$, where c is an additional letter which is not in Σ. Moreover the automaton $\mathcal{P}$ can only produce (Büchi) accepting runs for input words that are of the form ρc^ω, where ρ is in Σ^* (for each accepting state of the PFA $\mathcal{P}_{\mathsf{fin}}$ add a c-transition with probability one to a new state q_F which then is the only accepting state of the PBA $\mathcal{P}$ and has a c-loop attached to it with probability one).

In this section we will show some undecidability results for PBA that have an immediate consequence for related problems of POMDPs (as PBA can be seen as a special instance of POMDPs). We will moreover proof the almost-sure Büchi objective to be decidable for POMDPs which then implies the decidability of the emptiness problem for PBA under an almost-sure semantics.

4.4.1. Undecidability results

In contrast to finite probabilistic automata where the threshold $\lambda = 0$ makes the emptiness problem trivially decidable as a PFA with that threshold accepts the same language as its underlying deterministic finite automaton, we will show in this subsection that the emptiness problem for PBA (and threshold 0) is undecidable. We also discuss the consequences of this result for other problems for PBA and POMDPs.

4.4.1.1. Emptiness problem for PBA

The proof for the undecidability of the emptiness problem for PBA relies on a reduction from a variant of the emptiness problem for PFA, using the fact that modifying the transition probabilities can affect the accepted language of a PBA (Theorem 4.2.6). The emptiness problem for PFA is known to be undecidable [Paz71]. Here we use the following variant of this result, due to Madani, Hanks and Condon [MHC03].

Theorem 4.4.1 (Undecidability result for PFA, cf. [MHC03]).
The following problem is undecidable: Given a constant $0 < \varepsilon < 1$ and a PFA that either accepts some string with probability at least $1 - \varepsilon$ or accepts all strings with probability at most ε, decide which is the case.

Using this result we are able to show

Theorem 4.4.2 (Undecidability of the emptiness problem for PBA).
Checking emptiness is undecidable for PBA.

Proof. To provide an undecidability proof of the emptiness problem for PBA, we reduce the variant of the emptiness problem for PFA recalled in Theorem 4.4.1 to the *intersection problem for PBA* which takes as input two PBA $\mathcal{P}_1$ and $\mathcal{P}_2$ and asks whether $\mathcal{L}(\mathcal{P}_1) \cap \mathcal{L}(\mathcal{P}_2)$ is empty. As PBA are closed under intersection (see subsection 4.3.1), this will complete the proof for Theorem 4.4.2.

Let $\mathcal{R}$ be a PFA over some alphabet Σ and $0 < \varepsilon < \frac{1}{2}$ as in Theorem 4.4.1, i.e. such that either there exists some word ρ accepted by $\mathcal{R}$ with probability strictly greater than

$1-\varepsilon$, or all words are accepted with probability less than ε. For $\rho \in \Sigma^*$, let $\mathsf{Pr}^{\mathcal{R}}(\rho)$ denote the probability that the word ρ is accepted by $\mathcal{R}$. From the PFA $\mathcal{R}$ and the constant ε we construct two PBA $\mathcal{P}_1$ and $\mathcal{P}_2$ such that

$$\mathcal{L}^{>\varepsilon}(\mathcal{R}) = \emptyset \text{ if and only if } \mathcal{L}(\mathcal{P}_1) \cap \mathcal{L}(\mathcal{P}_2) = \emptyset,$$

where $\mathcal{L}^{>\varepsilon}(\mathcal{R}) = \{\rho \in \Sigma^* \mid \mathsf{Pr}^{\mathcal{R}}(\rho) > \varepsilon\}$. The alphabet for both $\mathcal{P}_1$ and $\mathcal{P}_2$ arise from the alphabet Σ of $\mathcal{R}$ by adding two new symbols $\sharp$ and $\$$, that is, $\mathcal{P}_1$ and $\mathcal{P}_2$ are PBA over the alphabet $\Sigma' = \Sigma \cup \{\sharp, \$\}$. The rough idea is to use the somehow complementary acceptance behavior of the automata $\mathcal{P}_\lambda$ and $\widetilde{\mathcal{P}}_\lambda$ (see Figure 4.3 and 4.4 on page 68, resp. 69). The automata $\mathcal{P}_1$ and $\mathcal{P}_2$ are designed to read words of the form $\rho_1^1\sharp\rho_2^1\sharp\cdots\rho_{k_1}^1\$\$\rho_1^2\sharp\rho_2^2\sharp\cdots\rho_{k_2}^2\$\$\cdots$ where $\rho_i^j \in \Sigma^*$. Roughly speaking, $\mathcal{P}_1$ will mimick the automaton $\mathcal{P}_\lambda$ and $\mathcal{P}_2$ will mimick $\widetilde{\mathcal{P}}_\lambda$, where reading a word $\rho_i^j\sharp$ in $\mathcal{P}_1$ (resp. $\mathcal{P}_2$) corresponds to reading a single letter a in $\mathcal{P}_\lambda$ (resp. $\widetilde{\mathcal{P}}_\lambda$). Recall that $\mathcal{P}_\lambda$ and $\widetilde{\mathcal{P}}_\lambda$ accept infinite words of the form $a^{k_1}ba^{k_2}b\ldots$ (depending on the k_i). The two $\$$-symbols serve as a separator for $\mathcal{P}_1$ and $\mathcal{P}_2$, just like the letter b does for $\mathcal{P}_\lambda$ and $\widetilde{\mathcal{P}}_\lambda$. Thus, the number of $\sharp$-symbols between the $(j-1)$st and the jth occurence of $\$\$$ (and therefore the number of words ρ_i^j) corresponds to the value of k_j. Automaton $\mathcal{P}_1$ evolves from the automaton $\mathcal{P}_\lambda$ by replacing each of its two states p_0, p_1 by a copy of the PFA $\mathcal{R}$. The transitions for the $\sharp$-symbol will be defined, such that after reading a word $\rho_i^j\sharp$ in the copy of $\mathcal{R}$ that corresponds to the state p_0 (recall that this corresponds to reading a single letter a in state p_0 of in $\mathcal{P}_\lambda$) the automaton $\mathcal{P}_1$ is still in this copy of $\mathcal{R}$ with probability $1 - \mathsf{Pr}^{\mathcal{R}}(\rho_i^j)$ and has moved to the other copy with probability $\mathsf{Pr}^{\mathcal{R}}(\rho_i^j)$, similar to the behavior of automaton $\mathcal{P}_\lambda$ upon reading the letter a in state p_0 (it stays in p_0 with probability $1-\lambda$ and moves to p_1 with probability λ). The structure of $\mathcal{P}_1$ and $\mathcal{P}_2$ is shown in Figure 4.10 and 4.11, respectively. The PBA $\mathcal{P}_1$ is com-

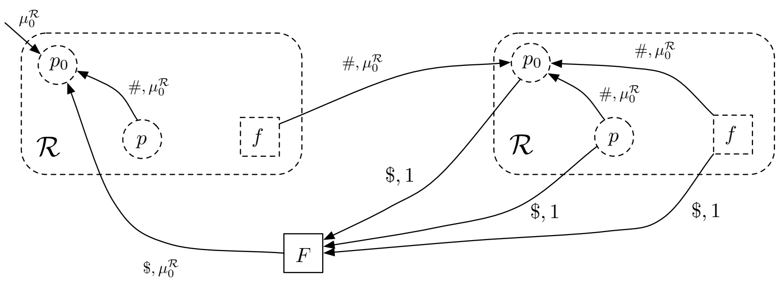


Figure 4.10: PBA $\mathcal{P}_1$

posed of two copies of the PFA $\mathcal{R}$ (respresented in dashed lines) augmented with new edges using the additional symbols $\sharp$ and $\$$. The initial states of $\mathcal{P}_1$ are the initial states of the first copy of $\mathcal{R}$ according to the initial distribution of $\mathcal{R}$. Reading the symbol $\sharp$ in any final state of the first copy of $\mathcal{R}$, the PBA $\mathcal{P}_1$ proceeds to the initial state of $\mathcal{R}$ in the second copy according to the initial distribution of $\mathcal{R}$. Reading the symbol $\sharp$ in any non-final state of the first copy of $\mathcal{R}$, the PBA $\mathcal{P}_1$ proceeds to the initial state of $\mathcal{R}$ in the first copy according to the initial distribution of $\mathcal{R}$. Consuming the symbol $\$$ in some (final or non-final) state of the second copy of $\mathcal{R}$, $\mathcal{P}_1$ moves with probability 1 to the special state F, which is the unique accepting state of $\mathcal{P}_1$. Reading the second \$ symbol, $\mathcal{P}_1$ proceeds on to an initial state according to the initial distribution of $\mathcal{R}$.

The accepted language of this PBA is (see end of this section (page 94))

$$L_1 = \Big\{\rho_1^1\sharp\rho_2^1\sharp\ldots\rho_{k_1}^1\$\$\,\rho_1^2\sharp\rho_2^2\sharp\ldots\rho_{k_2}^2\$\$\,\ldots \mid \rho_i^j \in \Sigma^* \text{ and } \prod_{j\geq 1}\Big(1 - \big(\prod_{i=1}^{k_j-1}(1-\mathsf{Pr}^{\mathcal{R}}(\rho_i^j))\big)\Big) > 0\Big\}.$$

PBA $\mathcal{P}_2$ (Fig. 4.11) does not depend on the structure of the given PFA $\mathcal{R}$, but only on ε and the alphabet Σ.

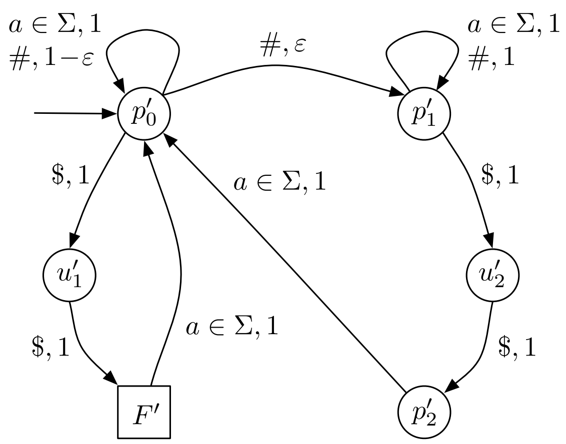


Figure 4.11: PBA $\mathcal{P}_2$

Its accepted language is (see end of this section (page 94))

$$L_2 = \Big\{v_1\$\$\,v_2\$\$\,\ldots \mid v_i \in (\Sigma\cup\{\sharp\})^* \text{ and } \prod_{i\geq 1}\big(1-(1-\varepsilon)^{|v_i|_\sharp}\big) = 0\Big\},$$

where $|v|_\sharp$ is the number of $\sharp$ symbols in the word $v \in (\Sigma\cup\{\sharp\})^*$.

We now show that the language $\mathcal{L}^{>\varepsilon}(\mathcal{R}) = \{\rho \in \Sigma^* \mid \mathsf{Pr}^{\mathcal{R}}(\rho) > \varepsilon\}$ of $\mathcal{R}$ for the threshold ε is empty if and only if $L_1 \cap L_2 = \emptyset$.

"$\Longrightarrow$": Assume that $\mathcal{L}^{>\varepsilon}(\mathcal{R})$ is empty, i.e. for all finite words $\rho \in \Sigma^*$ it holds that $\mathsf{Pr}^{\mathcal{R}}(\rho) \leq \varepsilon$. Let $\tilde{\omega} \in L_2$. The goal is to prove that $\tilde{\omega} \notin L_1$. Since $\tilde{\omega} \in L_2$, $\tilde{\omega}$ can be written as

$$\tilde{\omega} = v_1\$\$\,v_2\$\$\,\ldots \text{ with } v_i \in (\Sigma\cup\{\sharp\})^* \text{ and } \textstyle\prod_i\big(1-(1-\varepsilon)^{|v_i|_\sharp}\big) = 0.$$

The subwords v_i can be decomposed according to the occurrences of the symbol $\sharp$. That is,

$$\tilde{\omega} = \rho_1^1\sharp\rho_2^1\sharp\ldots\rho_{k_1}^1\$\$\,\rho_1^2\sharp\rho_2^2\sharp\ldots\rho_{k_2}^2\$\$\,\ldots \text{ with } |v_i|_\sharp = k_i - 1.$$

Hence $\tilde{\omega} \in L_2$ implies $\prod_i \big(1 - (1-\varepsilon)^{k_i - 1}\big) = 0$. However:

$$\begin{aligned}
\prod_j \Big(1 - \prod_{i=1}^{k_j - 1} \big(1 - \mathsf{Pr}^{\mathcal{R}}(\rho_i^j)\big)\Big) &\leq \prod_j \Big(1 - \prod_{i=1}^{k_j-1} (1-\varepsilon)\Big) \quad \text{since } \mathcal{L}^{>\varepsilon}(\mathcal{R}) = \emptyset \\
&= \prod_j \Big(1 - (1-\varepsilon)^{k_j - 1}\Big) \\
&= 0 \qquad\quad \text{since } \tilde{\omega} \in L_2.
\end{aligned}$$

Hence, $\tilde{\omega} \notin L_1$. Since this holds for any $\tilde{\omega} \in L_2$, we conclude that $L_1 \cap L_2 = \emptyset$.

"$\Longleftarrow$": Assume now that $\mathcal{L}^{>\varepsilon}(\mathcal{R}) \neq \emptyset$. By assumption on the PFA $\mathcal{R}$, this means that there exists a finite word $\rho \in \Sigma^*$ such that $\mathsf{Pr}^{\mathcal{R}}(\rho) > 1 - \varepsilon$. We define

$$\tilde{\omega}_{k_1,k_2,\ldots} = (\rho\sharp)^{k_1} \rho\$\$ \, (\rho\sharp)^{k_2} \rho\$\$ \, \ldots,$$

and prove that there exists a sequence $k_1, k_2, \ldots$, such that $\tilde{\omega}_{k_1,k_2,\ldots} \in L_1 \cap L_2$. The acceptance probability of $\tilde{\omega}_{k_1,k_2,\ldots}$ in $\mathcal{P}_1$ is

$$\begin{aligned}
\prod_j \Big(1 - \prod_{i=1}^{k_j} \big(1 - \mathsf{Pr}^{\mathcal{R}}(\rho)\big)\Big) &= \prod_j \Big(1 - \big(1 - \mathsf{Pr}^{\mathcal{R}}(\rho)\big)^{k_j}\Big) \\
&> \prod_j \Big(1 - \big(1 - (1-\varepsilon)\big)^{k_j}\Big) \\
&= \prod_j \Big(1 - \varepsilon^{k_j}\Big)
\end{aligned}$$

On the other hand, the word $\tilde{\omega}_{k_1,k_2,\ldots}$ can be written as $v_1\$\$v_2\$\$\ldots$ with $v_i \in (\Sigma \cup \{\sharp\})^*$ and $|v_i|_\sharp = k_i$. Hence, $\prod_i \big(1 - (1-\varepsilon)^{|v_i|_\sharp}\big) = \prod_i \big(1 - (1-\varepsilon)^{k_i}\big)$. We finally apply Lemma 4.2.7 (with $n = 2$) which yields the existence of a sequence $(k_i')_{i \geq 1}$ that will ensure at the same time (recall that $0 < \varepsilon < \frac{1}{2}$)

$$\prod_{j \geq 1} \Big(1 - \varepsilon^{k_j'}\Big) > 0 \text{ and } \prod_{i \geq 1} \Big(1 - (1-\varepsilon)^{k_i'}\Big) = 0.$$

Hence, $\tilde{\omega}_{k_1',k_2',\ldots} \in L_1 \cap L_2$ and $\mathcal{L}(\mathcal{P}_1) \cap \mathcal{L}(\mathcal{P}_2) \neq \emptyset$.

This completes the proof of Theorem 4.4.2. □

4.4.1.2. Consequences of the undecidability of the emptiness problem

Since complementation is effective for PBA, the undecidability of the emptiness problem yields immediately that many other interesting algorithmic problems for PBA are undecidable too.

Corollary 4.4.3 (Other undecidability results for PBA). Given two PBA $\mathcal{P}_1$ and $\mathcal{P}_2$, the following problems are undecidable.

$$\begin{array}{rlcl} \text{universality:} & \mathcal{L}(\mathcal{P}_1) & = & \Sigma^\omega? \\ \text{equivalence:} & \mathcal{L}(\mathcal{P}_1) & = & \mathcal{L}(\mathcal{P}_2)? \\ \text{inclusion:} & \mathcal{L}(\mathcal{P}_1) & \subseteq & \mathcal{L}(\mathcal{P}_2)? \end{array}$$

Another immediate consequence of Theorem 4.4.2 is that the verification problem for finite nondeterministic transition systems $\mathcal{T}$ and PBA-specifications is undecidable. Here we assume that the states in $\mathcal{T}$ are labeled with sets of atomic propositions of some finite set AP and consider the traces of the paths in $\mathcal{T}$ that arise by the projection to the labels of the states. Furthermore, we assume that the given PBA has the alphabet 2^{AP}.

Corollary 4.4.4 (Verification against PBA-specifications (i)). The following problems are undecidable.

(a) Given a finite transition system $\mathcal{T}$ and a PBA $\mathcal{P}$, is there a path in $\mathcal{T}$ whose trace is in $\mathcal{L}(\mathcal{P})$?

(b) Given a finite transition system $\mathcal{T}$ and a PBA $\mathcal{P}$, do the traces of all paths in $\mathcal{T}$ belong to $\mathcal{L}(\mathcal{P})$?

Proof. Consider a transition system $\mathcal{T}$ such that each infinite word over the alphabet of $\mathcal{P}$ is a trace of $\mathcal{T}$ and vice versa. Then the emptiness problem for PBA reduces to (a) and the universality problem for PBA reduces to (b). We define $\mathcal{T}$ as follows. Given a PBA with the alphabet $\Sigma = 2^{\mathsf{AP}} = \{a_1, \ldots, a_n\}$, we define the state set of $\mathcal{T}$ to be $S = \{s_1, \ldots, s_n\}$ and the set of actions to be $\mathsf{Act} = \{\alpha_1, \ldots, \alpha_n\}$. Each state s_i is labeled with the set of atomic propositions a_i. There is a transition from each s_i to each s_j, $1 \leq i, j \leq n$ via action α_j and every state is an initial state of $\mathcal{T}$. Thus $\Sigma^\omega = \{trace(\pi) \mid \pi \text{ is an infinite path in } \mathcal{T}\}$. □

As transition systems are special instances of state-labeled Markov decision processes, the following four cases of the qualitative verification problem for finite state-labeled Markov decision processes $\mathcal{M}$ and PBA-specifications $\mathcal{P}$ are undecidable too.

Corollary 4.4.5 (Verification against PBA-specifications (ii)). The following problems are undecidable. Given a finite state-labeled Markov decision process $\mathcal{M}$ and a PBA-specification $\mathcal{P}$, is there a scheduler $\mathcal{U}$ for $\mathcal{M}$ such that

$$\begin{array}{llll} (i) & \mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\mathcal{L}(\mathcal{P})) > 0? & (ii) & \mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\mathcal{L}(\mathcal{P})) = 1? \\ (iii) & \mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\mathcal{L}(\mathcal{P})) < 1? & (iv) & \mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\mathcal{L}(\mathcal{P})) = 0? \end{array}$$

Proof. Indeed, problem (a) of Corollary 4.4.4 reduces to (i) and problem (b) reduces to (iii) when $\mathcal{T}$ is viewed as an MDP $\mathcal{M}_{\mathcal{T}}$, where we assume the initial distribution to be uniform over the initial states of $\mathcal{T}$. To reduce problem (a) of Corollary 4.4.4 to (ii) and problem (b) to (iv) we have to use a little trick, as there is no proper nondeterminism in the initial states of an MDP (but only probabilistic choice). Consider a PBA $\mathcal{P}$ over $\Sigma = \{a, b\}$, such that $\mathcal{L}(\mathcal{P}) = \{a^\omega\}$ and the following transition system $\mathcal{T}$ where the state names are associated with their labels. Then there is a path in $\mathcal{T}$ whose trace is in $\mathcal{L}(\mathcal{P})$,

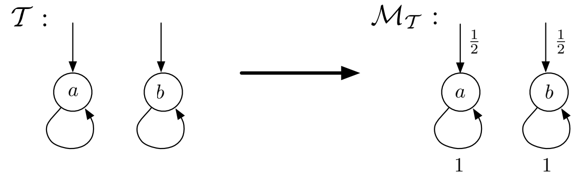


Figure 4.12: From $\mathcal{T}$ to $\mathcal{M}_{\mathcal{T}}$

but there is no scheduler in the MDP $\mathcal{M}_{\mathcal{T}}$ such that (ii) holds. Similarly, not all traces of $\mathcal{T}$ belong to $\mathcal{P}$, but there is no scheduler in $\mathcal{M}_{\mathcal{T}}$ such that (iv) holds. If a scheduler could also choose an initial distribution then we would be fine. We find a remedy for this situation by adjusting the given automaton and the MDP such that one irrelevant step is prepended. This yields the possibility to have a nondeterministic choice for the initial states of $\mathcal{T}$ in the MDP after the first irrelevant step. More precisely, let a labeled transition system $\mathcal{T} = (S, \mathsf{Act}, \delta_{\mathcal{T}}, S_0, \mathsf{AP}, L_{\mathcal{T}})$ and a PBA $\mathcal{P} = (Q, \Sigma, \delta_{\mathcal{P}}, \mu_{\mathcal{P}}, F)$ be given. Let the set of initial states of $\mathcal{T}$ be $S_0 = \{s_1, \ldots, s_n\}$. We define the labeled MDP $\mathcal{M}' = (S \mathbin{\dot\cup} \{s_{\mathsf{start}}\}, \mathsf{Act} \mathbin{\dot\cup} \{\alpha_1, \ldots, \alpha_n\}, \delta_{\mathcal{M}'}, \mu_{\mathcal{M}'}, \mathsf{AP} \mathbin{\dot\cup} \{a_{\mathsf{start}}\}, L_{\mathcal{M}'})$ with $\mu_{\mathcal{M}'}(s_{\mathsf{start}}) = 1$, $\delta_{\mathcal{M}'}(s_{\mathsf{start}}, \alpha_i, s_i) = 1, 1 \leq i \leq n$ and $\delta_{\mathcal{M}'}(s, \alpha, t) = 1$ if $t = \delta_{\mathcal{T}}(s, \alpha)$ for $s, t \in S$ and $\alpha \in \mathsf{Act}$. Moreover $L_{\mathcal{M}'}(s_{\mathsf{start}}) = \{a_{\mathsf{start}}\}$ and $L_{\mathcal{M}'}(s) = L_{\mathcal{T}}(s)$ for $s \in S$. Thus $\mathcal{M}'$ does an auxiliary first step and then mimics $\mathcal{T}$ (choosing an initial state of $\mathcal{T}$with a nondeterministic choice). We therefore have to adjust the automaton as well and define $\mathcal{P}' = (Q \mathbin{\dot\cup} \{q_{\mathsf{start}}\}, \Sigma \mathbin{\dot\cup} \{a_{\mathsf{start}}\}, \delta_{\mathcal{P}'}, \mu_{\mathcal{P}'}, F)$ with $\mu_{\mathcal{P}'}(q_{\mathsf{start}}) = 1$, $\delta_{\mathcal{P}'}(q_{\mathsf{start}}, a_{\mathsf{start}}, p) = \mu_{\mathcal{P}}(p)$ and $\delta_{\mathcal{P}'}(q, a, p) = \delta_{\mathcal{P}}(q, a, p)$ for $q, p \in Q$ and $a \in \Sigma$.

Now, problem (a) of Corollary 4.4.4 for $\mathcal{T}$ and $\mathcal{P}$ reduces to problem (ii) for $\mathcal{M}'$ and $\mathcal{P}'$ and problem (b) for $\mathcal{T}$ and $\mathcal{P}$ reduces to (iv) for $\mathcal{M}'$ and $\mathcal{P}'$. $\square$

Since PBA are a special case of partially observable Markov decision processes (POMDPs, see Definition 2.2.14 on page 13) our results immediately imply undecidability results for POMDPs and qualitative properties. In the literature, some undecidability results for POMDPs (or similar models) and quantitative properties (e.g. expected rewards, approximation of the maximal reachability problem) can be found [MHC03, GD07]. However, as far as we know, the undecidability of qualitative ω-regular properties for POMDPs is a new result. As POMDPs are $1\frac{1}{2}$-player games, the following results also apply to the setting of stochastic multi-player games with incomplete information.

Corollary 4.4.6 (**Undecidability results for POMDPs).** The following problems are undecidable:

(a) Given $(\mathcal{M}, \sim)$ a finite POMDP and F a set of states in $\mathcal{M}$, is there a deterministic observation-based scheduler $\mathcal{U}$ for $(\mathcal{M}, \sim)$ such that $\Pr^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) > 0$?

(b) Given $(\mathcal{M}, \sim)$ a finite POMDP and F a set of states in $\mathcal{M}$, is there a deterministic observation-based scheduler $\mathcal{U}$ for $(\mathcal{M}, \sim)$ such that $\Pr^{\mathcal{M},\mathcal{U}}(\Diamond\Box F) = 1$?

Proof. Given a total PBA $\mathcal{P}$ (i.e. a PBA that has transitions for each pair of a state and input letter) we define the equivalence relation $\sim\, = Q \times Q$. Note that each PBA can be trivially transformed into an equivalent total PBA. The pair $(\mathcal{P}, \sim)$ forms a POMDP with the action set Σ where a deterministic observation-based scheduler $\mathcal{U}$ represents an input word $\omega_{\mathcal{U}}$ for the PBA $\mathcal{P}$ (and vice versa). Consider F to be the set of accepting states of $\mathcal{P}$.

The undecidability of (a) is an immediate consequence of the undecidability of the emptiness problem for PBA as $\mathsf{Pr}^{\mathcal{P},\mathcal{U}}(\Box\Diamond F) = \mathsf{Pr}^{\mathcal{P}}(\omega_{\mathcal{U}})$.

The undecidability of (b) follows from the undecidability of the universality problem for PBA.

$$\mathsf{Pr}^{\mathcal{P}}(\omega_{\mathcal{U}}) \;=\; \mathsf{Pr}^{\mathcal{P},\mathcal{U}}(\Box\Diamond F) \;=\; 1 - \mathsf{Pr}^{\mathcal{P},\mathcal{U}}(\Diamond\Box(Q \setminus F)).$$

With $(\mathcal{P}, \sim)$ and the state set $Q \setminus F$ the answer to (b) is "yes" if and only if $\mathcal{L}(\mathcal{P}) \neq \Sigma^{\omega}$. As the universality problem for PBA is undecidable, this shows the claim. $\square$

4.4.1.3. Skipped proofs: accepted languages of $\mathcal{P}_1$ and $\mathcal{P}_2$

We now prove that $\mathcal{L}(\mathcal{P}_i) = L_i, i = 1, 2$ (see Figure 4.10, page 89, resp. Figure 4.11, page 90 for the automaton $\mathcal{P}_1$, resp. $\mathcal{P}_2$). Although the computations are analog to the computations for the accepted languages of the automata $\mathcal{P}_\lambda$ and $\widetilde{\mathcal{P}}_\lambda$, we provide them here for the sake of completeness.

$$\mathcal{L}(\mathcal{P}_1) = \Big\{\rho_1^1\sharp\rho_2^1\sharp\ldots\rho_{k_1}^1\$\$\,\rho_1^2\sharp\rho_2^2\sharp\ldots\rho_{k_2}^2\$\$\;\ldots \mid \rho_i^j \in \Sigma^* \\ \text{and } \prod_{j\geq 1}\Big(1 - \big(\prod_{i=1}^{k_j-1}(1 - \mathsf{Pr}^{\mathcal{R}}(\rho_i^j))\big)\Big) > 0\Big\}.$$

Proof. Starting in the first copy of $\mathcal{R}$, $1 - \mathsf{Pr}^{\mathcal{R}}(\rho)$ is the probability for reading the word ρ and ending in some non-final state p. Hence, $\prod_{i=1}^{k_j-1}(1 - \mathsf{Pr}^{P}(\rho_i^j))$ represents the probability to stay in the first copy of $\mathcal{R}$ after having read the finite word $\rho_1^j\sharp\rho_2^j\sharp\cdots\rho_{k_j-1}^j\sharp$. The complement of this probability is then exactly the probability to jump to the second copy at some point before reading $\rho_{k_j}^j$. This corresponds to the probability to be able to read the symbol \$ after the prefix $\rho_1^j\sharp\rho_2^j\sharp\cdots\rho_{k_j-1}^j\sharp\rho_{k_j}^j$. Thus, the infinite product $\prod_j\Big(1 - \big(\prod_{i=1}^{k_j-1}(1 - \mathsf{Pr}^{\mathcal{R}}(\rho_i^j))\big)\Big)$ is the probability to be able to read the two \$ symbols each time they appear in the input word. This agrees with the probability to visit infinitely often the final state F. This shows that the given expression for $\mathcal{L}(\mathcal{P}_1)$ is correct. $\square$

$$\mathcal{L}(\mathcal{P}_2) = \Big\{v_1\$\$\,v_2\$\$\;\ldots \mid v_i \in (\Sigma \cup \{\sharp\})^* \text{ and } \prod_{i\geq 1}\big(1 - (1-\varepsilon)^{|v_i|_\sharp}\big) = 0\Big\},$$

where $|v|_\sharp$ is the number of $\sharp$ symbols in word $v \in (\Sigma \cup \{\sharp\})^*$.

Proof. Starting in p'_0, $(1-\varepsilon)^{|v_i|_\sharp}$ is the probability to stay in p'_0 while reading the word v_i. Hence, $1 - (1-\varepsilon)^{|v_i|_\sharp}$ represents the probability to be in state p'_1 after the input word v_i. As a consequence $\prod_i\big(1 - (1-\varepsilon)^{|v_i|_\sharp}\big)$ is the probability to avoid forever the final state

F'. The probability to visit F' after reading the word $v_1\$\$v_2\$\$\cdots v_{N-1}\$\$$ and to avoid F' from then on is therefore

$$(1-\varepsilon)^{|v_{N-1}|_\sharp}\prod_{i\geq N}\big(1-(1-\varepsilon)^{|v_i|_\sharp}\big)$$

with the convention $|v_0|_\sharp = 0$. Hence, the probability to avoid F' from some point on is

$$\sum_N\Big((1-\varepsilon)^{|v_{N-1}|_\sharp}\prod_{i\geq N}\big(1-(1-\varepsilon)^{|v_i|_\sharp}\big)\Big).$$

To prove that $\mathcal{L}(\mathcal{P}_2)$ is as indicated above, we need to show that:

$$1-\sum_N\Big((1-\varepsilon)^{|v_{N-1}|_\sharp}\prod_{i\geq N}\big(1-(1-\varepsilon)^{|v_i|_\sharp}\big)\Big)>0 \iff \prod_i\Big(1-(1-\varepsilon)^{|v_i|_\sharp}\Big)=0$$

$\Leftarrow$: First assume that $\prod_i\Big(1-(1-\varepsilon)^{|v_i|_\sharp}\Big)=0$. Then, for all $N\in\mathbb{N}$,

$$\textstyle\prod_{i\geq N}\Big(1-(1-\varepsilon)^{|v_i|_\sharp}\Big)=0,$$

and thus $\sum_N\Big((1-\varepsilon)^{|v_{N-1}|_\sharp}\prod_{i\geq N}\big(1-(1-\varepsilon)^{|v_i|_\sharp}\big)\Big)=0$.

$\Rightarrow$: The other implication is more involved. Assume that $\prod_i\Big(1-(1-\varepsilon)^{|v_i|_\sharp}\Big)>0$. We have to show that

$$\sum_N\Big((1-\varepsilon)^{|v_{N-1}|_\sharp}\prod_{i\geq N}\big(1-(1-\varepsilon)^{|v_i|_\sharp}\big)\Big)=1.$$

With $\theta_i = 1-(1-\varepsilon)^{|v_i|_\sharp}$ we obtain:

$$\begin{aligned}
\sum_N\Big((1-\varepsilon)^{|v_{N-1}|_\sharp}\prod_{i\geq N}\big(1-(1-\varepsilon)^{|v_i|_\sharp}\big)\Big) &= \sum_N\Big((1-\theta_{N-1})\prod_{i\geq N}\theta_i\Big)\\
&= \sum_N\Big(\prod_{i\geq N}\theta_i-\theta_{N-1}\prod_{i\geq N}\theta_i\Big)\\
&= \sum_N\Big(\prod_{i\geq N}\theta_i-\prod_{i\geq N-1}\theta_i\Big)\\
&= \lim_{N\to\infty}\prod_{i\geq N}\theta_i \qquad \text{since } \theta_0=0
\end{aligned}$$

To conclude, we show that $\lim_{N\to\infty}\prod_{i\geq N}\theta_i=1$, using the assumption $\prod_i\theta_i>0$.

$$\begin{aligned}
&\prod_i\theta_i>0\Rightarrow\log(\prod_i\theta_i)<\infty\Rightarrow\sum_i\log(\theta_i)<\infty\Rightarrow\lim_{N\to\infty}\sum_{i\geq N}\log(\theta_i)=0\\
&\Rightarrow\lim_{N\to\infty}\log(\prod_{i\geq N}\theta_i)=0\Rightarrow\lim_{N\to\infty}\prod_{i\geq N}\theta_i=1.
\end{aligned}$$

This completes the proof that

$$\mathcal{L}(\mathcal{P}_2) = \Big\{ v_1\$\$\, v_2\$\$\, \ldots \mid v_i \in (\Sigma \cup \{\sharp\})^* \text{ and } \prod_{i \geq 1} \big(1 - (1-\varepsilon)^{|v_i|_\sharp}\big) = 0 \Big\}.$$

□

4.4.1.4. The threshold semantics

Another immediate implication of Theorem 4.4.2 is the undecidability of the emptiness problem when using a *threshold* semantics for PBA. For the variants of probabilistic ω-automata we studied so far, the accepted language has been defined as the set of infinite words over the input alphabet such that the probability measure of the set of accepting runs is nonzero. However, we did not require any positive lower bound for the acceptance probabilities. Following the concept of PFA [Rab63], PBA can also be equipped with a threshold $\gamma \in [0,1[$ for the accepted words. Given a PBA $\mathcal{P}$ and a threshold $0 \leq \gamma < 1$, we define the threshold language $\mathcal{L}^{>\gamma}(\mathcal{P})$ of $\mathcal{P}$ with respect to γ as

$$\mathcal{L}^{>\gamma}(\mathcal{P}) \; = \; \big\{ \omega \in \Sigma^\omega \mid \mathsf{Pr}^{\mathcal{P}}(\omega) > \gamma \big\}.$$

Note that with $\gamma = 0$ we obtain the standard semantics of PBA, i.e. $\mathcal{L}(\mathcal{P}) = \mathcal{L}^{>0}(\mathcal{P})$.

Theorem 4.4.7 (Undecidability of the emptiness problem for PBA under the threshold semantics).
Checking emptiness is undecidable for PBA under the threshold semantics.

Proof. We reduce the standard emptiness problem for PBA to the emptiness problem under the threshold semantics. Let a PBA $\mathcal{P} = (Q, \Sigma, \delta, \mu_0, F)$ and a threshold $\gamma < 1$ be given. We define $\mathcal{P}' = (Q', \Sigma', \delta', \mu_0', F')$ as follows. $Q' = Q \;\dot\cup\; \{q_F\}$, $\Sigma' = \Sigma$ and $F' = F \;\dot\cup\; \{q_F\}$. Moreover $\mu_0'(q_F) = \gamma$ and $\mu_0'(p) = (1-\gamma) \cdot \mu_0(p), p \in Q$. We define δ' as follows. $\delta'|_{Q \times \Sigma \times Q} = \delta$, $\delta'(q_F, a, q_F) = 1$ and $\delta'(q_F, a, p) = \delta'(p, a, q_F) = 0$ for all $a \in \Sigma$ and $p \in Q$. Obviously, $\mathcal{L}(\mathcal{P}) = \mathcal{L}^{>\gamma}(\mathcal{P}')$ which shows the claim. □

The proof of the above Theorem shows, that given any PBA $\mathcal{P}$ and any threshold γ, we can construct another PBA $\mathcal{P}'$ such that $\mathcal{L}^{>\gamma}(\mathcal{P}') = \mathcal{L}(\mathcal{P})$, that is $\mathbb{L}(\mathsf{PBA}) \subseteq \mathbb{L}(\mathsf{PBA}^{>\gamma})$ for any threshold γ. This raises the question whether PBA with the standard semantics are as expressive as PBA with the threshold semantics. This is not the case and we show

Theorem 4.4.8 (Existence of threshold-languages that are not PBA-recognizable).
There exists a real number $\gamma \in]0,1[$ such that $\mathbb{L}(\mathsf{PBA}^{>\gamma}) \not\subseteq \mathbb{L}(\mathsf{PBA})$.

Proof. The proof is based on an adaption of arguments provided by Paz [Paz71] for probabilistic finite automata (PFA). We identify any real number $\gamma \in]0,1[$ with the infinite word $a_1 a_2 a_3 \ldots \in \{0,1\}^\omega$ obtained by its binary representation $\gamma \; = \; \sum_{i=1}^{\infty} a_i 2^{-i} \; = \; 0.a_1 a_2 a_3 \ldots$ (where we assume that $a_i \neq 0$ for infinitely many indices i). We now consider the following languages $K_\gamma \subseteq \{0,1\}^*$:

$$K_\gamma \;=\; \big\{\, b_1 \dots b_n \,:\, b_1, \dots, b_n \in \{0,1\},\; \sum_{i=1}^{n} b_i 2^{-i} > \gamma \,\big\}$$

Paz [Paz71] has shown that K_γ is regular if and only if γ is rational. Rabin [Rab63] provided a PFA $\mathcal{R}$ such that, for all finite words $\rho \in \{0,1\}^*$, $\mathsf{Pr}^{\mathcal{R}}(\rho) > \gamma$ if and only if $\rho \in K_\gamma$. We modify this PFA $\mathcal{R}$ to a PBA $\mathcal{P}$ over the alphabet $\Sigma \;=\; \{0,1,c\}$ that under the threshold semantics accepts the language

$$L_\gamma \;=\; K_\gamma c^\omega \;=\; \{\rho c^\omega \mid \rho \in K_\gamma\}$$

when dealing with the threshold γ.

For this, we add a new accepting state q_{acc} with a c-self-loop and no other transitions (i.e. we set $\delta_{\mathcal{P}}(q_{\mathsf{acc}}, c, q_{\mathsf{acc}}) = 1$ and $\delta_{\mathcal{P}}(q_{\mathsf{acc}}, b, \cdot) = 0$ for $b \in \{0,1\}$) and c-transitions from each final state p in $\mathcal{R}$ to q_{acc} (i.e. we set $\delta_{\mathcal{P}}(p, c, q_{\mathsf{acc}}) = 1$ for all final states p of $\mathcal{R}$). The remaining transitions are as in $\mathcal{R}$. Finally the acceptance set of $\mathcal{P}$ is defined as $F = \{q_{\mathsf{acc}}\}$ and the initial distribution of $\mathcal{P}$ is the same as in $\mathcal{R}$. It then holds for all words $\rho \in \{0,1\}^*$ that $\mathsf{Pr}^{\mathcal{P}}(\rho c^\omega) \;=\; \mathsf{Pr}^{\mathcal{R}}(\rho)$. Furthermore, $\mathsf{Pr}^{\mathcal{P}}(\omega) = 0$ if ω contains infinitely many 0's or 1's. Thus:

$$\mathcal{L}^{>\gamma}(\mathcal{P}) \;=\; \{\, \rho c^\omega \,:\, \mathsf{Pr}^{\mathcal{R}}(\rho) > \gamma\} \;=\; K_\gamma c^\omega \;=\; L_\gamma.$$

Now fix an arbitrary irrational number $\gamma \in]0,1[$. Thus, by Paz [Paz71], K_γ non-regular. It remains to show that there is no PBA $\mathcal{P}'$ such that $\mathcal{L}(\mathcal{P}') = L_\gamma$. The intuitive argument will be the following. Suppose by contradiction that $\mathcal{P}'$ is a PBA with $\mathcal{L}(\mathcal{P}') = L_\gamma \subseteq \{0,1\}^* c^\omega$. Thus each word that will be accepted with a positive probability has a suffix consisting only of c's. But then, there is some "kind of underlying (finite word) automaton" in $\mathcal{P}'$, that decides which of the prefixes "to accept". Although this is a probabilistic automaton, as the acceptance threshold in $\mathcal{P}'$ is 0, this "underlying (finite word) automaton" will accept a regular language, which will contradict the assumption that γ is irrational.

More formally, we first observe that whenever (T, A) is an accepting end component of $\mathcal{P}'$ with $\mathsf{Pr}^{\mathcal{P}',\omega}((T, A)) > 0$ for some word $\omega \in \Sigma^\omega$ then $A(p) = \{c\}$ for all states $p \in T$. (Otherwise $\mathcal{P}'$ would accept some words that do not have a suffix consisting of c's.) Let T_0 be the set of states p in $\mathcal{P}'$ such that $p \in T$ for some accepting end component (T, A) of $\mathcal{P}'$ with $\mathsf{Pr}^{\mathcal{P}',\omega}((T, A)) > 0$ for some word $\omega \in \Sigma^\omega$. Furthermore, let T_0^+ be the set of all states q in $\mathcal{P}'$ such that $p \in \delta_{\mathcal{P}'}(q, c^n)$ for some $n \geq 0$ and $p \in T_0$. That is T_0^+ consists of all states from which a relevant accepting end component can be reached via a finite sequence of c's. Whenever $\rho \in \{0,1\}^*$ such that $\mathsf{Pr}^{\mathcal{P}'}(q_0 \xrightarrow{\rho} q) > 0$ for some initial state q_0 and some state $q \in T_0^+$ then

$$\mathsf{Pr}^{\mathcal{P}'}(\rho c^\omega) \;\geq\; \mathsf{Pr}^{\mathcal{P}'}(q_0 \xrightarrow{\rho} q) \cdot \mathsf{Pr}_q^{\mathcal{P}'}(q \xrightarrow{c^n} p) \;>\; 0$$

for some $n \in \mathbb{N}_{\geq 0}$, some state $q \in T_0^+$ and some state $p \in \delta_{\mathcal{P}'}(q, c^n)$, where p is in a relevant accepting end component. This yields

$$\rho c^\omega \;\in\; \mathcal{L}(\mathcal{P}') \;=\; L_\gamma \;= K_\gamma c^\omega,$$

and therefore $\rho \in K_\gamma$. Vice versa, if $\rho \in K_\gamma$ then $\rho c^\omega \in L_\gamma = \mathcal{L}(\mathcal{P}')$. Hence, there exists a state $q \in T_0^+$ such that $q \in \delta_{\mathcal{P}'}(q_0, \rho)$ for some initial state q_0.

This shows that K_γ agrees with the set of finite words $\rho \in \{0,1\}^*$ such that $\delta_{\mathcal{P}'}(q_0,\rho) \cap P_0^+ \neq \emptyset$. But then, K_γ agrees with the language of the NFA resulting from $\mathcal{P}'$ by discarding all c-transitions, interpreting the probabilistic branches by nondeterministic choices and declaring the states in P_0^+ to be final. Thus, K_γ is regular. This contradicts the assumption that γ is irrational in which case K_γ is not regular, as shown by Paz [Paz71]. $\square$

It remains to investigate PBA under a semantics that requires a word to generate accepting runs with probability 1 in order to be accepted by the automaton.

4.4.2. Decidability results for an almost-sure semantics

So far, we discussed PBA with a threshold semantics for thresholds $0 \leq \gamma < 1$. In this section we study PBA under the *almost-sure* semantics, where a word is accepted by a PBA $\mathcal{P}$, if it generates an almost-sure set of accepting runs, that is

$$\mathcal{L}^{=1}(\mathcal{P}) \;=\; \big\{\omega \in \Sigma^\omega \mid \mathsf{Pr}^{\mathcal{P}}(\omega) = 1\big\}.$$

4.4.2.1. Expressiveness of PBA under the almost-sure semantics

We first observe that for probabilistic Büchi automata, the switch from the standard semantics (which requires positive acceptance probability) to the almost-sure semantics leads to a loss of expressiveness, and that the class of probabilistic Büchi automata under the almost-sure semantics is not closed under complementation. Nevertheless, the class of languages definable by PBA under the almost-sure semantics is not included in the class of ω-regular languages.

Theorem 4.4.9 (Expressiveness of PBA under the almost-sure semantics).

(a) $\mathbb{L}(\mathsf{PBA}^{=1}) \;\subsetneq\; \mathbb{L}(\mathsf{PBA})$

(b) $\mathbb{L}(\omega\text{-reg}) \;\not\subseteq\; \mathbb{L}(\mathsf{PBA}^{=1})$

(c) $\mathbb{L}(\mathsf{PBA}^{=1}) \;\not\subseteq\; \mathbb{L}(\omega\text{-reg})$

(d) $\mathbb{L}(\mathsf{PBA}^{=1})$ is not closed under complementation.

(e) $\overline{\mathbb{L}(\mathsf{PBA}^{=1})} \subsetneq \mathbb{L}(\mathsf{PBA})$

(f) $\mathbb{L}(\omega\text{-reg}) \;\not\subseteq \overline{\mathbb{L}(\mathsf{PBA}^{=1})}$

(g) $\overline{\mathbb{L}(\mathsf{PBA}^{=1})} \not\subseteq \mathbb{L}(\omega\text{-reg})$

Proof.

(a) Let $\mathcal{P} = (Q, \Sigma, \delta, \mu_0, F)$ be a PBA and $L = \mathcal{L}^{=1}(\mathcal{P}) = \{\omega \in \Sigma^\omega : \mathsf{Pr}^{\mathcal{P}}(\omega) = 1\}$ the language of $\mathcal{P}$ under the almost sure semantics. We will transform $\mathcal{P}$ into an equivalent 0/1 PBA $\mathcal{P}'$. The idea to define $\mathcal{P}'$ is to pick at random some word position i where $\mathcal{P}$ could be in a state $p \in Q \setminus F$ and to check whether from this position i on, the probability in $\mathcal{P}$ for the event $\Box\neg F$ is positive. If so, then the input word is rejected by $\mathcal{P}$ with positive probability, and therefore, it does not belong to L. Formally, we define the PBA $\mathcal{P}'$ as $(Q', \Sigma', \delta', \mu_0', F')$ where

$$Q' \;=\; 2^Q \cup Q \times 2^Q, \quad F' = 2^Q$$

and $\mu_0'(Q_{\mathsf{init}}) = 1$ where $Q_{\mathsf{init}} = \{q \in Q : \mu_0(q) > 0\}$. The transition probabilities in $\mathcal{P}'$ are defined as follows. If $R \subseteq Q$ and $a \in \Sigma$ then $\delta'(R, a, S) = 1$ if $S = \delta(R, a) \subseteq F$. For $S = \delta(R, a)$ and $S \setminus F \neq \emptyset$ we define

$$\delta'(R, a, S) = \tfrac{1}{2}, \;\; \delta'(R, a, (p, S)) = \tfrac{1}{2 \cdot |S \setminus F|} \text{ for all } p \in S \setminus F.$$

For $p \in R \setminus F$, $q \in Q$ and $S = \delta(R, a)$ we set $\delta'((p, R), a, (q, S)) \;=\; \delta(p, a, q)$. For $p \in R \cap F$ and $S = \delta(R, a)$ we set $\delta'((p, R), a, S) = 1$. In all remaining cases, we set $\delta'(\cdot) = 0$.

Now assume $\omega \in \mathcal{L}^{=1}(\mathcal{P})$, thus $\mathsf{Pr}^{\mathcal{P},\omega}(\Box\Diamond F) = 1$. This implies $\mathsf{Pr}_t^{\mathcal{P},\omega\uparrow_i}(\Box\Diamond F) = 1$ for all $i \in \mathbb{N}_{\geq 1}$ and all states $t \in \delta(s, \omega\uparrow^{i-1})$, where $\mu_0(s) > 0$. Thus whenever $\mathcal{P}'$ enters its $Q \times 2^Q$ part while reading ω, it will afterwards reach a state (p, R) where p is an accepting state of $\mathcal{P}$ with probability one. As from such states $\delta'(.)$ leads to an accepting state of $\mathcal{P}'$ with probability 1 (in one step), this shows that $\mathsf{Pr}^{\mathcal{P}',\omega}(\Box\Diamond F') = 1$, so $\omega \in \mathcal{L}(\mathcal{P}')$.

Assume $\omega \notin \mathcal{L}^{=1}(\mathcal{P})$, thus $\mathsf{Pr}^{\mathcal{P},\omega}(\Diamond\Box\neg F) > 0$. Then there exists an $i \in \mathbb{N}_{\geq 1}$ such that $\mathsf{Pr}^{\mathcal{P},\omega}(\Diamond^{=i}\Box\neg F) > 0$, where $\Diamond^{=i}\Box\neg F$ denotes the event that after the $(i-1)$st step only states of $\neg F$ will be visited. Obviously

(i) $\mathsf{Pr}^{\mathcal{P},\omega}(\Diamond^{=j}\Box\neg F) \geq \mathsf{Pr}^{\mathcal{P},\omega}(\Diamond^{=i}\Box\neg F) > 0$ for all $j > i$.

Let $\theta := \mathsf{Pr}^{\mathcal{P},\omega}(\Diamond^{=i}\Box\neg F)$. As $\theta > 0$ it holds that

(ii) for all $j > i$ and all runs $q_0', q_1', q_2', \ldots$ of ω in $\mathcal{P}'$: $q_j'|_{2^Q} \cap \neg F \neq \emptyset$,

where $q_j'|_{2^Q} \;=\; R_j$ if $q_j' \;=\; R_j$ is a state in the 2^Q part of $\mathcal{P}'$ and $q_j'|_{2^Q} \;=\; R_j$ if $q_j' = (r_j, R_j)$ is a state in the $Q \times 2^Q$ part of $\mathcal{P}'$. Note that for the second statement (ii) the existence of a single run of ω in $\mathcal{P}$ satisfying $\Diamond^{=i}\Box\neg F$ suffices as the automaton $\mathcal{P}'$ performs a standard powerset construction on the 2^Q component in all its states. Examining the construction of $\mathcal{P}'$ shows that after reading the first i letters of ω, whenever $\mathcal{P}'$ is in its accepting 2^Q part, it will move with probability $\frac{1}{2}$ to its non-accepting $Q \times 2^Q$ part (because of (ii)) where it will stay forever with probability at least θ (because of (i) and the fact that the non-accepting $Q \times 2^Q$ part can only be left via a state (r, R) where $r \in F$). But this means that after the ith step (after reading $\omega\uparrow^i$), the automaton $\mathcal{P}'$ will almost surely reach its nonaccepting part and stay there forever which shows $\mathsf{Pr}^{\mathcal{P}',\omega}(\Box\Diamond F') = 0$, thus $\omega \notin \mathcal{L}(\mathcal{P}')$.

This shows that $\mathcal{L}(\mathcal{P}') = \mathcal{L}^{=1}(\mathcal{P})$. The strictness of the inclusion in (a) follows from (b) and the fact that $\mathbb{L}(\omega\text{-reg}) \;\subseteq\; \mathbb{L}(\mathsf{PBA})$.

(b) Let L be the language defined by the ω-regular expression $(a+b)^*a^\omega$. Suppose by contradiction that there is a PBA $\mathcal{P} = (Q, \{a,b\}, \delta, \mu_0, F)$ such that $\mathcal{L}^{=1}(\mathcal{P}) = L$. Without loss of generality we may assume that all states $p \in Q$ are reachable from some initial state. Let $\rho_p \in \{a,b\}^*$ be a finite word such that $p \in \delta(q_{\mathsf{init}}, \rho_p)$ where q_{init} is an initial state, i.e. $q_{\mathsf{init}} \in Q_{\mathsf{init}} = \{q_0 \in Q : \mu_0(q_0) > 0\}$. Since the word $\rho_p a^\omega$ belongs to L, it holds that $\mathsf{Pr}^{\mathcal{P}}(\rho_p a^\omega) = 1$. But then the set of accepting runs for a^ω starting in p must have probability measure 1, i.e. $\mathsf{Pr}_p^{\mathcal{P}}(a^\omega) = 1$. Thus, there exists $n_p \in \mathbb{N}_{\geq 1}$ such that $\delta(p, a^{n_p}) \cap F \neq \emptyset$. Let $n = \max_{p \in Q} n_p$ and $\tilde{\omega} = (a^n b)^\omega$. Then, $\tilde{\omega} \notin L$ and therefore $\mathsf{Pr}^{\mathcal{P}}(\tilde{\omega}) < 1$. For $p \in Q$, let θ_p be the probability to visit at least once an accepting state $q \in F$ when scanning the word a^n from p. Note that $\theta_p > 0$ since $n_p \leq n$ and $\delta(p, a^{n_p}) \cap F \neq \emptyset$. Let

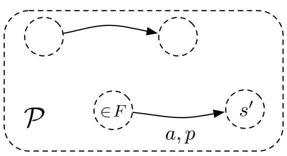
$$\theta = \min_{p \in Q} \theta_p$$

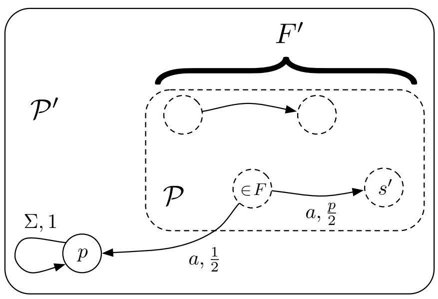
Then, $\theta > 0$ and for each $k \geq 0$ and each state $p \in \delta(Q_{\mathsf{init}}, (a^n b)^k)$, the probability to enter F at least once while reading a^n from p is at least θ. But then almost all runs for $\tilde{\omega}$ visit F infinitely often. That is, $\mathsf{Pr}^{\mathcal{P}}(\tilde{\omega}) = 1$, which contradicts the assumption that $\mathcal{L}^{=1}(\mathcal{P}) = (a+b)^*a^\omega$.

(c) The PBA $\widetilde{\mathcal{P}}_\lambda$ of Figure 4.4 recognizes a non-ω-regular language and enjoys the property that each word is either accepted with probability 0 or 1 (see Remark 4.2.5 on page 70), thus $\mathcal{L}^{=1}(\widetilde{\mathcal{P}}_\lambda) = \mathcal{L}(\widetilde{\mathcal{P}}_\lambda)$. This shows that $\widetilde{\mathcal{P}}_\lambda$ with the almost-sure semantics accepts a non-ω-regular language.

(d) It is evident that each DBA $\mathcal{P}$ can be viewed as a PBA and that $\mathcal{L}(\mathcal{P}) = \mathcal{L}^{=1}(\mathcal{P})$. Consider the language $(a^*b)^\omega$. It can be recognized by a DBA and hence by a PBA under the almost-sure semantics. However, its complement $(a+b)^*a^\omega$ cannot be recognized by a PBA with the almost-sure semantics (see part (b)). Hence the class of languages $\mathcal{L}^{=1}(\mathcal{P})$, $\mathcal{P}$ a PBA is not closed under complementation.

(e) The inclusion follows immediately from (a) and the fact that PBA are closed under complementation. Note that the construction in (a) as well as the complementation each impose an exponential blow-up. However given a PBA $\mathcal{P} = (Q, \Sigma, \delta, \mu_0, F)$ we can trivially construct a PBA $\mathcal{P}' = (Q', \Sigma, \delta', \mu_0', F')$ of the same size such that the complement of the language $\mathcal{L}^{=1}(\mathcal{P}) = \{\omega \in \Sigma^\omega : \mathsf{Pr}^{\mathcal{P}}(\omega) = 1\}$ is accepted by $\mathcal{P}'$ under the standard semantics. We define $\mathcal{P}'$ as indicated in the following picture.





$Q' = Q \cup \{p\}$, where $p \notin Q$. For $q_2 \in Q$, we set $\delta'(q_1, a, q_2) = \delta(q_1, a, q_2)$ if $q_1 \in Q \setminus F$ and $\delta'(q_1, a, q_2) = \frac{1}{2} \cdot \delta(q_1, a, q_2)$ if $q_1 \in F$. For $q_1 \in F$ we set $\delta'(q_1, a, p) = \frac{1}{2}$. Moreover $\delta'(p, a, p) = 1$ for all $a \in \Sigma$, $\mu'_0(q) = \mu_0(q)$ for $q \in Q$ (thus $\mu'_0(p) = 0$) and $F' = Q$.

Now assume $\omega \in \mathcal{L}^{=1}(\mathcal{P})$, thus $\mathsf{Pr}^{\mathcal{P},\omega}(\Box\Diamond F) = 1$. Thus reading ω, the automaton $\mathcal{P}'$ will almost surely reach the non-accepting state p and will then loop in p forever. So $\mathsf{Pr}^{\mathcal{P}',\omega}(\Box\Diamond F') = 0$ and $\omega \notin \mathcal{L}(\mathcal{P}')$.

Assume $\omega \notin \mathcal{L}^{=1}(\mathcal{P})$, thus $\mathsf{Pr}^{\mathcal{P},\omega}(\Diamond\Box\neg F) > 0$. Then there exists an $i \in \mathbb{N}_{\geq 1}$ such that $\mathsf{Pr}^{\mathcal{P},\omega}(\Diamond^{=i}\Box\neg F) > 0$, where $\Diamond^{=i}\Box\neg F$ denotes the event that after the $(i-1)$st step only states of $\neg F$ will be visited. But $\mathsf{Pr}^{\mathcal{P}',\omega}(\Box\Diamond F') \geq \mathsf{Pr}^{\mathcal{P}',\omega}(\Diamond^{=i}\Box F') \geq \left(\frac{1}{2}\right)^i \cdot \mathsf{Pr}^{\mathcal{P},\omega}(\Diamond^{=i}\Box\neg F) > 0$, so $\omega \in \mathcal{L}(\mathcal{P}')$.

This shows that $\mathcal{L}(\mathcal{P}') = \overline{\mathcal{L}^{=1}(\mathcal{P})}$. The strictness of the inclusion in (e) follows from (a) and the example in the reasoning for (b).

(f) Consider the ω-regular language $(a^*b)^\omega$. Its complement $(a+b)^*a^\omega$ cannot be recognized by a PBA with the almost-sure semantics (see part (b)).

(g) The PBA $\widetilde{\mathcal{P}}_\lambda$ of Figure 4.3 recognizes a non-ω-regular language and enjoys the property that each word is either accepted with probability 0 or 1 (see Remark 4.2.5 on page 70), so $\mathcal{L}^{=1}(\widetilde{\mathcal{P}}_\lambda) = \mathcal{L}(\widetilde{\mathcal{P}}_\lambda)$. Thus the complement language $\overline{\mathcal{L}^{=1}(\widetilde{\mathcal{P}}_\lambda)}$ is also non-ω-regular which shows the claim.

$\square$

Remark 4.4.10. It is worth noting that the almost-sure semantics does not lead to a loss of expressiveness if Streett or Rabin acceptance is considered. That is

$$\mathbb{L}(\mathsf{PSA}) = \mathbb{L}(\mathsf{PSA}^{=1}) = \mathbb{L}(\mathsf{PRA}^{=1}) = \mathbb{L}(\mathsf{PRA}).$$

This follows from the duality of the Streett and Rabin acceptance conditions and the results presented in section 4.3.2 as every PRA (resp. PSA) can be transformed into an equivalent PBA ([BG05]). More precisely, we show a ring of inclusions (i) $\mathbb{L}(\mathsf{PSA}) \subseteq \mathbb{L}(\mathsf{PSA}^{=1})$, (ii) $\mathbb{L}(\mathsf{PSA}^{=1}) \subseteq \mathbb{L}(\mathsf{PRA}^{=1})$, (iii) $\mathbb{L}(\mathsf{PRA}^{=1}) \subseteq \mathbb{L}(\mathsf{PRA})$ and (iv) $\mathbb{L}(\mathsf{PRA}) \subseteq \mathbb{L}(\mathsf{PSA})$.

(i) Let a PSA $\mathcal{P}_S$ be given. By Theorem 4.3.4 there exists a PBA $\mathcal{P}_B$ such that $\mathcal{L}(\mathcal{P}_S) = \mathcal{L}(\mathcal{P}_B)$. By Theorem 4.3.1 there exists a PBA $\overline{\mathcal{P}}_B$ such that $\overline{\mathcal{L}(\mathcal{P}_B)} = \mathcal{L}(\overline{\mathcal{P}}_B)$. This PBA can be transformed into an equivalent PRA $\overline{\mathcal{P}}_R$ (Remark 2.2.16). Thus

$$\mathcal{L}_{\mathit{Streett}}(\mathcal{P}_S) = \mathcal{L}_{\mathit{Büchi}}(\mathcal{P}_B) = \overline{\mathcal{L}_{\mathit{Büchi}}(\overline{\mathcal{P}}_B)} = \overline{\mathcal{L}_{\mathit{Rabin}}(\overline{\mathcal{P}}_R)} = \mathcal{L}^{=1}_{\mathit{Streett}}(\overline{P}_R).$$

Note that $\overline{\mathcal{L}_{\mathit{Rabin}}(\mathcal{P})} = \mathcal{L}^{=1}_{\mathit{Streett}}(\mathcal{P})$ holds for any given Rabin or Streett automaton as $\mathsf{Pr}^{\mathcal{P}}_{\mathit{Rabin}}(\omega) = 1 - \mathsf{Pr}^{\mathcal{P}}_{\mathit{Streett}}(\omega)$.

(ii) Let a PSA $\mathcal{P}_S$ be given. Then $\mathcal{L}^{=1}_{\mathit{Streett}}(\mathcal{P}_S) = \overline{\mathcal{L}_{\mathit{Rabin}}(\mathcal{P}_S)}$. Interpreting $\mathcal{P}_S$ as a Rabin automaton, there exists an equivalent Büchi automaton $\mathcal{P}_B$ (see [BG05]) which can be complemented into $\overline{\mathcal{P}}_B$ (Theorem 4.3.1). Theorem 4.3.2 proposes an equivalent 0/1-Rabin automaton $\mathcal{P}_R$ which yields

$$\mathcal{L}^{=1}_{\mathit{Streett}}(\mathcal{P}_S) \;=\; \overline{\mathcal{L}_{\mathit{Rabin}}(\mathcal{P}_S)} \;=\; \overline{\mathcal{L}_{\mathit{Büchi}}(\mathcal{P}_B)} \;=\; \mathcal{L}_{\mathit{Büchi}}(\overline{\mathcal{P}}_B) \;=\; \mathcal{L}_{\mathit{Rabin}}(\mathcal{P}_R) \;=\; \mathcal{L}^{=1}_{\mathit{Rabin}}(P_R).$$

Note that the last equation holds, because $\mathcal{P}_R$ is a 0/1-automaton, i.e. each input word is either accepted with probability 0 or 1.

(iii) Let a PRA $\mathcal{P}_R$ be given. Then $\mathcal{L}^{=1}_{\mathit{Rabin}}(\mathcal{P}_R) = \overline{\mathcal{L}_{\mathit{Streett}}(\mathcal{P}_R)}$. Interpreting $\mathcal{P}_R$ as a Streett automaton, there exists an equivalent Büchi automaton $\mathcal{P}_B$ (Theorem 4.3.4) which can be complemented into $\overline{\mathcal{P}}_B$ (Theorem 4.3.1). Remark 2.2.16 proposes an equivalent automaton $\mathcal{P}'_R$ which yields

$$\mathcal{L}^{=1}_{\mathit{Rabin}}(\mathcal{P}_R) \;=\; \overline{\mathcal{L}_{\mathit{Streett}}(\mathcal{P}_R)} \;=\; \overline{\mathcal{L}_{\mathit{Büchi}}(\mathcal{P}_B)} \;=\; \mathcal{L}_{\mathit{Büchi}}(\overline{\mathcal{P}}_B) \;=\; \mathcal{L}_{\mathit{Rabin}}(\mathcal{P}'_R).$$

(iv) Let a PRA $\mathcal{P}_R$ be given. $\mathcal{P}_R$ can be transformed into an equivalent Büchi automaton $\mathcal{P}_B$ (see [BG05]) which can be seen as a PSA $\mathcal{P}_S$ (Remark 2.2.16). This yields

$$\mathcal{L}_{\mathit{Rabin}}(\mathcal{P}_R) \;=\; \mathcal{L}_{\mathit{Büchi}}(\mathcal{P}_B) \;=\; \mathcal{L}_{\mathit{Streett}}(\mathcal{P}_S).$$

Note how the result

$$\mathbb{L}(\mathsf{PSA}) \;=\; \mathbb{L}(\mathsf{PSA}^{=1}) \;=\; \mathbb{L}(\mathsf{PRA}^{=1}) \;=\; \mathbb{L}(\mathsf{PRA}) \;=\; \mathbb{L}(\mathsf{PBA}) \;\supset\; \mathbb{L}(\mathsf{PBA}^{=1})$$

compares to the non-probabilistic setting of deterministic and nondeterministic ω-automata where

$$\mathbb{L}(\mathsf{NSA}) \;=\; \mathbb{L}(\mathsf{DSA}) \;=\; \mathbb{L}(\mathsf{DRA}) \;=\; \mathbb{L}(\mathsf{NRA}) \;=\; \mathbb{L}(\mathsf{NBA}) \;\supset\; \mathbb{L}(\mathsf{DBA}).$$

The previous theorem shows that PBA under the almost-sure semantics are less expressive then standard PBA. One benefit that we will draw out of this is the decidability of the emptiness problem for PBA under the almost-sure semantics. Nevertheless alike in the setting of standard PBA, the precise probability also matter for PBA under the almost-sure semantics.

Theorem 4.4.11 (The precise probabilities matter under the almost-sure semantics). For $0 < \lambda < \frac{1}{2} < \eta < 1$, $\quad \mathcal{L}^{=1}(\widetilde{\mathcal{P}}_\lambda) \neq \mathcal{L}^{=1}(\widetilde{\mathcal{P}}_\eta)$. (see Figure 4.4 for $\widetilde{\mathcal{P}}_\lambda$)

Proof. As $\mathcal{L}^{=1}(\widetilde{\mathcal{P}}_\lambda) = \mathcal{L}(\widetilde{\mathcal{P}}_\lambda)$ (see Remark 4.2.5 on page 70), the claim follows immediately from Lemma 4.2.7 (for n=2). $\square$

Thus modifying the transition probabilities can affect the accepted language of a PBA under the almost-sure semantics. However the emptiness problem "Given a PBA, does $\mathcal{L}^{=1}(\mathcal{P}) = \emptyset$?" for PBA under the almost-sure semantics is decidable. We will show a more general result, namely the decidability of the almost-sure repeated reachability problem for POMDPs (which asks whether, for a given POMDP $(\mathcal{M}, \sim)$ and a state set F, there exists an observation-based scheduler $\mathcal{U}$ such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) = 1$).

4.4.2.2. Decidability results in the general framework of POMDPs

Theorem 4.4.12 (Decidability results for POMDP).
Let a POMDP $(\mathcal{M}, \sim)$ and a state set $F \subseteq S$ be given. It is decidable,

(a) whether there exists an observation-based scheduler $\mathcal{U}$ for $(\mathcal{M}, \sim)$ such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) = 1$.

(b) whether there exists an observation-based scheduler $\mathcal{U}$ for $(\mathcal{M}, \sim)$ such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond\Box F) > 0$.

Proof.

(a) The proof of (a) splits into two steps. We first show (Lemma 4.4.13) that the almost-sure repeated reachability problem for POMDPs reduces to the almost-sure reachability problem for POMDPs (and vice versa) and then we proof the decidability of the latter problem (Theorem 4.4.15).

(b) It holds that

$$\begin{array}{lll}
\exists \mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)} \text{ s.th.} & \mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond\Box F) > 0 & \Leftrightarrow \\
\neg\big(\forall \mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)}. & \mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond\Box F) = 0\big) & \Leftrightarrow \\
\neg\big(\forall \mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)}. & \mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond\neg F) = 1\big) & \overset{\text{(see Remark 4.4.14)}}{\Leftrightarrow} \\
\neg\big(\forall \mathcal{U}' \in \mathsf{Sched}^{(\mathcal{M}',\sim')}. & \mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') = 1\big) & \Leftrightarrow \\
\exists \mathcal{U}' \in \mathsf{Sched}^{(\mathcal{M}',\sim')} \text{ s.th.} & \mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') < 1 & \Leftrightarrow \\
\exists \mathcal{U}' \in \mathsf{Sched}^{(\mathcal{M}',\sim')} \text{ s.th.} & \mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Box\neg F') > 0 &
\end{array}$$

The latter problem (confinement with positive probability: $\mathsf{Pr}^{\mathcal{U}}(\Box F) > 0$) has been proven to be EXPTIME-complete by de Alfaro [dA99].

$\square$

Lemma 4.4.13. The two following problems are reducible to each other:

(i) Given a POMDP $(\mathcal{M}, \sim)$ and a set of states F, is there an observation-based scheduler $\mathcal{U}$ with $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) = 1$?

(ii) Given a POMPD $(\mathcal{M}, \sim)$ and a set of states F, is there an observation-based scheduler $\mathcal{U}$ with $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond F) = 1$?

Proof.

- Problem (ii) reduces to (i) in a straightforward manner: given an instance for (ii) we transform it into an instance for (i) by making all F-states absorbing, i.e. by removing all outgoing edges from states in F, and adding self loops for all actions with probability one (to the states of F).

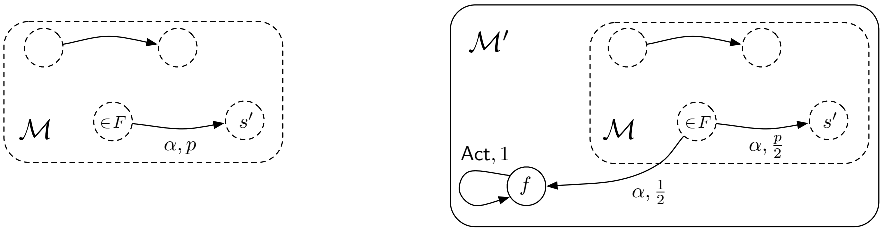


Figure 4.13: Transformation from $\mathcal{M}$ to $\mathcal{M}'$

- We now show that problem (i) is reducible to problem (ii). Let $(\mathcal{M}, \sim), F$ be an instance for (i). We define $\mathcal{M}'$ as follows: $\mathcal{M}'$ consists of a copy of $\mathcal{M}$ and some additional state f. All transitions (s, α, s') in $\mathcal{M}$ with $s \notin F$ are left unchanged. The transitions (s, α, s') in $\mathcal{M}$ with $s \in F$ are kept, but their probabilities are divided by 2 in $\mathcal{M}'$. In $\mathcal{M}'$, we add a self-loop with probability 1 to state f for all action $\alpha \in \mathsf{Act}$. Finally, for all $s \in F$ and $\alpha \in \mathsf{Act}$, we add a new transition (s, α, f) with probability $\frac{1}{2}$. The transformation is depicted in Figure 4.13. The equivalence relation $\sim'$ on $S \,\dot\cup\, \{f\}$ agrees with $\sim$ on S and $\{f\}$ forms its own equivalence class, i.e. $[s]_{\sim'} = [s]_{\sim}$ for $s \in S$ and $[f]_{\sim'} = \{f\}$. With $F' = \{f\}$, $(\mathcal{M}', \sim'), F'$ is an instance for problem (ii) satisfying the equivalence:

  $$\exists \mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)}.\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) = 1 \Leftrightarrow \exists \mathcal{U}' \in \mathsf{Sched}^{(\mathcal{M}',\sim')}.\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') = 1.$$

  Indeed if F is visited almost surely infinitely often in $\mathcal{M}$ under the scheduler $\mathcal{U}$, F' will be almost surely visited in $\mathcal{M}'$ under the scheduler $\mathcal{U}'$ that mimics $\mathcal{U}$. That is $\mathcal{U}'(\pi') = \mathcal{U}(\pi')$, if π' is not only a finite path in $\mathcal{M}'$ but also in $\mathcal{M}$ and $\mathcal{U}(\pi') = \alpha$ if $\mathsf{last}(\pi') = f$ (where $\alpha \in \mathsf{Act}$ is arbitrary). Note that all other cases (π' does not end in f and is not a path in $\mathcal{M}$) are irrelevant.

  Conversely, given $\mathcal{U}' \in \mathsf{Sched}^{(\mathcal{M}',\sim')}$ with $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') = 1$, we define $\mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)}$ to be the restriction of $\mathcal{U}'$ on the set of path of $\mathcal{M}$, that is $\mathcal{U}(\pi) = \mathcal{U}'(\pi)$ for all $\pi \in \mathsf{Path}^{\mathcal{M}}_{\mathsf{fin}}$. Then $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) = 1$, since $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond\Box\neg F) > 0$ implies $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Box\neg F') > 0$. The last claim is easy to see. We denote by $[(F)_{=j}(\neg F)_{>j}]$ the set of infinite paths π such that $\pi^j \in F$ and $\pi^k \notin F, k > j$. But then it holds that

  $$\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Box\neg F') \;\geq\; \tfrac{1}{2} \cdot \mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}([(F)_{=j}(\neg F)_{>j}]) \;\geq\; \tfrac{1}{2^{j+1}} \cdot \mathsf{Pr}^{\mathcal{M},\mathcal{U}}([(F)_{=j}(\neg F)_{>j}]).$$

  As $\{\pi \mid \pi \models \Diamond\Box\neg F\} = \dot\bigcup_{j \geq -1} [(F)_{=j}(\neg F)_{>j}]$, assuming $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond\Box\neg F) > 0$ yields the existence of an index k, such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}([(F)_{=k}(\neg F)_{>k}]) > 0$ which, together with the above chain of inequalities (for j=k), shows the claim.

  Note, that $\mathcal{U}$ and $\mathcal{U}'$ are of the same type, i.e. either both are deterministic, resp. memoryless or they are not.

$\square$

Remark 4.4.14. Note that the construction in Figure 4.13 also ensures that

$$\forall \mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)}.\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) = 1 \Leftrightarrow \forall \mathcal{U}' \in \mathsf{Sched}^{(\mathcal{M}',\sim')}.\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') = 1.$$

Indeed, let us assume that there exists an observation-based scheduler $\mathcal{U}$ of $\mathcal{M}$ such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) < 1$. By Lemma 2.2.13 it follows that there exists an end component (T, A) of $\mathcal{M}$ with $T \cap F = \emptyset$ such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\{\pi \in \mathsf{Path}^{\mathcal{M}}_{\mathsf{inf}} \mid \mathsf{Lim}(\pi) = (T, A)\}) > 0$. This immediately shows that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\{\pi \in \mathsf{Path}^{\mathcal{M}'}_{\mathsf{inf}} \mid \mathsf{Lim}(\pi) = (T, A)\}) > 0$ and therefore $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') < 1$. Here $\mathcal{U}'$ is the scheduler of $\mathcal{M}'$ that mimics $\mathcal{U}$ (as in the proof of Lemma 4.4.13). On the other hand assume that there exists an observation-based scheduler $\mathcal{U}'$ of $\mathcal{M}'$ such that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') < 1$. Note that for each end component (T, A) of $\mathcal{M}'$, either $T = F'$ or $T \cap F = \emptyset$. By Lemma 2.2.13 it follows that there exists an end component (T, A) of $\mathcal{M}'$ with $F' \neq T$ and $T \cap F = \emptyset$ such that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\{\pi \in \mathsf{Path}^{\mathcal{M}'}_{\mathsf{inf}} \mid \mathsf{Lim}(\pi) = (T, A)\}) > 0$. Thus for the restriction $\mathcal{U}$ of $\mathcal{U}'$ to $\mathcal{M}$ we derive that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\{\pi \in \mathsf{Path}^{\mathcal{M}}_{\mathsf{inf}} \mid \mathsf{Lim}(\pi) = (T, A)\}) > \mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\{\pi \in \mathsf{Path}^{\mathcal{M}'}_{\mathsf{inf}} \mid \mathsf{Lim}(\pi) = (T, A)\}) > 0$ and therefore $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\Diamond F) < 1$ which was to show.

By Lemma 4.4.13 we can reduce the almost-sure repeated reachability problem for POMDPs to the almost-sure reachability problem for POMDPs for which we now show decidability (see [ACY95, Lit96] for related results). The related problem (invariant with positive probability: $\mathsf{Pr}^{\mathcal{U}}(\Box F) > 0$) has been proven to be EXPTIME-complete by de Alfaro [dA99].

Theorem 4.4.15 (Decidability of the almost-sure reachability problem for POMDP). Let a POMDP $(\mathcal{M}, \sim)$ and a state set $F \subseteq S$ be given. It is decidable, whether there exists an observation-based scheduler $\mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)}$ such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond F) = 1$.

Proof. We reduce the almost-sure reachability problem for POMDPs to the almost-sure reachability problem for (fully observable) MDPs, which is known to be solvable by means of graph-algorithms. Let $\mathcal{M} = ((S, \mathsf{Act}, \delta, \mu), \sim)$ be a (w.l.o.g. total) POMDP and $F \subseteq S$. W.l.o.g. we assume that the states in F are absorbing, i.e. for all states $q \in F$, $\delta(q, \alpha, q) = 1$ for all $\alpha \in \mathsf{Act}$. We define an MDP $\mathcal{M}' = (S', \mathsf{Act}, \delta', \mu')$ as follows. The set of states S' of $\mathcal{M}$ consists of pairs (r, R) with $r \in R \subseteq [r]_\sim$ and an extra state q_F that has a self-loop with probability one for all $\alpha \in \mathsf{Act}$. Given $\alpha \in \mathsf{Act}$ and $R \subseteq S$, let $R' = \delta(R \backslash F, \alpha)$.

If $\delta(r, \alpha) \cap F = \emptyset$ then $\delta'((r, R), \alpha, (r', R' \cap [r']_\sim) = \delta(r, \alpha, r')$ for each $r' \in S$.

If $\delta(r, \alpha) \cap F \neq \emptyset$ then $\delta'((r, R), \alpha, (r', R' \cap [r']_\sim)) = \frac{1}{2\cdot|R' \backslash F|}$ for all $r' \in R' \backslash F$ and $\delta'((r, R), \alpha, q_F) = \frac{1}{2}$ (in case $R' \backslash F = \emptyset$, $\delta'((r, R), \alpha, q_F) = 1$).

Moreover $\mu'(q, [q]_\sim) = \mu(q)$ for all $q \notin F$ and $\mu'(q_F) = \Sigma_{r \in F}\mu(r)$. We set $F' = \{q_F\}$.

Before we show that this construction ensures that there exists an observation-based scheduler $\mathcal{U}$ of $\mathcal{M}$ with $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond F) = 1$ if and only if there exists a scheduler $\mathcal{U}'$ of $\mathcal{M}'$ such that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') = 1$, we fix some notation. For each action α we define the set of pre-final states of $\mathcal{M}'$ as $F'_{\mathsf{pre}}(\alpha) = \{(r, R) \mid \delta(r, \alpha) \cap F \neq \emptyset\}$. So $F'_{\mathsf{pre}}(\alpha)$ is the set of states ($\neq q_F$) from which $\mathcal{M}'$ reaches its accepting state via the action α. Given a position in some path π we denote by $\mathsf{Next}_{\mathsf{Act}}$ the action that occurs after this position in π. So $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}\Big(\Box\Diamond\big(\vee_\alpha(F'_{\mathsf{pre}}(\alpha) \wedge \mathsf{Next}_{\mathsf{Act}} = \alpha)\big)\Big)$ denotes the probability under the scheduler

$\mathcal{U}'$ of the set of paths in which infinitely often a pre-final state for some action α appears and is followed by the action α, i.e. it denotes the value

$$\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}\big(\{\pi' \mid \overset{\infty}{\exists}\, i : \vee_\alpha(\pi'_i \in F'_{\mathsf{pre}}(\alpha) \wedge \mathsf{Act}_{i+1}(\pi') = \alpha)\}\big).$$

Similarly $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}\Big(\Diamond\Box\big(\wedge_\alpha(\neg F'_{\mathsf{pre}}(\alpha) \vee \mathsf{Next}_{\mathsf{Act}} \neq \alpha)\big)\Big)$ denotes the probability under the scheduler $\mathcal{U}'$ of the set of paths for which from some point it holds that whenever a pre-final state for some action α appears then the following action is not α.

Now assume that there exists an observation-based scheduler $\mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)}$ such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond F) = 1$. We define $\mathcal{U}' \in \mathsf{Sched}^{\mathcal{M}'}$ as follows:

$$\mathcal{U}'((r_0, R_0) \xrightarrow{\alpha_1} (r_1, R_1) \ldots \xrightarrow{\alpha_n} (r_n, R_n)) \;=\; \mathcal{U}([r_0]_\sim \xrightarrow{\alpha_1} [r_1]_\sim \ldots \xrightarrow{\alpha_n} [r_n]_\sim)$$

We claim that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}\Big(\Box\Diamond\big(\vee_\alpha(F'_{\mathsf{pre}}(\alpha) \wedge \mathsf{Next}_{\mathsf{Act}} = \alpha)\big) \;\vee\; \Diamond\Box q_F\Big) = 1$. Assume the contrary. So $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}\Big(\Diamond\Box\big(\wedge_\alpha(\neg F'_{\mathsf{pre}}(\alpha) \vee \mathsf{Next}_{\mathsf{Act}} \neq \alpha)\big) \;\wedge\; \Box\Diamond\neg q_F\Big) > 0$. As q_F is absorbing this implies $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}\Big(\Diamond\Box\big(\wedge_\alpha(\neg F'_{\mathsf{pre}}(\alpha) \vee \mathsf{Next}_{\mathsf{Act}} \neq \alpha)\big) \;\wedge\; \Box\neg q_F\Big) > 0$. Since $\mathcal{M}'$ is a finite state system there exists a finite path $\tilde{\pi}' = (r_0, R_0), (r_1, R_1), \ldots, (r_n, R_n)$ of $\mathcal{M}'$ such that

$$\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}\Big(\big\{\pi' \mid \pi'{\uparrow}^n = \tilde{\pi}' \;\wedge \pi' \models \Diamond^{=n}\Box\big(\wedge_\alpha(\neg F'_{\mathsf{pre}}(\alpha) \vee \mathsf{Next}_{\mathsf{Act}} \neq \alpha)\big) \;\wedge\; \Box\neg q_F\big\}\Big) > 0.$$

Then

$$\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'_{\tilde{\pi}'}}_{(r_n,R_n)}\Big(\Box\big(\wedge_\alpha(\neg F'_{\mathsf{pre}}(\alpha) \vee \mathsf{Next}_{\mathsf{Act}} \neq \alpha)\big) \;\wedge\; \Box\neg q_F\Big) > 0,$$

where $\mathcal{U}'_{\tilde{\pi}'}(\hat{\pi}') = \mathcal{U}'(\tilde{\pi}'\hat{\pi}')$ for all finite paths $\hat{\pi}'$ with $\mathsf{first}(\hat{\pi}') = \mathsf{last}(\tilde{\pi}')$. For all other paths $\hat{\pi}'$ with $\mathsf{first}(\hat{\pi}') \neq \mathsf{last}(\tilde{\pi}')$ let $\mathcal{U}'_{\tilde{\pi}'}(\hat{\pi}')$ be defined arbitrarily. Note that

$$\mathsf{Pr}^{\mathcal{M},\mathcal{U}_{\tilde{\pi}}}_{r_n}(\Box\neg F) \geq \mathsf{Pr}^{\mathcal{M}',\mathcal{U}'_{\tilde{\pi}'}}_{(r_n,R_n)}\Big(\Box\big(\wedge_\alpha(\neg F'_{\mathsf{pre}}(\alpha) \vee \mathsf{Next}_{\mathsf{Act}} \neq \alpha)\big) \;\wedge\; \Box\neg q_F\Big) > 0 \qquad (+)$$

where $\tilde{\pi}$ is the state-wise projection of $\tilde{\pi}'$ to its first component, i.e. $\tilde{\pi} = r_0, r_1, \ldots, r_n$. This implies

$$\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Box\neg F) \;\geq\; \underbrace{\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\{\pi \mid \pi{\uparrow}^n = \tilde{\pi}\})}_{>0} \cdot \underbrace{\mathsf{Pr}^{\mathcal{M},\mathcal{U}_{\tilde{\pi}}}_{r_n}(\Box\neg F)}_{>0} > 0$$

which is a contradiction as we assumed $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond F) = 1$. This shows our claim that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}\Big(\Box\Diamond\big(\vee_\alpha(F'_{\mathsf{pre}}(\alpha) \wedge \mathsf{Next}_{\mathsf{Act}} = \alpha)\big) \;\vee\; \Diamond\Box q_F\Big) = 1$. Inspecting the construction of $\mathcal{M}'$ it easily follows that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond\Box q_F) = 1$, so $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond q_F) = 1$, which we wanted to show. It remains to show (+), that is

$$\mathsf{Pr}^{\mathcal{M},\mathcal{U}_{\tilde{\pi}}}_{r_n}(\Box\neg F) \geq \mathsf{Pr}^{\mathcal{M}',\mathcal{U}'_{\tilde{\pi}'}}_{(r_n,R_n)}\Big(\Box\big(\wedge_\alpha(\neg F'_{\mathsf{pre}}(\alpha) \vee \mathsf{Next}_{\mathsf{Act}} \neq \alpha)\big) \;\wedge\; \Box\neg q_F\Big)$$

Indeed, consider the infinite Markov chains $\mathcal{M}'_{\mathcal{U}'_{\tilde{\pi}'}}$ and $\mathcal{M}_{\mathcal{U}_{\tilde{\pi}}}$ that evolve when applying the scheduler $\mathcal{U}'_{\tilde{\pi}'}$ to $\mathcal{M}'$ and the scheduler $\mathcal{U}_{\tilde{\pi}}$ to $\mathcal{M}$. Then the state-wise projection on the first component of each path π' of $\mathcal{M}'_{\mathcal{U}'_{\tilde{\pi}'}}$ is also a path of $\mathcal{M}_{\mathcal{U}_{\tilde{\pi}}}$. Moreover the construction of $\mathcal{M}'$ ensures that if π' satisfies $\Box\big(\wedge_\alpha(\neg F'_{\mathsf{pre}}(\alpha) \vee \mathsf{Next}_{\mathsf{Act}} \neq \alpha)\big) \ \wedge \ \Box\neg q_F$ [2] then the transition probabilities of π' in $\mathcal{M}'_{\mathcal{U}'_{\tilde{\pi}'}}$ agree with the transition probabilities of its projection in $\mathcal{M}_{\mathcal{U}_{\tilde{\pi}}}$. As the projection of each such path satisfies $\Box\neg F$, this shows (+).

We now show the other direction. So we assume that there exists a scheduler $\mathcal{U}'$ of $\mathcal{M}'$ such that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}'}(\Diamond F') = 1$. We have to construct an observation-based scheduler $\mathcal{U} \in \mathsf{Sched}^{(\mathcal{M},\sim)}$ such that $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond F) = 1$. Note that given a standard MDP $\tilde{\mathcal{M}}$ and a state set $\tilde{F}$, the existence of a scheduler under which $\tilde{\mathcal{M}}$ reaches $\tilde{F}$ almost-surely also ensures the existence of a memoryless deterministic scheduler under which $\tilde{\mathcal{M}}$ reaches $\tilde{F}$ almost-surely ([HSP83, BdA95]). So w.l.o.g. we assume that $\mathcal{U}'$ is memoryless and deterministic.

Let $S = S_1 \,\dot{\cup}\, \ldots \,\dot{\cup}\, S_n$ be the partition of the state set of $\mathcal{M}$ with respect to $\sim$, i.e. for all $p \in S_i$ it holds that $[p]_\sim = S_i$. For each equivalence class S_i and each set $R \subseteq S_i$ we define a representative $p_i^R \in S_i$ such that the state (p_i^R, R) is reachable in $\mathcal{M}'$ (if possible). If no such state exists, the representative is undefined (R is then of no importance w.r.t. to the equivalence class S_i). First we define a new scheduler $\mathcal{U}''$ of $\mathcal{M}'$ that makes the same decision for states of $\mathcal{M}'$ that have a state of the same equivalence class in their first component and have the same second component. That is

$$\mathcal{U}''((p,R)) \ := \ \mathcal{U}'((p_i^R, R)),$$

where the index i is such that $p \in S_i$. Note that $R \subseteq [p]_\sim = [p_i^R]_\sim$ and that the scheduler $\mathcal{U}''$ is memoryless and deterministic. The construction of $\mathcal{M}'$ ensures that $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}''}(\Diamond F') = 1$. Now we define a scheduler $\mathcal{U}$ for $\mathcal{M}$ for all finite paths $p_0 \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} p_n$ of M with $p_1, \ldots, p_n \notin F$ (recall that the states in F are absorbing). For such a path there is a unique corresponding run

$$(p_0, [p_0]_\sim) \ \xrightarrow{\alpha_1} \ (p_1, R_1) \ \xrightarrow{\alpha_2} \ \ldots \ \xrightarrow{\alpha_n} \ (p_n, R_n)$$

in $\mathcal{M}'$. We define the scheduler $\mathcal{U}$ of $\mathcal{M}$ as

$$\mathcal{U}(p_0 \xrightarrow{\alpha_1} p_1 \xrightarrow{\alpha_2} \ldots \xrightarrow{\alpha_n} p_n) \ := \ \mathcal{U}''((p_n, R_n)).$$

Note that $\mathcal{U}$ is not only an observation-based scheduler, but also $\mathsf{Pr}^{\mathcal{M},\mathcal{U}}(\Diamond F) = 1$. This can be seen as follows. Any infinite path of $\mathcal{M}$ that never visits the set F has a corresponding path in $\mathcal{M}'$. As $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}''}(\Diamond F') = 1$, such a path almost-surely satisfies the condition that under the scheduler $\mathcal{U}''$ at infinitely many indices, the next action had the state q_F as a successor (since $\mathsf{Pr}^{\mathcal{M}',\mathcal{U}''}\big(\Box\Diamond\big(\vee_\alpha(F'_{\mathsf{pre}}(\alpha) \wedge \mathsf{Next}_{\mathsf{Act}} = \alpha)\big) \ \vee \ \Diamond\Box q_F\big) = 1$). But this means that the original path in $\mathcal{M}$ (which never visits F) almost-surely satisfies the condition that under the scheduler $\mathcal{U}$ at infinitely many indices the next action had a successor in F. Since $\mathcal{M}$ is finite all the transition probabilities are bounded below by some $\epsilon > 0$. This then ensures that the set of infinite paths never visiting F has measure zero under the scheduler $\mathcal{U}$. $\Box$

[2] Note that the states of $\mathcal{M}'_{\mathcal{U}'_{\tilde{\pi}'}}$ are finite paths of $\mathcal{M}'$. A state $x_1, x_2, \ldots, x_n$ of $\mathcal{M}'_{\mathcal{U}'_{\tilde{\pi}'}}$ is said to satisfy a property, if the last $\mathcal{M}'$-state of its sequence, namely x_n, satisfies the property.

Our algorithm uses a powerset construction and hence runs in time exponential in the size of the given POMDP. However, given the EXPTIME-hardness results established by Reif [Rei84] and by [CDHR06] for 2-player games with incomplete information and by de Alfaro [dA99] for POMDPs, we do not expect more efficient algorithms.

Remark 4.4.16. Inspecting the proof of Theorem 4.4.15, we see that given a POMDP $(\mathcal{M}', \sim')$ and a state set F', the existence of a scheduler under which $\mathcal{M}'$ reaches F' almost-surely also ensures the existence of a finite-memory deterministic scheduler under which $\mathcal{M}'$ reaches F' almost-surely. But then the construction used in the proof of Lemma 4.4.13 ensures that given a POMDP $(\mathcal{M}'', \sim'')$ and a state set F'', the existence of a scheduler under which $\mathcal{M}''$ repeatedly reaches F'' almost-surely also ensures the existence of a finite-memory deterministic scheduler under which $\mathcal{M}''$ repeatedly reaches F'' almost-surely.

Theorem 4.4.17 (Decidability of the emptiness problem for PBA under the almost-sure semantics).
Checking emptiness is decidable for PBA under the almost-sure semantics.

Proof. As PBA are a special case of POMDPs ($\sim = Q \times Q$), the claim is an immediate consequence of Theorem 4.4.12 and Remark 4.4.16 (since each deterministic scheduler can be seen as an input word). □

Remark 4.4.18. Solving the almost-sure reachability problem for standard MDPs is done by means of graph algorithms [HSP83, Var85, CY95] that do not take into account the precise transition probabilities (just, whether they are $\neq 0$). Thus ignoring the precise transition probabilities in the construction in the proof of Theorem 4.4.15 is legal. Moreover this shows that the almost-sure repeated reachability problem for POMDPs and therefore the emptiness problem for PBA under the almost-sure semantics do not depend on the precise transition probabilities. That means that although the accepted language of a PBA $\mathcal{P}$ under the almost-sure semantics depends on the precise transition probabilities, it holds that $\mathcal{L}^{=1}(\mathcal{P}) \neq \emptyset \Leftrightarrow \mathcal{L}^{=1}(\mathcal{P}') \neq \emptyset$ for each PBA $\mathcal{P}'$ that evolves from $\mathcal{P}$ by altering the transition probabilities in a legal way, that is $\delta(s, \alpha, t) > 0$ if and only if $\delta'(s, \alpha, t) > 0$. It even holds that if $\mathcal{L}^{=1}(\mathcal{P}) \neq \emptyset$, then there exists a word in $\mathcal{L}^{=1}(\mathcal{P})$ that is contained in $\mathcal{L}^{=1}(\mathcal{P}')$ for all such PBA $\mathcal{P}'$, i.e.

$$\mathcal{L}^{=1}(\mathcal{P}) \neq \emptyset \qquad \Longrightarrow \qquad \bigcap_{\substack{\mathcal{P}':\mathcal{P}' \text{ evolves from } \mathcal{P} \\ \text{by legally altering the trans. prob.}}} \mathcal{L}^{=1}(\mathcal{P}') \neq \emptyset.$$

This follows immediately from Remark 4.4.16, as $\mathcal{L}^{=1}(\mathcal{P}) \neq \emptyset$ ensures the existence of a finite-memory word ω, such that $\mathsf{Pr}^{\mathcal{P}}(\omega) = 1$. The behavior of $\mathcal{P}$ under this finite-memory word can be described by a finite Markov chain $\mathcal{P}^{\omega}$ and the almost-sure acceptance of the word in $\mathcal{P}$ is equivalent to the almost-sure repeated reachability of a set F' in $\mathcal{P}^{\omega}$. As the latter does not depend on the exact transition probabilities of $\mathcal{P}^{\omega}$, but only on its underlying graph [HSP83], this shows the claim.

4.5. Conclusion

We introduced and studied probabilistic ω-automata, particularly probabilistic Büchi automata. We investigated different acceptance semantics, namely positive acceptance, almost-sure acceptance and threshold acceptance. Before we summarize our results, Figure 4.14 shows an overview of the expressiveness that the different semantics apply to PBA (and Rabin, resp. Streett automata).

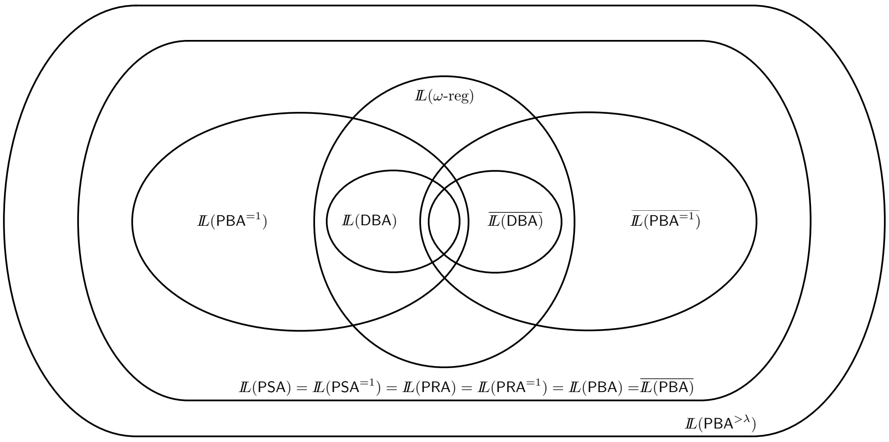


Figure 4.14: Overview of expressiveness of variants of probabilistic ω-automata

We moreover showed the following results.

- We showed that probabilistic Büchi automata with the acceptance criterion “the accepting runs have a positive probability measure” are more powerful than ω-regular languages. This stands in contrast to the facts that (1) deterministic Büchi automata do not have the full power of ω-regular languages and (2) PFA with the acceptance criterion “the accepting runs have a positive probability measure” can be viewed as nondeterministic finite automata, and hence, have exactly the power of regular languages.

- The intersection and union of PBA can be realized as in the nondeterministic case. For the complementation of PBA, we proposed a technique that relies on the switch to an equivalent probabilistic Rabin automaton that accepts all words either with probability 0 or 1 and whose size is exponential in the size of the original PBA. For this switch we used an advanced powerset construction that shares its basic ideas with Safra’s determinization procedure [Saf88]. Using the duality of Rabin and Streett acceptance and a polynomial transformation from probabilistic Streett automata to PBA this yields a method for the complementation of PBA with a possible exponential blow-up. The complexity of the latter transformation might be surprising, as in the nondeterministic case the switch from Streett to Büchi acceptance can cause an exponential blow-up [SV89].

- We showed the undecidability of the emptiness problem for PBA and of related problems using the fact that the accepted language of a PBA does not only depend on the topological structure of the automaton, but also on the precise transition probabilities.

- We considered PBA under a threshold semantics and showed that the class of recognizable languages might be (depending on the threshold) a proper superset of the class of languages that are recognizable by PBA with the standard semantics (i.e. the threshold equals zero).

- We also investigated the so-called almost-sure semantics which requires that "the accepting runs have measure one". Switching to the almost sure semantics, the Büchi acceptance criterion is no longer powerful enough for the full class of ω-regular languages. However, for the Rabin and Streett acceptance criterion the class of recognizable languages agrees with the class of PBA-recognizable languages under the standard semantics.

- Although the accepted language of an almost-sure PBA does not only depend on the topological structure of the automaton, but also on the precise transition probabilities, we established the decidability of the emptiness problem for PBA with the almost-sure semantics. We moreover showed that a nonempty almost-sure PBA recognizable language contains a finite-memory word.

- In the more general framework of POMDPs we showed that

  - the positive Büchi objective is undecidable (concerning deterministic schedulers).
  - the almost-sure co-Büchi objective is undecidable (concerning deterministic schedulers).
  - the almost-sure Büchi objective is decidable.
  - the positive co-Büchi objective is decidable.

Although PBA cannot be used for the qualitative verification of MDPs, they do apply to the qualitative verification of Markov chains [BG05]. Moreover, as PBA are a special case of POMDPs, the established undecidability results have a relevance for partial information games with ω-regular winning objectives [CDHR06] as well as POMDPs [Son71, Mon82, PT87, Lov91, BdRS96], which are used to model a wide range of applications, such as mobile robot navigation, probabilistic planning task, elevator control, etc. PBA also find an application in randomized monitoring [CSV08].

But there are also many open questions. For example, we would like to have a different characterization of the class of PBA-acceptable languages. That could be a logical characterization by means of a probabilistic variant of MSO or an algebraic characterization, e.g. an extension of ω-regular expressions as in [BC06b]. One can also add the concept of nondeterminism to probabilistic ω-automata, i.e. being in a state of the automaton, a letter of the alphabet does not impose a certain probability distribution over the successor states, but leads to a nondeterministic choice over several such distributions. We claim that such non-probabilistic PBAs under the semantics that a word is accepted if there exists a scheduler compliant to the word such that the set of accepting runs has positive measure are more expressive than standard PBA.

Bibliography

[ACY95] R. Alur, C. Courcoubetis, and M. Yannakakis. Distinguishing tests for nondeterministic and probabilistic machines. In *Proc. of the 27th ACM Symposium on Theory of Computing (STOC '95)*, pages 363–372. ACM, 1995.

[AF98] A. Ambainis and R. Freivalds. 1-way quantum finite automata: strengths, weaknesses and generalizations. In *Proc. of the 39th Symposium on Foundations of Computer Science (FOCS '98)*. IEEE Computer Society Press, 1998.

[ASBB95] A. Aziz, V. Singhal, F. Balarin, and R.K. Brayton. It usually works: The temporal logic of stochastic systems. In *Proc. of the 7th International Conference on Computer Aided Verification (CAV '95)*, volume 939 of *Lecture Notes in Computer Science*, pages 155–165. Springer, 1995.

[Bai98] C. Baier. *On the algorithmic verification of probabilistic systems*. Universität Mannheim, 1998. Habilitation Thesis.

[Bau78] H. Bauer. *Wahrscheinlichkeitstheorie und Grundzüge der Maßtheorie*. de Gruyter, 1978.

[BBG08] C. Baier, N. Bertrand, and M. Grösser. On decision problems for probabilistic Büchi automata. In *Proc. of the 11th International Conference on Foundations of Software Science and Computation Structures (FOSSACS '08)*, volume 4962 of *Lecture Notes in Computer Science*, pages 287–301. Springer, 2008.

[BC03] V. Blondel and V. Canterini. Undecidable problems for probabilistic finite automata. *Theory of Computer Systems*, 36:231–245, 2003.

[BC06a] C. Baier and F. Ciesinski. LiQuor: A tool for qualitative and quantitative linear time analysis of reactive systems. In QEST 2006 [QES06].

[BC06b] M. Bojanczyk and T. Colcombet. Bounds in omega-regularity. In *Proc. of the 21st IEEE Symposium on Logic in Computer Science (LICS '06)*, pages 285–296. IEEE Computer Society Press, 2006.

[BCG05] C. Baier, F. Ciesinski, and M. Grösser. Quantitative analysis of distributed randomized protocols. In *Proc. of the 10th International Workshop on Formal Methods for Industrial Critical Systems (FMICS 2005)*. ACM Press, 2005.

[BCHG+97] C. Baier, E. Clarke, V. Hartonas-Garmhausen, M. Kwiatkowska, and M. Ryan. Symbolic model checking for probabilistic processes. In *Proc. of the 24th International Colloqium on Automata, Languages and Programming (ICALP '97)*, volume 1256 of *Lecture Notes in Computer Science*, pages 430–440. Springer, 1997.

[BCM+92] J. Burch, E. Clarke, K. McMillan, D. Dill, and L. Hwang. Symbolic model checking: 10^{20} states and beyond. *Information and Computing*, 98(2):142–170, 1992.

[BdA95] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proc. of the 15th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '95)*, volume 1026 of *Lecture Notes in Computer Science*, pages 499–513. Springer, 1995.

[BDG06] C. Baier, P.R. D'Argenio, and M. Grösser. Partial order reduction for probabilistic branching time. In *Proc. of the 3rd Workshop on Quantitative Aspects of Programming Languages (QAPL '05)*, volume 153(2) of *Electronic Notes in Theoretical Computer Science*, pages 97–116. Elsevier, 2006.

[BdRS96] D. Burago, M. de Rougemont, and A. Slissenko. On the complexity of partially observed markov decision processes. *Theoretical Computer Science*, 157(2):161–183, 1996.

[Bea02] D. Beauquier. Markov decision processes and deterministic Büchi automata. *Fundamenta Informaticae*, 50(1):1–13, 2002.

[BEMC00] C. Baier, B. Engelen, and M. Majster-Cederbaum. Deciding bisimularity and similarity for probabilistic processes. *Jounal of Computer and System Sciences*, 60:187–231, 2000.

[BG05] C. Baier and M. Grösser. Recognizing ω-regular languages with probabilistic automata. In *Proc. of the 20th IEEE Symposium on Logic in Computer Science (LICS '05)*, pages 137–146. IEEE Computer Society Press, 2005.

[BGC04] C. Baier, M. Grösser, and F. Ciesinski. Partial order reduction for probabilistic systems. In QEST 2004 [QES04], pages 230–239.

[BHH+03] C. Baier, B. Haverkort, H. Hermanns, J.-P. Katoen, and M. Siegle, editors. *Validation of Stochastic Systems*, volume 2925 of *Lecture Notes in Computer Science*. Springer, 2003.

[BK98] C. Baier and M. Kwiatkowska. Model checking for a probabilistic branching time logic with fairness. *Distributed Computing*, 11, 1998.

[BM88] L. Babai and S. Moran. Arthur-Merlin games: a randomized proof system, and a hierarchy of complexity classes. *Journal of Computer Science*, 36:254–276, 1988.

[BM99] M. Bozga and O. Maler. On the Representation of Probabilities over Structured Domains. In *Proc. of the 11th International Conference on Computer Aided Verification (CAV '99)*, volume 1633 of *Lecture Notes in Computer Science*, pages 261–273. Springer, 1999.

[BRV04] D. Bustan, S. Rubin, and M. Y. Vardi. Verifying ω-regular properties of Markov chains. In *Proc. of the 16th International Conference on Computer Aided Verification (CAV '04)*, volume 3114 of *Lecture Notes in Computer Science*, pages 189–201. Springer, 2004.

[BS98] D. Beauquier and G. Slissenko. Polytime model checking for timed probabilistic computation tree logic. *Acta Informatica 35*, pages 645–664, 1998.

[BT00] V. Blondel and J. Tsitsiklis. A survey of computational complexity results in systems and control. *Automatica*, 36(9):1249–1274, 2000.

[CDHR06] K. Chatterjee, L. Doyen, T.A. Henzinger, and J.-F. Raskin. Algorithms for ω-regular games with imperfect information. In *Proc. of the 20th International Workshop on Computer Science Logic (CSL '06)*, volume 4207 of *Lecture Notes in Computer Science*, pages 287–302. Springer, 2006.

[CE81] E. Clarke and E. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Proc. of the Workshop on Logics of Programs 1981*, volume 131 of *Lecture Notes in Computer Science*, pages 52–71. Springer, 1981.

[CGP99] E. Clarke, O. Grumberg, and D. Peled. *Model Checking*. MIT Press, 1999.

[CS02] S. Cattani and R. Segala. Decision algorithms for probabilistic bisimulation. In *Proc. of the 13th International Conference on Concurrency Theory (CONCUR '02)*, volume 2421 of *Lecture Notes in Computer Science*, pages 371–385. Springer, 2002.

[CSS03] J.-M. Couvreur, N. Saheb, and G. Sutre. An optimal automata approach to LTL model checking of probabilistic systems. In *Proc. of the 10th International Conference on Logic for Programming, Artificial Intelligence and Reasoning (LPAR)*, volume 2850 of *Lecture Notes in Artificial Intelligence*, pages 361–375, 2003.

[CSV08] R. Chadha, A.P. Sistla, and M. Viswanathan. On the expressiveness and complexity of randomization in finite state monitors. In *Proc. of the 23rd IEEE Symposium on Logic in Computer Science (LICS '08)*, pages 18–29. IEEE Computer Society Press, 2008.

[CY90] C. Courcoubetis and M. Yannakakis. Markov decision processes and regular events. In *Proc. of the 17th International Colloquium on Automata, Languages and Programming (ICALP '90)*, volume 443 of *Lecture Notes in Computer Science*, pages 336–349. Springer, 1990.

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

[dA97a] L. de Alfaro. *Formal Verification of Probabilistic Systems*. PhD thesis, Stanford University, Department of Computer Science, 1997.

[dA97b] L. de Alfaro. Temporal logics for the specification of performance and reliability. In *Proc. of the 14th Annual Symposium on Theoretical Aspects of Computer Science (STACS '97)*, volume 1200 of *Lecture Notes in Computer Science*, pages 165–176. Springer, 1997.

[dA98] L. de Alfaro. Stochastic transition systems. In *Proc. of the 9th International Conference on Concurrency Theory (CONCUR '98)*, volume 1466 of *Lecture Notes in Computer Science*, pages 423–438. Springer, 1998.

[dA99] L. de Alfaro. The verification of probabilistic systems under memoryless partial-information policies is hard. In *Proc. of the 2nd International Workshop on Probabilistic Methods in Verification (ProbMiV '99)*, pages 19–32. Birmingham University, Research Report CSR-99-9, 1999.

[Des99] J. Desharnais. *Labelled Markov Processes*. PhD thesis, McGill University, Montreal, 1999.

[DGJP02] J. Desharnais, V. Gupta, R. Jagadeesan, and P. Panangaden. Weak bisimulation is sound and complete for pctl*. In *Proc. of the 13th International Conference on Concurrency Theory (CONCUR '02)*, volume 2421 of *Lecture Notes in Computer Science*. Springer, 2002.

[DJJL01] P.R. D'Argenio, B. Jeannet, H. Jensen, and K. Larsen. Reachability analysis of probabilistic systems by successive refinements. In *Proc. of the 1st Joint International Workshop on Process Algebra and Performance Modelling and Probabilistic Methods in Verification (PAPM-PROBMIV '01)*, volume 2399 of *Lecture Notes in Computer Science*, pages 57–76. Springer, 2001.

[DK03] M. Droste and D. Kuske. Skew and infinitary formal power serie. In *Proc. of the 30th International Colloquium on Automata, Languages and Programming (ICALP '03)*, volume 2719 of *Lecture Notes in Computer Science*, pages 426–438. Springer, 2003.

[DN04] P.R. D'Argenio and P. Niebert. Partial order reduction on concurrent probabilistic programs. In QEST 2004 [QES04], pages 240–249.

[DP04] M. Droste and U. Püschmann. Weighted Büchi automata. In *Workshop on Weighted Automata–Theory and Applications*, page 56, Dresden, 2004.

[DS90] C. Dwork and L. Stockmeyer. A time-complexity gap for two-way probabilistic finite state automata. *SIAM Journal of Computing*, 19:1011–1023, 1990.

[FOS03] R. Freund, M. Oswald, and L. Staiger. Omega-P automata with communication rules. In *Workshop on Membrane Computing*, volume 2933 of *Lecture Notes in Computer Science*, pages 203–217. Springer, 2003.

[Fre81] R. Freivalds. Probabilistic two-way machines. In *Proc. of the 10th International Symposium on Mathematical Foundations of Computer Science (MFCS '81)*, volume 118 of *Lecture Notes in Computer Science*, pages 33–45. Springer, 1981.

[GD07] S. Giro and P.R. D'Argenio. Quantitative model checking revisited: neither decidable nor approximable. In *Proc. of the 5th International Conference on Formal Modelling and Analysis of Timed Systems (FORMATS '07)*, volume 4763 of *Lecture Notes in Computer Science*, pages 179–194. Springer, 2007.

[GKPP95] R. Gerth, R. Kuiper, D. Peled, and W. Penczek. A partial order approach to branching time logic model checking. In *Proc. of the 3rd Israel Symposium on the Theory of Computing Systems (ISTCS '95)*, pages 130–139. IEEE Computer Society Press, 1995.

[GNB+06] M. Grösser, G. Norman, C. Baier, F. Ciesinski, M. Kwiatkowska, and D. Parker. On reduction criteria for probabilistic reward models. In *Proc. of the 26th Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS '06)*, volume 4337 of *Lecture Notes in Computer Science*, pages 309–320. Springer, 2006.

[GO01] P. Gastin and D. Oddoux. Fast LTL to Büchi automata translation. In *Proc. of the 13th International Conference on Computer Aided Verification (CAV '01)*, volume 2102 of *Lecture Notes in Computer Science*, pages 53–65. Springer, 2001.

[God96] P. Godefroid. *Partial Order Methods for the Verification of Concurrent Systems: An Approach to the State Explosion Problem*, volume 1032 of *Lecture Notes in Computer Science*. Springer, 1996.

[GPS96] P. Godefroid, D. Peled, and M. Staskauskas. Using partial-order methods in the formal validation of industrial concurrent programs. In *Proc. of the International Symposium on Software Testing and Analysis (ISSTA '96)*, pages 261–269. ACM Press, 1996.

[GPVW96] R. Gerth, D. Peled, M.Y. Vardi, and P. Wolper. Simple on-the-fly automatic verification of linear temporal logic. In *Proc. of the 15th IFIP WG6.1 International Symposium on Protocol Specification, Testing and Verification XV*, pages 3–18. Chapman & Hall, Ltd., 1996.

[GTW02] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[Han94] H. A. Hansson. Time and probability in formal design of distributed systems. SICS Dissertation Series 05, Swedish Institute of Computer Science, 1994.

[HGCC99] V. Hartonas-Garmhausen, S. Campos, and E. Clarke. Probverus: Probabilistic symbolic model checking. In *Proc. of the 5th International AMAST on Formal Methods for Real-Time and Probabilistic Systems (ARTS '99)*, volume 1601 of *Lecture Notes in Computer Science*, pages 96–110. Springer, 1999.

[HJ94] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[HKN$^+$03] H. Hermanns, M. Kwiatkowska, G. Norman, D. Parker, and M. Siegle. On the use of mtbdds for performability analysis and verification of stochastic systems. *Journal of Logic and Algebraic Programming: Special Issue on Probabilistic Techniques for the Design and Analysis of Systems*, 56:23–67, 2003.

[HMPS94] G. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Probabilistic Analysis of Large Finite State Machines. In *Proc. of the 31st Design Automation Conference (DAC '94)*. ACM Press, 1994.

[Hol03] G. Holzmann. *The SPIN Model Checker, Primer and Reference Manual*. Addison Wesley, 2003.

[HP94] G. Holzmann and D. Peled. An improvement in formal verification. In *Proc. of the 7th International Conference on Formal Description Techniques for Distributed Systems and Communications Protocols (FORTE '94)*, pages 197–211. Chapman & Hall, 1994.

[HS02] H. Hermanns and R. Segala, editors. *Proc. of the 2nd Joint International Workshop on Process Algebra and Performance Modelling and Probabilistic Methods in Verification (PAPM-PROBMIV '02)*, volume 2399 of *Lecture Notes in Computer Science*. Springer, 2002.

[HSP83] S. Hart, M. Sharir, and A. Pnueli. Termination of probabilistic concurrent programs. *ACM Transactions on Programming Languages and Systems*, 5(3):356–380, 1983.

[HT92] T. Huynh and L. Tian. On some equivalence relations for probabilistic processes. *Fundamenta Informaticae*, 17:211–234, 1992.

[Hut02] M. Huth. Possibilistic and probabilistic abstraction-based model checking. In [HS02], pages 115–134, 2002.

[Hut05] M. Huth. Abstraction and probabilities for hybrid logics. In *Proc. of the 2nd workshop on Quantitative Aspects of Programming Languages (QAPL '04)*, volume 112 of *Electronic Notes in Theoretical Computer Science*, pages 61–76. Elsevier, 2005.

[JDL02] B. Jeannet, P.R. D'Argenio, and K.G. Larsen. RAPTURE: A tool for verifying Markov decision processes. In *Tools Day '02*, Technical Report. Masaryk University Brno, 2002.

[JL91] B. Jonsson and K. Larsen. Specification and refinement of probabilistic processes. In *Proc. of the 6th IEEE Symposium on Logic in Computer Science (LICS '91)*, pages 266–277. IEEE Computer Society Press, 1991.

[KB05] J. Klein and C. Baier. Experiments with deterministic ω-automata for formulas of linear temporal logic. In *Proc. of the 10th International Conference on Implementation and Application of Automata (CIAA '05)*, volume 3845 of *Lecture Notes in Computer Science*, pages 199–212. Springer, 2005.

[KB07] J. Klein and C. Baier. On-the-fly stuttering in the construction of deterministic ω-automata. In *Proc. of the 12th International Conference on Implementation and Application of Automata (CIAA '07)*, volume 4783 of *Lecture Notes in Computer Science*, pages 51–61. Springer, 2007.

[KKLW07] Joost-Pieter Katoen, Daniel Klink, Martin Leucker, and Verena Wolf. Three-valued abstraction for continuous-time Markov chains. In *Proc. of the 19th International Conference on Computer Aided Verification (CAV '07)*, volume 4590 of *Lecture Notes in Computer Science*, pages 316–329. Springer, 2007.

[KNP02] M. Kwiatkowska, G. Norman, and D. Parker. PRISM: Probabilistic Symbolic Model Checker. In *Proc. of the 12th International Conference on Modelling Tools and Techniques for Computer and Communication System Performance Evaluation (TOOLS '02)*, volume 2324 of *Lecture Notes in Computer Science*, pages 113–140. Springer, 2002.

[KNP04] M. Kwiatkowska, G. Norman, and D. Parker. Probabilistic symbolic model checking with PRISM: A hybrid approach. *International Journal on Software Tools for Technology Transfer (STTT)*, 6(2):128–142, 2004.

[KNP06a] M. Kwiatkowska, G. Norman, and D. Parker. Game-based abstraction for Markov decision processes. In QEST 2006 [QES06], pages 157–166.

[KNP06b] M. Kwiatkowska, G. Norman, and D. Parker. Symmetry reduction for probabilistic model checking. In *Proc. of the 18th International Conference on Computer Aided Verification (CAV '06)*, volume 4114 of *Lecture Notes in Computer Science*, pages 234–248. Springer, 2006.

[KSK66] J. G. Kemeny, J. L. Snell, and A. W. Knapp. *Denumerable Markov chains*. D. Van Nostrand Co., 1966.

[KV98] O. Kupferman and M. Y. Vardi. Freedom, weakness, and determinism: From linear-time to branching-time. In *Proc. of the 13th IEEE Symposium on Logic in Computer Science (LICS '98)*, pages 81–92. IEEE Computer Society Press, 1998.

[KW97] A. Kondacs and J. Watrous. On the power of quantum finite state automata. In *Proc. of the 38th Symposium on Foundations of Computer Science (FOCS '97)*, pages 66–75. IEEE Computer Society Press, 1997.

[Lam83] L. Lamport. Specifying concurrent program modules. *ACM TOPLAS*, 5(2):190–222, 1983.

[Lit96] M. Littman. *Algorithms for Sequential Decision Making*. PhD thesis, Brown University, Department of Computer Science, 1996.

[Lov91] W. Lovejoy. A survey of algorithmic methods for partially observable Markov decision processes. *Annals of Operations Research*, 28(1):47–65, 1991.

[LS91] K. Larsen and A. Skou. Bisimulation through probabilistic testing. *Information and Computation*, 94(1):1–28, 1991.

[MHC99] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and infinite-horizon partially observable Markov chains. In *Proc. of the 16th National Conference on Artificial Intelligence (AAAI '99)*, pages 541–548, 1999.

[MHC03] O. Madani, S. Hanks, and A. Condon. On the undecidability of probabilistic planning and related stochastic optimization problems. *Artificial Intelligence*, 147(1-2):5–34, 2003.

[Mic88] M. Michel. Complementation is more difficult with automata on infinite words. *CNET, Paris*, 1988.

[MLMdCT02] L. Mora-Lopez, R. Morales, M. Sidrach de Cardona, and F. Triguero. Probabilistic finite automata and randomness in nature: a new approach in the modelling and prediction of climatic parameters. In *Proc. of the International Environmental Modelling and Software Congress (iEMSs '02)*, pages 78–83, 2002.

[Mon82] G. Monahan. A survey of partially observable Markov decision processes: Theory, models, and algorithms. *Management Science*, 28(1):1–16, 1982.

[MP03] A. Miner and D. Parker. Symbolic representations and analysis of large probabilistic systems. In [BHH$^+$03], 2003.

[MS08] A. Morgenstern and K. Schneider. From LTL to symbolically represented deterministic automata. In *Proc. of the 9th International Conference on Verification, Model Checking, and Abstract Interpretation (VMCAI '08)*, volume 4905 of *Lecture Notes in Computer Science*, pages 279–293. Springer, 2008.

[Par02] D. Parker. *Implementation of Symbolic Model Checking for Probabilistic Systems*. PhD thesis, University of Birmingham, 2002.

[Paz66] A. Paz. Some aspects of probabilistic automata. *Information and Control*, 9, 1966.

[Paz71] A. Paz. *Introduction to probabilistic automata*. Academic Press Inc., 1971.

[Pel93] D. Peled. All from one, one for all: On model checking using representatives. In *Proc. of the 5th International Conference on Computer Aided Verification (CAV '93)*, volume 697 of *Lecture Notes in Computer Science*, pages 409–423. Springer, 1993.

[Pel97] D. Peled. Partial order reduction: Linear and branching time logics and process algebras. In [PPH97], pages 233–257, 1997.

[Pit06] N. Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *Proc. of the 21st IEEE Symposium on Logic in Computer Science (LICS '06)*, pages 255–264. IEEE Computer Society Press, 2006.

[PLS00] A. Philippou, I. Lee, and O. Sokolsky. Weak bisimulation for probabilistic systems. In *Proc. of the 11th International Conference on Concurrency Theory (CONCUR '00)*, volume 1877 of *Lecture Notes in Computer Science*, pages 334–349. Springer, 2000.

[Pnu77] A. Pnueli. The temporal logic of programs. In *Proc. of the 18th Symposium on the Foundations of Computer Science (FOCS '77)*, pages 46–57. IEEE Computer Society Press, 1977.

[PPH97] D. Peled, V. Pratt, and G. Holzmann, editors. *Partial Order Methods in Verification*, volume 29(10) of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1997.

[PT87] C. Papadimitriou and J. Tsitsiklis. The comlexity of Markov decision processes. *Mathematics of Operations Research*, 12(3), 1987.

[Put94] M. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley and Sons, 1994.

[PZ93] A. Pnueli and L. D. Zuck. Probabilistic verification. *Information and Computation*, 103(1):1–29, 1993.

[QES04] *Proceedings of the 1st International Conference on Quantitative Evaluation of SysTems (QEST '04)*. IEEE Computer Society Press, 2004.

[QES06] *Proceedings of the 3rd International Conference on Quantitative Evaluation of SysTems (QEST '06)*. IEEE Computer Society Press, 2006.

[Rab63] M. O. Rabin. Probabilistic automata. *Information and Control*, 6(3):230–245, 1963.

[Rei84] J. H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer System Sciences*, 29(2):274–301, 1984.

[Rei98] R. D. Reisz. A characterization theorem for probabilistic automata over infinite words. In *50 de ani de la Infiintarea Faculatatii de Matematica, Conference Timisoara*, 1998.

[Rei99] R. D. Reisz. Decomposition theorems for probabilistic automata over infinite objects. *Informatica, Lithuanian Academy of Science*, 10(4):427–440, 1999.

[RST96] D. Ron, Y. Singer, and N. Tishby. The power of amnesia: Learning probabilistic automata with variable memory length. *Machine Learning*, 25(2–3):117–149, 1996.

[Saf88] S. Safra. On the complexity of ω-automata. In *Proc. of the 29th Symposium on Foundations of Computer Science (FOCS '88)*, pages 319–327. IEEE Computer Society Press, 1988.

[Sch97] K. Schneider. Translating linear temporal logic to deterministic ω-automata. In R. Hagelauer and M. Pfaff, editors, *Methoden des Entwurfs und der Verifikation digitaler Systeme*, pages 149–158, Linz, Österreich, 1997. Universitätsverlag Rudolf Trauner.

[Seg95] R. Segala. *Modeling and Verification of Randomized Distributed Real-Time Systems*. PhD thesis, Massachusetts Institute of Technology, 1995.

[SL95] R. Segala and N. Lynch. Probabilistic simulations for probabilistic processes. *Nordic Journal of Computing*, 2(2):250–273, 1995.

[Son71] E. J. Sondik. *The Optimal Control of Partially Observable Markov Processes*. PhD thesis, Stanford University, 1971.

[SV89] S. Safra and M. Y. Vardi. On ω-automata and temporal logic. In *Proc. of the 21st ACM Symposium on Theory of Computing (STOC '89)*, pages 127–137. ACM, 1989.

[Tho90] W. Thomas. Automata on infinite objects. In *Handbook of Theoretical Computer Science*, volume B, chapter 4, pages 133–191. Elsevier Science, 1990.

[Val90] A. Valmari. A stubborn attack on state explosion. In *Proc. of the 2nd International Conference on Computer Aided Verification (CAV '90)*, volume 531 of *Lecture Notes in Computer Science*. Springer, 1990.

[Val92] A. Valmari. A stubborn attack on state explosion. *Formal Methods in System Design*, 1:297–322, 1992.

[Val94] A. Valmari. State of the art report: Stubborn sets. *Petri-Net Newsletters*, 46:6–14, 1994.

[Val97] A. Valmari. Stubborn set methods for process algebras. In [PPH97], pages 213–231, 1997.

[Var85] M. Y. Vardi. Automatic verification of probabilistic concurrent finite-state programs. In *Proc. of the 26th Symposium on Foundations of Computer Science (FOCS '85)*, pages 327–338. IEEE Computer Society Press, 1985.

[Var96] M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Proc. of the Banff Higher Order Workshop on Logics for Concurrency: Structure versus Automata*, pages 238–266. Springer-Verlag, 1996.

[vGSST90] R. van Glabbeek, S. Smolka, B. Steffen, and C. Tofts. Reactive, generative, and stratified models of probabilistic processes. In *Proc. of the 5th IEEE Symposium on Logic in Computer Science (LICS '90)*, pages 130–141. IEEE Computer Society Press, 1990.

[VW86] M. Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. In *Proc. of the 1st IEEE Symposium on Logic in Computer Science (LICS '86)*, pages 332–345. IEEE Computer Society Press, 1986.