Masaryk University
Faculty of Informatics

# Decidability and Complexity of Finite-State Stochastic Games

Master's thesis

**Vojtěch Forejt**

Brno, 2007

# DECLARATION

Hereby I declare, that this paper is my original authorial work, which I have worked out by my own. All sources, references and literature used or excerpted during elaboration of this work are properly cited and listed in complete reference to the due source.

Brno, January 4, 2007
Vojtěch Forejt

**Advisor:** Mgr. Tomáš Brázdil

# Acknowledgement

I am grateful to my adviser and co-worker Tomáš Brázdil. Without his advices, guidance and encouragement, this thesis would hardly ever been written. My thesis is mainly based on the work of our team, so I would like to thank my other co-workers, Václav Brožek and Antonín Kučera. I would also like to thank Jan Holeček for his moral support and for sharing his experience with writing Master's thesis.

Lastly, and most importantly, I wish to thank my wife Lucie, my parents and my whole family for their encouragement and for providing a loving environment.

# ABSTRACT

We study stochastic turn based games played by two players ($\Box$ and $\Diamond$) and environment $\bigcirc$. The aim of the player $\Box$ is to provide a strategy that ensures satisfying of an objective no matter what strategy the player $\Diamond$ provides. Objectives are specified by the formulae of the logic PCTL (probabilistic extension of CTL) or PECTL$^*$ (logic similar to PCTL$^*$ with Büchi automata as temporal operators).

These games are not determined and the problem of synthesizing a winning strategy is undecidable. We prove that if we restrict ourselves to memoryless strategies, problems become decidable. For certain subclasses, we give completeness results. Our main result is synthesizing finite memory winning strategies for $1\frac{1}{2}$-player games and PECTL$^*$ objectives. We show that this problem is decidable in 2-**EXPTIME**.

# Keywords

Controller-synthesis, probability, game theory, PCTL, Büchi automata

# CONTENTS

CHAPTER 1

# INTRODUCTION

In 1936 Alan Turing proved programs have properties that cannot be verified by computers [19]. However, it is very useful to verify some properties. For example, verification of programs could prevent some software or hardware bugs or even very serious disasters (e.g. Ariane 5 Flight 501 [18]). For decades computer scientists have studied various modelling methods of programs and their properties. The aim is to prove decidability or undecidability of the verification problems and to determine the complexity of decidable problems.

In this thesis we concentrate on the finite state stochastic games. This formalism can be used for describing reactive systems or systems for which the exact behavior is unknown.

In our case, a $2\frac{1}{2}$**-player game** is a finite graph whose vertices are owned by one of two players (namely $\Box$ and $\Diamond$) or by environment $\bigcirc$. A play is initiated in a distinguished vertex of a graph and players take their turns in the vertices they own according to a chosen **strategy**. The aim of the player $\Box$ is to **synthesize** a strategy so that a given **property** is satisfied no matter what the player $\Diamond$ and the ENVIRonment do. Properties studied in this thesis are given in the logic **PCTL**, which is a probabilistic equivalent of CTL [7] or in the logic PECTL$^*$, which is the variant of the logic PCTL$^*$ that uses Büchi automata instead of temporal operators.

**Related work.** Stochastic games have been widely studied in recent years. There are other works dealing with the PCTL objectives (e.g. [8]). The properties expressed in linear time logics have also been studied (e.g. [5], [9]). Some works study more general games such as incomplete information games ([10], [11]) and concurrent games ([12], [3]). One can also consider games in which a state space is not finite, but infinite (for example described by BPA or PDA [2], [13], [20], [21]).

**Our contribution.** We show that $2\frac{1}{2}$-player games are not determined even for some restricted fragment of PCTL. For general strategies and whole PCTL the problem of synthesizing a winning strategy is undecidable. Undecidability holds even for **finite memory** strategies (the strategies that can be effectively implemented on computer) [1]. This gives a motivation to study fragments of games and logics. One of the possibilities are so-called **memoryless** strategies. These are strategies which do not need the history of the play, but only the current state. We show that the controller-synthesis problem for **deterministic** strategies is $\mathbf{NP}^{\mathbf{coNP}}$-complete. With the **NP**-completeness result for $1\frac{1}{2}$-player games [8], we obtain the complete classification for memoryless deterministic strategies. The lower complexity bound carries over to memoryless **randomized** strategies. Best known upper bound for $1\frac{1}{2}$-player and $2\frac{1}{2}$-player games is **PSPACE** and **EXPTIME**, respectively [14]. We were not able to improve any lower and upper bounds. Instead, we show that the lower bounds are likely to be outside both **NP** and $\mathbf{NP}^{\mathbf{coNP}}$ classes. This is done by the reduction from the *square root sum problem*, which is known to be in **PSPACE**, but its exact complexity is long-standing open problem in computational geometry.

The main result of this thesis is the decidability of synthesizing a winning finite memory strategy for $1\frac{1}{2}$-player games and the qualitative PECTL$^*$ objectives. Our proof uses ideas of the proof for the PCTL given in [1]. Nevertheless, the proof for PECTL$^*$ is much more technical and complicated. We also obtain a different complexity: double exponential for PECTL$^*$ in contrast to single exponential for PCTL. This corresponds to the fact that PECTL$^*$ is much stronger than PCTL.

**Structure of this thesis.** In Chapter 2 we introduce some notions and formalisms used for describing systems and their properties. Chapter 3 contains undecidability results and proof of non-determinacy. Chapter 4 contains results related to memoryless strategies and Chapter 5 contains the result related to finite memory strategies. In Chapter 6 we conclude our work and sketch some possible future work.

CHAPTER 2

# DEFINITIONS

In the first part of this chapter, we introduce basic notions of probability theory. Then we continue by explaining some types of probabilistic transition systems. At the end of this chapter, we show how to describe properties of such systems.

## 2.1 Probability

A **probability distribution** on a finite set $A$ is a function $f : A \to [0,1]$ where the condition $\Sigma_{a\in A} f(a) = 1$ holds. In this thesis, we mainly concentrate ourselves in rational probability distributions, i.e. $f(a) \in \mathbb{Q}$ for all $a \in A$. If we concern different type of distributions, we mention it explicitly. A probability distribution is Dirac if $f(a) = 1$ for some $a \in A$.

A **σ-field** over $X$ is a set $F \subseteq 2^X$ that includes X and is closed under complement and finite union. A **measurable space** is a pair $(X,F)$ in which $X$ is called sample space and $F$ is a σ-field over X. A probability measure over measurable space $(X,F)$ is a function $\mathcal{P} : F \to \mathbb{R}^{\geq 0}$ where $\mathcal{P}(X) = 1$ and for each countable collection $\{X_i \mid i \in I\}$ of pairwise elements of $F$ we have $\mathcal{P}(\bigcup_{i\in I} X_i) = \sum_{i\in I} \mathcal{P}(X_i)$. A **probabilistic space** is a triple $(X,F,\mathcal{P})$ where $(X,F)$ is a measurable space and $\mathcal{P}$ is a probability measure over $(X,F)$.

## 2.2 Probabilistic transition systems

### 2.2.1 Markov chains

Formally, a **Markov chain** is a triple $\mathcal{T} = (S, \to, Prob)$ such that

- $S$ is a finite or countably infinite set of **states**. We say that $\mathcal{T}$ is finite if $S$ is finite.

- $\to \subseteq S \times S$ is a **transition relation**. Instead of $(s,t) \in \to$ we write $s \to t$,

- *Prob* is a function which to each transition $s \to t$ assigns its probability $Prob(s \to t) \in (0,1]$. Moreover, for each $s \in S$ we have $\sum_{s \to t} Prob(s \to t) = 1$. For convenience, we write $s \xrightarrow{x} t$ instead of $Prob(s \to t) = x$ in the rest of this thesis.

**Runs and measuring their probability**

A **path** in a Markov chain $T$ is a finite or infinite sequence $w = s_0, s_1, \ldots$ of states such that $s_i \to s_{i+1}$ for every $i$. In this thesis we use $w(i)$ to denote the state $s_i$ of $w$. We also use $w^{\triangleleft i}$ to denote the prefix $s_0, \ldots s_i$ and $w^{i \triangleright}$ to denote suffix $s_i \ldots$.

A **run** is an infinite path. The set of all finite paths (resp. all runs) of $T$ is denoted $FPath$ (resp. $Run$). The set of all finite paths (resp. runs) that start in a given $s \in S$ is denoted $FPath(s)$ (resp. $Run(s)$).

Each $w \in FPath$ uniquely determines a basic cylinder $Run(w)$ which consists of all runs that start with $w$. To every $s \in S$ we associate the probabilistic space $(Run(s), F, \mathcal{P})$ where

- $F$ is the $\sigma$-field generated by all basic cylinders $Run(w)$ where $w(0) = s$.
- $\mathcal{P} : F \to [0,1]$ is the probability function in which for each $w = s_0, \ldots, s_m$ we have $\mathcal{P}(Run(w)) = \prod_{i=0}^{m-1} x_i$ if $s_i \xrightarrow{x_i} s_{i+1}$ for every $0 \leq i < m$. If the length of $w$ is 0, we put $\mathcal{P}(Run(w)) = 1$.

**Definition 2.2.1.** *Let $\mathcal{T} = (S, \to, Prob)$ be a finite Markov chain. The set $C \subseteq S$ is called bottom strongly connected component (b.s.c.c.) if for all $s, t \in C$ there is a path from $s$ to $t$ and whenever there is a path from $s \in C$ to $t \in S$, then $t \in C$.*

If we restrict states of Markov chain to some b.s.c.c., we obtain Markov chain that is strongly connected (in the graph theoretic sense).

**Lemma 2.2.2.** **[22]** *Let $\mathcal{T} = (S, \to, Prob)$ be a finite Markov chain. Let $R(s)$ be the set of all runs that are initiated in $s \in S$ and enter some b.s.c.c.. Then $\mathcal{P}(R(s)) = 1$.*

**Lemma 2.2.3.** **[22]** *Let $\mathcal{T} = (S, \to, Prob)$ be a strongly connected finite Markov chain. Let $R(s)$ be the set of all runs that are initiated in $s \in S$ and go through every state $t \in S$ infinitely often. Then $\mathcal{P}(R(s)) = 1$.*

**Definition 2.2.4.** *Let $\mathcal{T} = (S, \to, Prob)$ and $\mathcal{T}' = (S', \to, Prob')$ be Markov chains and let $\Theta : S \to S'$ be a surjective function. We say that $\Theta$ is a* **quotient** *of $\mathcal{T}$ onto $\mathcal{T}'$ if for every $s \in S$ and every $s' \in S'$ such that $\Theta(s) \xrightarrow{x} s'$ there is $t \in \Theta^{-1}(s')$ satisfying $s \xrightarrow{x} t$.*

We can naturally extend $\Theta$ to $Path$ of $\mathcal{T}$ so that for all $v \in Path$ and $0 \leq i \leq |v|$ we have $\Theta(v)(i) = \Theta(v(i))$.

**Lemma 2.2.5.** **[21]** *Let $\Theta$ be a quotient of $M$ onto $M'$ and let $s \in S$. Then $\Theta$ maps $Run(s)$ isomorphically onto $Run(\Theta(s))$. Moreover, $A \subseteq Run(s)$ is measurable if and only if $\Theta(A) \subseteq Run(\Theta(s))$ is measurable. In such case, $\mathcal{P}(A) = \mathcal{P}(\Theta(A))$*

### 2.2.2 $2\frac{1}{2}$-player games

This thesis in focused on $2\frac{1}{2}$-player games. Informally, these games are transition systems with three types of states. Each type of states is owned by one of three **players**, namely $\Box$, $\Diamond$ and $\bigcirc$. Player $\Box$ is "good", $\Diamond$ is the "bad" and $\bigcirc$ simulates random events in environment. In fact, player $\bigcirc$ never has a real choice (that's why he is sometimes called "half player"). Play of a game starts with a token in some distinguished vertex. Token is then repeatedly moved to other vertices. Each player controls the token only in his vertex. The aim of the player $\Box$ is to provide strategy such that a given objective is satisfied no matter what strategy the player $\Diamond$ provides.

Formally, $2\frac{1}{2}$-player game $G$ is a tuple $(V, E, (V_\Box, V_\Diamond, V_\bigcirc), Prob)$ where

- $V$ is a finite set of **states**,
- $E \subseteq V \times V$ is a set of **transitions**. We assume that each vertex has at least one outgoing transition,
- $(V_\Box, V_\Diamond, V_\bigcirc)$ is a partition of $V$ (where any of three components can be an empty set),
- $Prob$ assigns to each $v \in V_\bigcirc$ a rational probability over outgoing transitions of $v$.

If we restrict $V_\Diamond$ to be $\emptyset$, we obtain class of so called $1\frac{1}{2}$-player games, sometimes called Markov decision processes. If we restrict both $V_\Box$ and $V\Diamond$ to be $\emptyset$ we obtain class of all finite Markov chains.

When drawing graphs of a game, states of each player are depicted by appropriate shape. Probabilities are not drawn when they are not important for the context.

### 2.2.3 Strategies

If we want to measure probabilities in $2\frac{1}{2}$-player games, we have to resolve the nondeterminism in the vertices of $V_\Box$ and $V_\Diamond$. This can be done using **strategies**. Informally, a strategy advises a player which transitions should he use in which state, depending on the "history" of the "play".

Formally, having a $2\frac{1}{2}$-player game $G$, a strategy for player $\Box$ is a function $\sigma$ which to each $ws \in V^*V_\Box$ assigns a probability distribution over outgoing vertices of $s$.

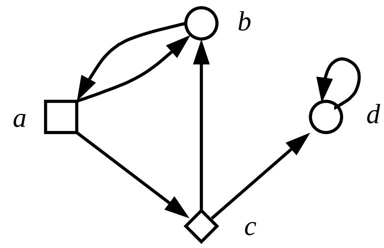



Figure 2.1: Example of $2\frac{1}{2}$-player game

In many examples and proofs we consider only restricted classes of strategies. Possible restrictions are:

**Restriction of randomization** Strategy that always advises Dirac distribution is called **deterministic** (denoted D). Strategies that are not necessarily deterministic are called **randomized** (denoted R).

**Restriction of memory** We can restrict the strategies so that $\sigma(ws) = \sigma(w's)$ for all $w, w' \in V^*$ and $s \in V$. We call this restricted strategies **memoryless** (denoted M). Strategies that are not necessarily memoryless are called **history-dependent** (denoted H). Other possible restriction is **finite memory** (denoted F). Strategy has finite memory, if there is a deterministic finite automaton $A = (Q, V, \delta, q_0)$ such that if $\delta(q_0, ws) = \delta(q_0, w's)$, then $\sigma(ws) = \sigma(w's)$.

Having the above given notation, we use the abbreviations MD, MR, FD, FR, HD and HR to denote memoryless deterministic, memoryless randomized, finite memory deterministic, finite memory randomized, history-dependent deterministic and history-dependent randomized strategies. Reader can easily verify that $MD \subseteq MR \subseteq FR \subseteq HR$, $MD \subseteq FD \subseteq HD \subseteq HR$ and $FD \subseteq FR$, but others are incomparable.

Strategies $\pi$ for player $\Diamond$ are defined similarly.

#### From $2\frac{1}{2}$-player games to Markov chains

Every tuple $(\sigma, \pi)$ of strategies for player $\Box$ and $\Diamond$ determines a Markov chain $G(\sigma, \pi)$ (abbreviated to $G(\sigma)$ in $1\frac{1}{2}$-player games) in which $V^+$ is a set of states and $ws \xrightarrow{x} wss'$ if and only if $(s, s') \in E$ and one of the following conditions is satisfied:

- $s \in V_{\bigcirc}$ and $Prob(s, s') = x$,
- $s \in V_{\Box}$ and $\sigma(ws)$ assigns $x$ to $(s, s')$,

- $s \in V_\Diamond$ and $\pi(ws)$ assigns $x$ to $(s, s')$.

We use $last(ws)$ to denote $s \in V$ for run $ws$. Sometimes we do not use *last* and write only $s$ instead of $ws$ when the formal meaning is clear from the context. As an example, consider the game from Figure 2.1 and MR strategy $\sigma$ for player $\Box$ which (in state $a$) chooses transition $a \xrightarrow{\frac{1}{3}} b$ and $a \xrightarrow{\frac{2}{3}} c$. Strategy $\pi$ for player $\Diamond$ always chooses $c \xrightarrow{1} b$. Part of the resulting Markov chain $G(\sigma, \pi)$ is depicted in Figure 2.2. Note that $G(\sigma, \pi)$ is a forest, but only a tree with the root $a$ is depicted here.

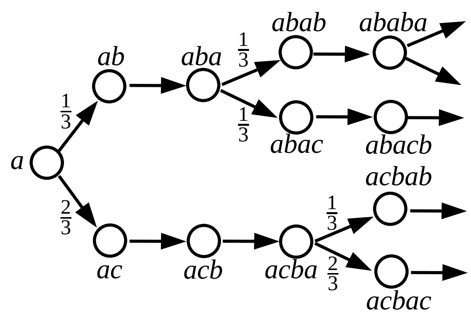



Figure 2.2: $G(\sigma, \pi)$

### 2.2.4 Finite automata on infinite words

Finite automata on infinite words are extension of finite automata. Instead of regular languages they accept $\omega$-regular languages [15]. Formally, finite automaton on infinite words is a tuple $\mathcal{A} = (\Sigma, S, \delta, s_0, F)$ where

- $\Sigma$ an is input alphabet,
- $S$ is a finite set of states,
- $\delta : S \times \Sigma \to 2^S$ is a transition function,
- $s_0 \in S$ is an initial state,
- $F$ is an accepting condition (which varies depending on the type of automaton we use).

Given a word $w \in \Sigma^*$, run of $\mathcal{A}$ on $w$ is a sequence $s_0 s_1 \dots$ such that $s_{i+1} \in \delta(s_i, w(i))$. Word $w$ is accepted if there is an accepting run (notion of accepting run will be introduced later). By $\mathsf{infi}(s_0 s_1 \dots)$ we denote set of all states that occur infinitely often in $s_0 s_1 \dots$.

We say that an automaton $\mathcal{A}$ is **deterministic** if $|\delta(s)| \leq 1$ for all $s$. Otherwise, an automaton is nondeterministic. If automaton is deterministic, there is at most one run $s_0, s_1 \dots$ on $w$ for each infinite word $w$. We write $w_{\mathcal{A}}$ to denote such run.

Now we introduce several kinds of automata that differ in the accepting condition.

### Büchi automata

Büchi automaton $\mathcal{B}$ is an automaton on infinite words for which accepting condition $F$ is a subset of $S$. A run $w$ is accepting if and only if $F \cap \mathsf{infi}(w) \neq \emptyset$.

It is known that the class of languages accepted by nondeterministic Büchi automata is not equivalent to the class of languages accepted by deterministic Büchi automata [15].

### Rabin automata

Rabin automaton is an automaton on infinite words for which an accepting condition $F$ is a set of tuples $(A,B)$ where $A,B \subseteq S$. Run $w$ is accepting if there is $(A,B) \in F$ such that we have $A \cap \mathsf{infi}(w) \neq \emptyset$ and $B \cap \mathsf{infi}(w) = \emptyset$.

In case of Rabin automata, the class of languages accepted by deterministic ones is equal to the class of languages accepted by nondeterministic ones. Moreover, these classes are equivalent to the class of languages accepted by nondeterministic Büchi automata [15].

### Streett automata

Streett automaton is an automaton for which an accepting condition $F$ is a set of tuples $(A,B)$ where $A,B \subseteq S$. Run $w$ is accepting if for each $(A,B) \in F$ we have $A \cap \mathsf{infi}(w) \neq \emptyset$ or $B \cap \mathsf{infi}(w) = \emptyset$.

In case of Streett automata, the class of languages accepted by deterministic ones is equal to the class of languages accepted by nondeterministic ones. Moreover, these classes are equivalent to the class of languages accepted by nondeterministic Büchi automata [15].

### Converting and complementation

In our proofs we often use a conversion between different types of automata. We also complement them. Next lemmas reveal the complexities of these operations, proofs can be found in [15].

**Lemma 2.2.6.** *Given a Büchi automaton $\mathcal{B}$, one can construct a Büchi automaton $\mathcal{B}'$ such that $L(\mathcal{B}) = co - L(B')$ Problem of constructing such automaton is*

***EXPTIME**-complete.*

**Lemma 2.2.7.** *Given a Büchi automaton $\mathcal{B}$, one can construct a Rabin automaton $\mathcal{R}$ such that $L(\mathcal{B}) = L(\mathcal{R})$. This construction can be done in **EXPTIME**.*

**Lemma 2.2.8.** *Given a Rabin automaton $\mathcal{R}$, one can construct a deterministic Streett automaton $\overline{\mathcal{R}}$ such that $L(\mathcal{R}) = co - L(\overline{\mathcal{R}})$. This construction can be done in polynomial time.*

*Proof.* Let $F$ be an accepting condition of $\mathcal{R}$. Automaton $\overline{\mathcal{R}}$ looks exactly as $\mathcal{R}$, except that the accepting condition contains $(B,A)$ for each $(A,B) \in F$. ☐

**Additional notation**

To simplify our notation we will use $States(\mathcal{A})$, $\delta_{\mathcal{A}}$, $Init(\mathcal{A})$ and $Final(\mathcal{A})$ to denote the sets $S$, $\delta$, $q_0$ and $F$ of automaton $\mathcal{A} = (\Sigma, S, \delta, q_0, F)$. We also suppose that there is (arbitrary but fixed) ordering on $F$ and we write $F(i)$ to denote $i$-th element of $F$. If these elements are tuples, we use 1 and 2 to denote first and second component, respectively.

We also use $\overline{\mathcal{A}}$ to denote (Streett or Rabin) automaton that is complement of a given (Rabin or Streett) automaton $\mathcal{A}$. By $\mathcal{A}_q$ we denote automaton that looks exactly as $\mathcal{A}$, except that initial state is changed to $q$. It is easy to see that if $\mathcal{A}$ accepts $w$ and $\delta_{\mathcal{A}}(Init(\mathcal{A}), w(0)) = q$, then $\mathcal{A}_q$ accepts $w^{1\triangleright}$.

## 2.3 Winning objectives

In this section we introduce some methods of describing probabilistic properties. These properties are called **objectives**. Formally, an objective for a $2\frac{1}{2}$-player game $G$ is a pair $(\nu, O)$ where $\nu : Ap \to 2^V$ is a valuation and $O$ is a winning condition expressed by any of the methods given further. Note that each valuation $\nu$ determines a valuation $\overline{\nu} : Ap \to 2^{V^+}$ defined by $\overline{\nu} = \{ws \in V^+ | s \in \nu(s)\}$.

A $(\nu, O)$-winning strategy for player $\Box$ in a vertex $v \in V$ is a strategy $\sigma$ such that for every strategy $\pi$ of player $\Diamond$ we have that $v \models^{\nu}_{G(\sigma,\pi)}$. Similarly, a $(\nu, O)$-winning strategy for player $\Diamond$ in a vertex $v \in V$ is a strategy $\pi$ such that for every strategy $\sigma$ of player $\Box$ we have that $v \not\models^{\nu}_{G(\sigma,\pi)} O$.

### 2.3.1 Büchi winning condition

Büchi winning condition is a set $B \subseteq Ap$. This winning condition is satisfied for run $w$ if and only if $\mathsf{infi}(w) \cap B \neq \emptyset$ where $\mathsf{infi}(\rho) = \{a \in Ap | \forall k \exists l > k. a \in \nu(w(k))\}$. Informally Büchi objective $O$ is satisfied if and only if some propositions of $B$ are satisfied infinitely often along the run.

In the non-probabilistic setting, we can ask whether the winning condition is satisfied for every run. This winning condition is called **sure Büchi** winning condition.

In the probabilistic setting, we can ask whether the measure of runs satisfying the winning condition $B$ is equal to ($<$, $\leq$ etc.) some rational number $p$. Winning condition where the relation is *equal to* and $p = 1$ is called **almost-sure Büchi** winning condition.

In [1], we introduced a notion of **mixed Büchi** winning condition. This winning condition is defined as a tuple $(B_S, B_{AS})$ where $B_S$ is sure Büchi winning condition and $B_{AS}$ is almost sure Büchi winning condition. Mixed Büchi winning conditionobjective is satisfied if both its components are satisfied.

For example, in the state $a$ of the Markov chain shown in Figure 2.2, Büchi winning condition $\{a,b,d\}$ is surely satisfied and Büchi winning condition $\{d\}$ is satisfied with probability equal to 0.

### 2.3.2 The logic PCTL

The logic PCTL, the probabilistic extension of CTL, was introduced by Hansson & Jonsson in [16]. Let $Ap = \{p,q,\dots\}$ be a countably infinite set of *atomic propositions*. The syntax of PCTL formulae is given by the following abstract syntax equations:

$$\Phi \quad ::= \quad p \mid \neg p \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \mathrm{X}^{\bowtie\rho}\Phi \mid \Phi_1 \mathrm{U}^{\bowtie\rho}\Phi_2$$

Here $p \in Ap$, $\rho \in [0,1]$, and $\bowtie \in \{\leq,<,\geq,>,=,\neq\}$.

Let $\mathcal{T} = (S, \rightarrow, Prob)$ be a Markov chain, and let $\nu : Ap \rightarrow 2^S$ be a **valuation**. The semantics of PCTL is defined below.

| | | |
|---|---|---|
| $s \models^\nu p$ | iff | $s \in \nu(p)$ |
| $s \models^\nu \neg p$ | iff | $s \notin \nu(p)$ |
| $s \models^\nu \Phi_1 \vee \Phi_2$ | iff | $s \models^\nu \Phi_1$ or $s \models^\nu \Phi_2$ |
| $s \models^\nu \Phi_1 \wedge \Phi_2$ | iff | $s \models^\nu \Phi_1$ and $s \models^\nu \Phi_2$ |
| $s \models^\nu \mathrm{X}^{\bowtie\rho}\Phi$ | iff | $\mathcal{P}(\{w \in Run(s) \mid w(1) \models^\nu \Phi\}) \bowtie \rho$ |
| $s \models^\nu \Phi_1 \mathrm{U}^{\bowtie\rho}\Phi_2$ | iff | $\mathcal{P}(\{w \in Run(s) \mid \exists j \geq 0 : w(j) \models^\nu \Phi_2$ and $w(i) \models^\nu \Phi_1$ for all $0 \leq i < j\}) \bowtie \rho$ |

Note that in our version of PCTL syntax, the negation can be applied only to atomic propositions. This is no restriction because the syntax is closed under dual connectives and relations: For every $\bowtie \in \{\leq,<,\geq,>,=,\neq\}$, let $\overline{\bowtie}$ be the complement of $\bowtie$ (for example, if $\bowtie$ is $\leq$, then $\overline{\bowtie}$ is $>$). The negation of $X^{\bowtie\rho}\Phi$ and $\Phi_1\mathrm{U}^{\bowtie\rho}\Phi_2$ then corresponds to $X^{\overline{\bowtie}\rho}\Phi$ and $\Phi_1\mathrm{U}^{\overline{\bowtie}\rho}\Phi_2$, respectively. The $\mathrm{F}^{\bowtie\rho}$ and $\mathrm{G}^{\bowtie\rho}$ operators are defined in the standard way: $\mathrm{F}^{\bowtie\rho}\Phi$ stands for $\mathtt{tt}\,\mathrm{U}^{\bowtie\rho}\Phi$, and $\mathrm{G}^{\bowtie\rho}\Phi$ stands for $\mathtt{tt}\,\mathrm{U}^{\widehat{\bowtie}1-\rho}\neg\Phi$, where $\widehat{\bowtie}$ is $<$, $>$, $\leq$, $\geq$, $=$, or $\neq$, depending on whether $\bowtie$ is $>$, $<$, $\geq$, $\leq$, $=$, or $\neq$, respectively.

As an example, consider the Markov chain shown in 2.2 again. In the vertex $a$ of this Markov chain, the formula $\mathrm{G}^{>0.7}\mathrm{F}^{\geq 0.9}a$ is satisfied, while the formula $\mathrm{F}^{>0.1}(b \wedge X^{>0}a) \wedge G^{=0}a$ is not.

Various natural fragments of PCTL can be obtained by restricting the PCTL syntax to certain modal connectives and/or certain operator/number combinations. For example, the **qualitative** fragment of PCTL is obtained by restricting the allowed operator/number combinations to '$\bowtie 0$' and '$\bowtie 1$'. Hence, $a\mathrm{U}^{<1}b \vee \mathrm{F}^{>0}c$ is a qualitative PCTL formula. In this paper we also consider fragments with unary reachability and safety connectives. Formally, for each tuple $Y_1, \cdots, Y_n$, where each $Y_i$ is of the form $\mathrm{X}^{\bowtie\rho}$, $\mathrm{F}^{\bowtie\rho}$, or $\mathrm{G}^{\bowtie\rho}$, we define the $\mathcal{L}(Y_1, \cdots, Y_n)$ fragment of PCTL:

$$\Phi \quad ::= \quad p \;|\; \neg p \;|\; \Phi_1 \vee \Phi_2 \;|\; \Phi_1 \wedge \Phi_2 \;|\; Y_1\Phi \;|\; \cdots \;|\; Y_n\Phi$$

For example, $\mathrm{F}^{>0}(b \vee \mathrm{G}^{\geq 0.4}(\neg c \wedge \mathrm{F}^{<0.5}d))$ is a formula of $\mathcal{L}(\mathrm{F}^{>0}, \mathrm{G}^{\geq 0.4}, \mathrm{F}^{<0.5})$. Sometimes we also use formulae of the form $p \Rightarrow \Phi$ which stand for $\neg p \vee \Phi$.

As a remark we note here that there is also a different semantics for PCTL. It is defined an used in [2] or [17], for example.

### 2.3.3 The logic PECTL$^*$

Now we introduce the logic PECTL$^*$ that differs from PCTL in the set of connectives it uses. Connectives U and X are substituted with Büchi automata. With this substitution PECTL$^*$ becomes strictly stronger than PCTL and PCTL* (probabilistic equivalent of CTL*).

The formal syntax of PECTL$^*$ is following:

$$\Phi \quad ::= \quad p \;|\; \neg\Phi \;|\; \Phi_1 \vee \Phi_2 \;|\; \Phi_1 \wedge \Phi_2 \;|\; \mathcal{B}^{\bowtie\rho}(\phi_1, \ldots, \phi_n)$$

where $\mathcal{B}$ is a Büchi automaton over the alphabet $2^{\{1\ldots n\}}$, $\bowtie \in \{\leq, <, =, >, \geq\}$ and $\rho$ is a rational number from the interval $[0,1]$. We can define qualitative fragment of PECTL$^*$by restricting $\rho$ to be 0 or 1.

Given a Markov chain $\mathcal{T} = (S, \rightarrow, Prob)$ and valuation $\nu : Ap \rightarrow 2^S$, the semantics is following:

| | | |
|---|---|---|
| $s \models^\nu p$ | iff | $s \in \nu(p)$ |
| $s \models^\nu \neg\Phi$ | iff | $s \not\models^\nu \Phi$ |
| $s \models^\nu \Phi_1 \vee \Phi_2$ | iff | $s \models^\nu \Phi_1$ or $s \models^\nu \Phi_2$ |
| $s \models^\nu \Phi_1 \wedge \Phi_2$ | iff | $s \models^\nu \Phi_1$ and $s \models^\nu \Phi_2$ |
| $s \models^\nu \mathcal{B}^{\bowtie\rho}(\phi_1, \ldots, \phi_n)$ | iff | $\mathcal{P}(\{w \in Run(s) \mid w_{\mathcal{B}} \in \mathcal{L}(\mathcal{B})\}) \bowtie \rho$ |
| | | where $w_{\mathcal{B}}(i) = \{k \in \{1\ldots n\} \mid w(i) \models^\nu \Phi_k\}$ |

Closer look at the semantics reveals that the connectives $\vee$ and $\wedge$ are in fact only a syntactic sugar, because they can be expressed using a product Büchi automaton. Moreover, we can define equivalent logic in which negations are pushed to atomic propositions. Converting any PECTL$^*$ formula to this forms can be done in polynomial time.

An example of the formula PECTL$^*$ that has not an equivalent PCTL* (and thus PCTL) formula is the formula $\mathcal{B}_{alter}^{>0}(a,b)$ with the automaton $\mathcal{B}_{alter}$ from the Figure 2.3. This formula says that the probability of runs in which $a$ holds in odd states and $b$ holds in even states is greater than zero.

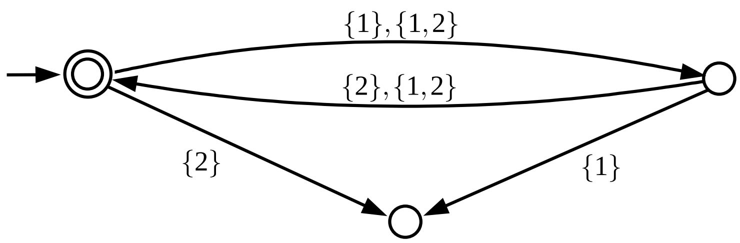



Figure 2.3: Büchi automaton $\mathcal{B}_{alter}$

**Qualitative Rabin-Streett-PECTL$^*$**

Later in this thesis, we prove that given a qualitative PECTL$^*$ objective and $1\frac{1}{2}$-player game, it is decidable whether there is a finite-memory strategy that ensures satisfying the objective.

In our proof we use a variant of qualitative PECTL$^*$ that uses deterministic Rabin and Streett automata instead of Büchi automata. We call this logic qualitative Rabin-Streett-PECTL$^*$. The syntax is following:

$$\Phi \quad ::= \quad p \mid \neg p \mid \quad \mid \mathcal{A}^{\bowtie\rho}(\Phi_1,\ldots,\Phi_n)$$

where $\mathcal{A}(\Phi_1,\ldots,\Phi_n)$ is a deterministic Rabin or Street automaton over the alphabet $2^{\{1,\ldots,n\}}$ and $\bowtie \rho \in \{=1, >0\}$. Semantics of the formula is straightforward. In what follows, we will use $w_{\mathcal{A}}$ to denote run of $\mathcal{A}$ on $w_{\{1\ldots n\}}$, where $w$ is a run of Markov chain $\mathcal{T}$. We will also write $\overline{>0}$ (resp. $\overline{=1}$) to denote $=1$ (resp. $>0$).

**Lemma 2.3.1.** *Every formula $\Phi$ of qualitative PECTL$^*$ can be transformed to qualitative Rabin-Streett-PECTL$^*$ formula in **EXPTIME**.*

*Proof.* First of all, we transform $\Phi$ so that it does not contain connectives $\vee$ and $\wedge$ and that negations are pushed to atomic propositions. Next, we replace each Büchi automaton with an equivalent Rabin automaton. This transformation can be done in **EXPTIME**. It remains to remove $< 1$ and $= 0$ probabilities. This can be done by substituting Rabin automaton $R$ by Street automaton $\overline{R}$ and by altering the guarding probabilities. ☐

Let $\Phi$ be a Rabin-Streett-PECTL$^*$ formula, $\mathcal{T} = (S, \rightarrow, Prob)$ a Markov chain and let $M$ be the smallest set that is closed under complement and contains all automata $\mathcal{A}_q$ where $\mathcal{A}$ is (Rabin or Streett) automaton that occurs in $\Phi$ and $q \in States(\mathcal{A})$. We define the function $\gamma : M \times S \rightarrow 2^M \times \{= 1, > 0\}$ by

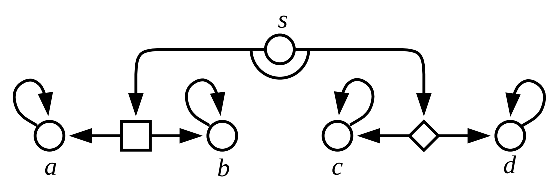
$$\begin{aligned}\gamma(\mathcal{A}, u) = \quad & \{(\mathcal{A}_i, \bowtie \rho) \mid 1 \leq i \leq n, u \models \Phi_i, \Phi_i \equiv \mathcal{A}_i^{\bowtie \rho}\} \\ & \cup \{\overline{(\mathcal{A}_i}, \overline{\bowtie \rho}) \mid 1 \leq i \leq n, u \not\models \Phi_i, \Phi_i \equiv \mathcal{A}_i^{\bowtie \rho}\}\end{aligned}$$

We also define the function $\beta$ as the closure of $\gamma$. $\beta(\mathcal{A}, u)$ is the smallest set such that $\gamma(\mathcal{A}, u) \subseteq \beta(\mathcal{A}, q, u)$ and for each $(\mathcal{A}', \bowtie \rho) \in \beta(\mathcal{A}, u)$ we have that $\gamma(\mathcal{A}', u) \subseteq \beta(\mathcal{A}, u)$.

## Chapter 3

# Non-determinacy and undecidability

We start by showing that the stochastic games with PCTL objectives are **not determined** even if these objectives are taken from the $\mathcal{L}(F^{=1}, F^{>0})$ fragment of PCTL. Consider the following game:




Let $\nu$ be a valuation which defines the validity of the propositions $a, b, c, d$ as indicated in the above figure, and let $\varphi \equiv F^{=1}(a \vee c) \vee F^{=1}(b \vee d) \vee \left(F^{>0}c \wedge F^{>0}d\right)$. Now it is easy to check that none of the two players has a $(\nu, \varphi)$-winning strategy in the vertex $s$, regardless whether we consider MD, MR, HD, or HR strategies.

Next important facts about stochastic games with PCTL objectives are stated in next two theorems.

**Theorem 3.0.2. [1]** *The problem whether there is a winning strategy for player $\Box$ in $1\frac{1}{2}$-player game with a PCTL objectives is undecidable.*

**Theorem 3.0.3. [1]** *The problem whether there is a winning strategy with finite memory for player $\Box$ in $1\frac{1}{2}$-player game with PCTL objectives is undecidable.*

Note that all results from this chapter can be easily extended to PECTL$^*$ objectives.

CHAPTER 4

# MEMORYLESS STRATEGIES AND PCTL

## 4.1 Deterministic strategies

In this chapter, we summarize some results about memoryless strategies. These results were published in [1], but the proofs given here are more detailed.

**Theorem 4.1.1.** *The problem of existence of a winning MR strategy in $2\frac{1}{2}$-player game with $\mathcal{L}(\mathrm{F}^{>0},\mathrm{F}^{=1})$ objective is* $\boldsymbol{NP^{coNP}}$*-hard.*

We give the proof by reduction from the following $\mathbf{NP^{coNP}}$-complete problem [4]: Determine whether a boolean formula $\Psi$ of the form

$$\exists x_1,\ldots,x_k \forall x_{k+1},\ldots,x_l \phi$$

(where $\phi$ is propositional formula in which negations are pushed to atomic propositions) is true.

Given a formula $\Psi$ we construct a game $G(\Psi) = (V, E, (V_\Box, V_\Diamond, V_\bigcirc), Prob)$ and formula $\varphi \in \mathcal{L}(\mathrm{F}^{>0},\mathrm{F}^{=1})$. The set $V$ consists of the vertices $a$, $b(i)$, $x(i)$ and $nx(i)$ where $i \in \{1,\ldots,l\}$.

Vertices $b(1),\ldots,b(k)$ form the set $V_\Box$ and vertices $b(k+1),\ldots,b(l)$ form the set $V_\Diamond$. The structure of $G(\Psi)$ is shown in Figure 4.1. Valuation $\nu$ assigns unique proposition to each vertex. For simplicity we assume that the name of a proposition is same as the name of corresponding vertex.

Now we construct formula $\varphi$ such that the player $\Box$ has a winning MR strategy in a vertex $a \in V$ if and only if $\Psi$ is true. Formula $\varphi$ looks as follows:

$$\varphi \;\equiv\; (\mathit{Satisfy} \wedge \mathit{NoRnd}_\Box) \vee \mathit{NoRnd}_\Diamond$$

The *Satisfy* part ensures that player $\Box$ can win the game if and only if formula $\Psi$ is true, the *NoRnd* parts prevent both players from selecting more than one transition from each vertex.

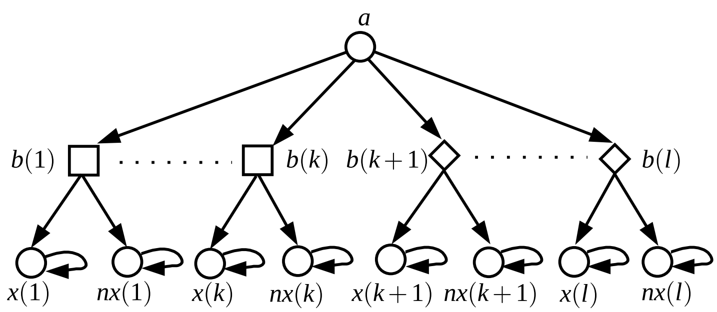



Figure 4.1: Structure of the game $G(\Psi)$

*NoRnd*$_\Box$ has the form

$$\bigwedge_{i=0}^{k} \mathrm{F}^{>0}(b(i) \wedge \mathrm{F}^{=1}x(i)) \vee \mathrm{F}^{>0}(b \wedge \mathrm{F}^{=1}nx(i))$$

. It says that player $\Box$ must choose deterministic transitions to satisfy the formula.

*NoRnd*$_\Diamond$ has the form

$$\bigvee_{i=k+1}^{l} \mathrm{F}^{>0}x(i) \wedge \mathrm{F}^{>0}nx(i)$$

Because player $\Diamond$ must avoid satisfying the formula $\varphi$, he has to prevent himself from satisfying *NoRnd*$_\Diamond$ subformula. This can be done only by choosing deterministic transitions.

*Satisfy* has exactly the same structure as the formula $\phi$ (which is propositional part of $\Psi$), except that each occurrence of the positive literal $x_i$ is replaced by $\mathrm{F}^{>0}x(i)$ and each occurrence of the negative literal $\neg x_i$ is replaced by $\mathrm{F}^{>0}nx(i)$. Observe that player $\Box$ in fact chooses values for variables guarded by existential quantifier and player $\Diamond$ chooses values for variables guarded by general quantifier. Thus, the formula $\varphi$ is satisfied in $a$ if and only if $\Phi$ is satisfied.

**Lemma 4.1.2.** *Formula $\Psi$ is true if and only if player $\Box$ has $(\nu,\Phi)$-winning MR strategy in state $a$ of $G(\Psi)$.*

*Proof.* $\Rightarrow$: Let $\overline{x_1},\ldots,\overline{x_k} \in \{0,1\}$ be the values of $x_1,\ldots,x_k$ such that for every values $\overline{x_{k+1}},\ldots,\overline{x_l} \in \{0,1\}$ of $x_{k+1},\ldots,x_l$ formula $\Psi$ is satisfied. Now we construct a strategy for player $\Box$ such that $\varphi$ is satisfied in $a$. For each $i \in \{1,\ldots,k\}$ player $\Box$ chooses transition from $b(i)$ to $x(i)$ if $\overline{x_i} = 1$ and transition from $b(i)$ to $nx(i)$ otherwise. Observe that $x_i = 1$ if and only if the formula $\mathrm{F}^{>0}x(i)$ is satisfied, and $x_i = 0$

if and only if formula $\mathrm{F}^{>0} nx(i)$ is satisfied. Situation for player $\Diamond$ is similar. Following the way of constructing formula $Satisfy$, it is easy to see that player $\Diamond$ can't avoid satisfying of the formula *Satisfy*.

$\Leftarrow$: Assume there is a strategy for player $\Box$ such that for every strategy of player $\Diamond$ formula $\varphi$ is satisfied. Obviously, this strategy is deterministic. If player $\Box$ chooses transition from $b(i)$ to $x(i)$, we set $x_i = 1$, otherwise $x_i = 0$. All possible strategies of player $\Diamond$ correspond to all valuations of $x_{k+1}, \ldots, x_l$ in $\Psi$, and thus formula $\Psi$ is true. ☐

Theorem 4.1.1 follows directly from previous lemma.

**Theorem 4.1.3.** *The existence of a winning MD strategy in $2\frac{1}{2}$ games with $\mathcal{L}(\mathrm{F}^{>0})$ objectives is* $\mathbf{NP^{coNP}}$*-hard.*

*Proof.* In case of MD strategy, we can remove *NoRnd* parts from previous proof. Thus, there is no need of $\mathrm{F}^{=1}$ operator. ☐

**Lemma 4.1.4.** *Problem whether there is a winning MD strategy in $2\frac{1}{2}$ games with PCTL objectives is* $\mathbf{NP^{coNP}}$*-complete.*

*Proof.* Hardness is shown in Theorem 4.1.3, here we show that the problem is in $\mathbf{NP^{coNP}}$. Problems from this class can be solved in polynomial time by a nondeterministic Turing machine with **coNP** oracle. We show that problem of existence of MD strategy can be solved by this automaton.

Let $G(\Psi) = (V, E, (V_\Box, V_\Diamond, V_\bigcirc), Prob)$ and $\varphi \in PCTL$. At first step, the Turing machine nondeterministically guesses strategy of player $\Box$ and then asks oracle whether there is a strategy for player $\Diamond$ such that the formula $\phi$ is not satisfied. This question is in **coNP**, because there is polynomially verifiable counter-example. If answer from oracle is "yes", the Turing machine answers "no" and vice versa. ☐

## 4.2 Randomized strategies

As mentioned in introduction, we did not manage to prove exact complexity bound for synthesizing MR strategies in $2\frac{1}{2}$-player games. Best known approximations are **NP**-hardness for $1\frac{1}{2}$-player games [8] and $\mathbf{NP^{coNP}}$-hardness for $2\frac{1}{2}$-player games (Theorem 4.1.1). In [14], upper bounds are set to **PSPACE** for $1\frac{1}{2}$-player games and to **EXPTIME** for $2\frac{1}{2}$-player games.

Here we show that problem is likely to be harder than known lower bounds. We show this by reduction from **square root sum problem**, whose lower bound is an open problem in computational geometry. We start by showing that there are games in which randomization can come up with irrational numbers.

**Lemma 4.2.1.** *There exists a $1\frac{1}{2}$-player game such that the only winning MR strategy for player $\Box$ chooses irrational distribution.*

*Proof.* We give the proof by example.

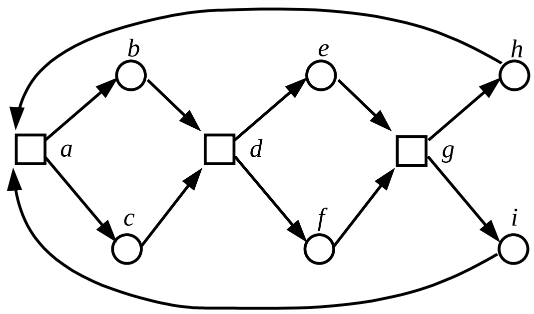



Figure 4.2: The structure of $H$

We construct a game $H = (V, E, (V_{\Box}, V_{\bigcirc}), Prob)$ and formula $\varphi$. All components of $H$ are given by Figure 4.2. Structure of the formula $\phi$ is following:

$$\varphi = FirstSecond \wedge SecondThird \wedge ThirdFirst$$

where

$$\begin{aligned} FirstSecond &= (a \vee b \vee d)\mathrm{U}^{=1/2} e \\ SecondThird &= \mathrm{F}^{>0}(d \wedge ((d \vee e \vee g)\mathrm{U}^{=1/2} h)) \\ ThirdFirst &= \mathrm{F}^{>0}(g \wedge ((g \vee h \vee a)\mathrm{U}^{=1/2} b)) \end{aligned}$$

It is obvious that player $\Box$ has a winning strategy – it is sufficient to set probability of $a \to b$, $d \to e$ and $g \to h$ equal to $\sqrt{1/2}$. But this is the only existing solution: suppose there is another one and $a \to b$ has probability greater than $\sqrt{1/2}$. To satisfy $FirstSecond$, $d \to e$ must have probability lower than $\sqrt{1/2}$. To satisfy $SecondThird$, $g \to h$ must have probability greater than $\sqrt{1/2}$. But to satisfy $ThirdFirst$, $a \to b$ must have probability lower than $\sqrt{1/2}$. And this is not possible, as probability of $a \to b$ is already set to the value greater than $\sqrt{1/2}$.

Similar reasoning can be used for every modification of the strategy. $\Box$

**Definition 4.2.2. (Square root sum problem)** *Given the vector $(x_1, \ldots, x_n, z)$ of natural numbers, decide whether $\sum_{i=1}^{n} \sqrt{x_i} \leq z$.*

In the following text, we propose reduction from this problem to the problem of existence of winning MR strategy in $1\frac{1}{2}$ games.

Let $Y = (x_1, \ldots, x_n, z)$ be an instance of the square root sum problem. Dividing both sides of $\sum_{i=1}^{n} \sqrt{x_i} \leq z$ by a natural number $q = \sum_{i=1}^{n} x_i$, we obtain $\sum_{i=1}^{n} \sqrt{x_i}/q \leq$

$z/q$. Obviously, the second inequality holds if and only if the first inequality holds. Moreover we have that $\sum_{i=1}^{n}\sqrt{x_i}/q$ is lower than 1.

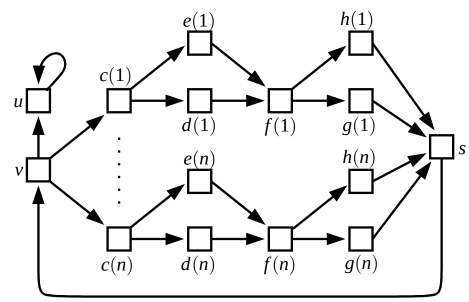



Figure 4.3: The structure of $I(Y)$

Now we can use the technique presented in previous lemma to create the game $I(Y) = (V, E, (V_\Box, V_\bigcirc), Prob)$. Schema of $I(Y)$ is shown in Figure 4.3, reader can easily imagine the game for arbitrary $n$.

Now we construct formula $\varphi$ such that player $\Box$ has MR winning strategy in $v$ if and only if $\sum_{i=1}^{n}\sqrt{x_i} \leq z$.

$$\varphi = Result \wedge \bigwedge_{j=1}^{n} ProbX_j$$

Intended behavior is as follows: $ProbX_j$ forces player $\Box$ to set probability of $v \to c(j)$ to $\sqrt{x_j}/q$. Transition $h \to i$ will then have probability $p = 1 - \sum_{i=1}^{n}\sqrt{x_i}/q$. Using formula $Result$, we ask whether $p < 1 - z/q$. Formally,

$$ProbX_j = A \wedge B \wedge C$$

where

$$\begin{aligned} A &= (v \vee c(j))\mathrm{U}^{=r} e(j) \\ B &= F^{>0}(c(j) \wedge ((c(j) \vee e(j) \vee f(j))\mathrm{U}^{=r} h(j))) \\ C &= F^{>0}(f(j) \wedge ((f(j) \vee h(j) \vee s \vee v)\mathrm{U}^{=r} c(j))) \end{aligned}$$

for $r = x_j/q^2$.

$$Result = \mathrm{X}^{\geq 1 - z/q} u$$

**Lemma 4.2.3.** *Player $\Box$ has winning MR strategy for $\varphi$ in $I(Y)$ if and only if $Y$ is true.*

*Proof.* $\Leftarrow$: Let $x_1,\ldots,x_n$ and $z$ be the integers such that $\sum_{i=1}^{n}\sqrt{x_i} \leq z$.

Winning strategy must set the probability of $v \to c(i)$ to $\sqrt{x_i}/q$. This is possible because both $\sqrt{x_i}/q$ and $\sum_{i=1}^{n}\sqrt{x_i}/q$ are less than 1. Probability of $v \to u$ is then set to $1-\sum_{i=1}^{n}\sqrt{x_i}/q$, because sum of probabilities of outgoing transition from each vertex must be equal to 1. It is easy to see that formulas $ProbX_j$ are surely satisfied.

Now we analyze the the *Result* formula. It asks whether the probability of $v \to u$ is high enough. Because $Y$ is true, we know that $\sum_{i=1}^{n}\sqrt{x_i}/q \leq z/q$ and also $1-\sum_{i=1}^{n}\sqrt{x_i}/q \geq 1-z/q$. Left side of this inequality is exactly the probability of reaching $u$ as the next state. That's why the formula *Result* is satisfied.

$\Rightarrow$: Let there be the winning MR strategy for player $\Box$. Using same argument as in Lemma 4.2.1, we know that transitions $v \to c(i)$ have probabilities equal to $\sqrt{x_i}/q$. Because sum of outgoing transitions from one vertex is 1, probability of $v \to u$ is $1-\sum_{i=1}^{n}\sqrt{x_i}/q$. Because formula φ is satisfied, formula *Result* is satisfied too. This means that $1-\sum_{i=1}^{n}\sqrt{x_i}/q \geq 1-z/q$ from which we obtain $\sum_{i=1}^{n}\sqrt{x_i}/q \leq z/q$. □

**Corollary 4.2.4.** *There is a strong evidence for* ***problem of determining whether there is a winning MR strategy for player*** $\Box$ ***in a*** $1\frac{1}{2}$ ***games*** *to be outside both of* ***NP*** *and* ***coNP*** *classes.*

CHAPTER 5

# FINITE MEMORY AND QUALITATIVE PECTL$^*$

In this chapter we show that the problem of deciding whether there is a $\Psi$-winning finite-memory strategy for a $1\frac{1}{2}$-player game and PECTL$^*$ formula $\Psi$ is decidable in 2-**EXPTIME**. Next, we show that the problem is **EXPTIME**-hard. The proof of hardness given here is the proof of **EXPTIME**-hardness for PCTL taken from [1]. Proof for PECTL$^*$ follows directly.

## 5.1 Upper bound

In this section, we prove that given a $1\frac{1}{2}$-player game $G = (V, E, (V_{\Box}, \emptyset, V_{\bigcirc}), Prob)$, $s_{in} \in V$ and qualitative PECTL$^*$ objective $(\nu, \Psi)$, the problem whether there is a deterministic finite-memory strategy $\sigma$ such that $s_{in} \models^{\nu}_{G(\sigma)} \Psi$ is decidable.

Let $\Psi$ be a formula of qualitative PECTL$^*$. We transform $\Psi$ to an equivalent Rabin-Streett-PECTL$^*$ formula $\Phi$. According to the Lemma 2.3.1, this can be done in **EXPTIME**.

Our proof will proceed as follows: First of all, we prove some lemmas that follow from the fact that $G(\sigma)$ is necessarily a finite Markov chain. Then we use these lemmas in a construction of $1\frac{1}{2}$-player game $G'$ and mixed Büchi objective $(\nu', (P, Q))$. We show that there is a $(\nu', (P, Q))$-winning strategy for $G'$ if and only if there is a strategy $\sigma$ such that $s \models^{\nu}_{G(\sigma)} \Phi$. Proposed construction will be done in **EXPTIME** and thus we yield 2-**EXPTIME** for the whole algorithm.

### 5.1.1 Accepting in finite Markov chains

For this subsection, let us fix a finite Markov chain $\mathcal{T} = (T, \rightarrow, Prob)$, some state $t_0 \in T$, set of atomic propositions $Ap$, valuation $\nu : Ap \rightarrow 2^T$, deterministic Rabin automaton $\mathcal{R} = (2^{\{1...|Ap|\}}, R, \delta_{\mathcal{R}}, r_0, F_{\mathcal{R}})$ and deterministic Streett automaton $\mathcal{S} = (2^{\{1...|Ap|\}}, S, \delta_{\mathcal{S}}, s_0, F_{\mathcal{S}})$.

Next two lemmas reveal important fact that helps us to find runs accepted by Rabin or Streett automaton. They state that after reaching some b.s.c.c., all runs are (in some sense) similar.

**Lemma 5.1.1.** *Following two statements are equivalent:*

1. $t_0 \models^{\nu}_{\mathcal{T}} \mathcal{R}^{=1}$
2. *For almost every run $w$ of $\mathcal{T}$ there is $k \in \mathbb{N}$ and $(A,B) \in F_{\mathcal{R}}$ such that*
   - (a) *for every run $u$ with prefix $w(0)\dots w(k)$ no state of $B$ occurs in $u^{k\triangleright}_{\mathcal{R}}$ and*
   - (b) *for almost every run $u$ with prefix $w(0)\dots w(k)$ some state of $A$ occurs infinitely often in $u$.*

*Proof.* $1 \Rightarrow 2$: We construct a Markov chain $\mathcal{T}' = (T \times R, \mapsto, Prob')$ in which $(t,r) \overset{x}{\mapsto} (t',r')$ if and only if $t \overset{x}{\to} t'$ and $\mathcal{R}$ goes from $r$ to $r'$ after reading $t$. Valuation $\nu' : Ap \to 2^{(T\times R)}$ is given by $\nu'(a) = \{(t,r) \mid t \in \nu(a)\}$. Obviously, there is a quotient $\Theta$ of $\mathcal{T}'$ onto $\mathcal{T}$ given by $\Theta((t,r)) = s$.

Consider the part of $T'$ that is reachable from $t'_0 = (t_0,r_0)$. Obviously, $t'_0 \models^{\nu'} \mathcal{R}^{=1}$ and $w \in T'$ is accepted by $\mathcal{R}$ if and only if $\Theta(w)$ is accepted by $\mathcal{R}$.

Now to each run $w$ of $\mathcal{T}'$ that reaches some b.s.c.c. we assign $k$ such that $w(k)$ is the first state on $w$ that is in the b.s.c.c. There is a run $w'$ such that $w(i) = w'(i)$ for $0 \le i \le k$ which goes through all states of b.s.c.c. infinitely often. This run must be accepted (because probability of such runs is greater than zero) and thus there is $(A,B) \in F$ such that none of states from $B$ occurs as second component of any state of the b.s.c.c. Also there must be a state $a \in A$ such that $a$ is a second component of some state of the b.s.c.c.

It is easy to see that $(A,B)$ suffices as accepting condition for all runs that start with $w(0)\dots w(k)$ and go through all states of b.s.c.c. infinitely often. Also, no run initiated in $w(0)\dots w(k)$ can force $\mathcal{R}$ to go through $B$ after $k$ steps.

Finally we just use $\Theta$ to transform runs of $\mathcal{T}'$ to runs of $\mathcal{T}$.

$2 \Rightarrow 1$ is obvious. □

**Lemma 5.1.2.** *Following two statements are equivalent:*

1. $t_0 \models^{\nu}_{\mathcal{T}} \mathcal{S}^{=1}$.
2. *For almost every run $w$ of $\mathcal{T}$ and for all $(A,B) \in F_{\mathcal{S}}$ there is $k \in \mathbb{N}$ such that one of the following holds:*
   - (a) *For every run $u$ with prefix $w(0)\dots w(k)$ no state of $B$ occurs in $u^{k\triangleright}_{\mathcal{S}}$*
   - (b) *For almost every run $u$ with prefix $w(0)\dots w(k)$ some state of $A$ occurs infinitely often in $u_{\mathcal{S}}$*

*Proof.* Proof is similar to the proof of Lemma 5.1.1 □

Following lemmas give a clue on how to transform "$> 0$" probability to "$= 1$" probability using a properties of finite Markov chains.

**Lemma 5.1.3.** *Following two statements are equivalent:*

1. $t_0 \models^{\nu}_{\mathcal{T}} \mathcal{R}^{>0}$

2. *There is a finite path* $u = t_0, \ldots, t_k$ *such that* $t_k \models^{\nu}_{\mathcal{T}} \mathcal{R}'^{=1}$ *where* $\mathcal{R}'$ *is* $\mathcal{R}$ *with initial state* $u_{\mathcal{R}}(k)$.

*Proof.* $1 \Rightarrow 2$: As in Lemma 5.1.1 we construct a Markov chain $\mathcal{T}' = (T \times R, \mapsto, Prob')$ in which $(t,r) \overset{x}{\mapsto} (t',r')$ if and only if $t \overset{x}{\to} t'$ and $\mathcal{R}$ goes from $r$ to $r'$ after reading $t$. Valuation $\nu' : Ap \to 2^{(T\times R)}$ is given by $\nu'(a) = \{(t,r) | t \in \nu(a)\}$. Again, there is a quotient $\Theta$ of $\mathcal{T}'$ onto $\mathcal{T}$ given by $\Theta((t,r) = t$.

Consider the part of $T'$ that is reachable from $t'_0 = (t_0, r_0)$. Obviously, $t'_0 \models^{\nu'} \mathcal{R}^{>0}$ and $w \in T'$ is accepted by $\mathcal{R}$ if and only if $\Theta(w)$ is accepted by $\mathcal{R}$.

Surely there is a run $w = t'_0, \ldots, t'_k, t'_{k+1} \ldots$ and b.s.c.c. $\mathcal{C}$ of $\mathcal{T}'$ such that $w$ enters $\mathcal{C}$ in $t'_k$, visits every state of $\mathcal{C}$ infinitely often and $w_{\mathcal{R}}$ is accepted by $\mathcal{R}$. Obviously, $w' = t'_k \ldots$ is accepted by $\mathcal{R}'$ with initial state $w'_{\mathcal{R}}(k)$. Let $(A,B)$ be the accepting condition that accepts $w'$. This condition must also accept all runs $w''$ initiated in $t'_k$ that visit each state of $C$ infinitely often. And these runs have probability equal to 1.

$2 \Rightarrow 1$ is obvious. □

**Lemma 5.1.4.** *Following two statements are equivalent:*

1. $t_0 \models^{\nu}_{\mathcal{T}} \mathcal{S}^{>0}$

2. *There is a finite path* $u = t_0, \ldots, t_k$ *such that* $t_k \models^{\nu}_{\mathcal{T}} \mathcal{S}'^{=1}$ *where* $\mathcal{S}'$ *is* $\mathcal{S}$ *with initial state* $u_{\mathcal{S}}(k)$.

*Proof.* Proof is similar to the proof of Lemma 5.1.3. □

Lemma 5.1.3 and Lemma 5.1.4 claim the existence of a path to some b.s.c.c. Next definition defines a function that returns a length of the shortest of all such paths.

**Definition 5.1.5.** *Let* $\mathcal{A}$ *be (Rabin or Streett) automaton. Function* $\alpha_{\mathcal{A}} : T \to \mathbb{N}_0$ *is defined by* $\alpha_{\mathcal{A}}(t) = min(\{i \mid \exists t'.t \to^i t', t' \models \mathcal{A}^{=1}\}$.

### 5.1.2 The reduction

Suppose that we are given a $1\frac{1}{2}$-player game $G = (V, E, (V_\Box, \emptyset, V_\bigcirc), Prob)$ and a Rabin-Streett-PECTL* winning condition $\Phi$. We construct a $1\frac{1}{2}$-player game $G' = (V', E', (V'_\Box, \emptyset, V'_\bigcirc), Prob')$ and a mixed Büchi objective $(P, O)$ such that there is $\Phi$-winning strategy for $G$ if and only there is $(P, O)$-winning strategy for $G'$.

First of all we try to intuitively explain construction of $G'$, then we give formal definition and proof.

#### Idea of the reduction

In $G'$, the aim of the player $\Box$ is to propose a strategy for $G$ and to show that this strategy is winning. He shows this by simulating runs of Rabin and Streett automata that represents the subformulae of $\Phi$. For each vertex of $G(\sigma)$, player $\Box$ must guess proper transition of each automaton. He proves the correctness of his guess by showing that chosen subformulae are satisfied in the current vertex. For this, he will keep a list of all 'to-be-satisfied' subformulae (represented by automata) along each run and he will remember the states in which automata occur after reading prefix of current run. To show that runs of all automata are accepting, the player $\Box$ will be required to satisfy $(P, O)$ winning condition.

More formally, $G'$ contains 5 types of vertices: *a*-, *b*-, *c*-, *d*- and *e*-vertices. Each vertex (except *e*-vertices) is a 5-tuple in which the first component represents a vertex of $G$, second and third component hold a list of automata that are satisfied (we say that automaton is satisfied if the formula it represents is satisfied) with probability equal to one and fourth and fifth component hold a list of automata that are satisfied with probability greater than zero in the vertex of $G$. Structure of *e*-vertices is slightly more complicated, automata in fourth and fifth components are divided among all possible successors of the current vertex.

The reason for having two components of the same purpose is following: We have an event for *each element* of the list and we want to check that each event occurs *infinitely often* (in our case, the elements are automata and event is 'enters some state'). Moreover, elements can be added to the list at any time. Unfortunately, Büchi winning condition allows us to 'check' just one infinite occurrence of event. That's why we use the following trick: We keep two different lists of element and we add new elements only to the second one. We check elements in the first list and wait until all elements fire an event. At that moment, we copy the second list to the first and wait again. Büchi winning condition checks whether we move the second list to the first infinitely often.

Play starts in a *a*-vertex that represents initial vertex of $G$ and has the only automaton representing $\Phi$ in the 'list'.

On a step from *a* to *b*, player $\Box$ can use Lemma 5.1.3 and Lemma 5.1.4 and move some automata from fourth component to third component. For future use, he can also add some automata to the lists.

On a step from $b$ to $c$, player simulates a step of all automata in the list. For each automaton, there is only one possible transition, but to find it, we need to know all subformulae that are satisfied. Thus, if player $\Box$ decides that a subformula is satisfied, he must prove it, and if he decides that it is not, he must prove that negation is satisfied. He can prove both by adding appropriate automata to the list and by simulating their steps. In this step, player $\Box$ can also use Lemma 5.1.1 and Lemma 5.1.2 and assign some accepting tuple(s) to some automaton. From this moment, this automaton cannot enter the state that should be visited finitely often.

Step from $c$ to $d$ is controlled by player $\bigcirc$. This fact is not important as there is only one possible successor. On this step, fifth component is moved to fourth if the fourth is empty (see the step $a$ to $b$). Also, third component is moved to second if all automata in second component have assigned an accepting condition and have recently visited some state that should be visited infinitely often.

If the vertex of $G$ specified by first component of current vertex is from $V_{\bigcirc}$, player $\Box$ can divide lists in fourth and fifth component on the step from $d$ to $e$. This division is then used to determine to which successor the automata should be assigned (note that we do not assign all automata to all successors, because Lemma 5.1.1 and Lemma 5.1.2 state that we need to find one path).

On the step from $e$ to $a$, step of $G$ is simulated.

Mixed Büchi condition states that we surely move fifth component to fourth infinitely often and that we almost surely move third component to second infinitely often. This corresponds to the fact that *every* automaton must be eventually removed from fourth component, but there can be non-accepting runs for automata in second component.

**Formal definition**

To $V'$ we put following vertices:

- $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_a$
- $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_b$
- $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_c$
- $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_d$
- $(s, X^{=1}, \overline{X}^{=1}, (X^{>0}_{t_1}, \ldots, X^{>0}_{t_n}), (\overline{X}^{>0}_{t_1}, \ldots, \overline{X}^{>0}_{t_n}))_e$ where $t_1, \ldots, t_n$ are all successors of $s$

in which elements of $X^{=1}$ are some tuples $(\mathcal{R}, q, r, \mathsf{flash})$ and $(\mathcal{S}, q, \Theta, \Gamma)$ where

- $\mathcal{R}$ (resp. $\mathcal{S}$) is a Rabin (resp. Streett) automaton that occurs in $\Psi$ or $\overline{\mathcal{R}}$ (resp. $\overline{\mathcal{S}}$) is a Streett (resp. Rabin) automaton that occurs in $\Psi$ or
- $q$ is a state of the automaton
- $r \in Final(\mathcal{R}) \cup \emptyset$
- $\mathsf{flash} \in \{\text{FALSE}, \text{TRUE}\}$
- $\Theta$ is $\emptyset$ or a tuple $(x_1, \ldots, x_m)$ where $x_i \in \{1,2\}$ and $m = |Final(\mathcal{S})|$
- $\Gamma$ is a tuple $(x_1, \ldots, x_m)$ where $x_i \in \{\text{TRUE}, \text{FALSE}\}$ and $m = |Final(\mathcal{S})|$. By $\Gamma_0$ we denote the tuple $(\text{FALSE}, \ldots, \text{FALSE})$.

Elements of $\overline{X}^{=1}$, $X^{>0}$ and $\overline{X}^{>0}$ are tuples $(\mathcal{A}, q)$ where $\mathcal{A}$ is Rabin or Streett automaton and $q \in States(\mathcal{A})$.

To $E'$ we put the following transitions:

- $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_a \to (s, Y^{=1}, \overline{Y}^{=1}, Y^{>0}, \overline{Y}^{>0})_b$ if and only if all the following conditions hold
  - $X^{=1} = Y^{=1}$
  - $X^{>0} \subseteq Y^{>0} \cup \overline{Y}^{=1}$
  - $\overline{X}^{=1} \subseteq \overline{Y}^{=1}$
  - $\overline{X}^{>0} \subseteq \overline{Y}^{>0}$
- $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_b \to (s, Y^{=1}, \overline{Y}^{=1}, Y^{>0}, \overline{Y}^{>0})_c$ if and only if
  - for each $(\mathcal{A}, q)$ in $\overline{X}^{=1}$ (resp. $X^{>0}$ or $\overline{X}^{>0}$) there is $(\mathcal{A}, q')$ in $\overline{Y}^{=1}$ (resp. $Y^{>0}$ or $\overline{Y}^{>0}$) such that $q' = \delta(q, g)$ for some $g \in 2^{\{1 \ldots n\}}$ and
    * if $i \in g$ and $\Phi_i \in Ap$, then $s \in \nu(\Phi_i)$
    * if $i \notin g$ and $\Phi_i \in Ap$, then $s \notin \nu(\Phi_i)$
    * if $i \in g$ and $\Phi_i$ is of the form $\mathcal{A}'^{\bowtie\rho}$, then $(\mathcal{A}', Init(\mathcal{A}')) \in \overline{X}^{\bowtie\rho}$
    * if $i \notin g$ and $\Phi_i$ is of the form $\mathcal{A}'^{\bowtie\rho}$, then $(\overline{\mathcal{A}'}, Init(\overline{\mathcal{A}'})) \in \overline{X^{\bowtie\rho}}$
  - for each $(\mathcal{R}, q, r, \mathsf{flash})$ in $X^{=1}$ there is $(\mathcal{R}, q', r', \mathsf{flash}')$ in $Y^{=1}$ such that
    * $\mathsf{flash}' = \text{TRUE}$ if and only if either $\mathsf{flash} = \text{TRUE}$ or $r = (A, B)$ and $q \in A$
    * if $r = (A, B)$, then $r' = (A, B)$
    * if $r = (A, B)$, then $q' \notin B$
    * $q' = \delta(q, g)$ for some $g \in 2^{\{1 \ldots n\}}$ and
      · if $i \in g$ and $\Phi_i \in Ap$, then $s \in \nu(\Phi_i)$

  - if $i \notin g$ and $\Phi_i \in Ap$, then $s \notin \nu(\Phi_i)$
  - if $i \in g$ and $\Phi_i$ is of the form $\mathcal{A}^{\bowtie\rho}$, then $(\mathcal{A}, Init(\mathcal{A})) \in \overline{X}^{\bowtie\rho}$
  - if $i \notin g$ and $\Phi_i$ is of the form $\mathcal{A}^{\bowtie\rho}$, then $(\overline{\mathcal{A}}, Init(\overline{\mathcal{A}})) \in \overline{X^{\bowtie\rho}}$

– for each $(\mathcal{S}, q, \Theta, \Gamma)$ in $X^{=1}$ there is $(\mathcal{S}, q', \Theta', \Gamma')$ in $Y^{=1}$ such that

  * for all $i$ we have $\Gamma'(i) = \text{TRUE}$ if and only if $\Gamma(i) = \text{TRUE}$ or $\Theta(i) = 2$ and $q \in A$
  * if $\Theta = (x_1, \ldots, x_n)$, then $\Theta = (x_1, \ldots, x_n)$
  * if $\Theta(i) = 2$, then $q' \notin B$
  * $q' = \delta(q, g)$ for some $g \in 2^{1\ldots n}$ and
    - if $i \in g$ and $\Phi_i \in Ap$, then $s \in \nu(\Phi_i)$
    - if $i \notin g$ and $\Phi_i \in Ap$, then $s \notin \nu(\Phi_i)$
    - if $i \in g$ and $\Phi_i$ is of the form $\mathcal{A}^{\bowtie\rho}$, then $(\mathcal{A}, Init(\mathcal{A})) \in \overline{X}^{\bowtie\rho}$
    - if $i \notin g$ and $\Phi_i$ is of the form $\mathcal{A}^{\bowtie\rho}$, then $(\overline{\mathcal{A}}, Init(\overline{\mathcal{A}})) \in \overline{X^{\bowtie\rho}}$

• $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_c \rightarrow (s, Y^{=1}, \overline{Y}^{=1}, Y^{>0}, \overline{Y}^{>0})_d$ if and only if

– if $X^{>0} = \emptyset$, then $\overline{Y}^{>0} = \emptyset$ and to $Y^{>0}$ we put $(\mathcal{R}, q, \emptyset, \text{FALSE})$ for each $(\mathcal{R}, q) \in \overline{X}^{>0}$ and $(\mathcal{S}, q, \emptyset, \Gamma_0)$ for each $(\mathcal{S}, q) \in \overline{X}^{=1}$.

– if for each $(\mathcal{R}, q, r, \mathsf{flash}) \in X^{=1}$ we have $\mathsf{flash} = \text{TRUE}$ and for each $(\mathcal{S}, q, \Theta, \Gamma) \in X^{=1}$ and $i$ we have $\Theta(i) = 2$ or $\Gamma(i) = \text{TRUE}$, then $\overline{Y}^{=1} = \emptyset$ and to $Y^{=1}$ we put

  * all elements of $X^{=1}$,
  * $(\mathcal{R}, q, \emptyset, \text{FALSE})$ for each $(\mathcal{R}, q) \in \overline{X}^{=1}$ and
  * $(\mathcal{S}, q, \emptyset, \Gamma_0)$ for each $(\mathcal{S}, q) \in \overline{X}^{=1}$.

• $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_d \rightarrow (s, X^{=1}, \overline{X}^{=1}, (Y_{t_1}^{>0}, \ldots, Y_{t_n}^{>0}), (\overline{Y}_{t_1}^{>0}, \ldots, \overline{Y}_{t_1}^{>0}))_e$ if and only if

– $\bigcup_{i=1}^{n} Y_{t_i}^{>0} = X^{>0}$
– $\bigcup_{i=1}^{n} \overline{Y}_{t_i}^{>0} = \overline{X}^{>0}$
– if $s \in V_\Box$, then $Y_{t_i}^{>0} = X^{>0}$ and $\overline{Y}_{t_i}^{>0} = \overline{X}^{>0}$ for all $1 \leq i \leq n$

• $(s, X^{=1}, \overline{X}^{=1}, (X_{t_1}^{>0}, \ldots, X_{t_n}^{>0}), (\overline{X}_{t_1}^{>0}, \ldots, \overline{X}_{t_n}^{>0}))_e \rightarrow (t_i, X^{=1}, \overline{X}^{=1}, X_{t_i}^{>0}, \overline{X}_{t_i}^{>0})_a$ if and only if $(s, t_i) \in E$

To $V'_\bigcirc$ we put all $c$ vertices and those $e$ vertices that have first component from $V_\bigcirc$. All other vertices are from $V_\Box$. *Prob'* preserves probabilities given by *Prob*.

As an initial vertex$s'_{in}$ we choose

- $(s_{in}, \{(\mathcal{R}, Init(\mathcal{R}), \emptyset, \text{FALSE})\}, \emptyset, \emptyset, \emptyset)$ if $\Psi \equiv \mathcal{R}^{=1}(1,\dots,n)$ and $\mathcal{R}$ is Rabin automaton,

- $(s_{in}, \{(\mathcal{S}, Init(\mathcal{S}), \emptyset, \Gamma_0)\}, \emptyset, \emptyset, \emptyset)$ if $\Psi \equiv \mathcal{S}^{=1}(1,\dots,n)$ and $\mathcal{S}$ is Street automaton,

- $(s_{in}, \emptyset, \emptyset, \{(\mathcal{A}, Init(\mathcal{A}))\}, \emptyset)$ if $\Psi \equiv \mathcal{A}^{>0}(1,\dots,n)$.

Valuation $\nu'$ assigns atomic proposition $o$ to all states $(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_c$ in which for each tuple $(\mathcal{R}, q, r, \mathsf{flash}) \in X^{=1}$ we have $\mathsf{flash} = \text{TRUE}$ and for each tuple $(\mathcal{S}, q, \Theta, \Gamma) \in X^{=1}$ we have $\Theta(i)) = 2$ or $\Gamma(i) = \text{TRUE}$. Atomic proposition $p$ is assigned to all states $(s, X^{=1}, \overline{X}^{=1}, \emptyset, \overline{X}^{>0})_c$.

The mixed Büchi winning condition $(P, O)$ states that every run visits $p$ infinitely often and almost every run visits $o$ infinitely often.

**Transferring winning strategies from $G'$ to $G$**

Let $\sigma'$ be a strategy such that $s'_{in} \models^{\nu'}_{G'(\sigma')} (P, O)$. We construct a strategy $\sigma$ such that $s_{in} \models^{\nu}_{G(\sigma)} \Phi$.

**Definition 5.1.6.** *Let $R$ be a set of a states reachable from $s'_{in}$. Function $\Lambda : R \to V^*$ is defined as follows:*

- $\Lambda(s'_{in}) = s_{in}$
- $\Lambda(w v_b v_c v_d v_e (t, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_a) = \Lambda(w).t$

It is easy to verify that $\Lambda$ is injective.

We construct strategy $\sigma : V^* V \Box \to V$ as follows: for a given $w \in \Lambda(R)$ such that $\sigma'(\Lambda^{-1}(w) v_b v_c v_d v_e) = (t, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_a$ we put $\sigma(w) = w.t$.

**Lemma 5.1.7.** $\Lambda(R)$ *is precisely the set of states that are reachable from $s_{in}$. Moreover, if there is a path from $v$ to $v'$, then $\mathcal{P}(v \to^* v') = \mathcal{P}(\Lambda(v) \to *\Lambda(v'))$*

*Proof.* Let $v \in \Lambda(R)$. We show that $v \xrightarrow{x} v'$ if and only if $v' \in \Lambda(R)$ and $\Lambda^{-1}(v) \xrightarrow{1} u_b \xrightarrow{1} u_c \xrightarrow{1} u_d \xrightarrow{1} u_e \xrightarrow{x} \Lambda^{-1}(v')$.

Assume that $last(v) = s$ and $last(\Lambda^{-1}(v)) = (s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_a$. There are two possibilities:

- If $s \in V_{\bigcirc}$, then we have $v \xrightarrow{x} v'$ if and only if $v' = v.t$ for some $t \in V$ such that $s \xrightarrow{x} t$ if and only if $\Lambda^{-1}(v) \xrightarrow{1} \Lambda^{-1}(v).u_b \xrightarrow{1} \Lambda^{-1}(v) u_b u_c \xrightarrow{1} \Lambda^{-1}(v) u_b u_c u_d \xrightarrow{1} \Lambda^{-1}(v) u_b u_c u_d u_e \xrightarrow{x} \Lambda^{-1}(v')$

- If $s \in V_{\Box}$, then we have that $v \xrightarrow{1} v'$ if and only if $v' = v.t$ for $t = \sigma(v)$ and $\sigma'(\Lambda^{-1}(v)) = (t, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_a$ if and only if $\Lambda^{-1}(v) \xrightarrow{1} \Lambda^{-1}(v).u_b \xrightarrow{1} \Lambda^{-1}(v)u_b u_c \xrightarrow{1} \Lambda^{-1}(v)u_b u_c u_d \xrightarrow{1} \Lambda^{-1}(v)u_b u_c u_d u_e \xrightarrow{1} \Lambda^{-1}(v')$

It follows that there is a (unique) path $v = v_0, \ldots, v_n = v'$ from $v$ to $v'$ if and only if there is a (unique) path $w$ from $\Lambda^{-1}(v)$ to $\Lambda^{-1}(v')$. Moreover we have that $w = \Lambda^{-1}(v_0), \ldots, \Lambda^{-1}(v_1), \ldots, \Lambda(v_n)$ and hence we obtain $\mathcal{P}(v \to^* v') = \mathcal{P}(\Lambda^{-1}(v) \to^* \Lambda^{-1}(v'))$. □

**Lemma 5.1.8.** *Let $s'_{0,a} = v(s_0, X_0^{=1}, \overline{X}_0^{=1}, X_0^{>0}, \overline{X}_0^{>0})_a$ be a state reachable from $s_{in}$. For each $(\mathcal{A}, q_0, M, N) \in X_0^{=1}$ and $(\mathcal{A}, q_0) \in \overline{X}^{=1}$ we have $s_0 \models^{\nu}_{G(\sigma)} \mathcal{A}^{=1}_{q_0}$ and for each $(\mathcal{A}, q_0) \in X^{>0}$ and $(\mathcal{A}, q_0) \in \overline{X}^{>0}$ we have and $s_0 \models^{\nu}_{G(\sigma)} \mathcal{A}^{>0}_{q_0}$.*

*Proof.* By induction to the structure of the formula $\Phi$.

As an induction hypothesis, suppose that the statement holds for all subautomata of $\mathcal{A}$.

Let $w$ be run of $G(\sigma')$ that is initiated in $s'_0$ and satisfies the almost sure Büchi condition. Let $(\mathcal{R}, q_0, r, \mathsf{flash})$ be an element of $X_0^{=1}$. Run $w$ is of the form

$$v(s_0, X_0^{=1}, \overline{X}_0^{=1}, X_0^{>0}, \overline{X}_0^{>0})_a, \ldots, v s'_{0,a} s'_{0,b} s'_{0,c} s'_{0,c} s'_{0,d} s'_{0,e} (s_1, X_1^{=1}, \overline{X}_1^{=1}, X_1^{>0}, \overline{X}_1^{>0})_a, \ldots$$

and for all $i$ there is $(\mathcal{A}, q_i) \in X_i^{=1}$ such that $q_{i+1} = \delta(q_i, g)$ where $g \in 2^{\{1,\ldots,n\}}$ and formulas identified by $g$ are exactly those that are satisfied in $s_i$ (here we use the induction hypothesis). It follows that there is a run $w' = s_0 s_1 \ldots$ in $G(\sigma)$ and that $w'_{\mathcal{R}} = q_0 q_1 \ldots$. From the fact that $w$ satisfies the almost sure winning condition, we have that in $w'$ there is a state $(s_i, X_i^{=1}, \overline{X}_i^{=1}, X_i^{>0}, \overline{X}_i^{>0})_a$ for which $(\mathcal{R}, q_i, (A, B), \mathsf{flash}) \in X_i^{=1}$. Thus, in the sequence $q_i q_{i+1} \ldots$ no state from $B$ occurs (this is ensured by structure of $E'$) and some state from $A$ occurs infinitely often ($w$ satisfies the almost sure Büchi objective, which means $\mathsf{flash}$ is TRUE infinitely often for the run of $\mathcal{R}$). The probability of runs initiated in $s'$ satisfying the almost sure Büchi condition is 1. From Lemma 5.1.7, we have $s \models^{\nu}_{G(\sigma)} \mathcal{A}^{=1}_q$. Proof for a Streett automaton is similar.

To prove the lemma for $\overline{X}^{=1}$, it suffices to see that each element from $\overline{X}^{=1}$ will almost surely "move" to $X^{=1}$.

Now let $(\mathcal{A}, q_0)$ be an element of $X_0^{>0}$. We show that $(\mathcal{A}, q)$ will be "removed" from $X^{>0}$ eventually. Let $k = \alpha(\mathcal{A}, s_0) - 1$. There is a finite sequence

$$v(s_0, X_0^{=1}, \overline{X}_0^{=1}, X_0^{>0}, \overline{X}_0^{>0})_a, \ldots, v'(s_k, X_{k+1}^{=1}, \overline{X}_{k+1}^{=1}, X_{k+1}^{>0}, \overline{X}_{k+1}^{>0})_a$$

in which for all $i \leq k$ we have $(\mathcal{A}, q_i) \in X_i^{>0}$ and $(\mathcal{A}, q_{k+1}) \in \overline{X}^{=1}$ and for $j \leq k$ we have $q_{j+1} = \delta_{\mathcal{A}}(q_j, g_j)$ for some $g_j \in 2^{\{1,\ldots,|AP|\}}$. In this, $g_j$ represents all subformulae valid in $s_j$. From previous we have that in $s_{k+1} \models^{\nu} \mathcal{A}^{=1}_{q_{k+1}}$. Moreover, there is a finite path $w = s_0, \ldots, s_{k+1}$ in $G(\sigma)$ such that automaton $\mathcal{A}_{q_0}$ ends in $q_{k+1}$ after reading $w$. From Lemma 5.1.3 and Lemma 5.1.4 we have that $s_0 \models^{\nu} \mathcal{A}_{q_0}^{>0}$.

To prove the lemma for $\bar{X}^{>0}$, note that each element from $\bar{X}^{>0}$ will surely "move" to $X^{>0}$. □

As a simple corollary of Lemma 5.1.8 we have the following theorem.

**Theorem 5.1.9.** $s_{in} \models^{\nu}_{G(\sigma)} \Phi$.

**Transferring winning strategies from $G$ to $G'$**

Suppose that there is a *Phi*-winning strategy $\sigma$ for $G$. We construct a strategy $\sigma'$ and show it is $(P,Q)$-winning for $G'$. Let $R$ denote states of $G(\sigma)$ that are reachable from $s_{in}$.

**Definition 5.1.10.** *Let $\mathcal{A}$ be a Rabin or Streett automaton. We define a function $\Xi_{\mathcal{A}} : R \times States(\mathcal{A}) \to R \times States(\mathcal{A}) \cup \{\text{TRUE}\}$ to be*

- $\Xi_{\mathcal{A}}(u,q) = \text{TRUE}$ *if $u \models A_q^{=1}$*
- $\Xi_{\mathcal{A}}(u,q) = (ws, q')$ *if $u \to us$, $\mathcal{A}_q$ goes to $q'$ after reading $u$ and $\alpha(\mathcal{A}_q, u) > \alpha(\mathcal{A}_{q'}, us)$. If there are more possibilities, we choose the value arbitrarily.*
- $\Xi_{\mathcal{A}}(u,q) = \bot$ *otherwise*

**Definition 5.1.11.** *Let $\mathcal{R}$ be a Rabin automaton. We define a function $\Delta_{\mathcal{R}} : R \times States(\mathcal{R}) \to Final(\mathcal{R}) \cup \{\emptyset\}$ by*

- $\Delta_{\mathcal{R}}(u,q) = (A,B)$ *if both following conditions hold*
  - *for every run $w$ initiated in $u$ we have that no state from $B$ occurs in $w_{\mathcal{R}_q}$,*
  - *for almost every run $w$ initiated in $u$, some state from $A$ occurs infinitely often in $w_{\mathcal{R}_q}$ and*
  - *$\Delta_{\mathcal{R}}(u',q) = (A,B)$, then $\Delta_{\mathcal{R}}(u,q) = (A,B)$ for each prefix $u'$ of $u$ (i.e. if the value is set once, it is preserved forever),*
- $\Delta_{\mathcal{R}}(u,q) = \emptyset$ *otherwise.*

**Definition 5.1.12.** *Let $\mathcal{S}$ be a Streett automaton. We define a function $\Delta_{\mathcal{S}} : R \times States(\mathcal{S}) \to \{(x_1, \ldots, x_n) \mid n = |Final(\mathcal{S})|, x_i \in \{1,2\}\} \cup \{\emptyset\}$ by*

- $\Delta_{\mathcal{S}}(u,q) = (x_1, \ldots, x_n)$ *if for each $1 \le i \le n$ following conditions hold:*
  - *if $x_i = 2$, then for every run $w$ initiated in $u$ we have that no state from $B$ occurs in $u_{\mathcal{S}_q}$, where $(A,B) = Final(\mathcal{S})(i)$,*

- *if* $x_i = 1$*, then for almost every run w initiated in u we have that some state from A occurs in* $u_{\mathcal{S}_q}$ *infinitely often, where* $(A,B) = Final(\mathcal{S})(i)$*,*
- *if* $\Delta_{\mathcal{R}}(u',q) = (x_1,\ldots,x_n)$*, then* $\Delta_{\mathcal{R}}(u,q) = (x_1,\ldots,x_n)$ *for each prefix* $u'$ *of u (i.e. if the value is set once, it is preserved forever),*

- $\Delta_{\mathcal{S}}(u,q) = \emptyset$ *otherwise.*

We use $\hat{w}$ to denote the finite sequence $s_0, s_1, \ldots, s_k$ for each finite sequence $w$ of the form

$$(s_0, X_0^{=1}, \overline{X}_0^{=1}, X_0^{>0}, \overline{X}_0^{>0})_a, \ldots, (s_1, X_1^{=1}, \overline{X}_1^{=1}, X_1^{>0}, \overline{X}_1^{>0})_a, \ldots, (s_k, X_k^{=1}, \overline{X}_k^{=1}, X_k^{>0}, \overline{X}_k^{>0})_a$$

We define the strategy $\sigma'$ inductively as follows:

- For each $w(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_a$ we choose the transition to the vertex $(s, Y^{=1}, \overline{Y}^{=1}, Y^{>0}, \overline{Y}^{>0})_b$ in which
  - $Y^{=1} = X^{=1}$
  - $\overline{Y}^{=1} = \overline{X}^{=1} \cup$
    $\{(\mathcal{A}, Init(\mathcal{A})) \mid (\mathcal{A}, =1) \in \beta(\mathcal{A}', s), ((\mathcal{A}', q, M, N) \in X^{=1} \vee (\mathcal{A}', q) \in \overline{X}^{=1} \cup X^{>0} \cup \overline{X}^{>0})\} \cup$
    $\{(\mathcal{A}, q) \mid (\mathcal{A}, q) \in X^{>0}, \Xi_{\mathcal{A}}(s,q) = \text{TRUE}\}$
  - $Y^{>0} = X^{>0} \setminus \{(\mathcal{A}, q) \mid (\mathcal{A}, q) \in X^{>0}, \Xi_{\mathcal{A}}(s,q) = \text{TRUE}\}$
  - $\overline{Y}^{>0} = \overline{X}^{>0} \cup$
    $\{(\mathcal{A}, Init(\mathcal{A})) \mid (\mathcal{A}, >0) \in \beta(\mathcal{A}', s), ((\mathcal{A}', q, M, N) \in X^{=1} \vee (\mathcal{A}', q) \in \overline{X}^{=1} \cup X^{>0} \cup \overline{X}^{>0})\}$

- For each $w(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_b$ we choose the transition to the vertex $(s, Y^{=1}, \overline{Y}^{=1}, Y^{>0}, \overline{Y}^{>0})_c$ where
  - $X^{=1} = \{(\mathcal{R}, q', r', \mathsf{flash}') \mid (\mathcal{R}, q, r, \mathsf{flash}) \in X^{=1}, \delta_{\mathcal{R}}(q,g) = q', r' = \Delta_{\mathcal{R}}(s,q'), \mathsf{flash}' = \mathsf{flash} \vee (r = (A,B) \wedge q \in A), g \in 2^{\{1,\ldots,n\}}\}$
    $\cup$
    $\{\mathcal{S}, q', \Theta', \Gamma') \mid (\mathcal{S}, q, \Theta, \Gamma) \in X^{=1}, \delta_{\mathcal{S}}(q,g) = q', \Theta' = \Delta_{\mathcal{S}}(s,q'), \Gamma'(i) = \Gamma(i) \vee (\Theta(i) = 1 \wedge q \in Final(\mathcal{S})(i)(1)), g \in 2^{\{1,\ldots,n\}}\}$
  - $\overline{Y}^{=1} = \{(\mathcal{A}, q') \mid (\mathcal{A}, q) \in \overline{X}^{=1}, \delta_{\mathcal{A}}(q,g) = q', g \in 2^{\{1,\ldots,n\}}\}$
  - $Y^{>0} = \{(\mathcal{A}, q') \mid (\mathcal{A}, q) \in X^{>0}, \delta_{\mathcal{A}}(q,g) = q', g \in 2^{\{1,\ldots,n\}}\}$
  - $\overline{Y}^{>0} = \{(\mathcal{A}, q') \mid (\mathcal{A}, q) \in \overline{X}^{>0}, \delta_{\mathcal{A}}(q,g) = q', g \in 2^{\{1,\ldots,n\}}\}$

In every above given set, $i \in g$ if and only if $s \models \Phi_i$.

- For each $w(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_d$ we choose the transition to the vertex $(s, X^{=1}, \overline{X}^{=1}, (Y^{>0}_{t_1}, \dots, (Y^{>0}_{t_n}), (\overline{Y}^{>0}_{t_1}, \dots, \overline{Y}^{>0}_{t_n}))_e$ such that
  - if $s \in V_\Box$, then $Y^{>0}_{t_i} = X^{>0}$ and $\overline{Y}^{>0}_{t_1} = \overline{X}^{>0}$.
  - if $s \in V_\bigcirc$, then for each $(\mathcal{A}, q) \in X^{>0}$ (resp. $\overline{X}^{>0}$) we put $(\mathcal{A}, q)$ to $Y^{>0}_{t_i}$ (resp. $\overline{Y}^{>0}_{t_i}$) where $\alpha(\mathcal{A}, \hat{w}t_i)$ is minimal for $1 \leq i \leq n$. If there are more possibilities, we choose $i$ arbitrarily.
- For each $w(s, X^{=1}, \overline{X}^{=1}, (X^{>0}_{t_1}, \dots, X^{>0}_{t_n}), (\overline{X}^{>0}_{t_1}, \dots, \overline{X}^{>0}_{t_n}))_e \in V'_\Box$ we choose the transition to $(t_i, X^{=1}, \overline{X}^{=1}, X^{>0}_{t_i}, \overline{X}^{>0}_{t_i})_a$ such that $\sigma(\hat{w}) = t_i$.

To complete our proof, we need to show that

1. $\sigma'$ is indeed a strategy and
2. $\sigma'$ is $(P, O)$-winning for $G'$.

**Lemma 5.1.13.** *For each $us \in V'^* V'_\Box$ we have $(s, \sigma(us)) \in E'$.*

*Proof.* We analyze lemma only for $b$ vertices here. Verifying other types of vertices is simple.

Let $v = w(s, X^{=1}, \overline{X}^{=1}, X^{>0}, \overline{X}^{>0})_b$ and $\sigma'(v) = (s, Y^{=1}, \overline{Y}^{=1}, Y^{>0}, \overline{Y}^{>0})_c$

- Let us fix $(\mathcal{A}, q)$ from $\overline{X}^{=1}$ (resp. $X^{>0}$, $\overline{X}^{>0}$). From definition of $\sigma'$ we have that there is $(\mathcal{A}, q') \in \overline{Y}^{=1}$ (resp. $Y^{>0}$, $\overline{Y}^{>0}$) such that $q' = \delta_\mathcal{A}(q, g)$ where $i \in g$ iff $s \models \Phi_i$.
  - If $i \in g$ and $\Phi_i \in Ap$ then $s \in \nu(\Phi_i)$
  - If $i \notin g$ and $\Phi_i \in Ap$ then $s \notin \nu(\Phi_i)$
  - If $i \in g$ and $\Phi_i$ is of the form $\mathcal{A}'^{\bowtie\rho}$, then $(\mathcal{A}', \bowtie \rho) \in \beta(\mathcal{A}_q, \hat{w}s)$ and thus $(\mathcal{A}', Init(\mathcal{A}')) \in X^{\bowtie\rho}$
  - If $i \notin g$ and $\Phi_i$ is of the form $\mathcal{A}'^{\bowtie\rho}$, then $(\overline{\mathcal{A}'}, \overline{\bowtie \rho}) \in \beta(\mathcal{A}_q, \hat{w}s)$ and thus $(\overline{\mathcal{A}'}, Init(\overline{\mathcal{A}'})) \in X^{\overline{\bowtie\rho}}$
- Let us fix $(\mathcal{R}, q, r, \mathsf{flash}) \in X^{=1}$. From definition of $\sigma'$ we have that there is $(\mathcal{R}, q', r', \mathsf{flash}) \in Y^{=1}$ such that
  - $\mathsf{flash}' = \mathsf{flash} \vee (r = (A, B) \wedge q \in A)$
  - $r' = \Delta_\mathcal{R}(s, q')$ and from definition of $\Delta_\mathcal{R}$ we have that if $r = (A, B)$, then $r' = (A, B)$.

- if $r' = (A,B)$, then $\Delta_{\mathcal{R}}(s,q') = (A,B)$ and thus no run initiated in $s$ forces $\mathcal{R}$ to go through $B$. In particular, $q \notin B$
- $q' = \delta_{\mathcal{R}}(q,g)$ where $i \in g$ iff $s \models \Phi_i$
  * If $i \in g$ and $\Phi_i \in Ap$ then $s \in \nu(\Phi_i)$
  * If $i \notin g$ and $\Phi_i \in Ap$ then $s \notin \nu(\Phi_i)$
  * If $i \in g$ and $\Phi_i$ is of the form $\mathcal{A}'^{\bowtie\rho}$, then $(\mathcal{A}', \bowtie \rho) \in \beta(\mathcal{R},q,\hat{w}s)$ and thus $(\mathcal{A}', Init(\mathcal{A}')) \in \bar{X}^{\bowtie\rho}$
  * If $i \notin g$ and $\Phi_i$ is of the form $\mathcal{A}'^{\bowtie\rho}$, then $(\overline{\mathcal{A}'}, \overline{\bowtie \rho}) \in \beta(\mathcal{R},q,\hat{w}s)$ and thus $(\overline{\mathcal{A}'}, Init(\overline{\mathcal{A}'})) \in \bar{X}^{\overline{\bowtie\rho}}$

- Let us fix $(\mathcal{S},q,\Theta,\Gamma) \in X^{=1}$. From definition of $\sigma'$ we have that there is $(\mathcal{S},q',\Theta',\Gamma') \in Y^{=1}$ such that
  - $\Gamma'(i) = \Gamma(i) \vee (\Theta(i) = 1 \wedge q \in Final(\mathcal{S})(i)(1))$,
  - if $\Theta = (x_1,\dots,x_n)$, then from definition of $\Delta_{\mathcal{S}}$ we have $\Theta' = (x_1,\dots,x_n)$,
  - if $\Theta(i) = 2$ and $B = Final(\mathcal{S})(i)(2)$, then no run can force $\mathcal{S}$ to go through $B$. In particular, $q \notin B$,
  - $q' = \delta_{\mathcal{S}}(q,g)$ where $i \in g$ iff $s \models \Phi_i$
    * If $i \in g$ and $\Phi_i \in Ap$ then $s \in \nu(\Phi_i)$
    * If $i \notin g$ and $\Phi_i \in Ap$ then $s \notin \nu(\Phi_i)$
    * If $i \in g$ and $\Phi_i$ is of the form $\mathcal{A}'^{\bowtie\rho}$, then $(\mathcal{A}', \bowtie \rho) \in \beta(\mathcal{S},q,\hat{w}s)$ and thus $(\mathcal{A}', Init(\mathcal{A}')) \in \bar{X}^{\bowtie\rho}$
    * If $i \notin g$ and $\Phi_i$ is of the form $\mathcal{A}'^{\bowtie\rho}$, then $(\overline{\mathcal{A}'}, \overline{\bowtie \rho}) \in \beta(\mathcal{S},q,\hat{w}s)$ and thus $(\overline{\mathcal{A}'}, Init(\overline{\mathcal{A}'})) \in \bar{X}^{\overline{\bowtie\rho}}$

□

**Theorem 5.1.14.** $s'_{in} \models^{\nu'}_{G'(\sigma')} (P,O)$

*Proof.* The proof consists of two parts. First, we show that $P$ is satisfied.

We show that for each run $w = s'_0 s'_1 \ldots \in Run(s'_{in})$ and for every $i \geq 0$ we have that there is $k > i$ such that $last(s'_k) = (s_k, X_k^{=1}, \bar{X}_k^{=1}, \emptyset, \bar{X}_k^{>0})_c$. In $G'(\sigma')$, elements are added to fourth component of every state only if the fourth component is $\emptyset$. Thus, it suffices to show that number of elements in the fourth component is lowering along each run.

Without loss of generality, suppose that $last(s'_i) = (s, X^{=1}, \bar{X}^{=1}, X^{>0}, \bar{X}^{>0})_c$. For each $(\mathcal{A},q) \in X^{>0}$ we have $\Xi_{\mathcal{A}}(s,q)$ defined. Let $(\mathcal{A},q)$ be a tuple for which the value $\alpha(\mathcal{A}_q, s)$ is minimal. It can be seen that for all $(s_Y, Y^{=1}, \bar{Y}^{=1}, Y^{>0}, \bar{Y}^{>0})_c$ that are descendants of $s'_i$ we have either $|Y^{>0}| < |X^{>0}|$ or $\alpha(\mathcal{A}'_{q'}, s_Y) < \alpha(\mathcal{A}_q, s)$ for

$(\mathcal{A}', q')$ which is minimal wrt $\alpha$. Now it only suffices to see that if $\alpha(\mathcal{A}'_{q'}, s) = 0$, then $(\mathcal{A}', q')$ is definitely removed.

Second part is to prove that $O$ is satisfied. We do this by showing that from each state $s'$ reachable from $s'_{in}$, state satisfying the atomic proposition $p$ is almost surely reached.

Without loss of generality, suppose that $last(s) = (s, X^{=1}, \bar{X}^{=1}, X^{>0}, \bar{X}^{>0})_c$. Observe that before reaching $p$, number of elements in second component will not rise. Let $n > 0$ be the number of elements of $X^{=1}$ that are of the form $(\mathcal{R}, q, \emptyset, \text{FALSE})$. Definition 5.1.11 together with Lemma 5.1.1 states that we almost surely reach a state in which the number of tuples of the form $(\mathcal{R}, q, \emptyset, \text{FALSE})$ will be strictly lower than $n$. From Definition 5.1.11 and Lemma 5.1.2 we get a similar argument for all tuples of the form $(\mathcal{S}, q, (0, \dots, 0), (\text{FALSE}, \dots, \text{FALSE}))$. If we use both these arguments inductively, we obtain that from $s'$ state $s'_i$ is reached such that $last(s'_i) = (s_i, X_i^{=1}, \bar{X}_i^{=1}, X_i^{>0}, \bar{X}_i^{>0})_c$ and $X_i^{=1}$ does not contain any tuple of the form $(\mathcal{R}, q, \emptyset, \mathsf{flash})$ or $(\mathcal{S}, q, \emptyset, \Gamma)$.

As a consequence of Definition 5.1.11 and Definition 5.1.12 we have that from $s'_i$ we almost surely reach a state $s'_k$ such that $last(s'_k) = (s_k, X_k^{=1}, \bar{X}_k^{=1}, X_k^{>0}, \bar{X}_k^{>0})_c$ and $X_k^{=1}$ contains only tuples that have the form $(\mathcal{S}, q, (x_1, \dots, x_m), (y_1, \dots, y_m))$ or $(\mathcal{R}, q, (A, B), \text{TRUE})$ where $y_i$ is FALSE only if $x_i = 1$ and thus we reach $p$ also. ☐

**Theorem 5.1.15. [1]** *Given a $1\frac{1}{2}$-player game, valuation $\nu$, vertex $s$ and mixed Büchi objective $(P, Q)$, the problem whether there is a $(\nu, (P, Q))$-winning strategy in $s$ is decidable in polynomial time.*

**Corollary 5.1.16.** *Given a $1\frac{1}{2}$-player game, valuation $\nu$, vertex $s$ and PECTL$^*$ formula $\Phi$, the problem whether there is a $(\nu, \Phi)$-winning strategy in $s$ is in 2-**EXPTIME**.*

## 5.2 Lower bound

In this section we show that problem of deciding whether there is a winning strategy that uses only finite memory for a $1\frac{1}{2}$-player game and qualitative PCTL objectives is **EXPTIME**-hard.

We reduce the acceptance problem for alternating LBA (which is known to be **EXPTIME**-complete [4]). An **alternating LBA** is a tuple $\mathcal{M} = (Q, \mathcal{A}, \Gamma, q_0, \vdash, \dashv, \delta, P)$ where $Q$ is a finite set of **control states**, $\mathcal{A}$ is a finite **input alphabet**, $\Gamma \supseteq \mathcal{A}$ is a finite **tape alphabet**, $q_0 \in Q$ is the **initial** control state, $\vdash, \dashv \in \Gamma$ are the left-end and the right-end markers, $\delta : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$ is a **transition function**, and $P = (Q_\forall, Q_\exists, Q_{acc}, Q_{rej})$ is a partition of $Q$ into **universal**, **existential**, **accepting**, and **rejecting** states, respectively. We can safely assume that $Q \cap \Gamma = \emptyset$, $q_0 \in Q_\exists$, $\delta(q, A) = \emptyset$ for all $q \in Q_{acc} \cup Q_{rej}$, and $\delta(q, A)$ has exactly two elements $(q_1, A_1, D_1)$,

$(q_2, A_2, D_2)$, where $q_1 \neq q_2$, for all $q \in Q_\forall \cup Q_\exists$. A **computational tree** for $\mathcal{M}$ on a word $u \in \mathcal{A}^*$ is a tree $T$ satisfying the following: the root of $T$ is (labeled by) the initial configuration for $u$, and if $N$ is a node of $T$ labeled by a configuration with a control state $q$, then the following holds:

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

*$\mathcal{M}$ accepts $u$* if and only if there is a finite computational tree $T$ such that all leafs of $T$ are accepting configurations. We can safely assume that *all* computational trees for $\mathcal{M}$ are finite.

Let $\mathcal{M} = (Q, \mathcal{A}, \Gamma, q_0, \vdash, \dashv, \delta, P)$ be an alternating LBA and $u \in \mathcal{A}^*$ an input word. We construt a $1\frac{1}{2}$ game $G(\mathcal{M}, u) = (V, E, (V_\Box, V_\bigcirc), Prob)$ and an objective $(\nu, \varphi)$ where $\varphi \in \mathcal{L}(\mathrm{F}^{=1}, \mathrm{G}^{=1})$ such that player $\Box$ has a $(\nu, \varphi)$-winning HD (or HR) strategy in a distinguished vertex $g(1,1) \in V$ if and only if $\mathcal{M}$ accepts $u$. Our construction is done in polynomial time. Configurations of $\mathcal{M}$ are written as

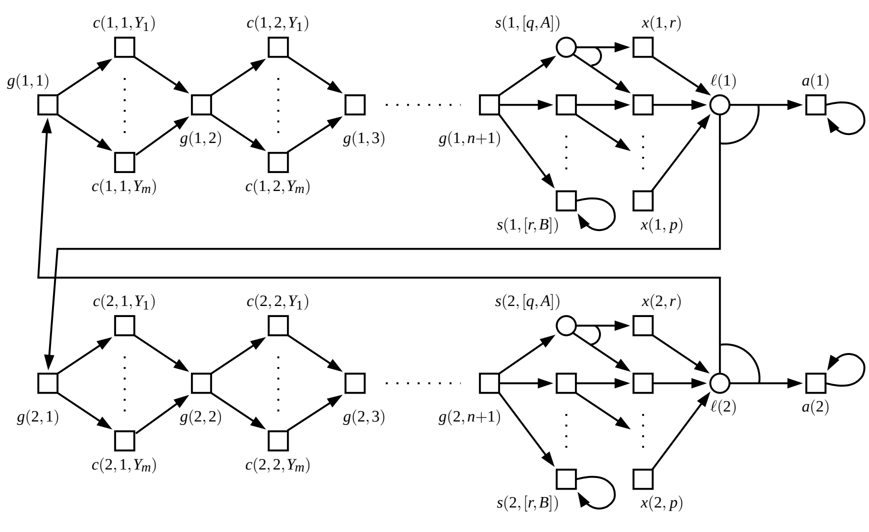



Figure 5.1: The structure of $G(\mathcal{M}, w)$

words over the alphabet $\Xi = Q \cup \Gamma$ in the standard way; for example, the initial configuration for $u$ is written as $q_0 \vdash u \dashv$. Another standard result is that one can efficiently compute the set $Comp(\mathcal{M}) \subseteq \Xi^6$ of all **compatible 6-tuples** such that for each configuration $c$ (written as a word over $\Xi$) we have that $c' \in \Xi^*$ is a one-step successor of $c$ iff $c'$ has the same length as $c$ and for all $1 \leq i \leq |c|-2$ we have that $(c(i), c(i{+}1), c(i{+}2), c'(i), c'(i{+}1), c'(i{+}2)) \in Comp(\mathcal{M})$.

Let $n = |u|+3$. The structure of $G(\mathcal{M},u)$ is shown in Figure 5.1. The set $V$ consists of the following vertices:

- $g(j,i)$, where $j \in \{1,2\}$ and $1 \leq i \leq n{+}1$;
- $c(j,i,Y)$, where $j \in \{1,2\}$, $1 \leq i \leq n{+}1$, and $Y \in \Xi$;
- $s(j,[q,A])$, where $j \in \{1,2\}$, $q \in Q$, and $A \in \Gamma$;
- $x(j,q)$, where $j \in \{1,2\}$ and $q \in Q$;
- $\ell(1)$, $\ell(2)$, $a(1)$, $a(2)$.

The set $E$ contains the following transitions:

- $g(j,i) \to c(j,i,Y)$ and $c(j,i,Y) \to g(j,i{+}1)$ for all $j \in \{1,2\}$, $i \in \{1,\cdots,n\}$, and $Y \in \Xi$;
- $g(j,n{+}1) \to s(j,[q,A])$ for all $j \in \{1,2\}$, $q \in Q$, and $A \in \Gamma$;
- $s(j,[q,A]) \to s(j,[q,A])$ for all $j \in \{1,2\}$, $A \in \Gamma$, and $q \in Q$ where $q$ is accepting or rejecting;
- $s(j,[q,A]) \to x(j,q')$ for all $j \in \{1,2\}$, $A \in \Gamma$, and $q,q' \in Q$ where $q$ is existential or universal and $\delta(q,A)$ contains a triple of the form $(q',B,D)$;
- $x(j,q) \to \ell(j)$ for all $j \in \{1,2\}$ and $q \in Q$;
- $\ell(1) \to g(2,1)$, $\ell(1) \to a(1)$, $\ell(2) \to g(1,1)$, $\ell(2) \to a(2)$;
- $a(1) \to a(1)$, $a(2) \to a(2)$.

The set $V_{\bigcirc}$ consists of $\ell(1), \ell(2)$ and all $s(j,[q,A])$ where $q \in Q_{\forall}$. The other vertices belong to $V_{\Box}$. The function *Prob* always assigns the uniform probability distribution over the set of outgoing transitions.

A play starts in $g(1,1)$. The intended scenario is the following: Player $\Box$ successively "guesses" the configurations of $\mathcal{M}$ by choosing appropriate moves in the vertices $g(1,1),\cdots,g(1,n)$ and $g(2,1),\cdots,g(2,n)$. In the states $g(1,n{+}1)$ and $g(2,n{+}1)$, player $\Box$ chooses the successor $s(1,[q,A])$ and $s(2,[q,A])$ where $q$ is the control state and $A$ the scanned tape symbol in the configuration just guessed. If $q$ is accepting or rejecting, there is a loop on the corresponding vertex (we call these vertices accepting/rejecting). If $q$ is existential, in the next move player $\Box$ chooses one of the two control states which can be entered by $\mathcal{M}$ after performing one computational step in the configuration just guessed. If $q$ is universal, this choice is random. In the next guessing phase, player $\Box$ will use the chosen control state and hence he "guesses" the configuration chosen in the previous round. This goes on until a loop is reached, which can happen either in a accepting/rejecting vertex, or in the vertices $a(1), a(2)$. The formula $\varphi$ constructed below ensures that player $\Box$ cannot violate this scenario, cannot use randomized moves, and has to enter $a(1)$, $a(2)$, or an accepting vertex with probability one. It turns out that $\mathcal{M}$ accepts $w$ if player $\Box$ has a HD (or HR) strategy such that $\varphi$ is satisfied in $g(1,1)$.

Now we describe the formula $\varphi$ in detail. For each $v \in V$ we fix a fresh atomic proposition $p_v$ which is valid only in $v$. Slightly abusing notation, we write $v$ instead of $p_v$. We put

$$\varphi \;\equiv\; \mathit{Init} \wedge \mathit{Succ} \wedge \mathit{Ctrl} \wedge \mathit{Choice} \wedge \mathit{Accept} \wedge \mathit{NoRnd}$$

The subformula *Init* says that the initial configuration $w = q_0 {\vdash} u {\dashv}$ is guessed from $g(1,1)$ at the beginning of a play. Hence, $\mathit{Init} \equiv \bigwedge_{i=1}^{n} \mathrm{F}^{=1} c(1,i,w(i))$. Note that if player $\Box$ selects, e.g., $c(1,1,\vdash)$ instead of $c(1,1,q_0)$, the formula $\mathrm{F}^{=1} c(1,1,q_0)$ is not satisfied in $g(1,1)$; this is because the vertex $c(1,1,q_0)$ can then be visited only after passing through the vertex $\ell(1)$, which enters the $a(1)$-loop with probability $1/2$.

The subformula *Succ* is of the form $\mathit{Succ}_1 \wedge \mathit{Succ}_2$. $\mathit{Succ}_1$ says that whenever the vertex $g(1,1)$ is entered, one of the following conditions holds:

- the control state of the configuration which is to be guessed from $g(1,1)$ is accepting;
- for every $1 \leq i \leq n-2$, the symbols chosen in $g(1,i)$, $g(1,i+1)$, $g(1,i+2)$ and in $g(2,i)$, $g(2,i+1)$, $g(2,i+2)$ form a compatible 6-tuple.

For all $X_1, X_2, X_3 \in \Xi$, let $\mathcal{C}(X_1,X_2,X_3)$ be the set of all triples $Y_1,Y_2,Y_3$ such that $(X_1,X_2,X_3,Y_1,Y_2,Y_3) \in \mathit{Comp}(\mathcal{M})$. The formula $\mathit{Succ}_1$ looks as follows:

$$\mathrm{G}^{=1}\Big(g(1,1) \Rightarrow \big(\mathit{Acc} \vee \bigwedge_{i=1}^{n-2} \bigvee_{\vec{X} \in \Xi^3} \mathit{Pos}(1,i,\vec{X})\big)\Big)$$

where $\mathit{Acc} \equiv \bigvee_{q \in Q_{\mathit{Acc}}, A \in \Gamma} \mathrm{F}^{=1} s(1,[q,A])$ and $\mathit{Pos}(1,i,\vec{X})$ stands for

$$\mathrm{F}^{=1} c(1,i,\vec{X}_1) \wedge \mathrm{F}^{=1} c(1,i+1,\vec{X}_2) \wedge \mathrm{F}^{=1} c(1,i+2,\vec{X}_3) \wedge \mathrm{F}^{=1} \psi$$

where $\psi$ is the formula

$$a(1) \vee \bigvee_{\vec{Y} \in \mathcal{C}(\vec{X})} c(2,i,\vec{Y}_1) \wedge \mathrm{F}^{=1}\big(c(2,i+1,\vec{Y}_2) \wedge \mathrm{F}^{=1} c(2,i+2,\vec{Y}_3)\big)$$

The formula $Succ_2$ says analogous conditions about the vertex $g(2,1)$ and is implemented similarly as $Succ_1$.

The subformula *Ctrl* is of the form $Ctrl_1 \wedge Ctrl_2$. $Ctrl_1$ says that the vertex chosen from $g(1,n{+}1)$ corresponds to the control state and the scanned tape symbol in the configuration just guessed. This can be written as follows:

$$\bigwedge_{\substack{1\le i<n\\ q\in Q}} \mathrm{G}^{=1}\Big(c(1,i,q) \Rightarrow \bigvee_{A\in\Gamma}\big(\mathrm{F}^{=1}c(1,i{+}1,A) \wedge \mathrm{F}^{=1}s(1,[q,A])\big)\Big)$$

$Ctrl_2$ encodes an analogous property for the vertex chosen from $g(2,n{+}1)$.

The subformula $Choice \equiv Choice_1 \wedge Choice_2$ says that whenever a vertex of the form $x(1,q)$ (or $x(2,q)$) is visited, then the configuration guessed next will have $q$ as its control state. We write just $Choice_1$ ($Choice_2$ is constructed analogously):

$$\bigwedge_{q\in Q} \mathrm{G}^{=1}\Big(x(1,q) \Rightarrow \mathrm{F}^{=1}\big(a(1) \vee \bigvee_{A\in\Gamma} s(2,[q,A])\big)\Big)$$

The subformula *Accept* says that the probability of visiting $a(1)$, $a(2)$, or one of the accepting vertices, is equal to one:

$$\mathrm{F}^{=1}\Big(a(1) \vee a(2) \vee \bigvee_{j\in\{1,2\},q\in Q_{acc},A\in\Gamma} s(j,[q,A])\Big)$$

Note that due to the assumption that every computational tree of $\mathcal{M}$ is finite, the previous formulae already guarantee that player $\Box$ *surely* (i.e., in the *non-probabilistic* sense) enters $a(1)$, $a(2)$, or an accepting/rejecting vertex after finitely many rounds. Hence, there is no infinite path in the computational tree constructed by the play, and the subformula *Accept* guarantees that all leafs in this tree are accepting.

Finally, the subformula *NoRnd* says that player $\Box$ does not use randomized moves. This subformula is actually needed only if player $\Box$ uses a HR strategy (*NoRnd* is redundant for HD strategies). This is implemented simply by saying that whenever a vertex of $V_\Box$ with more than one successor is visited, then one of its successors is visited with probability one in the next move. For example, for $g(1,1)$ the formula looks as follows:

$$G^{=1}\Big(g(1,1) \Rightarrow \bigvee_{\alpha\in\Xi} \mathrm{F}^{=1}c(1,2,\alpha)\Big)$$

The formulae for the other vertices of $V_\Box$ look similarly.

CHAPTER 6

# CONCLUSION AND FUTURE WORK

In this thesis, we have given almost complete classification of $2\frac{1}{2}$-player games, memoryless strategies and PCTL logic. We have also proved that the problem of determining whether there is a finite state winning strategy for $1\frac{1}{2}$-player game, qualitative PECTL$^*$ objective is in 2-**EXPTIME**. We did not manage to improve the lower bound for this problem. The best known result is **EXPTIME**-hardness. Future work is to prove matching lower bound for this problem. Let us note that we have not studied these problems for $2\frac{1}{2}$-player games. Other open problem is exact complexity of determining the existence of MR winning strategy for $1\frac{1}{2}$-player and $2\frac{1}{2}$-player games and PCTL objective.

One can also study different variants of games, as mentioned in introduction. Possible extensions are concurrent games or incomplete information games, for example.

# BIBLIOGRAPHY


[1] T. Br´azdil, V. Brožek, V. Forejt and A. Kučera. Stochastic Games with Branching-Time Winning Objectives. In *21th IEEE Symposium on Logic in Computer Science*, 2006

[2] T. Br´azdil, V. Brožek, V. Forejt and A. Kučera. Stochastic Games with Branching-Time Winning Objectives. In *21th IEEE Symposium on Logic in Computer Science*, 2006

[3] K. Etessami and M. Yannakakis. Recursive Concurrent Stochastic Games. In *33nd Int. Coll. on Automata, Languages and Programming*, 2006

[4] Ch. Papadimitriou. Computational complexity. *Addison*, 1994

[5] K. Chatterjee. Optimal Strategy Synthesis in Stochastic M¨uller Games. In *Foundations of Software Science and Computation Structures*, 2007

[6] K. Etessami and M. Yannakakis. Effi cient Qualitative Analysis of classes of Recursive Markov Decision Processes and Simple Stochastic Games. In *23rd Symp. on Theoretical Aspects of Computer Science*, 2006.

[7] E. Allen Emerson and Joseph Y. Halpern. Decision procedures and expressiveness in the temporal logic of branching time. In *Annual ACM Symposium on Theory of Computing*, 1982

[8] C.Baier, M. Gr¨oßer, M. Leucker, B. Bollig and F. Ciesinski. Controller synthesis for probabilistic systems. In *18th IFIP World Computer Congress, Theoretical Computer Science*, pp 439-506, 2004

[9] K. Chatterjee, L. de Alfaro and T. A. Henzinger. The Complexity of Quantitative Concurrent Parity Games. In *ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[10] K. Chatterjee, L. Doyen, T. A. Henzinger and J. Raskin. Algorithms for Omega-Regular Games with Imperfect Information. In *Computer Science Logic*, 2006

[11] K. Chatterjee and T. A. Henzinger. Semiperfect-Information Games. In *Foundations of Software Technology and Theoretical Computer Science*, 2005.

[12] K. Chatterjee, L. de Alfaro and T. A. Henzinger. The Complexity of Quantitative Concurrent Parity Games. *ACM-SIAM Symposium on Discrete Algorithms*, 2006.

[13] K. Etessami and M. Yannakakis. Recursive Markov Decision Processes and Recursive Stochastic Games. In *32nd Int. Coll. on Automata, Languages and Programming*, 2005.

[14] A. Kučera and O. Stražovsk´y. On the controller synthesis for fi nite-state Markov decision processes. In *Proceedings of FST&TCS 2005*, vol. 3821 of LNCS, Springer, 2005

[15] D. Perrin and J. Pin. Infi nite Words, Automata, Semigroups, Logic and Games. *Elsevier*, 2004

[16] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. In *Formal Aspects of Computing*, 1994

[17] A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Foundations of Software Technology and Theoretical Computer Science*, 1995.

[18] J. L. Lions et al.. Ariane 5 Flight Failure. *Ariane 501 Inquiry Board report*, 1996

[19] A. Turing. On computable numbers, with an application to the Entscheidungsproblem. In *Proceedings of the London Mathematical Society, Series 2, 42*, 1936

[20] V. Brožek. Decidability and complexity of infinite-state stochastic games. *Master's thesis*, to appear.

[21] T. Brázdil. Verification of Probabilistic Recursive Sequential Programs. *Ph.D. Thesis*, to appear.

[22] J. G. Kemeny and J. L. Snell. Finite Markov Chains: With a New Appendix "Generalization of a Fundamental Matrix". *Springer*, 1983