Strategy Synthesis for Markov Decision Processes and Branching-Time Logics

Tomáš Brázdil* and Vojtěch Forejt**

Faculty of Informatics, Masaryk University,
Botanická 68a, 60200 Brno,
Czech Republic.
{brazdil,forejt}@fi.muni.cz

Abstract. We consider a class of finite $1\frac{1}{2}$-player games (Markov decision processes) where the winning objectives are specified in the branching-time temporal logic qPECTL* (an extension of the qualitative PCTL*). We study decidability and complexity of existence of a winning strategy in these games. We identify a fragment of qPECTL* called detPECTL* for which the existence of a winning strategy is decidable in exponential time, and also the winning strategy can be computed in exponential time (if it exists). Consequently we show that every formula of qPECTL* can be translated to a formula of detPECTL* (in exponential time) so that the resulting formula is equivalent to the original one over finite Markov chains. From this we obtain that for the whole qPECTL*, the existence of a winning finite-memory strategy is decidable in double exponential time. An immediate consequence is that the existence of a winning finite-memory strategy is decidable for the qualitative fragment of PCTL* in triple exponential time. We also obtain a single exponential upper bound on the same problem for the qualitative PCTL. Finally, we study the power of finite-memory strategies with respect to objectives described in the qualitative PCTL.

1 Introduction

We study $1\frac{1}{2}$-player games (Markov decision processes), which have been applied in various contexts, from computer science and engineering (models of network systems, models of industrial processes, etc.) to biology [13, 9, 12]. A $1\frac{1}{2}$-player game G is a directed graph whose vertices are partitioned into two disjoint sets $V_\Box$ and $V_\bigcirc$. For each vertex of $V_\bigcirc$ there is a fixed probability distribution on outgoing transitions. A *play* is initiated by putting a token on some vertex. This token is then moved from vertex to vertex by one 'real' player $\Box$ and one 'virtual' player $\bigcirc$, who choose their moves in vertices of $V_\Box$ and $V_\bigcirc$, respectively. Player $\bigcirc$ chooses his moves randomly according to the fixed distribution. Player $\Box$ chooses his moves according to a *strategy*. Generally, strategies may depend on history of the play and may be either randomized or deterministic (we denote HR and HD the classes of the history-dependent randomized and deterministic strategies, respectively). In this paper we also consider *finite-memory* strategies that depend on a finite-state information about the history of the play [1]. The classes of randomized and deterministic finite-memory strategies are denoted FR and FD, respectively.

* Supported by "Institute for Theoretical Computer Science (ITI)", project No. 1M0545.
** Supported by the Czech Science Foundation, project No. 102/05/H050.

[1] More formally, a finite-memory (randomized) strategy is represented by a deterministic finite-state automaton and a function which assigns a distribution on outgoing transitions to the current vertex of the play and the state of the automaton after reading the history of the play.

Once player $\Box$ fixes his strategy σ for the game G, we obtain a Markov chain $G(\sigma)$ where the states are finite paths in G, and $ws \xrightarrow{x} wst$ if and only if (s,t) is a transition in G and x is either the fixed probability assigned to (s,t) (if $s \in V_{\bigcirc}$), or the probability of (s,t) assigned by player $\Box$ in ws. Now we may ask whether the resulting Markov chain $G(\sigma)$ satisfies a given property. A *winning objective* is a property of Markov chains to be achieved by player $\Box$. A strategy σ is called *winning* if the Markov chain $G(\sigma)$ satisfies the winning objective.

Winning objectives can be expressed using various formalisms. For example, various kinds of linear-time objectives, such as Büchi, parity, and Rabin objectives, were intensively studied in the past (see, e.g., [7, 8, 6]). In this paper we concentrate on a different kind of winning objectives specified by formulae of a branching-time temporal logic.

Let us note that the semantics of branching-time formulae can be defined directly for $1\frac{1}{2}$-player games (see, e.g., [2]). In that case strategies are chosen separately for each temporal operator occurring in a formula. This approach is different from the one taken in this paper and results on model-checking such games are not related to our results.

The problem of solving games with branching-time winning objectives was for the first time studied in [11], where the existence of a winning memoryless randomized strategy for objectives expressed in PCTL (see, e.g., [10]) was shown to be in **PSPACE**. Results of [11] were substantially extended in [3] where also history dependent strategies were taken into account. The most relevant results of [3] are the following. First, the existence of a winning HD (and also HR, FR and FD) strategy is undecidable for $1\frac{1}{2}$-player games with objectives specified in (quantitative) PCTL. Second, the problem of existence of a winning HD strategy is **EXPTIME**-complete for $1\frac{1}{2}$-player games with objectives specified in the $\mathcal{L}(\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0})$ [2] fragment of the qualitative PCTL.

The question is whether the positive result about the fragment $\mathcal{L}(\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0})$ can be extended to more expressive logics at least for finite-memory strategies. In this paper we address this problem and show that the existence of a winning finite-memory strategy is decidable even for a powerful temporal logic qPECTL*. We also show that the winning finite-memory strategy can always be effectively synthesized. This problem is well motivated because in practice one usually does not only want to know whether a strategy exists but also wants to implement the strategy. Finite-memory strategies have the advantage of being easy to implement.

The logic qPECTL* is the qualitative fragment of the logic PECTL* defined in [5]. PECTL* is a generalization of the logic PCTL* (see, e.g., [5, 2]) which is a probabilistic version of the well-known logic CTL*. Of course, PECTL* contains the logic PCTL. Hence, our results on qPECTL* have immediate consequences for the *qualitative* PCTL* (denoted qPCTL*) and the *qualitative* PCTL (denoted qPCTL).

Our contribution: The main results of this paper are summarized below.

– We show that the existence of a winning FR (or FD) strategy for objectives described by qPCTL, qPECTL*, and qPCTL* formulae is decidable in single exponential, double exponential, and triple exponential time, respectively. We also show that the winning strategy can effectively be computed with the same complexity. Moreover, we show that all these problems can be solved in time polynomial in the size of games.

[2] Formulae of $\mathcal{L}(\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0})$ are built up from literals using conjunction, disjunction, and the temporal operators $\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0}$ (negation is applied only to atomic propositions).

– In the course of the proof of the above results we identify a fragment of qPECTL*, called detPECTL*, and show that the existence of a winning HR (or HD) strategy for objectives described in detPECTL* is decidable in time exponential in the size of formulae and polynomial in the size of games. The fragment detPECTL* contains the logic $\mathcal{L}(\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0})$, and hence our results improve on the corresponding results of [3] by considering a more general logic, randomized strategies, and providing a polynomial time upper bound in the size of games.
– Finally, it has been shown in [3] that an infinite-memory strategy is needed for satisfying a formula of the fragment $\mathcal{L}(\mathrm{G}^{>0}, \mathrm{F}^{>0})$ of qPCTL. We extend this result and provide (in a sense) complete classification of the power of finite-memory strategies for various fragments of qPCTL.

Plan of the paper: In Section 2 we review basic definitions for Markov chains and games. We also introduce the logic qPECTL* and its fragments. In Section 3 we consider the problem of existence of a winning history-dependent strategy for objectives described in detPECTL*. In Section 4 we consider the same problem for finite-memory strategies and qPECTL*. Finally, Section 5 deals with the classification of fragments of qPCTL with respect to the power of finite-memory strategies.

2 Basic Definitions

In this section we introduce basic notions of Markov chains, probabilistic temporal logics, and games. Most definitions (except the definition of qPECTL*) are taken from [3].

We start by recalling basic notions of probability theory. Let A be a finite set. A *probability distribution* on A is a function $f : A \to [0,1]$ such that $\sum_{a \in A} f(a) = 1$. A distribution f is *Dirac* if $f(a) = 1$ for some $a \in A$. The set of all distributions on A is denoted $\mathcal{D}(A)$.

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

2.1 Markov Chains

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

In the rest of this paper we also write $s \xrightarrow{x} t$ instead of $Prob(s \to t) = x$. A *path* in M is a finite or infinite sequence $w = s_0, s_1, \ldots$ of states such that $s_i \to s_{i+1}$ for every i. The *length* of a given path w is the number of transitions in w. We also use $w(i)$ to denote the state s_i of w (by writing $w(i) = s$ we implicitly impose the condition that the length of w is at least i). The prefix $s_0, s_1, \ldots, s_i$ of w is denoted by w^i. A *run* is an infinite path. The sets of all finite paths and all runs of M are denoted $FPath$ and Run, respectively. Similarly, the

sets of all finite paths and runs that start in a given $s \in S$ are denoted $FPath(s)$ and $Run(s)$, respectively.

We say that a set $C \subseteq S$ is a bottom strongly connected component (BSCC) of M if for all $s, t \in C$ there is a path from s to t in M, and whenever there is a path from $s \in C$ to $t \in S$, then $t \in C$. Note that if we restrict the set of states of M to a BSCC C, we obtain a Markov chain.

Each $w \in FPath$ determines a *basic cylinder* $Run(w)$ which consists of all runs that start with w. To every $s \in S$ we associate the probability space $(Run(s), \mathcal{F}, \mathcal{P})$ where $\mathcal{F}$ is the σ-field generated by all basic cylinders $Run(w)$ where w starts with s, and $\mathcal{P} : \mathcal{F} \to [0,1]$ is the unique probability measure such that $\mathcal{P}(Run(w)) = \Pi_{i=0}^{m-1} x_i$ where $w = s_0, \cdots, s_m$ and $s_i \xrightarrow{x_i} s_{i+1}$ for every $0 \leq i < m$ (if m=0, we put $\mathcal{P}(Run(w)) = 1$).

2.2 The Logic qPECTL*

A Büchi automaton is a tuple $\mathcal{B} = (B, \Sigma, \delta, q_I, F)$, where Σ is a finite *alphabet*, B is a finite set of *states*, $\delta \subseteq B \times \Sigma \times B$ is a *transition relation* (we write $q \xrightarrow{a} q'$ instead of $(q, a, q') \in \delta$), q_I is the *initial state*, and $F \subseteq B$ is a set of accepting states. The automaton $\mathcal{B}$ is *deterministic* if for each $q \in B$ and each $a \in \Sigma$, there is at most one $q' \in B$ such that $q \xrightarrow{a} q'$.

The symbol Σ^ω denotes the set of all infinite words over the alphabet Σ. A *computation* of $\mathcal{B}$ on a word $w = w(0)w(1)\cdots \in \Sigma^\omega$ is a sequence $\omega = q_0, q_1, \ldots$ of states of $\mathcal{B}$ such that $q_0 = q_I$ and for all $i \geq 0$ we have $q_i \xrightarrow{w(i)} q_{i+1}$. A computation ω of $\mathcal{B}$ is *accepting* if a state of F occurs infinitely many times in ω. The automaton $\mathcal{B}$ accepts a word $w \in \Sigma^\omega$ if there exists an accepting computation of $\mathcal{B}$ on w. The set of all $w \in \Sigma^\omega$ accepted by $\mathcal{B}$ is denoted $\mathcal{L}(\mathcal{B})$.

The logic qPECTL* has the following syntax:

$$\Phi ::= a \mid \neg a \mid \mathcal{B}^{\sim\varrho}(\Phi_1, \cdots, \Phi_n)$$

Here a ranges over the set Ap of atomic propositions, $\sim\varrho \in \{=1, <1, >0, =0\}$, $n \geq 1$, $\mathcal{B}$ is a Büchi automaton over an alphabet $\Sigma \subseteq 2^{\{1,\ldots,n\}}$, and each Φ_i is a qPECTL* formula.

The semantics of qPECTL* formulae is defined below. Let $M = (S, \to, Prob)$ be a Markov chain and let $\nu : Ap \to 2^S$ be a valuation. We define $s \models^\nu a$ iff $s \in \nu(a)$, and $s \models^\nu \neg a$ iff $s \notin \nu(a)$. The semantics of a qPECTL* formula $\Phi = \mathcal{B}^{\sim\varrho}(\Phi_1, \cdots, \Phi_n)$, where $\mathcal{B}$ is a Büchi automaton with the alphabet $\Sigma \subseteq 2^{\{1,\ldots,n\}}$, is defined as follows: First, we can assume that the semantics of the qPECTL* formulae $\Phi_1, \ldots, \Phi_n$ has already been defined. For every state s of M, let $Run(s, \Phi)$ be the set of all runs $w \in Run(s)$ satisfying the following condition: There is a word $v \in \mathcal{L}(\mathcal{B})$ such that for all $i \geq 0$ and all $k \in v(i)$ holds $w(i) \models^\nu \Phi_k$. We stipulate that $s \models^\nu \Phi$ if and only if $\mathcal{P}(Run(s, \Phi)) \sim \varrho$.

We say that formulae Φ and Ψ are *equivalent* ($\Phi \equiv \Psi$) iff for each state s of an arbitrary Markov chain M and for arbitrary valuation ν holds: $s \models^\nu \Phi$ iff $s \models^\nu \Psi$.

Remark 1. Note that once a formula $\mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n)$, a Markov chain M, and a valuation ν are fixed, we can say that $\mathcal{B}$ (or any automaton with the alphabet $\Sigma \subseteq 2^{\{1,\ldots,n\}}$) accepts a run w of M if there is a word $v \in \mathcal{L}(\mathcal{B})$ such that for all $i \geq 0$ we have $\bigwedge_{k \in v(i)} w(i) \models^\nu \Phi_k$. Then, e.g., $\mathcal{P}(Run(s, \Phi))$ is the probability that $\mathcal{B}$ accepts a run of $Run(s)$. We can also say that the automaton $\mathcal{B}$ goes from a state q_0 to q_{i+1} after reading a finite path $s_0, \ldots, s_i$ in M

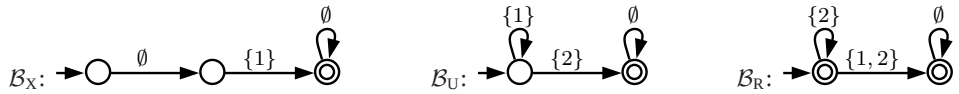
if there is a sequence $q_0, \ldots, q_{i+1}$ of states of $\mathcal{B}$ and a word $X_0, \ldots, X_i$ such that $q_j \xrightarrow{X_j} q_{j+1}$ and $\bigwedge_{k \in X_j} w(j) \models^{\nu} \Phi_k$ for all $0 \leq j \leq i$.

For computational purposes we assume that each formula is represented as a directed acyclic multigraph obtained from the parse tree of the formula by merging similar subtrees. For example, the formula $\mathcal{B}_1^{\sim\varrho}(\mathcal{B}_1^{\sim\varrho}(a,a,a), \mathcal{B}_1^{\sim\varrho}(a,a,a), \mathcal{B}_2^{\sim\varrho}(\mathcal{B}_1^{\sim\varrho}(a,a,a)))$ is represented by a multigraph with four nodes n_1, n_2, n_3, n_4 labeled with $\mathcal{B}_1^{\sim\varrho}, \mathcal{B}_2^{\sim\varrho}, \mathcal{B}_1^{\sim\varrho}, a$, respectively, and transitions: $n_1 \xrightarrow{1,2} n_3$ (here the numbers $1, 2$ stand for the first and the second argument), $n_1 \xrightarrow{3} n_2$, $n_2 \xrightarrow{1} n_3$, $n_3 \xrightarrow{1,2,3} n_4$. Here n_1 corresponds to the whole formula.

Expressing other operators in qPECTL*. The logic qPECTL*, as defined above, is very powerful and succinct, and hence ideal for theoretical considerations. However, it is easier to express complex properties when we have some additional operators. We show that all operators of qPCTL can be expressed in qPECTL*. We define automata $\mathcal{B}_\wedge$, $\mathcal{B}_\vee$ as follows:

It is easy to see that formulae $\mathcal{B}_\vee^{=1}(\Phi_1, \Phi_2)$ and $\mathcal{B}_\wedge^{=1}(\Phi_1, \Phi_2)$ are equivalent to logical disjunction and conjunction, respectively, of Φ_1 and Φ_2. Hence, in what follows we write $\Phi_1 \vee \Phi_2$ and $\Phi_1 \wedge \Phi_2$ instead of $\mathcal{B}_\vee^{=1}(\Phi_1, \Phi_2)$ and $\mathcal{B}_\wedge^{=1}(\Phi_1, \Phi_2)$, respectively.

We also define Büchi automata representing 'next', 'until' and 'release' (the dual of 'until') operators:



We write $\mathrm{X}^{\sim\varrho}\Phi_1$, $\Phi_1\mathrm{U}^{\sim\varrho}\Phi_2$ and $\Phi_1\mathrm{R}^{\sim\varrho}\Phi_2$ instead of $\mathcal{B}_{\mathrm{X}}^{\sim\varrho}(\Phi_1)$, $\mathcal{B}_{\mathrm{U}}^{\sim\varrho}(\Phi_1, \Phi_2)$ and $\mathcal{B}_{\mathrm{R}}^{\sim\varrho}(\Phi_1, \Phi_2)$, respectively. We also define 'future' and 'globally' operators as follows: Let $\mathtt{tt}$ and $\mathtt{ff}$ stand for $a \vee \neg a$ and $a \wedge \neg a$, respectively, for some $a \in Ap$. Let $\mathrm{F}^{\sim\varrho}\Phi$ stands for $\mathtt{tt}\mathrm{U}^{\sim\varrho}\Phi$, and let $\mathrm{G}^{\sim\varrho}\Phi$ stands for $\mathtt{ff}\mathrm{R}^{\sim\varrho}\Phi$.

Given a formula of the form $\mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n)$, we write $\neg\mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n)$ to stand for $\mathcal{B}^{\bowtie\varrho}(\Phi_1, \ldots, \Phi_n)$, where '$\bowtie\varrho$' is '$=1$', '$<1$', '$>0$', or '$=0$', depending on whether '$\bowtie\varrho$' is '<1', '$=1$', '$=0$', or '>0', respectively. This clearly corresponds to the logical operation of negation. Note that $\Phi_1\mathrm{R}^{\sim\varrho}\Phi_2$ is equivalent to $\neg(\neg\Phi_1\mathrm{U}^{\sim\varrho}\neg\Phi_2)$.

Now qPCTL is the fragment of qPECTL* consisting of all formulae of the following form:

$$\Phi ::= a \mid \neg a \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \mathrm{X}^{\sim\varrho}\Phi_1 \mid \Phi_1\mathrm{U}^{\sim\varrho}\Phi_2 \mid \Phi_1\mathrm{R}^{\sim\varrho}\Phi_2$$

Here $\sim\varrho$ ranges over $\{=1, =0, <1, >0\}$.

One can also show that all formulae of qPCTL* (for definition see, e.g., [5]) can be translated to equivalent qPECTL* formulae. This translation employs the algorithm for translating LTL formulae to Büchi automata (see, e.g., [15]) which results in a single exponential blow-up in the size of formulae.

The logic detPECTL*. Now we define the *deterministic* fragment of qPECTL* (called detPECTL*), which generalizes the fragment $\mathcal{L}(\mathrm{F}^{=1}, \mathrm{F}^{>0}, \mathrm{G}^{=1})$ defined in [3] (see also Section 5). This fragment (together with Theorem 6) plays the crucial role in the proof of Theorem 8.

Given an automaton $\mathcal{B}$ with an alphabet $\Sigma \subseteq 2^{\{1,\ldots,n\}}$, we say that a state q of $\mathcal{B}$ is *terminal* iff there is a transition $q \xrightarrow{\emptyset} q$ and no transition of the form $q \xrightarrow{X} q'$ where $q \neq q'$ in $\mathcal{B}$. A formula Φ of qPECTL* is a detPECTL* formula if all subformulae of Φ of the form $\mathcal{B}^{\sim\varrho}(\Phi_1, \cdots, \Phi_n)$ satisfy the following conditions:

1. '$\sim\varrho$' is either '$=1$' or '>0';
2. if '$\sim\varrho$' is '>0', then all accepting states of $\mathcal{B}$ are terminal;
3. All states q of $\mathcal{B}$ satisfy the following condition: For *distinct nonterminal* states q' and q'' such that $q \xrightarrow{A} q'$ and $q \xrightarrow{B} q''$, we have that $\bigwedge_{i \in A \cup B} \Phi_i$ is *not* satisfied in any state of any Markov chain for any valuation (this must hold even for $A = B$).

Observe that $\mathrm{X}^{=1}$, $\mathrm{X}^{>0}$, $\mathrm{U}^{=1}$, $\mathrm{U}^{>0}$, and $\mathrm{R}^{=1}$ are operators of detPECTL*.

2.3 Games and Strategies

A $1\frac{1}{2}$*-player game* (or *Markov decision process*) is a tuple $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ where V is a finite set of *vertices*, $E \subseteq V \times V$ is a set of *transitions*, $(V_\Box, V_\bigcirc)$ is a partition of V, and $Prob$ is a *probability assignment* which to each $v \in V_\bigcirc$ assigns a positive probability distribution on the set of its outgoing transitions. For technical convenience, we assume that each vertex has at least one outgoing transition.

The game is played by a player $\Box$ who selects the moves in the $V_\Box$ vertices, and a "virtual" player $\bigcirc$ who selects the moves in the $V_\bigcirc$ vertices according to the corresponding probability distribution.

A *strategy* for player $\Box$ is a function σ which to each $vs \in V^*V_\Box$ assigns a probability distribution on the set of outgoing transitions of s. We say that a strategy σ is *deterministic* if $\sigma(vs)$ is a Dirac distribution for each $vs \in V^*V_\Box$. Consistently with [1, 11, 3], we use HR and HD to denote the classes of all (history-dependent randomized) strategies and (history-dependent) deterministic strategies, respectively. A special type of strategies are strategies with *finite-memory*, which are formally defined as pairs $(\mathcal{A}, f)$ where $\mathcal{A} = (Q, V, \delta, q_0)$ is a deterministic finite-state automaton over the alphabet V of vertices and f is a function which to each pair $(q, s) \in Q \times V_\Box$ assigns a probability distribution on the set of outgoing transitions of s. The pair $(\mathcal{A}, f)$ determines a unique strategy $\sigma(\mathcal{A}, f)$ such that $\sigma(\mathcal{A}, f)(vs) = f(q, s)$, where $q = \delta(q_0, vs)$. Intuitively, the states of $\mathcal{A}$ represent a finite memory of size $|Q|$ where selected properties of the history of a play are stored. We denote FR and FD the classes of all finite-memory strategies and finite-memory deterministic strategies, respectively.

Each strategy σ for player $\Box$ determines a unique *play* of the game G, which is a Markov chain $G(\sigma)$ where V^+ is the set of states, and $vs \xrightarrow{x} vst$ iff $(s, t) \in E$ and one of the following conditions holds:

- $s \in V_\bigcirc$ and $Prob(s, t) = x$;
- $s \in V_\Box$ and $\sigma(vs)$ assigns x to (s, t).

For every $vs \in V^*V$ we denote $last(vs) = s$. For every run w of $G(\sigma)$ and every $i \geq 0$, we define $w[i] = last(w(i))$ (note that $w(i)$ is a finite sequence of vertices of the game G).

An *objective* is a pair (ν, Φ), where $\nu : Ap \to 2^V$ is a valuation and Φ a qPECTL* formula. Note that each valuation $\nu : Ap \to 2^V$ determines a valuation $\overline{\nu} : Ap \to 2^{V^+}$ defined by $\overline{\nu}(a) = \{vs \in V^*V \mid s \in \nu(a)\}$. For a given objective (ν, Φ), each state of $G(\sigma)$ either does or does not satisfy Φ. A (ν, Φ)-*winning strategy* for player $\Box$ in a vertex $s \in V$ is a strategy σ such that $s \models^\nu \Phi$. We are interested in the following problem:

Synthesis problem: Given a vertex s and an objective (ν, Φ),

is there a (ν, Φ)-winning strategy for player $\Box$ in s ?

Moreover, if the winning strategy exists, then return its finite representation.

Remark 2. Let σ be a finite-memory strategy determined by $(\mathcal{A}, f)$ where $\mathcal{A} = (Q, V, \delta, q_0)$. Observe, that the chain $G(\sigma)$ can be seen as an 'unfolding' of a finite Markov chain. Formally, let $\approx \subseteq V^+ \times V^+$ be an equivalence defined as follows: $u \approx v$ if and only if $\delta(q_0, u) = \delta(q_0, v)$ and $last(u) = last(v)$. Given $v \in V^+$ we denote $[v] = \{u \mid u \approx v\}$, the equivalence class of v, and we denote $V^+/\approx = \{[v] \mid v \in V^+\}$. Let us define a finite Markov chain $\bar{G}(\sigma)$ where $V^+/\approx$ is a set of states, and $[v] \xrightarrow{x} [vs]$ in $\bar{G}(\sigma)$ if and only if $v \xrightarrow{x} vs$ in $G(\sigma)$. Each valuation $\nu : Ap \to 2^V$ determines a valuation $\bar{\nu} : Ap \to 2^{V^+/\approx}$ defined by $\overline{\nu}(a) = \{[vs] \mid s \in \nu(a)\}$. Now, it is easy to verify that for every qPECTL* formula Φ, every valuation ν, and every state v of $G(\sigma)$, we have that $v \models^\nu \Phi$ iff $[v] \models^{\bar{\nu}} \Phi$.

3 The Synthesis Problem for detPECTL*

In this section we prove that the synthesis problem is decidable in exponential time for both HR and HD strategies and detPECTL* objectives. The proof follows similar lines as the analogous proof for HD strategies and the $\mathcal{L}(\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0})$ fragment of qPCTL (see [3]). However, new technical difficulties arise from the use of automata connectives and randomization.

Similarly to [3], we reduce the synthesis problem for detPECTL* to the problem of solving $1\frac{1}{2}$-player games for a different type of objectives (and strategies) defined as follows. Let $G = (V, E, (V_\Box, V_\bigcirc), Prob)$ be a $1\frac{1}{2}$-player game. A *mixed* Büchi objective is a pair (P, O) where $P, O \subseteq V$. A strategy σ is (P, O)-winning in a vertex s iff *all* runs in $G(\sigma)$ initiated in s visit a vertex of P infinitely often, and *almost all* runs initiated in s visit a vertex of O infinitely often. To be able to control randomization in games we introduce a new restriction on strategies defined as follows: Given a set $\Re \subseteq V_\Box$, we say that a (HR) strategy σ is $\Re$-*must* iff for every $v \in V^*$, each $s \in \Re$ and each $(s, t) \in E$ we have $\sigma(vs)(s, t) > 0$, and for each $t \in V_\Box \setminus \Re$ we have that $\sigma(vt)$ is a Dirac distribution. Intuitively, a strategy is $\Re$-must if it always assigns non-zero probability to all successors of vertices of $\Re$ and behaves deterministically in vertices of $V_\Box \setminus \Re$.

Let us fix a game $G = (V, E, (V_\Box, V_\bigcirc), Prob)$, a vertex s_{in} of G, a detPECTL* formula Φ, and a valuation ν. The following lemma allows us to assume that the branching degree of all vertices of G is at most two (we sketch the proof in [4]).

Lemma 3. *There is a $1\frac{1}{2}$-player game $\bar{G}$, a vertex $\bar{s}_{in}$, a formula $\bar{\Phi}$, and a valuation $\bar{\nu}$ (computable in polynomial time), such that each vertex of $\bar{G}$ has at most two successors, and there is a (ν, Φ)-winning strategy in s_{in} iff there is a $(\bar{\nu}, \bar{\Phi})$-winning strategy in $\bar{s}_{in}$.*

Moreover, each (ν, Φ)-winning FR (FD) strategy in s_{in} can be polynomially translated to a $(\bar{\nu}, \bar{\Phi})$-winning FR (FD) strategy in $\bar{s}_{in}$, and vice versa.

We construct a game G', a vertex s'_{in} of G', a mixed Büchi objective (P, O), and a set $\Re$, such that there is a (P, O)-winning $\Re$-must HR strategy in s'_{in} iff there is (ν, Φ)-winning HR strategy in s_{in}. The size of G' will be single exponential in the size of Φ and polynomial in the size of G.

To simplify our presentation we introduce some additional notation. We say that a Büchi automaton $\mathcal{B}$ *corresponds* to a formula Ψ if and only if Ψ is of the form $\mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n)$ (for some $\sim\varrho$ and $\Phi_1, \ldots, \Phi_n$). For technical convenience, we assume that each Büchi automaton corresponds to at most one subformula of Φ and that all automata occurring in Φ have pairwise disjoint sets of states. Let $States$ denote the set of all states of all automata occurring in Φ, and let $States^{>0}$ and $States^{=1}$ denote sets of states of all automata that correspond to subformulae of Φ of the form $\mathcal{B}^{>0}(\Phi_1, \ldots, \Phi_n)$ and $\mathcal{B}^{=1}(\Phi_1, \ldots, \Phi_n)$, respectively. By $L(s)$ we denote the set of all literals (i.e., atomic propositions and negated atomic propositions) satisfied in the vertex s.

Now we present a formal definition of the game G'. An intuition behind the definition is given below.

Formal definition of G'. We define $G' = (V', E', (V'_\Box, V'_\bigcirc), Prob')$ where the set V' consists of vertices of the following three forms:

- f-vertices are of the form $(s, A)^f$, where $s \in V$ and $A \subseteq States \times \{\circ, \star\}$.
- g-vertices are of the form $(s, \mathcal{D})^g$, where

$$\mathcal{D} \subseteq \{(t, B) \mid (s, t) \in E, B \subseteq States \times \{\circ, \star\}\}$$

  is a non-empty set, for each t there is at most one pair of the form (t, B) in $\mathcal{D}$, and if $s \in V_\bigcirc$ and $(s, t) \in E$, then $\mathcal{D}$ contains a pair of the form (t, B).
- distinguished vertices s'_{in} and $dead$.

To $V'_\bigcirc$ we put all g-vertices whose first component belongs to $V_\bigcirc$, and we put $V'_\Box = V' \setminus V'_\bigcirc$. To formally define E' we need some additional notation. Given a formula of the form $\Psi = \mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n)$, we denote $Rep(\Psi)$ the tuple $(q_0, \star)$ where q_0 is the initial state of $\mathcal{B}$. Given a literal Ψ, we define $Rep(\Psi) = \Psi$. Given a transition $q \xrightarrow{X} q'$ of an automaton corresponding to a formula of the form $\mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n)$, we denote $Start(q \xrightarrow{X} q')$ the set of all $Rep(\Phi_i)$ where $i \in X$. The set of transitions E' is defined as follows:

- $((s, A)^f, (s, \mathcal{D})^g) \in E'$ for *all* $(s, A)^f$ and $(s, \mathcal{D})^g$ satisfying the following: for *every* $(q, x) \in A$ there *exists* (q', x') satisfying $q \xrightarrow{X} q'$ and $Start(q \xrightarrow{X} q') \subseteq A \cup L(s)$ and

$$x' = \begin{cases} \star & \text{if } q' \text{ is accepting;} \\ \circ & \text{if } q' \in States^{=1} \text{ is not accepting and } A \cap (States^{=1} \times \{\circ\}) = \emptyset; \\ \circ & \text{if } q' \in States^{>0} \text{ is not accepting and } A \cap (States^{>0} \times \{\circ\}) = \emptyset; \\ x & \text{otherwise.} \end{cases}$$

  and either $(q', x') \in \bigcap_{(t,B)\in\mathcal{D}} B$ or $(q', x') \in \bigcup_{(t,B)\in\mathcal{D}} B$, depending on whether $q \in States^{=1}$ or $q \in States^{>0}$, respectively.

– $((s, A)^f, dead) \in E'$ for all f-vertices, and $(dead, dead) \in E'$;
– $((s, \mathcal{D})^g, (t, B)^f) \in E'$ for all $(t, B) \in \mathcal{D}$;
– $(s'_{in}, (s_{in}, A)^f) \in E'$ for all $A \subseteq States \times \{\circ, \star\}$ satisfying $Rep(\Phi) \in A$ (here we assume that Φ is not a literal, because otherwise the synthesis problem is trivially solved)

We define $Prob'((s, \mathcal{D})^g, (t, B)^f) = Prob(s, t)$ whenever $s \in V_{\bigcirc}$ and $(t, B) \in \mathcal{D}$.

Let P be the set of all vertices of the form $(s, A)^f$ such that A does not contain any pair of the form $(q, \circ)$ where $q \in States^{>0}$. Let O be the set of all vertices of the form $(s, A)^f$ such that A does not contain any pair of the form $(q, \circ)$ where $q \in States^{=1}$.

Let $\Re$ be the set of all g-vertices of the form $(s, \mathcal{D})^g$ where $s \in V_{\Box}$.

Intuition behind G'. The game G' simulates the game G (in the first component of vertices) and at the same time maintains some information about subformulae of Φ (the second component of vertices). Each step of the simulated play of G corresponds to two steps in G'. The first step, going from the current f-vertex to a g-vertex, updates the information about Φ. The next one, going from the g-vertex to an f-vertex, simulates a move in G.

While playing the game G', player $\Box$ simulates a play of the game G and at the same time simulates computations of Büchi automata corresponding to subformulae of Φ, verifying that these subformulae are satisfied in appropriate places in the simulated play. Given an f-vertex $(s, A)^f$, every pair $(q, x) \in A$ represents a running instance of an automaton which is in the state q. (Here x maintains the information whether this particular instance recently entered an accepting state.)

Going from the f-vertex $(s, A)^f$ to a g-vertex $(s, \{(t_1, B_1), (t_2, B_2)\})^g$, player $\Box$ simulates one computational step for each running instance $(q, x) \in A$. More concretely, let $(q, x) \in A$ where q is a state of an automaton corresponding to $\mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n)$. Player $\Box$ chooses a transition $q \xrightarrow{X} q'$ such that $Start(q \xrightarrow{X} q') \subseteq A \cup L(s)$, which intuitively means that the running instances of automata corresponding to formulae of $\{\Phi_i \mid i \in X\}$ have already been initiated in $(s, A)^f$. Then (q', x') (here x' is an appropriate update of x) is put either to both B_1, B_2 or to at least one of them, depending on whether '$\sim\varrho$' is either '$=1$' or '>0', respectively. Note that in the case '$=1$', the simulated computation goes to the same state q' for both successors. To ensure correctness of the simulation, we need Φ to be a detPECTL* formula (intuitively this means that there is at most one 'correct' non-terminal successor q' of q after reading the current state).

Note that the definition of x' and the winning objective (P, O) ensure that *all* running instances of automata corresponding to formulae of the form $\mathcal{B}^{=1}(\ldots)$ are almost surely accepting (i.e., enter an accepting state infinitely many times), and that *all* running instances of automata corresponding to formulae of the form $\mathcal{B}^{>0}(\ldots)$ are surely accepting (i.e., enter a terminal accepting state). Note that the sets B_1 and B_2 may, in addition to the obligatory contents, contain arbitrary pairs from $States \times \{\circ, \star\}$. This may be used by player $\Box$ to initiate new running instances needed in the next simulation step to perform transitions of Büchi automata (see above).

Finally, going from the g-vertex $(s, \{(t_1, B_1), (t_2, B_2)\})^g$ to an f-vertex, player $\Box$ randomly chooses one of the successors $(t_1, B_1)^f, (t_2, B_2)^f$, by which he chooses a successor in the simulated play. The $\Re$-must restriction ensures that each of the successors is chosen with non-zero probability, which prevents player $\Box$ from erasing pairs of $States^{>0} \times \{\circ, \star\}$.

The following lemma is proved in [4].

Lemma 4. *There is a (ν, Φ)-winning HR strategy in s_{in} if and only if there is a (P, O)-winning $\Re$-must HR strategy in s'_{in}. Moreover, each (P, O)-winning $\Re$-must FR strategy $(\mathcal{A}, f)$ in s'_{in} induces a (ν, Φ)-winning FR strategy in s_{in} computable in time polynomial in the size of $(\mathcal{A}, f)$.*

It has been shown in [3] that the existence of a winning HD strategy in $1\frac{1}{2}$-player games with mixed Büchi objectives is decidable in polynomial time, and moreover, that the existence of a winning HD strategy in such games implies the existence of a winning FD strategy computable in polynomial time. By a slight modification of the proof from [3] we obtain the following analogy for $\Re$-must HR strategies:

Lemma 5. *The existence of a winning $\Re$-must HR strategy in $1\frac{1}{2}$-player games with mixed Büchi objectives is decidable in polynomial time. Moreover, in these games, the existence of a winning $\Re$-must HR strategy implies the existence of a winning $\Re$-must FR strategy computable in polynomial time.*

Applying Lemma 4 and Lemma 5 we obtain the following theorem.

Theorem 6. *The existence of a winning HR (HD) strategy in $1\frac{1}{2}$-player games with detPECTL* objectives is decidable in time exponential in the size of formulae and polynomial in the size of games. Moreover, in these games, the existence of a winning HR (HD) strategy implies the existence of a winning FR (FD) strategy computable in time exponential in the size of formulae and polynomial in the size of games.*

Proof (Sketch). For HR strategies, the result follows immediately from Lemma 4 and Lemma 5. For HD strategies, it suffices to slightly modify the construction of the game G' by erasing all g-vertices $(s, \mathcal{D})^g$ such that $s \in V_\Box$ and $|\mathcal{D}| > 1$. Now for $\Re = \emptyset$, each $\Re$-must strategy in s'_{in} is deterministic. An inspection of the proof of Lemma 4 reveals that the lemma remains valid even for deterministic strategies. Now using Lemma 5 we obtain the desired result.

4 The Synthesis Problem for qPECTL* and Finite-Memory Strategies

In this section we show how to solve the synthesis problem for qPECTL* and finite-memory strategies. We show that, in fact, the logic qPECTL* and its fragment detPECTL* are expressively equivalent over finite Markov chains, and then obtain the solution to the synthesis problem as an immediate corollary of our previous results. To formally capture this equivalence, we write $\Phi \equiv_{fin} \Psi$ whenever for arbitrary state s of arbitrary *finite* Markov chain and arbitrary valuation ν, holds $s \models^\nu \Phi$ iff $s \models^\nu \Psi$. The main aim of this section is formalized by the following theorem.

Theorem 7. *For every qPECTL* formula Φ there is a detPECTL* formula Ψ, computable in exponential time, such that $\Phi \equiv_{fin} \Psi$. Moreover, if Φ is a qPCTL formula, then Ψ is computable in polynomial time.*

An immediate corollary of Theorem 7, Theorem 6, and Remark 2 is the following

Theorem 8. *For both FR and FD strategies, the synthesis problem for qPCTL, qPECTL*, and qPCTL* objectives can be solved in single exponential, double exponential, and triple exponential time, respectively. Moreover, in all these cases, the synthesis problem can be solved in time polynomial in the size of games.*

The rest of this section is devoted to the proof of Theorem 7. Let us fix a qPECTL* formula Φ. First, observe that we may assume (w.l.o.g.) that for each subformula of Φ of the form $\mathcal{B}^{\sim\varrho}(\Phi_1,\ldots,\Phi_n)$, every state s of an arbitrary Markov chain and an arbitrary valuation ν, there is *exactly one* letter A in the alphabet of $\mathcal{B}$ such that $s \models^\nu \bigwedge_{i\in A}\Phi_i$. Indeed, if Φ does not have this property, it suffices to substitute each subformula of Φ of the form $\mathcal{B}^{\sim\varrho}(\Phi_1,\ldots,\Phi_n)$ with a formula of the form $\bar{\mathcal{B}}^{\sim\varrho}(\Phi_1,\ldots,\Phi_n,\neg\Phi_1,\ldots,\neg\Phi_n)$, where $\bar{\mathcal{B}}$ is obtained from $\mathcal{B}$ by substituting each transition of the form $q \xrightarrow{A} q'$ with all transitions of the form $q \xrightarrow{A'\cup A''} q'$ where $A \subseteq A' \subseteq \{1,\ldots,n\}$ and $A'' = \{m+n | m \in \{1,\ldots,n\}\backslash A'\}$, and consequently making the transition function total. Observe that although the alphabet of $\bar{\mathcal{B}}$ may be exponentially larger than the alphabet of $\mathcal{B}$, the number of states increases only by 1 due to making the transition function total. Observe also that the size of the (graph representing) resulting formula is polynomial in the size of Φ.

Now, to determinize the formula Φ, it suffices to determinize (syntactically) transition relations of all Büchi automata in Φ. However, the problem is that deterministic Büchi automata are strictly weaker than non-deterministic Büchi automata. We solve this problem by first translating the Büchi automata to equivalent deterministic Rabin automata (using results of [14]) and then encoding the deterministic Rabin automata to detPECTL* formulae.

First, let us formally define the notion of deterministic Rabin automata. A *deterministic Rabin automaton* $\mathcal{R}$ is a tuple $(Q,\Sigma,\gamma,q_0,Acc)$, where Q is a finite set of states, Σ is an input alphabet, $\gamma : Q\times\Sigma\to Q$ is a transition function, q_0 is an initial state, and $Acc = \{(C_1,D_1),\ldots,(C_k,D_k)\}$, where $C_1,\ldots,C_k,D_1,\ldots,D_k \subseteq Q$, specifies the acceptance condition. A word $w\in\Sigma^\omega$ is accepted by $\mathcal{R}$ iff there is a sequence $\omega = q_0,q_1,\ldots$ of states of $\mathcal{R}$ such that for all $i\geq 0$ we have $\gamma(q_i,w(i)) = q_{i+1}$, and there is $j\in\{1,\ldots,k\}$ such that some state of C_j occurs infinitely often in ω and no state of D_j occurs infinitely often in ω. We denote $\mathcal{L}(\mathcal{R})$ the set of all words accepted by $\mathcal{R}$. The following proposition was proved in [14].

Theorem 9 ([14]). *Given a Büchi automaton $\mathcal{B} = (B,\Sigma,\delta,q_I,F)$ there is an effectively computable deterministic Rabin automaton $\mathcal{R} = (Q,\Sigma,\gamma,q_0,Acc)$ such that $\mathcal{L}(\mathcal{R}) = \mathcal{L}(\mathcal{B})$, $|R| = 2^{\mathcal{O}(|B|\log|B|)}$ and $|Acc| = \mathcal{O}(|B|)$.*

Now let $\mathcal{B}^{\sim\varrho}(\Phi_1,\ldots,\Phi_n)$ be a subformula of Φ and let us assume that $\Phi_1,\ldots,\Phi_n$ are already detPECTL* formulae. Let us denote Σ the alphabet of $\mathcal{B}$, and let us fix a Rabin automaton $\mathcal{R} = (Q,\Sigma,\gamma,q_0,Acc)$, where $Acc = \{(C_1,D_1),\ldots,(C_k,D_k)\}$, such that $\mathcal{L}(\mathcal{R}) = \mathcal{L}(\mathcal{B})$. Let us assume (w.l.o.g.) that the transition function of $\mathcal{R}$ is total.

Let us first assume that '$\sim\varrho$' is either of the form '>0' or '$=1$'. Based on $\mathcal{R}$, we define deterministic Büchi automata $\mathcal{B}_{fin}$ and $\mathcal{B}_{q,i}$, for all $q\in Q$ and $1\leq i\leq k$, such that

$$\mathcal{B}^{\sim\varrho}(\Phi_1,\ldots,\Phi_n) \equiv_{fin} \mathcal{B}^{\sim\varrho}_{fin}(\Phi_1,\ldots,\Phi_n,\Psi_1,\ldots,\Psi_\ell)$$

where each Ψ_j is of the form $\mathcal{B}^{=1}_{q,i}(\Phi_1,\ldots,\Phi_n)$ for one of the automata $\mathcal{B}_{q,i}$ (i.e., $\ell = |Q|\cdot k$). In what follows we denote $index(q,i)$ the number j such that Ψ_j is the formula $\mathcal{B}^{=1}_{q,i}(\Phi_1,\ldots,\Phi_n)$.

Before we formally define $\mathcal{B}_{fin}$ and $\mathcal{B}_{q,i}$, let us explain the intuition behind the definition. Let us fix a state s_0 of a finite Markov chain M and a valuation ν, and let us assume that $\mathcal{B}$ (and hence also $\mathcal{R}$) accepts a run of $Run(s_0)$ (see Remark 1) with a probability greater than 0 (the explanation is analogous for the probability $=1$). Because M is finite, there is a finite path $v\in FPath(s_0)$ such that $\mathcal{R}$ accepts almost all runs of $Run(v)$. Here, however, using

basic results of the theory of finite Markov chains, one can say even more. There is a finite path $v \in FPath(s_0)$ such that almost all $w \in Run(v)$ satisfy the following condition: the automaton $\mathcal{R}$, after reading the prefix v of w, enters a state of C_j infinitely often and no state of D_j at all, for a suitable j. We define $\mathcal{B}_{fin}$ and $\mathcal{B}_{q,i}$ so that $\mathcal{B}^{>0}_{fin}(\Phi_1, \ldots, \Phi_n, \Psi_1, \ldots, \Psi_\ell)$ expresses precisely this property.

The automata $\mathcal{B}_{fin}$ and $\mathcal{B}_{q,i}$ are formally defined as follows. Let $\mathcal{B}_{fin} = (Q \cup \{q_a\}, \Sigma \cup T, \delta_{fin}, q_0, \{q_a\})$, where $T = \{\{n+1\}, \ldots, \{n+\ell\}\}$, and transitions of $\mathcal{B}$ are defined as follows:

- $q \xrightarrow{A} q'$ for all $A \in \Sigma$ and $q, q' \in Q$ such that $\gamma(q, A) = q'$;
- $q \xrightarrow{\{n+index(q,i)\}} q_a$ for all $q \in Q$ and all $1 \leq i \leq k$;
- $q_a \xrightarrow{\emptyset} q_a$;
- nothing else is a transition.

We define $\mathcal{B}_{q,i} = (Q, \Sigma, \delta_{q,i}, q, C_i)$ where transitions are defined as follows: for all $q \in Q$ and all $A \in \Sigma$, we define $q \xrightarrow{A} q'$ if and only if $q, q' \notin D_i$ and $\gamma(q, A) = q'$ (i.e., there are no transitions leaving or entering states of D_i).

Lemma 10. *If '$\sim\varrho$' is either of the form '>0' or '$=1$', then*

$$\mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n) \equiv_{fin} \mathcal{B}^{\sim\varrho}_{fin}(\Phi_1, \ldots, \Phi_n, \Psi_1, \ldots, \Psi_\ell)$$

Moreover, the right hand side formula is in detPECTL*.

Proof (Sketch). The fact that $\mathcal{B}^{\sim\varrho}_{fin}(\Phi_1, \ldots, \Phi_n, \Psi_1, \ldots, \Psi_\ell)$ is a detPECTL* formula follows immediately from our assumption about Φ and from the fact that the automata $\mathcal{B}_{fin}$ and $\mathcal{B}_{q,i}$ are obtained from the deterministic automaton $\mathcal{R}$ either by deleting transitions or by adding transitions to the newly added terminal state q_a.

It remains to prove the equivalence. Let us fix a finite Markov chain M with the set of states S, a state s_0 of M, and a valuation ν. Observe that for every state s of M there is exactly one $A_s \in \Sigma$ such that $s \models \bigwedge_{i \in A_s} \Phi_i$. Now an automaton with the alphabet Σ accepts a run $s_0, s_1, \ldots$ of M if it accepts the word $A_{s_0} A_{s_1} \cdots$.

First, let us assume that $s_0 \models \mathcal{B}^{\sim\varrho}_{fin}(\Phi_1, \ldots, \Phi_n, \Psi_1, \ldots, \Psi_\ell)$. It follows immediately from the definitions that, with probability $\sim\varrho$, there is a path $s_0, s_1, \ldots, s_i, s_{i+1}$ in M satisfying the following conditions:

- the automaton $\mathcal{B}_{fin}$ (and hence also the automaton $\mathcal{R}$) moves from its initial state to a state r after reading the word $A_{s_0} \cdots A_{s_i}$;
- $s_{i+1} \models \mathcal{B}^{=1}_{r,j}(\Phi_1, \ldots, \Phi_n)$ for some $1 \leq j \leq k$ (and hence almost all runs of $Run(s_{i+1})$ are accepted by the Rabin automaton $\mathcal{R}$ initiated in r).

However, this immediately implies that, with probability $\sim\varrho$, the automaton $\mathcal{R}$ (and hence the automaton $\mathcal{B}$) accepts a run of $Run(s_0)$.

For the opposite direction, let $M \times \mathcal{R}$ be a Markov chain (the synchronous product of M and $\mathcal{R}$) whose set of states is $S \times Q$ and transitions are defined as follows: $(s, q) \xrightarrow{x} (t, r)$ iff $s \xrightarrow{x} t$ and $q \xrightarrow{A_s} r$. Given $1 \leq j \leq k$, we say that a BSCC C of M is j-accepting iff a state of C_j occurs in (the second component of a state of) C and no state of D_j occurs in C. Basic results of the theory of Markov chains imply that the probability measure of all runs of $Run(s)$ accepted by $\mathcal{R}$ (hence also by $\mathcal{B}$) is equal to the probability of reaching some j-accepting

BSCC of $M \times \mathcal{R}$. Thus, if $s_0 \models \mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n)$, then, with probability $\sim\varrho$, there is a path $(s_0, q_0), \ldots, (s_i, q_i)$ in $M \times \mathcal{R}$ such that (s_i, q_i) belongs to a j-accepting BSCC. It is easy to show that $s_i \models \mathcal{B}^{=1}_{q_i,j}(\Phi_1, \ldots, \Phi_n)$ and $\mathcal{B}_{fin}$ moves from q_0 to q_a after reading the word $A_{s_0}, \ldots, A_{s_{i-1}}, \{n + index(q_i, j)\}$. This implies that $s_0 \models \mathcal{B}^{\sim\varrho}_{fin}(\Phi_1, \ldots, \Phi_n, \Psi_1, \ldots, \Psi_\ell)$. □

Now, let us consider the case where '$\sim\varrho$' is either of the form '$=0$' or '<1'. We denote $\widehat{\sim\varrho}$ the 'dual' of '$\sim\varrho$', i.e., $\widehat{=0}$ is '$=1$', and $\widehat{<1}$ is '>0'. Using very similar arguments, we prove the following analogy of Lemma 10 (a proof can be found in [4]).

Lemma 11. *If '$\sim\varrho$' is either of the form '$=0$' or '<1', then there are Büchi automata $\mathcal{B}_{fin}$ and $\mathcal{B}_{q,i}$, for all $q \in Q$ and $1 \leq i \leq k$, such that*

$$\mathcal{B}^{\sim\varrho}(\Phi_1, \ldots, \Phi_n) \equiv_{fin} \mathcal{B}^{\widehat{\sim\varrho}}_{fin}(\Phi_1, \ldots, \Phi_n, \Psi_1, \ldots, \Psi_\ell)$$

where each Ψ_j is of the form $\mathcal{B}^{=1}_{q,i}(\Phi_1, \ldots, \Phi_n)$ for one of the automata $\mathcal{B}_{q,i}$ (i.e., $\ell = |Q| \cdot k$). Moreover, the right hand side formula is in detPECTL*.

Using Lemma 10 and Lemma 11 one can easily design an algorithm which transforms the formula Φ to a detPECTL* formula Ψ using appropriate substitutions in a 'bottom-up' manner.

For general qPECTL* formulae, the time complexity of the algorithm is single exponential in the size of Φ. Indeed, by [14] the Rabin automaton $\mathcal{R}$ can be computed in time exponential in the number of states of $\mathcal{B}$ and polynomial in the size of the alphabet, and consequently the automata $\mathcal{B}_{fin}$ and $\mathcal{B}_{q,i}$ are computable in single exponential time. It follows that the formula Ψ is computable in exponential time (here we make use of the representation of Ψ as a directed acyclic graph, i.e., we assume that several occurrences of the same subformula are represented by one vertex).

Now observe that in the case of PCTL formulae, there are only five distinct Büchi automata used to define Boolean connectives and operators X, U, and R. It follows that the corresponding Rabin automata have bounded size, and thus each PCTL formula can be translated to a detPECTL* formula in polynomial time. This finishes the proof of Theorem 7.

5 qPCTL and Finite-Memory Strategies

In this section we study the power of finite-memory strategies w.r.t. the synthesis problem for $1\frac{1}{2}$-player games and qPCTL objectives.

Given $\mathcal{Y} \subseteq \{\mathrm{X}^{\sim\varrho}, \mathrm{F}^{\sim\varrho}, \mathrm{G}^{\sim\varrho} \mid {\sim\varrho} \in \{=0, >0, <1, =1\}\}$, we denote $\mathcal{L}(\mathcal{Y})$ the fragment of qPCTL which consists of formulae of the following form:

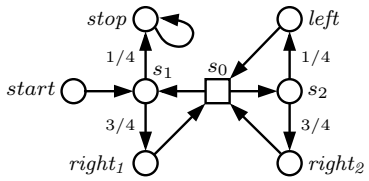
$$\Phi ::= a \mid \neg a \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid \mathrm{Y}^{\sim\varrho}\, \Phi_1$$

where $\mathrm{Y}^{\sim\varrho} \in \mathcal{Y}$. For example, the fragment $\mathcal{L}(\{\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0}\})$ (we usually omit the set brackets and write $\mathcal{L}(\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0})$) is the fragment of qPCTL whose formulae are built up from literals using conjunction, disjunction, and three temporal operators $\mathrm{F}^{=1}, \mathrm{G}^{=1}, \mathrm{F}^{>0}$ (there is no operation of complement). Note that we work with the operators $\mathrm{F}^{\sim\varrho}$ and $\mathrm{G}^{\sim\varrho}$ only for simplicity: *All* results of this section remain valid even if one replaces $\mathrm{F}^{\sim\varrho}$ with $\mathrm{U}^{\sim\varrho}$, and $\mathrm{G}^{\sim\varrho}$ with $\mathrm{R}^{\sim\varrho}$.

Definition 12. *A fragment $\mathcal{L}(\mathcal{Y})$ is* finitely determined *if for every formula Φ of $\mathcal{L}(\mathcal{Y})$, arbitrary vertex s of a $1\frac{1}{2}$-player game, and arbitrary valuation ν, the following holds: If there is a (ν, Φ)-winning strategy in s, then there is a (ν, Φ)-winning* finite-memory *strategy in s.*

First, we show which fragments are *not* finitely determined. Then we prove that no finitely determined fragment is more expressive than the fragment $\mathcal{L}(\mathrm{X}^{=1}, \mathrm{X}^{>0}, \mathrm{G}^{=1}, \mathrm{F}^{=1}, \mathrm{F}^{>0})$.

It has been shown in [3] that $\mathcal{L}(\mathrm{G}^{>0}, \mathrm{F}^{>0})$ is not finitely determined. We extend this result and show that also $\mathcal{L}(\mathrm{G}^{>0})$ is not finitely determined. Let us consider a game G depicted in the following figure (it is very similar to the corresponding game from [3]):



We use names of vertices as atomic propositions with an obvious semantics. Let us denote $\Phi = \mathrm{G}^{>0}(\neg stop \wedge (\neg left \vee \mathrm{G}^{>0}\neg right_2))$. The proof of the following lemma is presented in [4].

Lemma 13. *There is a (ν, Φ)-winning HD strategy in $start$. There is* no *(ν, Φ)-winning FR strategy in $start$.*

We continue by proving that also $\mathcal{L}(\mathrm{G}^{=0})$ is not finitely determined. Let us consider a formula $\Psi = \mathrm{G}^{>0}(\neg stop \wedge (\neg left \vee \mathrm{F}^{=1}\mathrm{G}^{>0}\neg right_2))$ and the game G defined above. It is easy to show, using arguments similar to the proof of Lemma 13, that there is a (ν, Ψ)-winning HD strategy in $start$, and no (ν, Ψ)-winning FR strategy in $start$. Now we transform Ψ to a $\mathcal{L}(\mathrm{G}^{=0})$ formula Ψ' such that for an arbitrary strategy we have $start \models^{\nu} \Psi$ iff $start \models^{\nu} \Psi'$. Using obvious equivalences $\neg\mathrm{G}^{>0}\phi \equiv \mathrm{G}^{=0}\phi$ and $\mathrm{F}^{=1}\phi \equiv \mathrm{G}^{=0}\neg\phi$, one can easily show that Ψ is equivalent to $\neg\chi$ where $\chi = \mathrm{G}^{=0}(\neg stop \wedge (\neg left \vee \mathrm{G}^{=0}\mathrm{G}^{=0}\neg right_2))$. Now it is easy to verify that $start \models^{\nu} \neg\chi$ if and only if $start \models^{\nu} \mathrm{G}^{=0}(\neg start \vee \chi)$. It follows that $\mathcal{L}(\mathrm{G}^{=0})$ is not finitely determined.

Next, let us consider the fragment $\mathcal{L}(\mathrm{F}^{<1})$. Using the equivalence $\mathrm{G}^{>0}\phi \equiv \mathrm{F}^{<1}\neg\phi$, one can easily show that Ψ is equivalent to $\mathrm{F}^{<1}(stop \vee (left \wedge \mathrm{F}^{<1}\mathrm{F}^{<1} right_2))$. It follows that $\mathcal{L}(\mathrm{F}^{<1})$ is not finitely determined.

The last fragments we analyze are $\mathcal{L}(\mathrm{X}^{=0}, \mathrm{F}^{=1})$, $\mathcal{L}(\mathrm{X}^{<1}, \mathrm{F}^{=1})$, $\mathcal{L}(\mathrm{F}^{=0}, \mathrm{F}^{=1})$ and $\mathcal{L}(\mathrm{G}^{<1}, \mathrm{F}^{=1})$. Using the equivalences $\mathrm{F}^{=1}\phi \equiv \mathrm{G}^{=0}\neg\phi$ and $\neg\mathrm{F}^{=1}\phi \equiv \mathrm{G}^{>0}\neg\phi$ one can show that $\Phi = \mathrm{G}^{>0}(\neg stop \wedge (\neg left \vee \mathrm{G}^{>0}\neg right_2))$ is equivalent to $\neg\chi$ where $\chi = \mathrm{F}^{=1}(stop \vee (left \wedge \mathrm{F}^{=1} right_2))$. Now, using a similar trick as above, we obtain $start \models^{\nu} \neg\chi$ iff $start \models^{\nu} \mathrm{X}^{=0}(\chi)$ iff $start \models^{\nu} \mathrm{X}^{<1}(\chi)$ iff $start \models^{\nu} \mathrm{F}^{=0}(start \wedge \chi)$ iff $start \models^{\nu} \mathrm{G}^{<1}(\neg start \vee \chi)$.

Now we can give a complete classification of finitely determined fragments.

Lemma 14. *The fragment $\mathcal{L}_1 = \mathcal{L}(\mathrm{X}^{=1}, \mathrm{X}^{<1}, \mathrm{X}^{>0}, \mathrm{X}^{=0}, \mathrm{G}^{=1}, \mathrm{F}^{>0}, \mathrm{F}^{=0}, \mathrm{G}^{<1})$ and the fragment $\mathcal{L}_2 = \mathcal{L}(\mathrm{X}^{=1}, \mathrm{X}^{>0}, \mathrm{G}^{=1}, \mathrm{F}^{>0}, \mathrm{F}^{=1})$ are maximal (w.r.t. inclusion) finitely determined fragments.*

Proof. We have proved above that the fragments $\mathcal{L}(\mathrm{G}^{>0})$, $\mathcal{L}(\mathrm{G}^{=0})$, $\mathcal{L}(\mathrm{F}^{<1})$, $\mathcal{L}(\mathrm{X}^{=0}, \mathrm{F}^{=1})$, $\mathcal{L}(\mathrm{X}^{<1}, \mathrm{F}^{=1})$, $\mathcal{L}(\mathrm{F}^{=0}, \mathrm{F}^{=1})$ and $\mathcal{L}(\mathrm{G}^{<1}, \mathrm{F}^{=1})$ are not finitely determined. Clearly, any fragment which contains one of these fragments is not finitely determined. On the other hand, it

follows from Theorem 6 that the fragment $\mathcal{L}_2$ *is* finitely determined. By a close inspection of various possibilities, we obtain that the only fragment we have not yet classified is $\mathcal{L}_1$ (and some of its subsets).

However, we show that each formula of $\mathcal{L}_1$ can efficiently be translated to $\mathcal{L}_2$, which implies that $\mathcal{L}_1$ is finitely determined. Let Ψ be a formula of $\mathcal{L}_1$. First, using the equivalences $\mathrm{X}^{=0}\Phi \equiv \mathrm{X}^{=1}\neg\Phi$, $\mathrm{X}^{<1}\Phi \equiv \mathrm{X}^{>0}\neg\Phi$, $\mathrm{F}^{=0}\Phi \equiv \mathrm{G}^{=1}\neg\Phi$, $\mathrm{G}^{<1}\Phi \equiv \mathrm{F}^{>0}\neg\Phi$, one can remove operators $\mathrm{X}^{=0}$, $\mathrm{X}^{<1}$, $\mathrm{F}^{=0}$, $\mathrm{G}^{<1}$ (introducing, however, some negations to the formula). The negations introduced in the previous step can be pushed to atomic propositions using equivalences $\neg\mathrm{X}^{=1}\phi \equiv \mathrm{X}^{>0}\neg\phi$, $\neg\mathrm{X}^{>0}\phi \equiv \mathrm{X}^{=1}\neg\phi$, $\neg\mathrm{G}^{=1}\phi \equiv \mathrm{F}^{>0}\neg\phi$, $\neg\mathrm{G}^{>0}\phi \equiv \mathrm{F}^{=1}\neg\phi$. □

Corollary 15. *A fragment $\mathcal{L}(\mathcal{Y})$, where $\mathcal{Y} \subseteq \{\mathrm{X}^{\sim\varrho}, \mathrm{F}^{\sim\varrho}, \mathrm{G}^{\sim\varrho} \mid {\sim\varrho} \in \{=0, >0, <1, =1\}\}$, is finitely determined if and only if each formula of $\mathcal{L}(\mathcal{Y})$ can be efficiently translated to an equivalent formula of $\mathcal{L}(\mathrm{X}^{=1}, \mathrm{X}^{>0}, \mathrm{G}^{=1}, \mathrm{F}^{=1}, \mathrm{F}^{>0})$.*

References

1. C. Baier, M. Größer, M. Leucker, B. Bollig, and F. Ciesinski. Controller synthesis for probabilistic systems. In *Proceedings of IFIP TCS'2004*. Kluwer, 2004.
2. A. Bianco and L. de Alfaro. Model checking of probabilistic and nondeterministic systems. In *Proceedings of FST&TCS'95*, vol. 1026 of *LNCS*, pp. 499–513. Springer, 1995.
3. T. Brázdil, V. Brožek, V. Forejt, and A. Kučera. Stochastic games with branching-time winning objectives. In *Proceedings of LICS 2006*. IEEE, 2006.
4. T. Brázdil and V. Forejt. Strategy synthesis for Markov decision processes and branching-time logics. Technical report FIMU-RS-2007-03, 2007.
5. T. Brázdil, A. Kučera, and O. Stražovský. On the decidability of temporal properties of probabilistic pushdown automata. In *Proceedings of STACS'2005*, vol. 3404 of *LNCS*, pp. 145–157. Springer, 2005.
6. K. Chatterjee, L. de Alfaro, and T. Henzinger. The complexity of stochastic Rabin and Streett games. In *Proceedings of ICALP 2005*, vol. 3580 of *LNCS*, pp. 878–890. Springer, 2005.
7. K. Chatterjee, M. Jurdzinski, and T. Henzinger. Simple stochastic parity games. In *Proceedings of CSL'93*, vol. 832 of *LNCS*, pp. 100–113. Springer, 1994.
8. K. Chatterjee, M. Jurdzinski, and T. Henzinger. Quantitative stochastic parity games. In *Proceedings of SODA 2004*, pp. 121–130. SIAM, 2004.
9. E. Feinberg and A. Shwartz, editors. *Handbook of Markov Decision Processes*. Kluwer, 2002.
10. H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
11. A. Kučera and O. Stražovský. On the controller synthesis for finite-state Markov decision processes. In *Proceedings of FST&TCS 2005*, vol. 3821 of *LNCS*, pp. 541–552. Springer, 2005.
12. S. Mahadevan. Partially observable semi-Markov decision processes: Theory and applications in engineering and cognitive science. In *AAAI: Fall Symposium on Planning with Partially Observable Markov Decision Processes*, 1998.
13. M.L. Puterman. *Markov Decision Processes*. Wiley, 1994.
14. S. Safra. *Complexity of automata on infinite objects*. PhD thesis, 1989.
15. M. Y. Vardi. An automata-theoretic approach to linear temporal logic. In *Banff Higher Order Workshop*, pp. 238–266, 1995.