



FI MU

Faculty of Informatics
Masaryk University Brno

Stochastic Games with Branching-Time Winning Objectives

by

Tomáš Brázdil
Václav Brožek
Vojtěch Forejt
Antonín Kučera

FI MU Report Series

FIMU-RS-2006-02

Copyright © 2006, FI MU

September 2006

**Copyright © 2006, Faculty of Informatics, Masaryk University.
All rights reserved.**

**Reproduction of all or part of this work
is permitted for educational or research use
on condition that this copyright notice is
included in any copy.**

**Publications in the FI MU Report Series are in general accessible
via WWW:**

`http://www.fi.muni.cz/reports/`

Further information can be obtained by contacting:

**Faculty of Informatics
Masaryk University
Botanická 68a
602 00 Brno
Czech Republic**

Stochastic Games with Branching-Time Winning Objectives*

Tomáš Brázdil Václav Brožek Vojtěch Forejt Antonín Kučera

Faculty of Informatics, Masaryk University,
Botanická 68a, 60200 Brno,
Czech Republic.

{brazdil,xbrozek,xforejt,kucera}@fi.muni.cz

Abstract

We consider stochastic turn-based games where the winning objectives are given by formulae of the branching-time logic PCTL. These games are generally not determined and winning strategies may require memory and/or randomization. Our main results concern history-dependent strategies. In particular, we show that the problem whether there exists a history-dependent winning strategy in $1\frac{1}{2}$ -player games is *highly undecidable*, even for objectives formulated in the $\mathcal{L}(F^{5/8}, F^=1, F^{>0}, G^=1)$ fragment of PCTL. On the other hand, we show that the problem becomes decidable (and in fact **EXPTIME**-complete) for the $\mathcal{L}(F^=1, F^{>0}, G^=1)$ fragment of PCTL, where winning strategies require only finite memory. This result is tight in the sense that winning strategies for $\mathcal{L}(F^=1, F^{>0}, G^=1, G^{>0})$ objectives may already require infinite memory.

1 Introduction

In this paper we consider stochastic turn-based games where the winning objectives are given by formulae of the branching-time logic PCTL. Formally, a $2\frac{1}{2}$ -player game G is a finite directed graph where the vertices are partitioned into three subsets V_{\square} , V_{\diamond} , V_{\circ} . A *play* is initiated by putting a token on some vertex. The token is then moved from

*The work is supported by the research center Institute for Theoretical Computer Science (ITI), project No. 1M0545.

vertex to vertex by two players, \square and \diamond , who are responsible for selecting outgoing transitions in the vertices of V_\square and V_\diamond , respectively. In the vertices of V_\circ , outgoing transitions are chosen randomly according to a fixed probability distribution. A *strategy* specifies how a player should play. In general, a strategy may or may not depend on the history of a play (we say that a strategy is *history-dependent (H)* or *memoryless (M)*), and the transitions may be chosen deterministically or randomly (*deterministic (D)* and *randomized (R)* strategies). In the case of randomized strategies, a player chooses a probability distribution on the set of outgoing transitions. Note that deterministic strategies can be seen as restricted randomized strategies, where one of the outgoing transitions has probability 1. Each pair of strategies (σ, π) for players \square and \diamond determines a unique Markov chain $G(\sigma, \pi)$ where the states are finite paths in G , and $wu \rightarrow wuu'$ with probability x iff (u, u') is a transition in the game and x is the probability chosen by player \square or \diamond (when $u \in V_\square$ or $u \in V_\diamond$, respectively), or the fixed probability of the transition (u, u') when $u \in V_\circ$. A *winning objective* for player \square is some property of Markov chains that is to be achieved. A *winning strategy* for player \square is a strategy σ such that for every strategy π of player \diamond the Markov chain $G(\sigma, \pi)$ has the desired property. Usually, the aim of player \diamond is to *falsify* this property, which means that his winning objective is dual. A winning strategy for player \diamond is a strategy π such that $G(\sigma, \pi)$ does *not* have the property for any strategy σ of player \square . A game is *determined* if one of the two players has a winning strategy in every vertex. $1\frac{1}{2}$ -player games are “restricted” $2\frac{1}{2}$ -player games where $V_\diamond = \emptyset$. All of the above introduced notions (except for determinacy) are applicable also to $1\frac{1}{2}$ -player games.

Infinite games have been studied in various fields of mathematics and computer science (recently written overviews are, e.g., [16, 5]). For example, model-checking problems for certain temporal logics (such as the modal μ -calculus) can be naturally reformulated as the questions to determine the winner in parity games, and a lot of research effort has been invested into this problem. Our work is mainly motivated by applications of games in system design, where systems are modeled as games, player \square corresponds to a “controller” which determines the system behaviour in a subset of controllable states, player \diamond models the environment, and the winning objectives for player \square correspond to the desired property of the system. The task is to find a controller (a strategy σ for player \square) such that the desired property holds no matter what the environment does (i.e., the strategy σ is winning). As for stochastic games, the majority of existing results concern games with *linear time* winning objectives which are specified by some property of runs in Markov chains. Examples include quantitative reachability objectives (the probability of all runs that hit a “good” state is at least ϱ), qualitative Büchi objectives (the probability of all runs along which a “good” vertex ap-

pears infinitely often is 1), qualitative/quantitative parity objectives [6, 7], Rabin and Street objectives [4], etc. In this paper we study *branching-time* objectives that are formalized as formulae of the branching-time probabilistic logic PCTL.

Previous and related work. In [1], it is shown that winning strategies for PCTL objectives may require memory and/or randomization in general. Hence, the MD, MR, HD, and HR strategies (see above) need to be considered separately. It is also proven that the problem whether there exists a winning MD strategy in a given $1\frac{1}{2}$ -player game for a given PCTL objective is **NP**-complete. MR strategies were considered in [14], where it is shown that the existence of a winning MR strategy in a given $2\frac{1}{2}$ -player game for a given PCTL objective is in **EXPTIME**. The construction also yields **PSPACE** upper bound for $1\frac{1}{2}$ -player games.

To prevent misunderstanding, we should say that the logic PCTL can also be interpreted directly on games (or Markov decision processes). The decidability of the model-checking problem for stochastic games and PCTL was established in [9] as a simple consequence of the results about quantitative ω -regular games. However, this is a different problem which is not directly related to the subject of this paper (as we shall, the results about stochastic games with branching-time winning objectives are quite different from the results about model-checking).

Main results. We start by observing that stochastic games with branching-time objectives are not determined, even if the objectives are formulae of the $\mathcal{L}(F^{=1}, F^{>0})$ fragment of PCTL (in general, $\mathcal{L}(Y_1, \dots, Y_n)$ denotes the fragment of PCTL containing the connectives Y_1, \dots, Y_n , conjunction, and disjunction (negation can be applied only to atomic propositions)). As a warm-up, we present some simple results about memory-less strategies in Section 3.1. We show that the problem whether player \square has a winning MD strategy in a given $2\frac{1}{2}$ -player game for a given PCTL objective is $\Sigma_2 = \mathbf{NP}^{\mathbf{NP}}$ complete. The Σ_2 lower bound holds even for the $\mathcal{L}(F^{=1}, F^{>0})$ fragment of PCTL. Since the existence of a winning MD strategy for player \square in $1\frac{1}{2}$ -player games with PCTL objectives is **NP**-complete [1], we yield a full complexity classification for MD strategies. The lower complexity bounds carry over to MR strategies and hold even for *qualitative* PCTL objectives for which we give the matching upper bounds—we show that the existence of a winning MR strategy for player \square in $1\frac{1}{2}$ -player (or $2\frac{1}{2}$ -player) games with *qualitative* PCTL objectives is **NP**-complete (or $\Sigma_2 = \mathbf{NP}^{\mathbf{NP}}$ complete, resp.). Let us note that randomized strategies are strictly more powerful than deterministic ones even for qualitative objectives (a simple example is given in Section 3.1). The existence of a winning MR strategy for player \square in $1\frac{1}{2}$ -player and $2\frac{1}{2}$ -player games with general PCTL objectives is known to be in **PSPACE** and **EXPTIME**, respectively [14]. We did not manage to lift the **NP** and Σ_2 lower bounds, and we also failed to improve the mentioned

upper bounds. On the other hand, there is some indication that lowering the bounds below **PSPACE** would be quite difficult. We use the same argument as Etessami & Yannakakis in [11], where it is shown that the SQUARE-ROOT-SUM problem is efficiently reducible to the quantitative reachability problem for one-exit recursive Markov chains. An instance of SQUARE-ROOT-SUM is a tuple (a_1, \dots, a_n, b) of integers. The question is whether $\sum_{i=1}^n \sqrt{a_i} \leq b$. This problem is known to be in **PSPACE**, but its exact complexity is a long-standing open problem in computational geometry. Hence, an efficient reduction of SQUARE-ROOT-SUM to another problem $P \in \mathbf{PSPACE}$ can be seen as an indication that the complexity of P is hard to improve. We show that SQUARE-ROOT-SUM is efficiently reducible to the problem whether player \square has a winning MR strategy in $1\frac{1}{2}$ -player games with PCTL objectives. Let us note that the technique used in the proof is different from the one of [11].

The main results of this paper concern history-dependent strategies. First, we answer the open question formulated in [1] by showing that the existence of a winning HD (or HR) strategy in $1\frac{1}{2}$ -player games is *highly undecidable* even for objectives of the $\mathcal{L}(F^{=5/8}, F^=1, F^{>0}, G^=1)$ fragment of PCTL. More precisely, we show that the above problem is complete for the Σ_1^1 level of the analytical hierarchy. This is already a deep result relying on specific tricks which were developed to encode and simulate a computation of a given nondeterministic Minsky machine. A slight modification of the proof reveals that the existence of a winning HD (or HR) strategy with *finite memory* in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^{=5/8}, F^=1, F^{>0}, G^=1)$ objectives is also undecidable (and complete for the Σ_1 level of the arithmetical hierarchy). The role of the quantitative $F^{=5/8}$ operator is very important in these undecidability results¹. In general, qualitative questions tend to be easier than quantitative ones (this also holds for PCTL and certain classes of infinite-state Markov chains [10, 3, 2]; note that the plays determined by history-dependent strategies are infinite-state Markov chains). Hence, we turn our attention to *qualitative* PCTL objectives. We start by examining the fragments with qualitative forms of reachability and safety connectives, i.e., $F^{\bowtie q}$ and $G^{\bowtie q}$, where $\bowtie \in \{=, >, <\}$ and $q \in \{0, 1\}$. Even in this simplified setting, the results are not uniform and different combinations of connectives lead to quite different results. First, we show that the role of $F^{=5/8}$ operator in the aforementioned undecidability proof is *provably* crucial in the sense that the existence of a winning HD strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^=1, F^{>0}, G^=1)$ objectives is **EXPTIME** complete. Let us note that

- the **EXPTIME** upper bound is proven in two phases. First, we show that the existence of a winning HD strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^=1, F^{>0}, G^=1)$ objectives

¹Let us note that $5/8$ is not some kind of “magic number”, it is just technically convenient. In principle, any operator of the form $F^{=x}$ where $0 < x < 1$ would suffice for our purposes.

is effectively reducible to the existence of a winning HD strategy in $1\frac{1}{2}$ -player games with *mixed linear-time* objectives, which are essentially conjunctions of one *qualitative-Büchi* and one *sure-Büchi* objective. This reduction is exponential. Then, we show that the existence of a winning HD strategy in $1\frac{1}{2}$ -player games with mixed linear-time objectives is in **P**. Note that if we had a conjunction of two qualitative-Büchi or two sure-Büchi objectives, we could simply apply known results. To the best of our knowledge, the games where the winning objectives are “mixtures” of stochastic and non-stochastic requirements have not yet been explicitly considered (perhaps due to the lack of motivation). The solution we provide is not trivial.

- The **EXPTIME** lower bound holds even for $\mathcal{L}(F^=1, G^=1)$ objectives and for both HD and HR strategies.

Our construction also reveals that a winning strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^=1, F^{>0}, G^=1)$ objectives needs only a finite memory whose size is exponential in the size of a given objective. This result does *not* hold for $\mathcal{L}(F^=1, F^{>0}, G^=1, G^{>0})$ objectives—we show that even $\mathcal{L}(F^{>0}, G^{>0})$ objectives require infinite memory in general. In this sense, the previous result is tight.

Many interesting questions remain open. For example, it is not clear whether the existence of a winning strategy in $1\frac{1}{2}$ -player games with qualitative PCTL objectives is decidable or not (all we know is that these strategies may require infinite memory). Another question is whether some of our positive results can be extended to $2\frac{1}{2}$ -player games and/or to *concurrent* stochastic games with branching-time winning objectives. Our knowledge about randomized strategies is also limited, we have not addressed the issue of fairness, and so on. These problems are left for future research.

2 Basic Definitions

We start by recalling basic notions of probability theory. Let A be a finite set. A *probability distribution* on A is a function $f : A \rightarrow [0, 1]$ such that $\sum_{a \in A} f(a) = 1$. A distribution f is *rational* if $f(a) \in \mathbb{Q}$ for every $a \in A$, and *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 measurable space (X, \mathcal{F}) is a function $\mathcal{P} : \mathcal{F} \rightarrow \mathbb{R}^{\geq 0}$ 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 *probabilis-*

tic 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}) .

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

In the rest of this paper we also write $s \xrightarrow{x} t$ instead of $Prob(s \rightarrow t) = x$. A *path* in \mathcal{T} is a finite or infinite sequence $w = s_0, s_1, \dots$ of states such that $s_i \rightarrow s_{i+1}$ for every i . We also use $w(i)$ to denote the state s_i of w (by writing $w(i) = s$ we implicitly impose the condition that the length of w is at least $i + 1$). The prefix s_0, s_1, \dots, 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 \mathcal{T} 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.

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

The logic PCTL. The logic PCTL, the probabilistic extension of CTL, was introduced by Hansson & Jonsson in [12]. 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 equation:

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

Here $p \in Ap$, $q \in [0, 1]$, 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.

$$\begin{aligned} s \models^{\nu} p & \quad \text{iff } s \in \nu(p) \\ s \models^{\nu} \neg p & \quad \text{iff } s \notin \nu(p) \\ s \models^{\nu} \Phi_1 \vee \Phi_2 & \quad \text{iff } s \models^{\nu} \Phi_1 \text{ or } s \models^{\nu} \Phi_2 \\ s \models^{\nu} \Phi_1 \wedge \Phi_2 & \quad \text{iff } s \models^{\nu} \Phi_1 \text{ and } s \models^{\nu} \Phi_2 \\ s \models^{\nu} X^{\bowtie q} \Phi & \quad \text{iff } \mathcal{P}(\{w \in Run(s) \mid w(1) \models^{\nu} \Phi\}) \bowtie q \\ s \models^{\nu} \Phi_1 U^{\bowtie q} \Phi_2 & \quad \text{iff } \mathcal{P}(\{w \in Run(s) \mid \exists j \geq 0 : w(j) \models^{\nu} \Phi_2 \text{ and } w(i) \models^{\nu} \Phi_1 \text{ for all } 0 \leq i < j\}) \bowtie q \end{aligned}$$

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 $\bar{\bowtie}$ be the complement of \bowtie (for

example, if \bowtie is \leq , then $\overline{\bowtie}$ is $>$). The negation of $X^{\bowtie\ell}\Phi$ and $\Phi_1 U^{\bowtie\ell} \Phi_2$ then corresponds to $X^{\overline{\bowtie}\ell}\Phi$ and $\Phi_1 U^{\overline{\bowtie}\ell} \Phi_2$, respectively. The $F^{\bowtie\ell}$ and $G^{\bowtie\ell}$ operators are defined in the standard way: $F^{\bowtie\ell}\Phi$ stands for $\text{tt } U^{\bowtie\ell} \Phi$, and $G^{\bowtie\ell}\Phi$ stands for $\text{tt } U^{\widehat{\bowtie}1-\ell} \neg\Phi$, where $\widehat{\bowtie}$ is $<$, $>$, \leq , \geq , $=$, or \neq , depending on whether \bowtie is $>$, $<$, \geq , \leq , $=$, or \neq , respectively.

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, $aU^{<1}b \vee F^{>0}c$ is a qualitative PCTL formula. In this paper we also consider fragments with unary reachability and safety connectives. Formally, for each tuple Y_1, \dots, Y_n , where each Y_i is of the form $X^{\bowtie\ell}$, $F^{\bowtie\ell}$, or $G^{\bowtie\ell}$, we define the $\mathcal{L}(Y_1, \dots, Y_n)$ fragment of PCTL:

$$\Phi ::= p \mid \neg p \mid \Phi_1 \vee \Phi_2 \mid \Phi_1 \wedge \Phi_2 \mid Y_1 \Phi \mid \dots \mid Y_n \Phi$$

For example, $F^{>0}(b \vee G^{\geq 0.43}(\neg c \wedge F^{<0.5}d))$ is a formula of $\mathcal{L}(F^{>0}, G^{\geq 0.43}, F^{<0.5})$. Sometimes we also use formulae of the form $p \Rightarrow \Phi$ which stand for $\neg p \vee \Phi$.

Games, strategies, and objectives. A $2\frac{1}{2}$ -player game is a tuple $G = (V, E, (V_{\square}, V_{\diamond}, V_{\circ}), \text{Prob})$ where V is a finite set of *vertices*, $E \subseteq V \times V$ is the set of *transitions*, $(V_{\square}, V_{\diamond}, V_{\circ})$ is a partition of V , and Prob is a *probability assignment* which to each $v \in V_{\circ}$ assigns a rational 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 two players, \square and \diamond , who move a single token from vertex to vertex along the transitions of E . Player \square selects the moves in the V_{\square} vertices, and player \diamond selects the moves in the V_{\diamond} vertices. Transitions in the V_{\circ} vertices are chosen randomly according to the corresponding probability distribution. Game graphs are drawn in the standard way; vertices of V_{\square} , V_{\diamond} , and V_{\circ} are depicted as squares, diamonds, and circles, respectively. Probability distributions are usually uniform, which is indicated by arcs connecting the outgoing transitions of V_{\circ} vertices. A *strategy* for player \square is a function σ which to each $wv \in V^*V_{\square}$ assigns a probability distribution on the set of outgoing transitions of v . We say that a strategy σ is *memoryless* (M) if $\sigma(wv)$ depends just on the last vertex v , and *deterministic* (D) if $\sigma(wv)$ is a Dirac distribution for each $wv \in V^*V_{\square}$. Consistently with [1, 14], strategies that are not necessarily memoryless are called *history-dependent* (H), and strategies that are not necessarily deterministic are called *randomized* (R). A special type of history-dependent 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 $(q, v) \in Q \times V_{\square}$ assigns a probability distribution on the set of outgoing transitions of v . The pair (\mathcal{A}, f) determines a unique strategy $\sigma(\mathcal{A}, f)$ such that $\sigma(\mathcal{A}, f)(wv) = f(q, v)$, where $q = \delta(q_0, wv)$. Intuitively, the

states of \mathcal{A} represent a finite memory of size $|Q|$ where selected properties of the history of a play are stored. Hence, we can define the following four classes of strategies: MD, MR, HD, and HR, where $\text{MD} \subseteq \text{HD} \subseteq \text{HR}$ and $\text{MD} \subseteq \text{MR} \subseteq \text{HR}$, but MR and HD are incomparable. Strategies for player \diamond are defined analogously. Each pair (σ, π) of strategies for player \square and \diamond determines a unique *play* of the game G , which is a Markov chain $G(\sigma, \pi)$ where V^+ is the set of states, and $wu \xrightarrow{x} wuu'$ iff $(u, u') \in E$ and one of the following conditions holds:

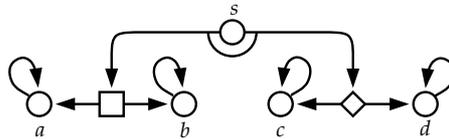
- $u \in V_{\circlearrowleft}$ and $\text{Prob}(u, u') = x$;
- $u \in V_{\square}$ and $\sigma(wu)$ assigns x to (u, u') ;
- $u \in V_{\diamond}$ and $\pi(wu)$ assigns x to (u, u') .

An *objective* is a pair (ν, φ) , where $\nu : Ap \rightarrow 2^V$ is a valuation and φ a PCTL formula. Note that each valuation $\nu : Ap \rightarrow 2^V$ determines a valuation $\bar{\nu} : Ap \rightarrow 2^{V^+}$ defined by $\bar{\nu}(a) = \{wu \in V^+ \mid u \in \nu(a)\}$. For a given objective (ν, φ) , each state of $G(\sigma, \pi)$ either does or does not satisfy φ . A (ν, φ) -*winning strategy* for player \square in a vertex $v \in V$ is a strategy σ such that for every strategy π of player \diamond we have that $v \models^{\nu} \varphi$. Similarly, a (ν, φ) -*winning strategy* for player \diamond in a vertex $v \in V$ is a strategy π such that for every strategy σ of player \square we have that $v \not\models^{\nu} \varphi$. The game G is (ν, φ) -*determined* if there is a (ν, φ) -*winning strategy* for one of the two players in every vertex v of G .

$1\frac{1}{2}$ -*player games* are $2\frac{1}{2}$ -player games where the set V_{\diamond} is empty. Formally, a $1\frac{1}{2}$ -player game is a tuple $G = (V, E, (V_{\square}, V_{\circlearrowleft}), \text{Prob})$ where all elements have the expected meaning.

3 The Results

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



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

3.1 Memoryless Strategies

In [1], it is shown that the problem whether there exists a winning MD strategy in a given $1\frac{1}{2}$ -player game for a given PCTL objective is **NP**-complete. In fact, the **NP** lower bound holds even for the $\mathcal{L}(F^{=1})$ fragment of PCTL. The following theorem gives a complexity classification for $2\frac{1}{2}$ -player games.

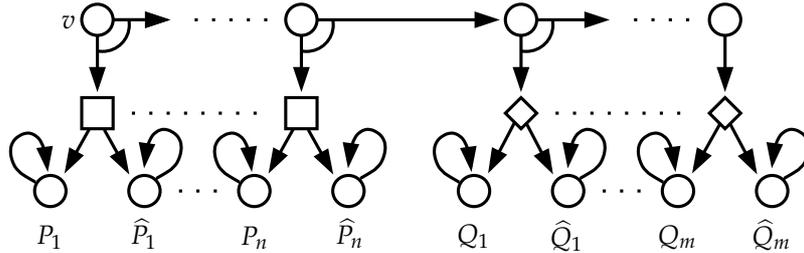
Theorem 3.1. *The existence of a winning MD strategy for player \square in $2\frac{1}{2}$ -player games with PCTL objectives is $\Sigma_2 = \mathbf{NP}^{\mathbf{NP}}$ complete. The Σ_2 lower bound holds even for $\mathcal{L}(F^{=1}, F^{>0})$ objectives and for both MD and MR strategies.*

Proof. A Σ_2 formula is a formula of the form

$$\exists x_1, \dots, x_n \forall y_1, \dots, y_m B$$

where $n, m \in \mathbb{N}$ and B is a \wedge, \vee -expression over the (possibly negated) variables $x_1, \dots, x_n, y_1, \dots, y_m$. The problem whether a given Σ_2 formula is valid is Σ_2 -complete [15].

Let $\psi \equiv \exists x_1, \dots, x_n \forall y_1, \dots, y_m B$. We construct a $2\frac{1}{2}$ -player game $G(\psi)$, a valuation ν , and a formula $\varphi \in \mathcal{L}(F^{=1}, F^{>0})$ such that player \square has a (ν, φ) -winning MD (or MR) strategy in a distinguished vertex v of $G(\psi)$ iff ψ is valid. Let us fix two sets $P = \{p_i, \hat{p}_i \mid 1 \leq i \leq n\}$ and $Q = \{q_j, \hat{q}_j \mid 1 \leq j \leq m\}$ of fresh atomic propositions, and let $P_i = P \setminus \{p_i\}$, $\hat{P}_i = P \setminus \{\hat{p}_i\}$, $Q_j = P \cup \{q_j\}$, $\hat{Q}_j = P \cup \{\hat{q}_j\}$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$. The structure of $G(\psi)$ together with the valuation ν are shown in the following figure:



Let

$$\varphi \equiv \left(\bigvee_{j=1}^m (F^{>0} q_j \wedge F^{>0} \hat{q}_j) \right) \vee \left(\hat{B} \wedge \bigwedge_{i=1}^n (F^{=1} p_i \vee F^{=1} \hat{p}_i) \right)$$

where \hat{B} is the formula obtained from B by substituting each occurrence of x_i , $\neg x_i$, y_j , and $\neg y_j$ with $F^{=1} p_i$, $F^{=1} \hat{p}_i$, $F^{>0} q_j$, and $F^{>0} \hat{q}_j$, respectively. Intuitively, player \square chooses an assignment for the variables x_1, \dots, x_n (x_i is set to true or false by selecting the transition to a vertex satisfying p_i or \hat{p}_i , resp.). Note that player \square cannot use randomized moves because then the formula $F^{=1} p_i \vee F^{=1} \hat{p}_i$ would not hold. Similarly, player \diamond

chooses an assignment for y_1, \dots, y_m . Observe that player \diamond cannot use randomized moves either because this would make some $F^{>0}q_j \wedge F^{>0}\hat{q}_j$ true. Now it is easy to check that ψ is valid iff player \square has a (ν, φ) -winning MD (or MR) strategy in the vertex v . This establishes the Σ_2 lower bound.

The Σ_2 upper bound holds for all PCTL objectives. First, let us note that the model-checking problem for PCTL formulae and Markov chains is in **P** [12]. Hence, it suffices to “guess” a winning strategy σ for player \square , and then ask the **NP** oracle whether there is a strategy π of player \diamond such that $G(\sigma, \pi)$ does *not* satisfy a given objective. The answer of the oracle is then simply negated. \square

The complexity classification for MD strategies is thus established. As for MR strategies, the **NP** and Σ_2 lower bounds still hold. However, we managed to provide the matching upper bounds only for the subclass of *qualitative* PCTL objectives. Note that randomized strategies are more powerful than deterministic ones even for qualitative objectives—consider the formula $X^{>0}p_u \wedge X^{>0}p_v$ and a simple game G with three vertices $t, u, v \in V_\square$ where $t \rightarrow u$, $t \rightarrow v$, $u \rightarrow u$, and $v \rightarrow v$. The propositions p_u and p_v hold only in u and v , respectively. Obviously, there is no winning (ν, φ) -winning MD strategy, but there are many (ν, φ) -winning MR strategies.

Theorem 3.2. *The existence of a winning MR strategy for player \square in $1\frac{1}{2}$ -player (or $2\frac{1}{2}$ -player) games with qualitative PCTL objectives is **NP**-complete (or $\Sigma_2 = \mathbf{NP}^{\mathbf{NP}}$ complete, resp.).*

Proof. A straightforward induction on the structure of a qualitative PCTL formula φ shows that the (in)validity of φ does not depend on the exact values of transition probabilities. It only matters which of the transition have zero/positive probability. Hence, in the case of $1\frac{1}{2}$ -player games, it suffices to “guess” the subset of outgoing transitions in each vertex of V_\square which should have positive probability, and then verify that the guess was correct by a (polynomial time) PCTL model-checking algorithm [12]. The Σ_2 upper bound for $2\frac{1}{2}$ -player games is established analogously (see the proof of Theorem 3.1). \square

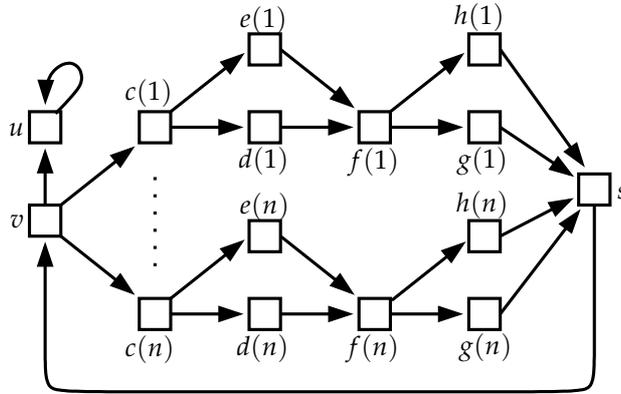
The existence of a winning MR strategy for player \square in $1\frac{1}{2}$ -player and $2\frac{1}{2}$ -player games with general PCTL objectives is known to be in **PSPACE** and **EXPTIME**, respectively [14]. We did not manage to lift the **NP** and Σ_2 lower bounds, and we also failed to improve the mentioned upper bounds. At least, we provide some evidence that lowering these bounds below **PSPACE** is difficult (see the discussion in Section 1). As a byproduct of this construction, we obtain an example of a $1\frac{1}{2}$ -player game (where $V_\circ = \emptyset$) and an objective (ν, φ) where $\varphi \in \mathcal{L}(X^{>0}, U^{=1/2})$ such that the only (ν, φ) -winning MR strategy assigns irrational probabilities to transitions.

Theorem 3.3. *The SQUARE-ROOT-SUM problem is efficiently reducible to the problem whether player \square has a winning MR strategy in $1\frac{1}{2}$ -player games with PCTL objectives.*

Proof. Let a_1, \dots, a_n, b be an instance of SQUARE-ROOT-SUM (see Section 1). Let G be a game where

- the set $V = V_\square$ contains the vertices $v, u, s, c(i), d(i), e(i), f(i), g(i)$, and $h(i)$ for all $1 \leq i \leq n$;
- the set of transitions contains $v \rightarrow u, u \rightarrow u, s \rightarrow u, v \rightarrow c(i), c(i) \rightarrow d(i) \rightarrow f(i), c(i) \rightarrow e(i) \rightarrow f(i), f(i) \rightarrow g(i) \rightarrow s$, and $f(i) \rightarrow h(i) \rightarrow s$ for all $1 \leq i \leq n$.

The structure of G is shown in the following figure:



We assume that for each vertex $t \in V$ there is an atomic proposition p_t which is valid only in t (thus we obtain our valuation v). Slightly abusing notation, we write t instead of p_t in our formulae.

Every strategy σ for player \square assigns (some) probabilities $p(c_i), p(e_i)$, and $p(h_i)$ to transitions $v \rightarrow c(i), c(i) \rightarrow e(i)$, and $f(i) \rightarrow h(i)$, respectively, where $1 \leq i \leq n$. Let $q = b + \sum_{i=1}^n a_i$. We construct a PCTL formula φ such that every (v, φ) -winning MR strategy in v has to assign $p(c_i) = p(e_i) = p(h_i) = \sqrt{a_i}/q$ for every $1 \leq i \leq n$. Then the probability of $v \rightarrow u$ must be $1 - \sum_{i=1}^n \sqrt{a_i}/q$. The formula φ contains the clause $X^{\geq 1-b/q}u$. Hence, player \square has a (v, φ) -winning MR strategy in v iff $1 - \sum_{i=1}^n \sqrt{a_i}/q \geq 1 - b/q$, i.e., iff $\sum_{i=1}^n \sqrt{a_i} \leq b$.

Now we describe the formula φ in greater detail. For every $1 \leq i \leq n$, let $\Phi_i \equiv (v \vee c(i)) \cup^{a_i/q^2} e(i)$. Note that $v \models^v \Phi_i$ iff $p(c_i) \cdot p(e_i) = a_i/q^2$. Similarly, we construct the formulae Ψ_i and Ξ_i such that $v \models^v \Psi_i$ and $v \models^v \Xi_i$ iff $p(e_i) \cdot p(h_i) = a_i/q^2$ and $p(h_i) \cdot p(c_i) = a_i/q^2$, respectively:

$$\begin{aligned} \Psi_i &\equiv X^{>0}(c(i) \vee (e(i) \vee f(i)) \cup^{a_i/q^2} h(i)) \\ \Xi_i &\equiv X^{>0}X^{>0}X^{>0}((f(i) \vee h(i) \vee s \vee v) \cup^{a_i/q^2} c(i)) \end{aligned}$$

Observe that if $p(c_i) \cdot p(e_i) = p(e_i) \cdot p(h_i) = p(h_i) \cdot p(c_i) = a_1/q^2$, then necessarily $p(c_i) = p(e_i) = p(h_i) = \sqrt{a_i}/q$. We put $\varphi \equiv X^{\geq 1-b/q}u \wedge \bigwedge_{i=1}^n (\Phi_i \wedge \Psi_i \wedge \Xi_i)$.

Let us consider the game obtained for $n = 1, a_1 = 2$, and $b = 0$. Then $\Phi_1 \wedge \Psi_1 \wedge \Xi_1 \in \mathcal{L}(X^{>0}, U^{1/2})$ and the only $(v, \Phi_1 \wedge \Psi_1 \wedge \Xi_1)$ -winning MR strategy in v assigns irrational probabilities to certain transitions. Thus, we obtain the example promised above. \square

3.2 History-Dependent Strategies

The results presented in this section constitute the main contribution of our paper. We start with the negative ones.

Theorem 3.4. *The existence of a winning HD (or HR) strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^{=5/8}, F^{=1}, F^{>0}, G^{=1})$ objectives is undecidable (and Σ_1^1 -hard).*

Proof. The result is obtained by reduction of the problem whether a given nondeterministic Minsky machine with two counters initialized to zero has an infinite computation such that the initial instruction is executed infinitely often (this problem is known to be Σ_1^1 -complete [13]). Formally, a nondeterministic Minsky machine with two counters c_1, c_2 is a finite sequence \mathcal{M} of numbered instructions $1:ins_1, \dots, n:ins_n$, where each ins_i is of one of the following forms (where $j \in \{1, 2\}$):

- $c_j := c_j + 1; goto\ k$
- *if* $c_j = 0$ *then* $goto\ k$ *else* $c_j := c_j - 1; goto\ m$
- $goto\ \{k\ or\ m\}$

Here the indexes k, m range over $\{1, \dots, n\}$. We can safely assume that

- $ins_i \neq ins_j$ for $i \neq j$ (in the rest of this proof we do not strictly distinguish between instructions and their corresponding indexes);
- ins_i does not contain the $goto\ i$ statement;
- $ins_1 \equiv c_1 := c_1 + 1; goto\ 2$.

A *configuration* of \mathcal{M} is a triple $[ins_i, v_1, v_2]$, where ins_i is the instruction to be executed, and $v_1, v_2 \in \mathbb{N}_0$ are the current values of c_1, c_2 . A *computational step* \hookrightarrow between configurations is defined in the expected way. A *recurrent computation* of \mathcal{M} is an infinite computation initiated in $[ins_1, 0, 0]$ along which ins_1 is executed infinitely often. As we already mentioned, the problem whether a given \mathcal{M} has a recurrent computation is Σ_1^1 -complete.

Let $\mathcal{M} \equiv 1:ins_1, \dots, n:ins_n$ be a nondeterministic Minsky machine. We construct a $1\frac{1}{2}$ game $G(\mathcal{M}) = (V, E, (V_\square, V_\circ), Prob)$ and a formula $\varphi \in \mathcal{L}(F^{=5/8}, F^{>0}, F^{=1}, G^{=1})$ such that player \square has a winning HD or HR strategy in a vertex $(q_0, 1, res_1) \in V$ iff \mathcal{M} has a recurrent computation.

We define the game $G(\mathcal{M})$ incrementally (the sets V and E are initially empty). For the sake of simplicity, we also introduce redundant vertices that are not reachable from the initial vertex $(q_0, 1, res_1)$. Let $S = \{inc_1, inc_2, dec_1, dec_2, res_1, res_2, nil\}$.

- For all $0 \leq i \leq n$, $j \in \{1, 2\}$, and $s \in S$ we add a vertex (q_i, j, s) to V_\circ , and vertices (t_i, j, s) , (r_i, s) to V_\square . We also fix fresh atomic propositions $a(i, j), b(i, j)$ which will be used later.
- The outgoing transitions of the (r_i, s) vertices are shown in Figure 1 (bottom left).
- For each (q_i, j, s) vertex we add four new vertices to V_\circ . The vertices are connected by transitions as shown in Figure 1 (left). Some of the newly added vertices satisfy the propositions $a(i, j), b(i, j)$ as indicated in the figure. Note that the structure for $i = 0$ is slightly different.
- For each (t_i, j, s) vertex we add either 7 or 9 new vertices (depending on j and s) to V_\circ and connect them as indicated in Figure 1 (right). The validity of the propositions $a(i, j), b(i, j)$ in these new vertices is also shown in the figure.

For all $i \in \{0, \dots, n\}$ and $j \in \{1, 2\}$ we fix a fresh atomic proposition $c(i, j)$ whose validity is defined as follows:

- all vertices which satisfy $b(i, j)$ satisfy also $c(i, j)$;
- if ins_i contains the *goto* k statement, then all vertices which satisfy $a(k, j)$ also satisfy $c(i, j)$. Moreover, for all $s \in S$ the vertex $(q_k, 2, s)$ satisfies $c(i, 1)$, and the vertex (r_k, s) satisfies $c(i, 2)$.
- no other vertices satisfy $c(i, j)$.

Finally, we assume that for each vertex v there is a fresh proposition p_v which is valid only in v (we write v instead of p_v in our formulae).

Our aim is to construct the formula φ so that each play of $G(\mathcal{M})$ which satisfies φ corresponds to a recurrent computation of \mathcal{M} . From this point on we restrict our attention to HD strategies (at the end of this proof we show that the use of randomized moves can easily be prohibited by φ , and hence the presented result also applies to HR strategies).

The structure of a typical play is shown in Figure 2 (to simplify the figure, the loops on vertices are not drawn). First, let us realize that *every* play of $G(\mathcal{M})$ can be identified with an infinite sequence

$$[ins_1, 0, 0], \dots, [ins_i, V_1, V_2], [ins_k, U_1, U_2], \dots$$

of *extended* configurations of \mathcal{M} , where the counters can also take the ω (i.e., “infinite”) value. Of course, this sequence does not necessarily correspond to a valid computation of \mathcal{M} . The way how a given play determines its associated sequence is indicated in

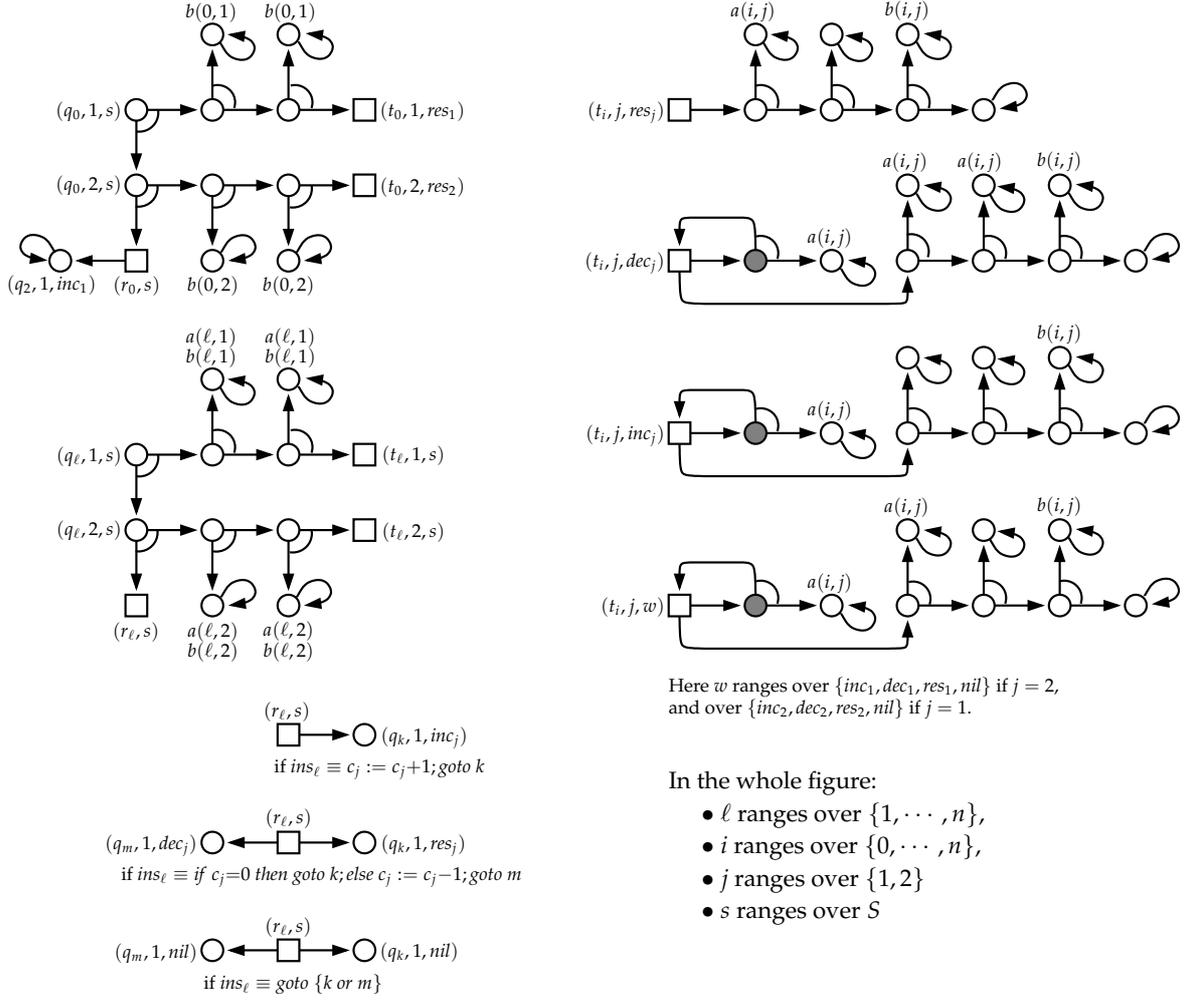


Figure 1: The structure of $G(\mathcal{M})$

Figure 2. Each extended configuration in the sequence corresponds to a block of vertices in the play. For example, a configuration $[ins_i, V_1, V_2]$, where $ins_i \equiv c_1 := c_1 + 1; goto k$, is represented by a block of vertices that starts in a $(q_i, 1, s)$ vertex (where s corresponds to the instruction of the configuration which immediately precedes $[ins_i, V_1, V_2]$ in the sequence). The counters are encoded as follows: First, observe that when a (t_i, j, s) vertex of the play is visited, player \square can choose between transitions leading to a “gray” or “white” vertex (see Figure 1). If he chooses a gray vertex, then with probability $1/2$ he will make another choice in two transitions. Thus, player \square may decide to visit a gray vertex C_j -times, where C_j ranges from 0 to infinity. This C_j represents the value of counter j . In the play of Figure 2, player \square has chosen a gray vertex from a $(t_i, 1, dec_2)$ vertex V_1 -times, and hence the value of the first counter is V_1 .

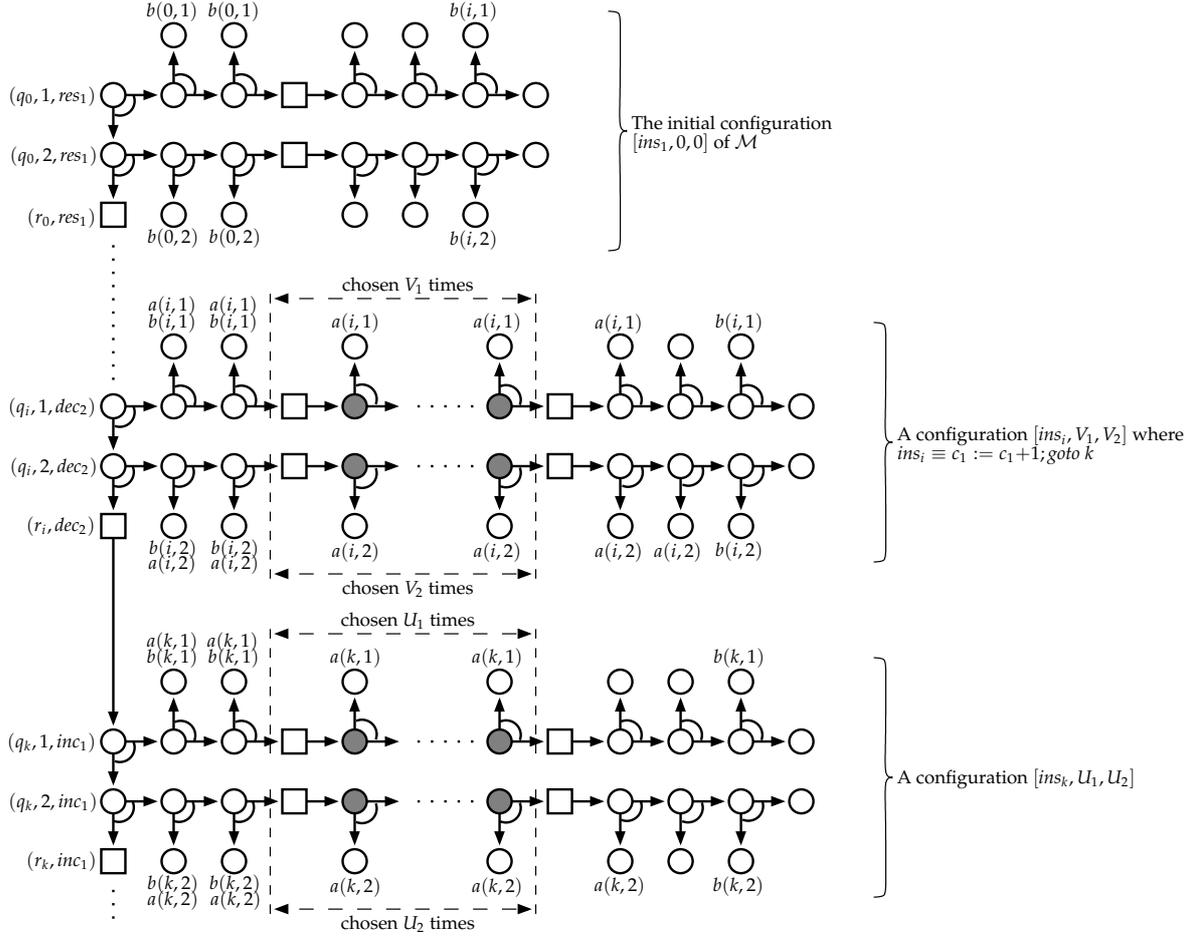


Figure 2: A play of $G(\mathcal{M})$ (the loops on vertices are not drawn).

The initial configuration $[ins_1, 0, 0]$ requires a special treatment, because the counters are initially zero but can become positive when ins_1 is revisited. This is the reason why we introduced the family of zero-indexed vertices such as (q_0, j, s) , (r_0, s) , etc.

It is easy to see that the sequence

$$[ins_1, 0, 0], \dots, [ins_i, V_1, V_2], [ins_k, U_1, U_2], \dots$$

does correspond to a recurrent computation of \mathcal{M} iff the following conditions are satisfied:

- (a) Counter values in all extended configurations of the sequence are finite.
- (b) The sequence contains infinitely many configurations of the form $[ins_1, \dots]$.
- (c) For each pair $[ins_i, V_1, V_2], [ins_k, U_1, U_2]$ of successive configurations we have that $[ins_i, V_1, V_2] \leftrightarrow [ins_k, U_1, U_2]$.

We show how to express these conditions in $\mathcal{L}(F^{=5/8}, F^{>0}, F^{=1}, G^{=1})$. Condition (a) is easy—it suffices to say that whenever a (t_i, j, s) vertex is hit, there must be a finite path to a vertex satisfying $b(i, j)$. This prevents player \square from “looping” at gray vertices forever (see Figure 1). Formally, this is encoded by the formula

$$Fin \equiv \bigwedge_{0 \leq i \leq n, j \in \{1,2\}, s \in S} G^{=1}((t_i, j, s) \Rightarrow F^{>0}b(i, j))$$

Condition (b) can be expressed by the formula

$$Rec \equiv G^{=1} \left(\left(\bigvee_{0 \leq i \leq n, s \in S} (q_i, 1, s) \right) \Rightarrow \left(F^{>0} \bigvee_{s' \in S} (q_1, 1, s') \right) \right)$$

Condition (c) requires more care, and this is where we need the $F^{=5/8}$ operator. Let $[ins_i, V_1, V_2], [ins_k, U_1, U_2]$ be successive configurations in our sequence, and let us first consider the case when $ins_i \equiv c_1 := c_1 + 1; goto\ k'$ (note that this case is visualized in Figure 2). The definition of $G(\mathcal{M})$ guarantees that $k' = k$ (see the outgoing transitions of (r_i, s) vertices in Figure 1). The $(q_k, 1, s)$ vertex in the block which corresponds to $[ins_k, U_1, U_2]$ must be of the form $(q_k, 1, inc_1)$. So, all we need to check is that $U_1 = V_1 + 1$ and $U_2 = V_2$. We claim that

- $U_1 = V_1 + 1$ iff the $(q_i, 1, s)$ vertex which corresponds to the configuration $[ins_i, V_1, V_2]$ satisfies the formula $F^{=5/8}c(i, 1)$.
- $U_2 = V_2$ iff the $(q_i, 2, s)$ vertex which corresponds to the configuration $[ins_i, V_1, V_2]$ satisfies the formula $F^{=5/8}c(i, 2)$.

Let us first verify the second claim. By definition, $c(i, 2)$ is valid in the (r_k, s) vertices and in all vertices that satisfy $b(i, 2)$ or $a(k, 2)$. All runs initiated in the $(q_i, 2, s)$ vertex which leave the two blocks of vertices corresponding to $[ins_i, V_1, V_2]$ and $[ins_k, U_1, U_2]$ inevitably go through (r_k, s) , and the total probability of all these runs is $1/8$ (here we need our assumption that ins_i does not contain the *goto* i statement; without this assumption, it could happen that $k = i$ and hence the considered probability could be $1/4$). So, the considered $(q_i, 2, s)$ vertex satisfies the formula $F^{=5/8}c(i, 2)$ iff the probability of hitting a vertex which satisfies $b(i, 2)$ or $a(k, 2)$ and lies *within* the two blocks of vertices corresponding to $[ins_i, V_1, V_2]$ and $[ins_k, U_1, U_2]$ is exactly $1/2$. A closer look reveals that this probability is equal to the following sum of two binary numbers:

$$0.011\underbrace{0 \dots 0}_{V_2}001 + 0.00011\underbrace{1 \dots 1}_{U_2}1$$

Obviously, this sum is equal to $1/2$ iff $U_2 = V_2$, and we are done.

The first claim is verified similarly. In this case, we obtain that the $(q_i, 1, s)$ vertex satisfies $F^{=5/8}c(i, 1)$ iff

$$0.011\underbrace{0 \dots 0}_{V_1}001 + 0.00011\underbrace{1 \dots 1}_{U_1}$$

is equal to $1/2$, which happens iff $U_1 = V_1 + 1$.

If $ins_i \equiv \text{if } c_1=0 \text{ then goto } k'; \text{else } c_j:=c_j-1; \text{goto } m'$ then $k = k'$ or $k = m'$ (again, this follows from the definition of $G(\mathcal{M})$). The block of vertices which corresponds to $[ins_k, U_1, U_2]$ then contains a vertex of the form $(q_k, 1, s)$, where s is either res_1 or dec_1 . We need to check the following conditions:

- (1) $U_2 = V_2$.
- (2) If $s = res_1$, then $U_1 = V_1 = 0$.
- (3) If $s = dec_1$, then $U_1 = V_1 - 1$.

Similarly as above, one can verify that Condition (1) holds iff the $(q_i, 2, s)$ vertex in the block of vertices corresponding to $[ins_i, V_1, V_2]$ satisfies the formula $F^{=5/8}c(i, 2)$. We claim that Conditions (2) and (3) hold iff the $(q_i, 1, s)$ vertex in the same block satisfies the formula $F^{=5/8}c(i, 1)$. Let us first consider Condition (2). The definition of $G(\mathcal{M})$ guarantees that $U_1 = 0$ (see the outgoing transitions of $(t_i, 1, res_1)$ in Figure 1). Hence, it actually suffices to check that $U_1 = V_1$, which is done by the formula $F^{=5/8}c(i, 1)$. Condition (3) is handled similarly.

If ins_i operates over the second counter, the arguments are the same as above (there is no need to change the formulae or the vertices in which they are supposed to hold). Finally, if $ins_i \equiv \text{goto } \{k' \text{ or } m'\}$, then $k = k'$ or $k = m'$. We need to check that $U_1 = V_1$ and $U_2 = V_2$, which is again implemented by the formulae $F^{=5/8}c(i, 1)$ and $F^{=5/8}c(i, 2)$.

So, Condition (c) can be encoded by the formula

$$Succ \equiv \bigwedge_{0 \leq i \leq n, j \in \{1,2\}, s \in S} G^1((q_i, j, s) \Rightarrow F^{=5/8}c(i, j))$$

Now we define the formula

$$\varphi \equiv Fin \wedge Rec \wedge Succ \wedge NoRnd$$

where the subformula *NoRnd* says that player \square cannot use randomized moves (the formula says that whenever a vertex $v \in V_\square$ is hit, there is an immediate successor of v which is visited with probability 1). This can be expressed using G^1 and F^1 operators. Hence, our proof applies both to HD and HR strategies. \square

On the other hand, the existence of a winning HD strategy in $1\frac{1}{2}$ games with general PCTL objectives can be encoded by a Σ_1^1 formula in a straightforward way. Hence, the problem is Σ_1^1 -complete.

A slight modification of the construction presented in Theorem 3.4 reveals the following:

Theorem 3.5. *The existence of a winning HD (or HR) strategy with finite memory in $1\frac{1}{2}$ games with $\mathcal{L}(F^{5/8}, F^1, F^{>0}, G^1)$ objectives is undecidable.*

Proof. First, let us realize that the problem is semidecidable (i.e., belongs to the Σ_1 level of the arithmetical hierarchy). Obviously, one can effectively enumerate all (\mathcal{A}, f) and for each such (\mathcal{A}, f) decide whether $\sigma(\mathcal{A}, f)$ is winning, because the corresponding play has only finitely many states (more precisely, the play is obtained as unfolding of an effectively constructible finite-state Markov chain). The undecidability result is obtained by a slight modification of the construction presented in Theorem 3.4. In this case, we reduce the halting problem for “ordinary” deterministic Minsky machines (i.e., there is no *goto* $\{k \text{ or } m\}$ instruction, and the last instruction is *halt*). The subformula *Rec* is replaced with a formula $Halt \equiv F^{>0} \bigvee_{s \in S} (q_n, 1, s)$, which says that a “halting state” is reachable with positive probability. Note that if a given Minsky machine halts, then it halts after finitely many steps and the corresponding winning strategy needs only finite memory (of course, there is no bound on its size). If the machine does not halt, there is no winning strategy at all. \square

Now we show that the previous undecidability results are tight in the sense that the existence of a winning HD strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^1, F^{>0}, G^1)$ objectives is decidable, and in fact **EXPTIME**-complete.

Let G be a $1\frac{1}{2}$ -player game where V is the set of vertices. A *mixed* objective is a pair (P, Q) where $P, Q \subseteq V$. A strategy σ for player \square is (P, Q) -winning in a vertex $v \in V$ iff all runs in $G(\sigma)$ initiated in v visit some state of P infinitely often, and the probability of all runs which visit some state of Q infinitely often is 1. Hence, a mixed objective is essentially a conjunction of a *sure-Büchi* objective specified by P and a *qualitative-Büchi* objective specified by Q . The first step towards the promised **EXPTIME** upper bound is the following:

Lemma 3.6. *Let G be a $1\frac{1}{2}$ -player game, s_{in} a vertex of G , and (v, ψ) an objective where $\psi \in \mathcal{L}(F^1, F^{>0}, G^1)$. Then there effectively exists a $1\frac{1}{2}$ -player game G' , a vertex s'_{in} of G' , and a mixed objective (P, Q) such that player \square has a (v, ψ) -winning HD strategy in the vertex s_{in} iff player \square has a (P, Q) -winning HD strategy in the vertex s'_{in} . Moreover, the G' , s'_{in} , and (P, Q) are computable in time which is linear in the size of G and exponential in the size of ψ .*

Since the proof of Lemma 3.6 is relatively long and relies on several auxiliary results, it has been shifted to Appendix A.1.

Lemma 3.6 shows that the problem of our interest is reducible to another game-theoretic problem, whose complexity is analyzed in our next lemma.

Lemma 3.7. *The existence of a winning strategy in $1\frac{1}{2}$ -player games with mixed objectives is decidable in polynomial time.*

A proof of Lemma 3.7 can be found in Appendix A.2.

A direct consequence of Lemma 3.6 and Lemma 3.7 is that the existence of a winning HD strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^{=1}, F^{>0}, G^{=1})$ objectives is in **EXPTIME**. It remains to establish the matching lower bound.

Lemma 3.8. *The existence of a winning HD (or HR) strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^{=1}, G^{=1})$ objectives is **EXPTIME**-hard.*

Proof. We reduce the acceptance problem for alternating LBA (which is known to be **EXPTIME**-complete [15]). 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, $\delta : Q \times \Gamma \rightarrow 2^{Q \times \Gamma \times \{L, R\}}$ 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 iff 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 construct (in polynomial time) a $1\frac{1}{2}$ game $G(\mathcal{M}, u) = (V, E, (V_{\square}, V_{\circ}), Prob)$ and an objective (v, φ) where $\varphi \in \mathcal{L}(F^{=1}, G^{=1})$ such that player \square has a (v, φ) -winning HD (or HR) strategy in a distinguished vertex $g(1, 1) \in V$ iff \mathcal{M} accepts u . Configurations of \mathcal{M} are written as 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 Ξ) we have that $c' \in \Xi^6$ 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 3. The set V consists of the following vertices:

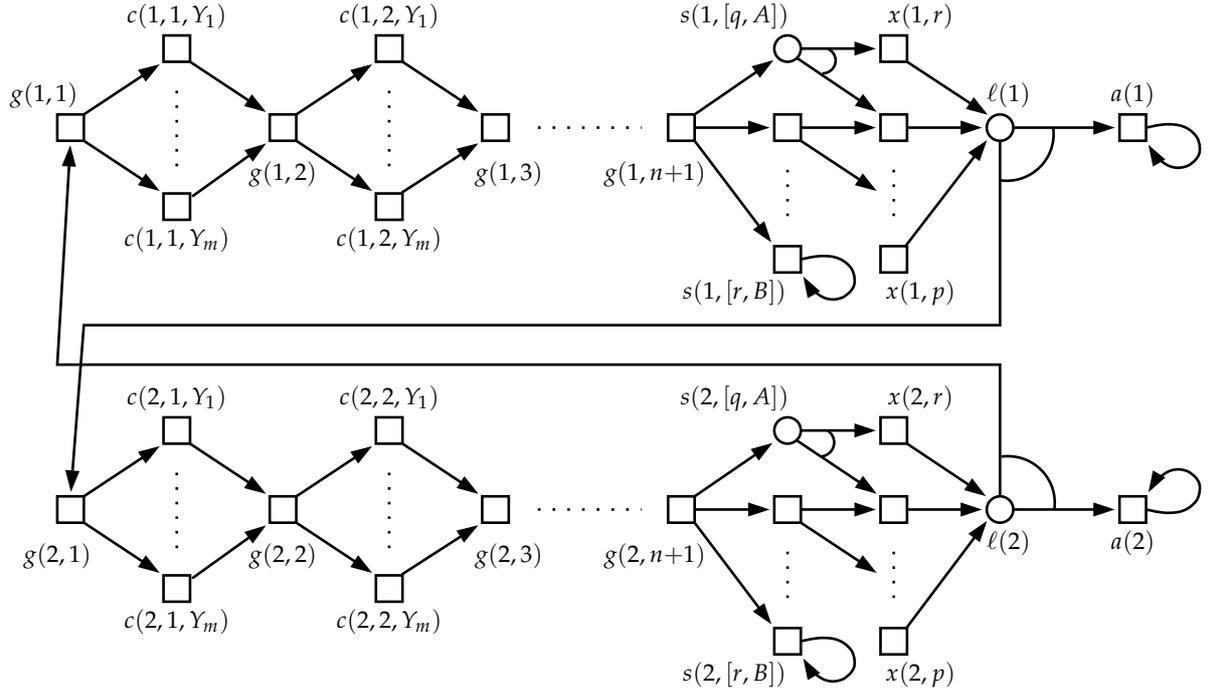


Figure 3: The structure of $G(\mathcal{M}, w)$

- $g(j, i)$, 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:

- $g(j, i) \rightarrow c(j, i, Y)$ and $c(j, i, Y) \rightarrow g(j, i+1)$ for all $j \in \{1, 2\}$, $i \in \{1, \dots, n\}$, and $Y \in \Xi$;
- $g(j, n+1) \rightarrow s(j, [q, A])$ for all $j \in \{1, 2\}$, $q \in Q$, and $A \in \Gamma$;
- $s(j, [q, A]) \rightarrow s(j, [q, A])$ for all $j \in \{1, 2\}$, $A \in \Gamma$, and $q \in Q$ where q is accepting or rejecting;
- $s(j, [q, A]) \rightarrow x(j, q')$ 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) ;
- $x(j, q) \rightarrow \ell(j)$ for all $j \in \{1, 2\}$ and $q \in Q$;
- $\ell(1) \rightarrow g(2, 1)$, $\ell(1) \rightarrow a(1)$, $\ell(2) \rightarrow g(1, 1)$, $\ell(2) \rightarrow a(2)$;
- $a(1) \rightarrow a(1)$, $a(2) \rightarrow a(2)$.

The set V_{\circ} consists of $\ell(1), \ell(2)$ and all $s(j, [q, A])$ where $q \in Q_{\forall}$. The other vertices belong to V_{\square} . 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 \square successively “guesses” the configurations of \mathcal{M} by choosing appropriate moves in the vertices $g(1,1), \dots, g(1,n)$ and $g(2,1), \dots, g(2,n)$. In the states $g(1,n+1)$ and $g(2,n+1)$, player \square 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 \square 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 \square 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 φ constructed below ensures that player \square 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 \square has a HD (or HR) strategy such that φ is satisfied in $g(1,1)$.

Now we describe the formula φ 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

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

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, $\text{Init} \equiv \bigwedge_{i=1}^n F^{-1}c(1,i,w(i))$. Note that if player \square selects, e.g., $c(1,1,\vdash)$ instead of $c(1,1,q_0)$, the formula $F^{-1}c(1,1,q_0)$ 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 $\text{Succ}_1 \wedge \text{Succ}_2$. 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 $(X_1, X_2, X_3, Y_1, Y_2, Y_3) \in \text{Comp}(\mathcal{M})$. The formula Succ_1 looks as follows:

$$G^{-1} \left(g(1,1) \Rightarrow \left(\text{Acc} \vee \bigwedge_{i=1}^{n-2} \bigvee_{\vec{X} \in \Xi^3} \text{Pos}(1,i,\vec{X}) \right) \right)$$

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

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

where ψ is the formula

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

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:

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

$Ctrl_2$ 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):

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

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

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

Note that due to the assumption that every computational tree of \mathcal{M} is finite, the previous formulae already guarantee that player \square *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 leaves in this tree are accepting.

Finally, the subformula $NoRnd$ says that player \square does not use randomized moves. This subformula is actually needed only if player \square uses a HR strategy ($NoRnd$ is redundant for HD strategies). This is implemented simply by saying that whenever a vertex of V_\square 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:

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

The formulae for the other vertices of V_\square look similarly. □

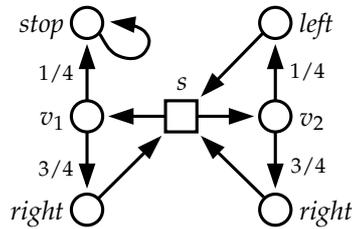
A simple corollary of Lemma 3.6, Lemma 3.7, and Lemma 3.8 is the following:

Theorem 3.9. *The existence of a winning HD strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^=1, F^{>0}, G^=1)$ objectives is EXPTIME-complete. The EXPTIME lower bound holds even for $\mathcal{L}(F^=1, G^=1)$ objectives.*

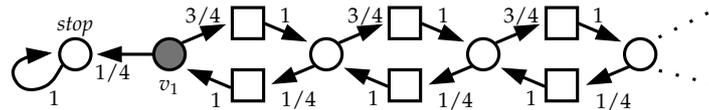
It follows from the proofs of Lemma 3.6 and Lemma 3.7 that a winning HD strategy in $1\frac{1}{2}$ -player games with $\mathcal{L}(F^=1, F^{>0}, G^=1)$ objectives actually requires only *finite memory* whose size is linear in the size of a given game and exponential in the size of a given objective. A natural question is whether Theorem 3.9 can be generalized to a larger class of qualitative PCTL objectives. One natural possibility is to add the $G^{>0}$ operator, which yields the $\mathcal{L}(F^=1, F^{>0}, G^=1, G^{>0})$ fragment. However, there is a strong evidence that the method of Lemma 3.6 cannot be generalized to this class of objectives. This is because these objectives may already require *infinite memory*, which is demonstrated in our last theorem:

Theorem 3.10. *A winning HD strategy in $1\frac{1}{2}$ games with $\mathcal{L}(F^{>0}, G^{>0})$ objectives may require infinite memory.*

Proof. Let $\varphi \equiv G^{>0}(\neg stop \wedge F^{>0} stop)$ and let G be the following game (the valuation v for atomic propositions *stop*, *left*, and *right* is also indicated in the figure):



First we show that there is a (v, φ) -winning HD strategy σ for player \square in the vertex v_1 . We define $\sigma(ws)$ to be the Dirac distribution which assigns 1 to the transition leading to v_1 or v_2 , depending on whether $\#_{right}(w) - \#_{left}(w) \leq 0$ or $\#_{right}(w) - \#_{left}(w) > 0$, respectively. Here $\#_{right}(w)$ denotes the number of occurrences of a state satisfying the proposition *right* in w . We claim that the state v_1 in the play $G(\sigma)$ satisfies the formula $G^{=2/3}(\neg stop \wedge F^{>0} stop)$ and hence also the formula φ . To see this, realize that the play $G(\sigma)$ corresponds to the unfolding of the following infinite Markov chain:



A standard calculation reveals that the probability of hitting the *stop* state from v_1 is equal to $1/3$. Hence, the probability of all runs initiated in v_1 which do *not* hit the *stop* state is $2/3$. All states in all these runs can reach the *stop* state with positive probability. Hence, v_1 satisfies the formula $G^{=2/3}(\neg stop \wedge F^{>0} stop)$.

Now we show that there is no (ν, φ) -winning HD strategy with finite memory. Suppose the converse. Let (\mathcal{A}, f) be such a strategy where the automaton $\mathcal{A} = (Q, V, \delta, q_0)$ has n states. We show that the state v_1 in the corresponding play satisfies the formula $G^{=0}(\neg stop \wedge F^{>0}stop)$, which means that v_1 does not satisfy φ . We say that a state w in the play $G(\sigma(\mathcal{A}, f))$ is *live* if there is a state $ww's$ such that $w \rightarrow^* ww's$ and $f(\delta(q_0, ww's), s)$ assigns 1 to the transition leading to v_1 . A state which is not live is *dead*. We claim that there is a fixed $\varepsilon > 0$ such that the probability of hitting a *stop* state from a given live state w is at least ε . To see this, it suffices to observe that whenever w is a live state, then there is a path from w to a *stop* state of length at most $3n+1$. Note that a state w is dead iff w is a *stop* state or w cannot reach a *stop* state at all. By applying standard arguments of Markov chain theory, we can now conclude that the probability of hitting a dead state from v_1 is equal to one. Since a dead state does not satisfy $\neg stop \wedge F^{>0}stop$, we obtain that v_1 satisfies $G^{=0}(\neg stop \wedge F^{>0}stop)$ and we are done. \square

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] T. Brázdil, J. Esparza, and A. Kučera. Analysis and prediction of the long-run behavior of probabilistic sequential programs with recursion. In *Proceedings of FOCS 2005*, pp. 521–530. IEEE Computer Society Press, 2005.
- [3] 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 *Lecture Notes in Computer Science*, pp. 145–157. Springer, 2005.
- [4] K. Chatterjee, L. de Alfaro, and T. Henzinger. The complexity of stochastic Rabin and Streett games. In *Proceedings of ICALP 2005*, vol. 3580 of *Lecture Notes in Computer Science*, pp. 878–890. Springer, 2005.
- [5] K. Chatterjee and T. Henzinger. Semiperfect-information games. In *Proceedings of FST&TCS 2005*, vol. 3821 of *Lecture Notes in Computer Science*, pp. 1–18. Springer, 2005.
- [6] K. Chatterjee, M. Jurdzinski, and T. Henzinger. Simple stochastic parity games. In *Proceedings of CSL'93*, vol. 832 of *Lecture Notes in Computer Science*, pp. 100–113. Springer, 1994.
- [7] K. Chatterjee, M. Jurdzinski, and T. Henzinger. Quantitative stochastic parity games. In *Proceedings of SODA 2004*, pp. 121–130. SIAM, 2004.

- [8] L. de Alfaro. Computing minimum and maximum reachability times in probabilistic systems. In *Proceedings of CONCUR'99*, vol. 1664 of *Lecture Notes in Computer Science*, pp. 66–81. Springer, 1999.
- [9] L. de Alfaro and R. Majumdar. Quantitative solution of omega-regular games. *Journal of Computer and System Sciences*, 68:374–397, 2004.
- [10] J. Esparza, A. Kučera, and R. Mayr. Model-checking probabilistic pushdown automata. In *Proceedings of LICS 2004*, pp. 12–21. IEEE Computer Society Press, 2004.
- [11] K. Etessami and M. Yannakakis. Recursive Markov chains, stochastic grammars, and monotone systems of non-linear equations. In *Proceedings of STACS'2005*, vol. 3404 of *Lecture Notes in Computer Science*, pp. 340–352. Springer, 2005.
- [12] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6:512–535, 1994.
- [13] D. Harel. Effective transformations on infinite trees with applications to high undecidability dominoes, and fairness. *Journal of the Association for Computing Machinery*, 33(1), 1986.
- [14] A. Kučera and O. Stražovský. On the controller synthesis for finite-state Markov decision processes. In *Proceedings of FST&TCS 2005*, vol. 3821 of *Lecture Notes in Computer Science*, pp. 541–552. Springer, 2005.
- [15] Ch. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.
- [16] W. Thomas. Infinite games and verification. In *Proceedings of CAV 2003*, vol. 2725 of *Lecture Notes in Computer Science*, pp. 58–64. Springer, 2003.

A Appendix

A.1 A Proof of Lemma 3.6

For the rest of this proof, let us fix a $1\frac{1}{2}$ -player game $G = (V, E, (V_{\square}, V_{\circ}), Prob)$, a vertex $s_{in} \in V$, and an objective (ν, ψ) where $\psi \in \mathcal{L}(F^{=1}, F^{>0}, G^{=1})$. For technical convenience, we assume that all subformulae of ψ are pairwise distinct (this can be achieved by replacing atomic propositions in φ with fresh propositions so that each proposition has a unique occurrence in ψ ; the valuation ν is extended accordingly). Our aim is to define another $1\frac{1}{2}$ -player game G' , a vertex s'_{in} of G' , and a mixed objective (P, Q) such that player \square has a (ν, ψ) -winning HD strategy in s_{in} iff player \square has a (P, Q) -winning HD strategy in s'_{in} .

Let L be the set of all *literals*, i.e., atomic propositions and their negations. Let \mathcal{S} be the set of all subformulae of ψ , where negation is not considered as a connective (for example, if $\psi \equiv F^{=1}\neg q$, then $\mathcal{S} = \{\neg q, F^{=1}\neg q\}$). For each connective $\alpha \in \{F^{=1}, F^{>0}, G^{=1}, \vee, \wedge\}$, we use \mathcal{S}_{α} to denote the subset of \mathcal{S} consisting of all formulae where the topmost connective is α . We also use \mathcal{S}_{Ap} , \mathcal{S}_F , \mathcal{S}_{Temp} , \mathcal{S}_{Bool} , and $\mathcal{S}_{\bar{F}}$ to denote the sets $\mathcal{S} \cap L$, $\mathcal{S}_{F^{=1}} \cup \mathcal{S}_{F^{>0}}$, $\mathcal{S}_{F^{=1}} \cup \mathcal{S}_{F^{>0}} \cup \mathcal{S}_{G^{=1}}$, $\mathcal{S}_{\vee} \cup \mathcal{S}_{\wedge}$, and $\{\bar{F}^{\times q} \varphi \mid F^{\times q} \varphi \in \mathcal{S}_F\}$, respectively. The purpose of “barred” formulae of $\mathcal{S}_{\bar{F}}$ becomes clear later.

In the following, we assume that $\mathcal{S} = \{\varphi_1, \dots, \varphi_n\}$ where $i < j$ implies that φ_i is *not* a subformula of φ_j . The first step towards the definition of G' is the function $\Theta : 2^{\mathcal{S}} \rightarrow 2^{2^{\mathcal{S} \cup \mathcal{S}_{\bar{F}}}}$ which decomposes subformulae of ψ into “subgoals”. Let $A \subseteq \mathcal{S}$. If $A \subseteq \mathcal{S}_{Ap}$, then $\Theta(A) = \{A\}$. Otherwise, let i be the least index such that $\varphi_i \in A \setminus \mathcal{S}_{Ap}$. We distinguish among the following possibilities:

- If $\varphi_i \equiv \varphi_k \vee \varphi_{\ell}$, then

$$\Theta(A) = \Theta((A \setminus \{\varphi_i\}) \cup \{\varphi_k\}) \cup \Theta((A \setminus \{\varphi_i\}) \cup \{\varphi_{\ell}\})$$

- If $\varphi_i \equiv \varphi_k \wedge \varphi_{\ell}$, then $\Theta(A) = \Theta((A \setminus \{\varphi_i\}) \cup \{\varphi_k, \varphi_{\ell}\})$
- If $\varphi_i \equiv G^{=1}\varphi_j$, then

$$\Theta(A) = \{D \cup \{G^{=1}(\varphi_j)\} \mid D \in \Theta((A \setminus \{\varphi_i\}) \cup \{\varphi_j\})\}$$

- $\varphi_i \equiv F^{\times q}\varphi_j$; then

$$\begin{aligned} \Theta(A) = & \{D \cup \{F^{\times q}(\varphi_j)\} \mid D \in \Theta(A \setminus \{\varphi_i\})\} \\ & \cup \{D \cup \{\bar{F}^{\times q}(\varphi_j)\} \mid D \in \Theta((A \setminus \{\varphi_i\}) \cup \{\varphi_j\})\} \end{aligned}$$

The intuition behind the function Θ is the following: to find out whether there is a (ν, ψ) -winning HD strategy in s_{in} , we extend each vertex of G (and hence each state of an arbitrary play of G) with a set A of subformulae of ψ that should be valid when the play is

in the state. Some of these formulae represent temporal “goals” which can be achieved either in the current state or in its successors. The function Θ “offers” all admissible possibilities how to distribute the goals among the current state and its successors so that all formulae in A are valid. Selecting the right alternative becomes the responsibility of player \square . For example, $\Theta(\{F^{-1}p\}) = \{\{F^{-1}p\}, \{\bar{F}^{-1}p, p\}\}$, because the “current” state satisfies $F^{-1}p$ iff either all of its successors satisfy $F^{-1}p$ (the goal is “postponed”), or the proposition p is satisfied in the current state (the goal is “achieved”). In the latter case, the function Θ also “marks” the current state with $\bar{F}^{-1}(p)$, which means that the goal $F^{-1}(p)$ has been achieved. The exact purpose of these marks will be clarified later.

Before defining the game G' , we formulate several properties of Θ that are useful in proofs. The next two lemmas are proven by a straightforward induction on the total number of connectives in the formulae of A .

Lemma A.1. *Let s be a state of some play of G , and let $A \subseteq \mathcal{S}$. Then s satisfies all formulae of A iff there is $D \in \Theta(A)$ such that s satisfies all formulae of $D \cap \mathcal{S}$.*

Lemma A.2. *For all non-empty disjoint sets $A, B \subseteq \mathcal{S}$ we have that $\Theta(A \cup B) = \{X \cup Y \mid X \in \Theta(A), Y \in \Theta(B)\}$.*

Corollary A.3. *Let $A \subseteq \mathcal{S}$ be a non-empty set.*

- If $G^{-1}\xi \in A$, then $\Theta(A)$ is equal to

$$\{\{G^{-1}\xi\} \cup X \cup Y \mid X \in \Theta(A \setminus \{G^{-1}(\xi)\}), Y \in \Theta(\{\xi\})\}$$

- If $F^{\times e}\xi \in A$, then $\Theta(A)$ is equal to

$$\begin{aligned} & \{\{F^{\times e}\xi\} \cup X \mid X \in \Theta(A \setminus \{F^{\times e}\xi\})\} \\ & \cup \{\{\bar{F}^{\times e}\xi\} \cup X \cup Y \mid X \in \Theta(A \setminus \{F^{\times e}\xi\}), Y \in \Theta(\{\xi\})\} \end{aligned}$$

The game $G' = (V', E', (V'_{\square}, V'_{\circ}), Prob')$ is defined as follows. The set of vertices V' consists of vertices of the following two forms (f -vertices and g -vertices):

- f -vertices are of the form $(s, A, B, C)^f$ where $s \in V$, $A \subseteq \mathcal{S}$, $B \subseteq \mathcal{S}_{F=1} \cup \{\bullet\}$, and $C \subseteq \mathcal{S}_{F>0}$. Intuitively, the set A consists of formulas that should be satisfied in the current state (see the intuitive description of Θ above). The sets B and C assure that all subgoals of the form $F^{-1}\varphi$ and $F^{>0}\varphi$ are eventually fulfilled (see the mixed winning objective defined below).
- g -vertices are of the form $(s, A, B, C, \vec{D})^g$ where $s \in V$, $A \subseteq \mathcal{S} \cup \mathcal{S}_{\bar{F}}$, $B \subseteq \mathcal{S}_{F=1} \cup \{\bullet\}$, $C \subseteq \mathcal{S}_{F>0}$, and $\vec{D} \in \prod_{t \in V} 2^{\mathcal{S}_{F>0}}$. The purpose of B and C is similar as in the case of f -vertices. The set A consists of subgoals that should be satisfied in successors of the current state. The vector \vec{D} is used to distribute the subgoals of the form $F^{>0}\varphi$ to the successors of the current state.

The set V_{\square}^f consists of all f -vertices and of all g -vertices of the form $(s, A, B, C, \vec{D})^g$ where $s \in V_{\square}$. The set V_{\circ}^g consists of all g -vertices of the form $(s, A, B, C, \vec{D})^g$ where $s \in V_{\circ}$. The set E' of transitions of G' is defined as follows:

1. $(s, A, B, C)^f \rightarrow (s, A', B', C', \vec{D})^g$ iff the following conditions are satisfied:

- $A' \in \Theta(A)$
- B' is equal to
 - $\{\bullet\}$, if $A' \cap Ap \not\subseteq v(s)$ or there is $\neg p \in A'$ such that $p \in v(s)$;
 - $A' \cap \mathcal{S}_{\bar{F}=1}$, if $B = \emptyset$;
 - $B \setminus \{F^{-1}\zeta \mid \bar{F}^{-1}\zeta \in A'\}$ otherwise.
- if $C = \emptyset$, then $C' = A' \cap \mathcal{S}_{F>0}$; otherwise $C' = C \setminus \{F^{>0}\zeta \mid \bar{F}^{>0}\zeta \in A'\}$.
- $\bigcup_{(s,t) \in E} \vec{D}_t = A' \cap \mathcal{S}_{\bar{F}>0}$
- if $s \in V_{\square}$ then for each $t \in V$ such that $(s, t) \in E$ we have that $D_t = A' \cap \mathcal{S}_{F>0}$.

Intuitively, the f -vertices are controlled by player \square who chooses a set of sub-goals $A' \in \Theta(A)$. The atomic propositions in A' are immediately verified (if there is some inconsistency then \bullet is put into B') while the other formulae in A' are passed to successors. The sets B' and C' are updated depending on which sub-goals (subformulae) are chosen by player \square as “achieved” in the current state (cf. the intuitive description of Θ above). Note that the vertex s is not changed in the successors of f -vertices. The transitions of G are simulated in g -vertices (see below).

2. $(s, A, B, C, \vec{D})^g \rightarrow (t, A', B', C')^f$ if $(s, t) \in E$, $A' = (A \setminus (\mathcal{S}_{F>0} \cup \mathcal{S}_{\bar{F}} \cup \mathcal{S}_{Ap})) \cup \vec{D}_t$, $B' = B$, and $C' = C \cap \vec{D}_t$

3. There are no other transitions in E' than those given by the rules 1. and 2.

$Prob'$ is defined as follows: For all $s \in V_{\circ}$, the probability of $(s, A, B, C, \vec{D})^g \rightarrow (t, A', B', C')^f$ is the same as the probability of $s \rightarrow t$ in G . We put $s'_{in} = (s_{in}, \{\psi\}, \emptyset, \emptyset)^f$. Finally, we define the mixed (P, Q) objective as follows:

- the set P consists of all vertices of the form $(s, A, B, \emptyset, \vec{D})^g$;
- the set Q consists of all vertices of the form $(s, A, \emptyset, C, \vec{D})^g$.

It remains to show that player \square has a (ν, ψ) -winning HD strategy in s'_{in} iff player \square has a (P, Q) -winning HD strategy in s'_{in} . We demonstrate both implications separately in subsequent subsections. First, we fix some notation which is used in both parts of our proof.

For a given a f -vertex $(s, A, B, C)^f$, the i -th component of the tuple (s, A, B, C) is denoted $\rho_i((s, A, B, C)^f)$. For example, $\rho_2((s, A, B, C)^f) = A$. An analogous notation

is used also for g -vertices. The last symbol of a given non-empty word w is denoted $last(w)$.

A.1.1 Transferring winning strategies from G' to G .

Let σ' be a (P, Q) -winning HD strategy in s'_{in} . The states in $G'(\sigma')$ of the form $u \cdot (s, A, B, C)^f$ are called f -states, and states of the form $u \cdot (s, A, B, C, \vec{D})^g$ are called g -states.

Let R be the set of all g -states of $G'(\sigma')$ that are reachable from s'_{in} . We define a function $\Lambda : R \rightarrow V^*$ inductively as follows:

- $\Lambda(s'_{in} \cdot \sigma'(s'_{in})) = s_{in}$
- $\Lambda(v \cdot (s, A, B, C)^f (s, A', B', C', \vec{D})^g) = \Lambda(v) \cdot s$

A routine check confirms the following:

Lemma A.4. Λ is injective.

We define a strategy $\sigma : V^*V_{\square} \rightarrow V$ as follows: For a given $v \in \Lambda(R)$ such that $\sigma'(\Lambda^{-1}(v)) = (t, A, B, C)^f$ we put $\sigma(v) = t$. For $v \in (V^*V_{\square}) \setminus \Lambda(R)$ we put $\sigma(v)$ to an arbitrary value (as we show these strings are not states of $G(\sigma)$).

Lemma A.5. $\Lambda(R)$ is precisely the set of states that are reachable from s_{in} in $G(\sigma)$, and for all $v, v' \in \Lambda(R)$ we have $\mathcal{P}(v \rightarrow^* v') = \mathcal{P}(\Lambda^{-1}(v) \rightarrow^* \Lambda^{-1}(v'))$.

Proof. Let $v \in \Lambda(R)$. We show that $v \xrightarrow{x} v'$ iff $v' \in \Lambda(R)$ and $\Lambda^{-1}(v) \xrightarrow{x} u \xrightarrow{1} \Lambda^{-1}(v')$, where u is the unique predecessor of $\Lambda^{-1}(v')$ in $G'(\sigma')$.

Assume that $last(v) = s$ and $last(\Lambda^{-1}(v)) = (s, A, B, C, \vec{D})^g$. There are two possibilities:

- If $s \in V_{\circ}$, then we have that $v \xrightarrow{x} v'$ iff $v' = v \cdot t$ for some $t \in V$ such that $s \xrightarrow{x} t$ iff $\Lambda^{-1}(v) \xrightarrow{x} \Lambda^{-1}(v) \cdot (t, A', B', C')^f \xrightarrow{1} \Lambda^{-1}(v')$
- If $s \in V_{\square}$, then we have that $v \xrightarrow{1} v'$ iff $v' = v \cdot t$ for $t = \sigma(v)$ and $\sigma'(\Lambda^{-1}(v)) = (t, A', B', C')^f$ iff $\Lambda^{-1}(v) \xrightarrow{1} \Lambda^{-1}(v) \cdot (t, A', B', C')^f \xrightarrow{1} \Lambda^{-1}(v')$

It follows that there is a (unique) path $v = v_0, \dots, v_n = v'$ from v to v' iff there is a (unique) path π from $\Lambda^{-1}(v)$ to $\Lambda^{-1}(v')$. Moreover, $\pi = \Lambda^{-1}(v_0), u_1, \Lambda^{-1}(v_1), u_2, \dots, \Lambda^{-1}(v_n)$ and hence by applying the above arguments we obtain that $\mathcal{P}(v \rightarrow^* v') = \mathcal{P}(Run(v_0, \dots, v_n)) = \mathcal{P}(Run(\Lambda^{-1}(v_0), u_1, \Lambda^{-1}(v_1), u_2, \dots, \Lambda^{-1}(v_n))) = \mathcal{P}(\Lambda^{-1}(v) \rightarrow^* \Lambda^{-1}(v'))$. \square

The last technical step is the following lemma:

Lemma A.6. Let $A \subseteq \mathcal{S}$ be a non-empty set, and let $D \in \Theta(A)$ be a set such that $G^{-1}\xi \in D$ or $\bar{F}^{\times\ell}\xi \in D$. Then there is $Y \in \Theta(\{\xi\})$ such that $Y \subseteq D$.

Proof. Let us first consider the case when $G^{-1}\xi \in D$. Let us assume that $G^{-1}\xi \equiv \varphi_i$, and let $\varphi_j \in A$ be the formula with the least index in A . Note that $j \leq i$ because $G^{-1}\xi$ is a subformula of some formula in A . We proceed by induction in $i - j$.

If $i = j$, then by Corollary A.3 we have that

$$\Theta(A) = \{\{G^{-1}\xi\} \cup X \cup Y \mid X \in \Theta(A \setminus \{G^{-1}\xi\}), Y \in \Theta(\xi)\}$$

from which the claim follows. The induction step for $j < i$ is completed simply by considering all possible forms of φ_j and applying the definition of Θ .

The case when $\bar{F}^{\times\ell}(\xi) \in D$ is handled similarly. \square

Now we show the main result of this subsection:

Lemma A.7. $s_{in} \models^v \psi$

Proof. By induction on the structure of ψ we show that for all $\varphi \in \mathcal{S}$ and for all states of R of the form $v = v' \cdot (s, A, B, C)^f (s, A', B', C', \bar{D})^g$ we have that $\Lambda(v) \models^v \varphi$ whenever $\varphi \in A' \cap \mathcal{S}$. Together with Lemma A.1, this implies that $s_{in} \models^v \psi$ because there is a state of the form $(s_{in}, \{\psi\}, \emptyset, \emptyset)^f (s_{in}, A'', B'', C'', \bar{D}')^g$ in R where $A'' \in \Theta(\{\psi\})$.

Since $(s, A, B, C)^f \rightarrow (s, A', B', C', \bar{D})^g$, we have that $A' \in \Theta(A)$. First, observe that if φ is a Boolean combination of some formulae, then $\varphi \notin A' \in \Theta(A)$ and we are done. The other cases are analyzed below.

- If $\varphi \in \mathcal{S}_{Ap}$, then $\Lambda(v) \models^v \varphi$ because otherwise $B' = \{\bullet\}$ by definition of G' . Note that if $B' = \{\bullet\}$ then σ' is not winning strategy for G' because \bullet is never removed and hence the (P, Q) -winning objective is not achieved.
- Let $\varphi \equiv G^{-1}\xi$. First we show that if $G^{-1}\xi \in A'$ then $\Lambda(v) \models^v \xi$. By Lemma A.6 we have that if $G^{-1}\xi \in A'$ then there is $D \in \Theta(\{\xi\})$ such that $D \subseteq A'$. Then, $\Lambda(v)$ satisfies all formulae of $D \cap \mathcal{S}$ by induction hypothesis (note that all formulae of $D \cap \mathcal{S}$ are subformulae of ξ) and hence $\Lambda(v) \models^v \xi$ by applying Lemma A.1. Now observe that for every u reachable from v we have that $G^{-1}\xi \in \rho_2(\text{last}(u))$ which implies that $\Lambda(u) \models \xi$. The rest now follows from Lemma A.5.
- Let $\varphi \equiv \bar{F}^{-1}\xi$. Similarly as for $G^{-1}\xi$ we show that whenever $\bar{F}^{-1}\xi \in A'$, then $\Lambda(v) \models^v \xi$. We prove that almost every run of $\text{Run}(v)$ contains a g -state u such that $\bar{F}^{-1}\xi \in \rho_2(\text{last}(u))$. Let ω be a run initiated in v that does not contain such a state. Observe that for all $i \geq 0$ we have $\bar{F}^{-1}\xi \in \rho_2(\text{last}(\omega(i)))$. However, this means that either $\rho_3(\text{last}(\omega(i))) \neq \emptyset$ for all $i \geq 0$, or there is $j \geq 0$ such that $\rho_3(\text{last}(\omega(j))) = \emptyset$ and for all $k > j$ we have $\bar{F}^{-1}\xi \in \rho_3(\text{last}(\omega(k)))$. In both cases, the run ω does not satisfy

the qualitative Büchi objective specified by the set Q in the mixed winning objective (P, Q) , and we know that the probability of all such runs is 0.

It follows that almost every run initiated in v reaches a state $u \in R$ such that $\Lambda(u) \models^v \zeta$. Hence, $\Lambda(v) \models^v F^{-1}\zeta$ due to Lemma A.5.

- Let $\varphi \equiv F^{>0}\zeta$. Similarly as in the previous cases we show that whenever $\bar{F}^{>0}\zeta \in A'$ then $\Lambda(v) \models^v \zeta$. We prove that there is a state u reachable from v such that $\bar{F}^{>0}\zeta \in \rho_2(\text{last}(u))$. By definition of G' we have that there is a run $\omega \in \text{Run}(v)$ such that either $\bar{F}^{>0}\zeta \in \rho_2(\omega(j))$ for some $j \geq 0$, or $F^{>0}\zeta \in \rho_2(\omega(j))$ for all $j \geq 0$. We show that the latter case contradicts the sure-Büchi objective specified by the set P in the mixed winning objective (P, Q) . There are two possibilities: either $\rho_4(\omega(i)) \neq \emptyset$ for all $i \geq 0$, or there is $j \geq 0$ such that $\rho_4(\omega(j)) = \emptyset$ and for all $k > j$ we have $F^{>0}\zeta \in \rho_4(\omega(j))$. In both cases, the run ω visits the states of P only finitely many times. The rest follows from Lemma A.5. \square

Transferring winning strategies from G to G' .

Let $\sigma : V^*V_\square \rightarrow V$ be a strategy such that $s_{in} \models^v \psi$. Let R be the set of states of $G(\sigma)$ that are reachable from s_{in} .

Lemma A.8. *For all $v \in R$ and $A \subseteq \mathcal{S}$ such that v satisfies all formulae of A there is $D \in \Theta(A)$ such that*

- v satisfies all formulae of $D \cap \mathcal{S}$;
- if $F^{\times e}\zeta \in D$ then $v \not\models^v \zeta$.

Proof. For each $X \in \Theta(A)$ we put $F(v, X) = \{F^{\times e}\zeta \in X \mid v \models^v \zeta\}$, and we denote $\text{rank}(X) = \min\{i \mid \varphi_i \in F(v, X)\}$ (we put $\text{rank}(X) = \infty$ if $F(v, X) = \emptyset$). By Lemma A.1, we know that there is $A' \in \Theta(A)$ such that v satisfies all formulae of $A' \cap \mathcal{S}$. Let us assume that $F(v, A') \neq \emptyset$ (otherwise we put $D = A'$ and we are done). We show that there is $A'' \in \Theta(A)$ such that v satisfies all formulae of $A'' \cap \mathcal{S}$ and $\text{rank}(A'') > \text{rank}(A')$.

Let $1 \leq i \leq n$ be the least index such that $\varphi_i = F^{\times e}\zeta \in F(v, A')$, and let $\varphi_j \in A$ be the formula with the least index in A . Clearly $j \leq i$. We show by induction on $j - i$ that there is $A'' \in \Theta(A)$ such that $\text{rank}(A'') > \text{rank}(A')$.

If $i = j$ then by Corollary A.3 we have that $A' = X \cup \{F^{\times e}\zeta\}$ where $X \in \Theta(A \setminus \{F^{\times e}\zeta\})$. Note that v satisfies all formulae of $X \cap \mathcal{S}$. However, by Corollary A.3 we have that $X \cup \{\bar{F}^{\times e}\zeta\} \cup Y \in \Theta(A)$ for every $Y \in \Theta(\{\zeta\})$. Since $v \models \zeta$, by applying Lemma A.1 we obtain that there is $Y \in \Theta(\{\zeta\})$ such that v satisfies all formulae of $Y \cap \mathcal{S}$, and hence all formulae of $(X \cup Y) \cap \mathcal{S}$. Hence, we put $A'' = X \cup \{\bar{F}^{\times e}\zeta\} \cup Y$.

Observe, that an index of arbitrary formula of $X \cup Y$ is greater than i . In particular, $\text{rank}(A'') > \text{rank}(A')$.

The induction step for $j < i$ is completed by considering all possible forms of φ_j and applying the definition of Θ . □

Definition A.9. Given a formula $\xi \in \mathcal{S}$ and a state v of $G(\sigma)$, we denote $\text{Dist}(v, \xi)$ the length of the shortest path from v to a state which satisfies ξ (we put $\text{Dist}(v, \xi) = \infty$ if there is no such state).

Lemma A.10. Let $vs \in R$ where $s \in V$ and $vs \not\models \xi$. Then the following holds:

- If $s \in V_{\square}$ and $vs \models^v F^{\times e} \xi$, then $vs\sigma(s) \models^v F^{\times e} \xi$.
- If $s \in V_{\circ}$ and $vs \models^v F^{-1} \xi$, then for all $t \in V$ such that $s \rightarrow t$ we have $vst \models^v F^{-1} \xi$.
- If $s \in V_{\circ}$ and $vs \models^v F^{>0} \xi$, then there is $t \in V$ such that $s \rightarrow t$, $vst \models^v F^{>0} \xi$ and $\text{Dist}(vst, \xi) < \text{Dist}(vs, \xi)$.

We define a function $\Gamma : R \rightarrow (V')^*$ and a strategy $\sigma' : (V')^*(V'_{\square}) \rightarrow V'$ inductively as follows: $\Gamma(s_{in}) = (s_{in}, \{\psi\}, \emptyset, \emptyset)^f$. For a given $v \in R$ such that $\text{last}(\Gamma(v)) = (s, A, B, C)^f$ where v satisfies all formulae in A we define $\sigma'(\Gamma(v))$ to be one of the vertices (chosen arbitrarily) of the form $(s, A', B', C', \vec{D})^g$ where

- $A' \in \Theta(A)$, v satisfies all formulae in $A' \cap \mathcal{S}$, and if $F^{\times e} \xi \in A'$ then $v \not\models \xi$.
- The sets B' and C' are determined by A' , B , and C (see the definition of G')
- \vec{D} is compatible with the definition of G' , and if $s \in V_{\circ}$ then $F^{>0} \xi \in \vec{D}_t$ implies that $vt \models F^{>0} \xi$ and $\text{Dist}(vt, \xi) < \text{Dist}(v, \xi)$.

Due to Lemma A.8 and Lemma A.10, there must be at least one vertex of the form $(s, A', B', C', \vec{D})^g$ satisfying the three properties above. Furthermore,

- if $s \in V_{\square}$, we put $\sigma'(\Gamma(v)) \cdot (s, A', B', C', \vec{D})^g = (t, A'', B'', C'')^f$ where $\sigma(v) = t$;
- if $s \in V_{\circ}$, then for all $t \in V$ such that $s \rightarrow t$ we put $\Gamma(vt) = \Gamma(v)(s, A', B', C', \vec{D})^g(t, A'', B'', C'')^f$;
- if $s \in V_{\square}$ and $t = \sigma(v)$, we put $\Gamma(vt) = \Gamma(v)(s, A', B', C', \vec{D})^g(t, A'', B'', C'')^f$

In all these cases, the sets A'', B'', C'' are uniquely determined (see the definition of G'). Note that the invariant “ v satisfies all formulae in A ” is maintained throughout the inductive definition. For other strings of $(V')^*(V'_{\square})$ we define σ' arbitrarily.

Lemma A.11. $\Gamma(R)$ is precisely the set of all f -states of $G'(\sigma')$ that are reachable from s'_{in} . Moreover, for all $v, v' \in R$ we have that $\mathcal{P}(v \rightarrow^* v') = \mathcal{P}(\Gamma(v) \rightarrow^* \Gamma(v'))$.

Proof. It is easy to show that for a given $v \in R$ we have $v \xrightarrow{x} v'$ iff $\Gamma(v) \xrightarrow{x} u \xrightarrow{1} \Gamma(v')$, where u is the unique predecessor of $\Gamma(v')$ in $G'(\sigma')$. The proof can be completed using the same arguments as in the proof of Lemma A.5. \square

Lemma A.12. *Let $\Gamma(v) \in \Gamma(R)$. Then almost all runs of $Run(\Gamma(v))$ contain a state u such that $\rho_3(last(u)) = \emptyset$.*

Proof. Let us suppose that $\sigma'(\Gamma(v)) = (s, A, B, C, \vec{D})^g$ and let $F^{=1}\zeta \in B \subseteq A$. Let v' be a state reachable from v such that $v' \models \zeta$. It follows from the definition of σ' that $\sigma'(\Gamma(v')) = (s, A', B', C', \vec{D})^g$ where $F^{=1}\zeta \notin B'$. Moreover, $\mathcal{P}(\Gamma(v) \rightarrow^* \Gamma(v')) = \mathcal{P}(v \rightarrow^* v')$ due to Lemma A.11. It follows that for every $F^{=1}\zeta \in B$ and for almost all runs of $Run(\Gamma(v))$ there is $i \geq 0$ such that $F^{=1}\zeta \notin \rho_3(last(\omega(i)))$. From this we can deduce that almost all $\omega \in Run(\Gamma(v))$ satisfy the following condition: for every formula $F^{=1}\zeta \in B$ there is $i \geq 0$ such that $F^{=1}\zeta \notin \rho_3(last(\omega(i)))$ (it follows from the fact that the intersection of finitely many sets which have probability 1 has also probability 1). However, this property together with the definition of G' implies that there is $j \geq 0$ such that $\rho_3(last(\omega(j))) = \emptyset$. \square

Lemma A.13. *Let $\Gamma(v) \in \Gamma(R)$. Then every run of $Run(\Gamma(v))$ contains a state u such that $\rho_4(last(u)) = \emptyset$.*

Proof. Let us suppose that $\sigma'(\Gamma(v)) = (s, A, B, C, \vec{D})^g$ and let $F^{>0}\zeta \in C \subseteq A$. Let $\omega \in Run(\Gamma(v))$ be a run such that for all $i \geq 0$ we have that $last(\omega(2i)) = (t_i, g, A_i, B_i, C_i, \vec{D}_i)$ where $F^{>0}\zeta \in C_i$. Since $v \models F^{>0}\zeta$, we have that $Dist(v, \zeta) < \infty$. However, by the definition of σ' we have that $Dist(vt_1 \cdots t_{i+1}, \zeta) < Dist(vt_1 \cdots t_i, \zeta)$ which contradicts the fact that $Dist$ is a non-negative function. From this we obtain that for every $\omega \in Run(\Gamma(v))$ and every formula $F^{>0}\zeta \in C$ there is $i \geq 0$ such that $F^{>0}\zeta \notin \rho_4(last(\omega(i)))$. The definition of G' implies that there is $j \geq 0$ such that $\rho_4(last(\omega(i))) = \emptyset$. \square

Corollary A.14. *The strategy σ' is a (P, Q) -winning strategy in s'_{in} .*

A.2 A Proof of Lemma 3.7

In order to prove Lemma 3.7, we need some technical tools that allow us to do modify Markov chains induced by strategies. Since these Markov chains are trees we start by introducing an auxiliary notion of *probabilistic tree*.

Probabilistic trees.

Let us fix a finite set V . A *probabilistic tree* is a pair $T = (X, P)$ where $X \subseteq V^+$ is a nonempty prefix-closed set containing exactly one $s \in V$ (the root of T) and

$P : (V^+ \times V^+) \rightarrow (0, 1]$ is a function with domain $\{(w, ws) \mid w, ws \in X, s \in V\}$ (i.e., P is defined exactly on edges of T) such that for each $w \in X$ we have $\sum_{ws \in T} P(w, ws) = 1$. Note that P is formally a subset of $(V^+ \times V^+) \times (0, 1]$, and hence we can apply set operations on P . Elements of X are called *nodes* of T , a node ws where $s \in V$ is a *successor* of w , and nodes without any successors are called *leaves*. *Infinite probabilistic trees* are probabilistic trees without leaves. Note that infinite probabilistic trees can be seen as Markov chains. In what follows we do not distinguish between infinite probabilistic trees and their corresponding Markov chains.

Given a node $ws \in X$ where $s \in V$, we define a *subtree* of T in ws as a tuple $T^{ws} = (X^{ws}, P^{ws})$ where $X^{ws} = \{sv \mid wsv \in T\}$ and P^{ws} is induced by P in the obvious way. Given a subtree T^w of T , we denote by $T \setminus T^w$ the tree obtained from T by cutting off the subtree T^w , where only the node w is left. Formally, $T \setminus T^w = (X', P')$ where $X' = X \setminus (\{w\} \cdot V^+)$ and P' is the induced restriction of P .

Given two trees T_1 and T_2 , we define the *concatenation* of T_1 and T_2 , denoted $T_1 \odot T_2$, to be the tree $(X_1 \odot X_2, P_1 \odot P_2)$ where

$$X_1 \odot X_2 = X_1 \cup \{wsv \mid ws \text{ is a leaf of } T_1, sv \in X_2\}$$

and $P_1 \odot P_2$ is induced by P_1 and P_2 in the obvious way.

Finally, given an infinite tree $T = (X, P)$ and two nodes $v, w \in X$ such that $\text{last}(v) = \text{last}(w)$, we denote $T^{w \rightarrow v} = (T \setminus T^v) \odot T^w$ the tree obtained from T by cutting off T^v and pasting T^w to its place.

The proof.

Let us fix a $1\frac{1}{2}$ -player game $G = (V, E, (V_{\square}, V_{\circ}), \text{Prob})$ and a mixed objective (P, Q) . In what follows we say that a strategy is (P, Q) -winning in a vertex s without explicitly mentioning the player who is always player \square . We denote $V^P = \{ws \mid w \in V^*, s \in P\}$ and $V^Q = \{ws \mid w \in V^*, s \in Q\}$.

Given a strategy σ and a vertex $s \in V$ we denote $\mathcal{T}[s, \sigma]$ the part of $G(\sigma)$ that is reachable from s . By a run in $\mathcal{T}[s, \sigma]$ we mean a run from the root s of the tree $\mathcal{T}[s, \sigma]$ unless explicitly stated otherwise. Observe that $\mathcal{T}[s, \sigma]$ is an infinite probabilistic tree.

We show that the existence of a (P, Q) -winning strategy in $s \in V$ it is decidable in polynomial time. We start with the following simple observation:

Lemma A.15. *Given a (P, Q) -winning strategy σ and states $w, v \in \mathcal{T}[s, \sigma]$ such that $\text{last}(w) = \text{last}(v)$, there is a (P, Q) -winning strategy σ' such that $\mathcal{T}[s, \sigma]^{w \rightarrow v} = \mathcal{T}[s, \sigma']$.*

Lemma A.16. *If there is a (P, Q) -winning strategy in $s \in V$, then there is a (P, Q) -winning strategy σ' in s such that there is a run in $\mathcal{T}[s, \sigma']$ that reaches the set V^Q in at most $2|V|$ steps and any run in $\mathcal{T}[s, \sigma']$ reaches V^P in at most $|V|$ steps.*

Proof. Let σ be a (P, Q) -winning strategy in s . First, observe that there is $k \geq 1$ such that an arbitrary run from s in $\mathcal{T}[s, \sigma]$ reaches a state of V^P in less than k steps. Indeed, if this was not true, then we could inductively define an infinite run in $\mathcal{T}[s, \sigma]$ which does not contain any state of V^P , which contradicts the assumption that σ is (P, Q) -winning.

Let us assume that there is a vertex $t \in V$ and a run in $\mathcal{T}[s, \sigma]$ which contains states of the form wt and $wtvt$ before the first occurrence of a state of V^P . Then we can apply Lemma A.15 and obtain a (P, Q) -winning strategy σ' such that $\mathcal{T}[s, \sigma'] = \mathcal{T}[s, \sigma]^{wtvt \rightarrow wt}$. Repeating the procedure finitely many times, we obtain a (P, Q) -winning strategy σ'' such that no run in $\mathcal{T}[s, \sigma'']$ contains a duplicated vertex before the first occurrence of a state of V^P . Hence, the distance of states of V^P from the root of $\mathcal{T}[s, \sigma'']$ is at most $|V|$.

Because σ'' is (P, Q) -winning in s , there is a path ω in $\mathcal{T}[s, \sigma'']$ from s to a state $v \in V^Q$. If there are no states of V^P before v in the path ω then we are done. Otherwise, let us assume that $\omega(i)$ is the first occurrence of a state of V^P in ω and let us consider the suffix $\omega' = \omega(i), \omega(i+1), \dots, \omega(|\omega|-1)$. Duplicate states in ω' can be eliminated using the above “copying” procedure, which yields a strategy σ''' which satisfies the desired properties. \square

Lemma A.17. *If there is a (P, Q) -winning strategy in $s \in V$, then there is a strategy σ' in s such that $\mathcal{T}[s, \sigma']$ satisfies the following conditions:*

1. *Along every run initiated in an arbitrary state of $\mathcal{T}[s, \sigma']$, a state of V^P is reached within at most $2|V|$ steps.*
2. *There is $\varepsilon > 0$ such that the probability of reaching V^Q from any state of $\mathcal{T}[s, \sigma']$ is at least ε .*

Proof. Let us denote $\bar{V} = \{t \mid wt \in X\}$ the set of vertices of G that occur in the play induced by σ . For a given $t \in \bar{V}$, we fix a node w_t of $\mathcal{T}[s, \sigma]$ such that $\text{last}(w_t) = t$. Observe that for each $t \in \bar{V}$ the strategy σ induces a (P, Q) -winning strategy in t which yields the tree T^{w_t} . By Lemma A.16 we obtain that there is a (P, Q) -winning strategy σ_t in t which satisfies the conclusion of Lemma A.16.

We denote $T_t = (X_t, P_t)$ a (finite) probabilistic tree obtained from $\mathcal{T}[t, \sigma_t]$ by cutting off all nodes whose distance from t is greater than $2|V|$. Now, we gradually construct a new infinite tree T' as follows: we start with T_s and then in every stage we concatenate all trees T_t to the current tree. Formally, we define sequences of trees K_0, K_1, K_2, \dots and $K_1^t = (X_1^t, P_1^t), K_2^t = (X_2^t, P_2^t), \dots$ where $K_0 = T_s$ and for all $i \geq 0$ and all $t \in \bar{V}$ we have

- $K_{i+1}^t = K_i \odot T_t$
- $K_{i+1} = (\bigcup_{t \in \bar{V}} X_{i+1}^t, \bigcup_{t \in \bar{V}} P_{i+1}^t)$

Note that $\bigcup_{t \in \bar{V}} P_{i+1}^t$ is a function for every $i \geq 0$, because for a fixed i the functions P_{i+1}^t agree on edges in T_i .

We claim that there is a strategy σ' such that $\mathcal{T}[s, \sigma'] = (\bigcup_{i=1}^{\infty} X_i, \bigcup_{i=1}^{\infty} P_i)$. Indeed, the strategy σ' behaves like σ_s until it reaches a leaf w_t of T_s (i.e., during the first $2|V|$ steps), then it starts to behave like the strategy σ_t for the next $2|V|$ steps, and so on.

Finally, we put $\varepsilon = \min\{p_t \mid t \in \bar{V}\}$ where each p_t equals the probability of reaching a state of V^Q in $\mathcal{T}[t, \sigma_t]$ in at most $2|V|$ steps. Now, it is easy to see that σ' satisfies the conditions 1. and 2. \square

Corollary A.18. *If there is a (P, Q) -winning strategy in $s \in V$, then there is a (P, Q) -winning strategy σ' such that along any run from an arbitrary state of $\mathcal{T}[s, \sigma']$ a state of V^P is reached within at most $2|V|$ steps.*

Proof. It follows from Lemma A.17 using standard tools of probability theory. \square

Theorem A.19. *Let G be a $1\frac{1}{2}$ -player game, s a vertex of G , and (P, Q) a mixed objective. If there is a (P, Q) -winning HD strategy in s , then there is a (P, Q) -winning HD strategy with finite memory (\mathcal{A}, f) whose size is polynomial in the size of G . Moreover, the existence of such a strategy is decidable in P .*

Proof. We construct a new game G' , its vertex s' , and a subset P' of vertices of G' such that player \square has a (P, Q) -winning HD strategy in s iff player \square has a (P', \emptyset) -winning strategy in s' . Thus, we “reduce” (P, Q) -winning objectives to qualitative Büchi objectives for which a polynomial-time algorithm exists.

Corollary A.18 implies that in order to ensure that vertices of P are entered infinitely often along a play, it suffices to count the number of vertices between two successive occurrences of vertices of P , and to give up the whole play whenever this number exceeds $2|V|$.

Let us define a new game $G' = (V', E', (V'_{\square}, V'_{\circ}), Prob')$ (which extends the game G with a counter) where $V' = V \times \{1, \dots, 2|V|, \perp\}$, $V'_{\square} = V_{\square} \times \{1, \dots, 2|V|, \perp\}$, $V'_{\circ} = V' \setminus V'_{\square}$, E' is the least set (w.r.t. \subseteq) such that for each $(u, v) \in E$ we have

- $((u, i), (v, 1)) \in E'$ if $u \in V^P$ and $i \leq 2|V|$
- $((u, i), (v, i+1)) \in E'$ if $u \in V \setminus P$ and $i < 2|V|$
- $((u, 2|V|), (v, \perp)) \in E'$ if $u \in V \setminus P$
- $(u, \perp), (v, \perp) \in E'$

The function $Prob'$ is defined for each $((u, \alpha), (v, \beta)) \in E'$ as $Prob'((u, \alpha), (v, \beta)) = Prob(u, v)$.

Observe that the game G' faithfully simulates the game G in the first component of states, and in the second component it merely counts the actual number of steps

after the last occurrence of a vertex of P . Hence, for every strategy σ' in G' there is a corresponding strategy in G which “forgets” the second component of vertices (and vice versa). If the number of steps outside of P exceeds $2|V|$ then the second component of a state changes to \perp , which remains there forever. Hence, a state containing \perp is reachable (with positive probability) in $\mathcal{T}[(s, 1), \sigma']$ iff there is a path of length greater than $2|V|$ in $\mathcal{T}[(s, 1), \sigma']$ which does not contain states of V^B . Let $P' = Q \times \{1, \dots, 2|V|\}$. Now it is easy to check that there is a (P, Q) -winning strategy in s iff there is a (P', \emptyset) -winning strategy in $s' = (s, 1)$.

The existence of a (P', \emptyset) -winning strategy in s' is decidable in polynomial time [8, 6]. Moreover, if there is *some* winning strategy, then there is also a memoryless and deterministic winning strategy σ' . Hence, we can construct a (P, Q) -winning HD strategy with finite memory for s from the strategy σ' . Since the only information that is kept in memory is the actual value of the counter, we obtain a polynomial upper bound on the size of the memory. \square