Deciding Maxmin Reachability in Half-Blind Stochastic Games*

Edon Kelmendi
LaBRI
Bordeaux, France
edon.kelmendi@labri.fr

Hugo Gimbert
LaBRI & CNRS
Bordeaux, France
hugo.gimbert@labri.fr

May 23, 2016

Abstract

Two-player, turn-based, stochastic games with reachability conditions are considered, where the maximizer has no information (he is blind) and is restricted to deterministic strategies whereas the minimizer is perfectly informed. We ask the question of whether the game has maxmin 1, in other words we ask whether for all $\epsilon > 0$ there exists a deterministic strategy for the (blind) maximizer such that against all the strategies of the minimizer, it is possible to reach the set of final states with probability larger than $1 - \epsilon$. This problem is undecidable in general, but we define a class of games, called leaktight half-blind games where the problem becomes decidable. We also show that mixed strategies in general are stronger for both players and that optimal strategies for the minimizer might require infinite-memory.

*This work was partially supported by the French ANR project "Stoch-MC" and "LaBEX CPU" of Universit de Bordeaux.

Contents

1 Introduction

Two-player stochastic games are a natural framework for modeling and verification in the presence of uncertainty, where the problem of control is reduced to the problem of optimal strategy synthesis [10]. There is a variety of two-player stochastic games that have been studied, depending on the information available to the players (perfect information or partial information), the winning objective (safety, reachability, etc.), the winning condition (surely, almost-surely, or limit-surely winning; probability higher than some quantity), whether the players choose actions concurrently or whether they take turns. Stochastic games with partial observation are particularly well suited for modeling many scenarios occurring in practice; normally we do not know the exact state of the system we are trying to model, e.g. we are aided by noisy sensors or by a software interface that provides only a partial picture. Unfortunately, compared to perfect information games, algorithmic problems on partial information games are substantially harder and often undecidable [3, 18, 16]. Assuming one player to be perfectly informed while the other player is partially informed (semiperfect-information games [5, 4]) brings some relief to the computational hardness as opposed to general partial information games.

In the present paper we consider half-blind stochastic games: one player has no information (he is blind) and plays deterministically while the other player is perfectly informed. We study half-blind games for the reachability objective and maxmin winning condition: we want to decide if for every $\epsilon > 0$ there exists a deterministic strategy for the maximizer such that against all strategies of the minimizer, the final states are reached with probability at least $1 - \epsilon$.

The maxmin condition for half-blind games is a generalization of the value 1 problem for probabilistic finite automata [20]. Most decision problems on probabilistic finite automata are undecidable, notably language emptiness [18, 1, 16], and the value 1 problem [16]. Consequently, stochastic games with partial information and quantitative winning conditions (the probability of fulfilling the winning objective is larger than some quantity) are undecidable. Nevertheless recently there has been some effort on characterizing decidable classes of probabilistic automata [16, 6, 2, 13, 11], with the leaktight class [13] subsuming the others [12].

Our results. In the present paper we show that a subclass of half-blind games called leaktight games have a decidable maxmin reachability problem. The game is abstracted through a finite algebraic structure called the belief monoid. This is an extension to the Markov monoid used in [13]. Indeed the elements of the belief monoid are sets of elements of the Markov monoid, and they contain information on the outcome of the game when one strategy choice is fixed. The algorithm builds the belief monoid and searches for particular elements which are witnesses that the set of final states is maxmin reachable. The proof of the correctness of the algorithm uses k-decomposition trees, a data structure used in [9] that is related to Simon's factorization forests. The k-decomposition trees are used to prove lower and upper bounds on certain outcomes of the game and show that it behaves as predicted by the belief monoid.

Comparison with previous work. The proof methods extends those developped in [13] in three aspects. First, we define a new monoid structure on top of the Markov monoid structure introduced in [13]. Second, we rely on the extension of Simon's factorization forest theorem [21] to k-factorization trees instead of 2-factorization trees in [13] in order to derive upper and lower bound on the actual probabilities abstracted by the belief monoid. Third, we rely on the leaktight hypothesis to prove both completeness and soundness, while in the case of probabilistic automata the soundness of the abstraction by the Markov monoid was for free.

Outline of the paper. We start by fixing some notions and notation in Section 2 as well as providing a couple of examples. In Section 3 we introduce the belief monoid algorithm and the Markov and belief monoids themselves. The k-decomposition tree data structure used in the proofs of correctness is introduced in Section 4, then in Section 5 the class of leaktight games is defined using the notion of a leak. The correctness of the algorithm is proved in Section 6, and finally we discuss the power of different types of strategies in Section 7 and conclude.

2 Half-Blind Games and the Maxmin Reachability Problem

Given a set X, we denote by $\Delta(X)$ the set of distributions on X, i.e. functions $f : X \to [0,1]$ such that $\sum_{x \in X} f(x) = 1$.

A *half-blind* game is a two-player, zero-sum, stochastic, turn-based game, played on a finite bipartite graph, where the maximizer has no information, whereas the minimizer has perfect information. Formally a game G is given by the tuple $G = (\mathbf{S_1}, \mathbf{S_2}, \mathbf{A_1}, \mathbf{A_2}, p, F)$. The finite set $\mathbf{S_i}$ is the states controlled by Player i, the finite set $\mathbf{A_i}$ is the actions available to Player i ($i = 1, 2$). Player 1 is the maximizer and Player 2 is the minimizer. The function p mapping $(\mathbf{S_1}, \mathbf{A_1})$ to $\Delta(\mathbf{S_2})$ and $(\mathbf{S_2}, \mathbf{A_2})$ to $\Delta(\mathbf{S_1})$ gives the dynamics of the game. The sets $\mathbf{S_1}, \mathbf{S_2}$ and $\mathbf{A_1}, \mathbf{A_2}$ are disjoint, i.e. $\mathbf{S_1} \cap \mathbf{S_2} = \emptyset$ and $\mathbf{A_1} \cap \mathbf{A_2} = \emptyset$. The set $F \subseteq \mathbf{S_1}$ is the set of final states.

A play of such a game takes place in turns. Initially the game is in some state $s_1 \in \mathbf{S_1}$, then the maximizer (a.k.a. player 1) chooses some action $a_1 \in \mathbf{A_1}$ which moves the game to some state $t_1 \in \mathbf{S_2}$ selected randomly according to the lottery $p(s_1, a_1)$. It is up to the minimizer (a.k.a. player 2) now to choose some action $b_1 \in \mathbf{A_2}$ which moves the game to some state $s_2 \in S_1$. Then again maximizer chooses some action $a_2 \in \mathbf{A_1}$ and so on, until the maximizer decides to stop, at which point, if the game is in a state that belongs to the set of final states F, the maximizer wins, otherwise it is the minimizer who wins. The maximizer is totally blind and does not know what happens, he does not know in which state the game is nor the actions played by minimizer. Moreover the maximizer plays in a deterministic way, he is not allowed to use a random generator to select his actions. As a consequence, the decisions of maximizer

only depend on the time elapsed and can be represented as words on $\mathbf{A_1}$. On the other hand, the minimizer has full information and is allowed to plays actions selected randomly.

Formally, the set of strategies for the maximizer is denoted by Σ_1 they consist of finite words, i.e. $\Sigma_1 = \mathbf{A_1}^*$. In order to emphasize that the strategies of the maximizer are words, elements of Σ_1 are usually denoted by w.

The minimizer's strategies are functions from $\mathcal{H} = (\mathbf{S_1}\mathbf{A_1}\mathbf{S_2}\mathbf{A_2})^*\mathbf{S_1}$ to $\Delta(\mathbf{A_2})$. Let Σ_2 be the set of such strategies. Its elements are typically denoted by τ.

Fixing strategies $w \in \Sigma_1$ of length n, $\tau \in \Sigma_2$ and an initial state $s \in \mathbf{S_1}$ gives a probability measure on the set $\mathcal{H}_n = (\mathbf{S_1}\mathbf{A_1}\mathbf{S_2}\mathbf{A_2})^n\mathbf{S_1}$ which is denoted by $\mathbb{P}_s^{w,\tau}$: for a history $h = s_1a_1t_1b_1 \cdots s_na_nt_nb_ns_{n+1} \in \mathcal{H}_n$,

$$\mathbb{P}_s^{w,\tau}(h) = \prod_{i=1}^{n} p(s_i, a_i)(t_i) \cdot \tau(h_i)(b_i) \cdot p(t_i, b_i)(s_{i+1})$$

if $s = s_1$ and $w = a_1 \cdots a_n$, and 0 otherwise, where $h_i = s_1a_1t_1b_1 \cdots s_ia_it_i$, $1 \leq i \leq n$.

For $t \in \mathbf{S_1}$, we will denote by $\mathbb{P}_s^{w,\tau}(t)$ the chance of ending up in state t after starting from state s and playing the respective strategies, i.e $\mathbb{P}_s^{w,\tau}(t) = \sum_{ht \in \mathcal{H}} \mathbb{P}_s^{w,\tau}(ht)$. Whereas for a set of states $R \subseteq \mathbf{S_1}$ let $\mathbb{P}_s^{w,\tau}(R) = \sum_{t \in R} \mathbb{P}_s^{w,\tau}(t)$.

2.1 The Maxmin Reachability Problem

Now we can introduce the maxmin reachability and for half-blind games, using the notation and notions just defined. Given a game with initial state $s \in \mathbf{S_1}$ and final states $F \subseteq \mathbf{S_1}$, the maxmin value $\underline{val}(s)$ is defined by

$$\underline{val}(s) = \sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(F) \ .$$

In case $\underline{val}(s) = 1$, we say that F is maxmin reachable from s.

Problem 1 (Maxmin reachability)**.** *Given a game, is the set of final states F maxmin reachable from the initial state s?*

There is no hope to decide this problem in general. The reason is that in the special case where the minimizer has no choice in any of the states that she controls, then Problem 1 is equivalent to the value one problem for *probabilistic finite automata* which is already known to be undecidable [16]. However, in the present paper, we establish that Problem 1 is decidable for a subclass of half-blind games called leaktight games.

2.2 Deterministic Strategies for the Minimizer

In general, strategies of the minimizer are functions from $\mathcal{H} = (\mathbf{S_1}\mathbf{A_1}\mathbf{S_2}\mathbf{A_2})^*\mathbf{S_1}$ to $\Delta(\mathbf{A_2})$. However, because in the present paper we focus on the maxmin reachability problem, we can assume that strategies of the minimizer have a

much simpler form: the choice of action by the minimizer is deterministic and only depends on the current state and on how much time has elapsed since the beginning of the play. Formally, we assume that minimizer strategies are functions $\mathbb{N} \to (\mathbf{S_2} \to \mathbf{A_2})$. Denote Σ_2^p the set of all such strategies. This restriction of the set of minimizer strategies does change the answer to the maxmin reachability problem because of the following theorem.

Theorem 1. *Given a game with initial state $s \in \mathbf{S_1}$ and final states $F \subseteq \mathbf{S_1}$ we have*

$$\sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2^p} \mathbb{P}_s^{w,\tau}(F) = \sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(F).$$

Proof. Fixing a word $w \in \Sigma_1$ of length n, one can construct an MDP of finite horizon with state-space $\mathbf{S_2} \times \{1, \ldots, n\}$ and safety objective. Stationary strategies suffice to reach the safety objective here (see e.g. [14]). A stationary strategy in this MDP is interpreted as a strategy in Σ_2^p for the half-blind game. □

2.3 Two Examples

The graph on which a half-blind game is played is visualized as in Figures 1 and 2. The circle states are controlled by the maximizer, and the square states are controlled by the minimizer, so for the example in Figure 1, $\mathbf{S_1} = \{i, f\}$ and $\mathbf{S_2} = \{1, 2\}$. We represent only edges (s, t) such that $p(s, a)(t) > 0$ for some action a and we label the edge (s, t) by a if $p(s, a)(t) = 1$ and by $(a, p(s, a, t))$ otherwise.

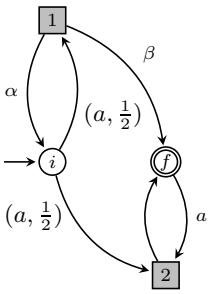

Figure 1: A half-blind game with $\underline{val}(i) = 1$.

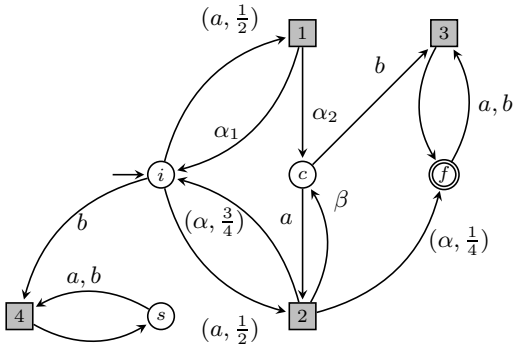

Figure 2: A half-blind game with $\underline{val}(i) < 1$.

For the game in Figure 1 it is easy to see that $\underline{val}(i) = 1$, since if the maximizer plays the strategy a^n, no matter what strategy the minimizer chooses the probability to be on the final state is at least $1 - \frac{1}{2^n}$. On the other hand in the game depicted on Figure 2, $\{f\}$ is not maxmin reachable from i. If

the maximizer plays a strategy of only a's then the minimizer always plays the action β and α_1 for example and the probability to be in the final state will be 0. Therefore the maximizer has to play a b at some point. But then the strategy of the minimizer will be to play β except against the action just before b, against that action the minimizer plays α letting at most $1/4$ of the chance to go to the final state, but making sure that the rest of the probability distribution is stuck in the sink state s. Consequently $\underline{val}(s) = 1/4$. It is interesting to note that in the example in Figure 2, if we fix a strategy for the minimizer first, then for all $\epsilon > 0$ the maximizer can make the probability of reaching the final state to be at least $1 - \epsilon$ by playing enough a's to make sure that the token is either in c or in f and at that point playing b, therefore f is minmax reachable from i, but it is not maxmin reachable. This is discussed in more details in Section 7.

We refer back to the game in Figure 2 in order to illustrate the belief monoid algorithm in the next section.

3 The Belief Monoid Algorithm

We abstract the game using two (finite) monoid structures that are constructed, one on top of the other. Given that the game belongs to the class of leaktight games, the monoids will contain enough information to decide maxmin reachability.

3.1 The Markov Monoid

The Markov monoid is a finite algebraic object that is in fact richer than a monoid; it is a *stabilisation* monoid (see [8]). The Markov monoid was used in [13] to decide the value 1 problem for leaktight probabilistic automata on finite words.

Elements of the Markov monoid are $\mathbf{S_1} \times \mathbf{S_1}$ binary matrices. They are typically denoted by capital letters such as U, V, W. The entry that corresponds to the states $s, t \in \mathbf{S_1}$ is denoted by $U(s,t)$. We will make use of the notation $s \xrightarrow{U} t$ in place of $U(s,t) = 1$, when it is helpful.

We define two operations on these matrices: the product and the iteration.

Definition 1. *Given two $\mathbf{S_1} \times \mathbf{S_1}$ binary matrices U, V, their* product *(denoted UV) is defined for all $s, t \in \mathbf{S_1}$ as*

$$UV(s,t) = \begin{cases} 1 & \text{if } \exists s' \in \mathbf{S_1},\ s \xrightarrow{U} s' \wedge s' \xrightarrow{V} t = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Given a $\mathbf{S_1} \times \mathbf{S_1}$ binary matrix U that is idempotent, i.e. $U^2 = U$, its iteration *(denoted $U^\#$) is defined for all $s, t \in \mathbf{S_1}$ as*

$$U^\#(s,t) = \begin{cases} 1 & \text{if } s \xrightarrow{U} t \text{ and } t \text{ is } U\text{-recurrent,} \\ 0 & \text{otherwise.} \end{cases}$$

We say that some state $t \in \mathbf{S_1}$ is U-recurrent, if for all $t' \in \mathbf{S_1}$, $t \xrightarrow{U} t' \implies t' \xrightarrow{U} t$. Otherwise we say that t is U-transient.

For a set X of binary matrices, we denote $\langle X \rangle$ the smallest set of binary matrices containing X and closed under product and iteration. Let $B^{a,\tau}$, $a \in \mathbf{A_1}$, $\tau \in \Sigma_2^p$ be a matrix defined by $s \xrightarrow{B^{a,\tau}} t \iff \mathbb{P}_s^{a,\tau}(t) > 0$, $s, t \in \mathbf{S_1}$. Now the definition of the Markov monoid can be given.

Definition 2 (Markov monoid)**.** *The Markov monoid denoted $\mathcal{M}$ is*

$$\mathcal{M} = \langle \{B^{a,\tau} \mid a \in \mathbf{A_1}, \tau \in \Sigma_2^p\} \cup \{\mathbf{1}\} \rangle \ ,$$

where $\mathbf{1}$ is the unit matrix.

3.2 The Belief Monoid

Roughly speaking, while the elements of the Markov monoid try to abstract the outcome of the game when both strategies are fixed, the belief monoid tries to abstract the *possible outcomes* of the game when only the strategy of the maximizer is fixed. Hence the elements of the belief monoid are subsets of $\mathcal{M}$, and they are typically denoted by boldfaced lowercase letters such as $\mathbf{u}, \mathbf{v}, \mathbf{w}$.

Given two elements of the belief monoid $\mathbf{u}$ and $\mathbf{v}$, their product is the product of their elements, while the iteration of some idempotent $\mathbf{u}$ is the sub-Markov monoid that is generated by $\mathbf{u}$ minus the elements in $\mathbf{u}$ that are not iterated.

Definition 3. *Given $\mathbf{u}, \mathbf{v} \subseteq \mathcal{M}$, their product (denoted $\mathbf{uv}$) is defined as*

$$\mathbf{uv} = \{UV \mid U \in \mathbf{u}, V \in \mathbf{v}\}.$$

Given $\mathbf{u} \subseteq \mathcal{M}$ that is idempotent, i.e. $\mathbf{u}^2 = \mathbf{u}$, its iteration (denoted $\mathbf{u}^\#$) is defined as

$$\mathbf{u}^\# = \left\langle \left\{ UE^\#V \mid U, E, V \in \mathbf{u}, EE = E \right\} \right\rangle \ .$$

Given $a \in \mathbf{A_1}$, let $\mathbf{a} = \{B^{a,\tau} \mid \tau \in \Sigma_2^p\}$; we give the definition of the belief monoid.

Definition 4 (Belief Monoid)**.** *The belief monoid, denoted $\mathcal{B}$, is the smallest subset of $2^{\mathcal{M}}$ that is closed under product and iteration and contains $\{\mathbf{a} \mid a \in \mathbf{A_1}\} \cup \{\{\mathbf{1}\}\}$, where $\mathbf{1}$ is the unit matrix.*

We are interested in a particular kind of elements in the belief monoid, called *reachability witnesses*.

Definition 5 (Reachability Witness)**.** *An element $\mathbf{u} \in \mathcal{B}$ is called a reachability witness if for all $U \in \mathbf{u}$, $s \xrightarrow{U} t \implies t \in F$, where s is the initial state of the game and F is the set of final states.*

We give an informal description of the way that the belief monoid abstracts the outcomes of the game. Roughly speaking the strategy choice of the maximizer corresponds to choosing an element $\mathbf{u} \in \mathcal{B}$ while the strategy choice of the minimizer corresponds to picking some $U \in \mathbf{u}$. Consequently under those strategy choices, U will tell us the outcome of the game, that is to say if for some $s, t \in \mathbf{S_1}$, if we have $s \xrightarrow{U} t$ then there is some positive probability (larger than a uniform bound) of going from the state s to the state t. In case of $s \not\xrightarrow{U} t$ we will be ensured that the probability of reaching the state t from s can be made arbitrarily small. Therefore if a reachability witness is found then we will know that for any strategy that the minimizer picks the probability of going to some non-final state from the initial state can be made to be arbitrarily small.

3.3 The Belief Monoid Algorithm

Algorithm 1: The belief monoid algorithm.

Data: A half-blind game.
Result: Answer to the Maxmin reachability problem.
$\mathcal{B} \leftarrow \{\mathbf{a} \mid a \in \mathbf{A_1}\}$.
Close $\mathcal{B}$ by product and iteration
Return true iff there is a reachability witness in $\mathcal{B}$

The belief monoid associated with a given game is computed by the belief monoid, see Algorithm 1. We will see later that under some condition, the belief monoid algorithm decides the maxmin rechability problem.

We illustrate the computation of the belief monoid with an example. Consider the game represented on Figure 2. The minimizer has four pure stationary strategies $\tau_{\alpha_1\alpha}$, mapping 1 to α_1 and 2 to α, and similarly the strategies $\tau_{\alpha_1\beta}, \tau_{\alpha_2\alpha}, \tau_{\alpha_2\beta}$. Now we compute $B^{a,\tau}$ where τ is one of the strategies above. Assume that we have the following order on the states: $i < c < s < f$, then $B^{a,\tau_{\alpha_1\alpha}} = \begin{bmatrix} 1&0&0&1\\1&0&0&1\\0&0&1&0\\0&0&0&1 \end{bmatrix}$, $B^{a,\tau_{\alpha_1\beta}} = \begin{bmatrix} 1&1&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{bmatrix}$, $B^{a,\tau_{\alpha_2\alpha}} = \begin{bmatrix} 1&1&0&1\\1&0&0&1\\0&0&1&0\\0&0&0&1 \end{bmatrix}$, and $B^{a,\tau_{\alpha_2\beta}} = \begin{bmatrix} 0&1&0&0\\0&1&0&1\\0&0&1&0\\0&0&0&1 \end{bmatrix}$. The set that contains these matrices is the set $\mathbf{a}$. We can verify that $\mathbf{a}$ is not idempotent, since $U = B^{a,\tau_{\alpha_1\alpha}} B^{a,\tau_{\alpha_1\beta}} \notin \mathbf{a}^2$, and the same for $V = B^{a,\tau_{\alpha_1\alpha}} B^{a,\tau_{\alpha_2\beta}}$. In fact $\mathbf{a}^2 = \mathbf{a} \cup \{U, V\}$. The set $\mathbf{a}^2$ on the other hand is closed under taking products, i.e. $\mathbf{a}^4 = \mathbf{a}^2$. Therefore we can take its iteration and compute the element $(\mathbf{a}^2)^\#$. The reader can verify that $(\mathbf{a}^2)^\#$ contains $(B^{a,\tau_{\alpha_1\alpha}})^\# = \begin{bmatrix} 0&0&0&1\\0&0&0&1\\0&0&1&0\\0&0&0&1 \end{bmatrix}$, $(B^{a,\tau_{\alpha_1\beta}})^\# = \begin{bmatrix} 0&1&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1 \end{bmatrix}$, V and $B^{a,\tau_{\alpha_2\beta}}$. But it also contains $(B^{a,\tau_{\alpha_1\beta}})^\# B^{a,\tau_{\alpha_1\alpha}} = B^{a,\tau_{\alpha_1\alpha}}$. Therefore $(\mathbf{a}^2)^\#\mathbf{b}$ is not a reachability witness because if we pick $A = B^{a,\tau_{\alpha_1\alpha}}$ in $(\mathbf{a}^2)^\#$ and some $B \in \mathbf{b}$, we will have $i \xrightarrow{AB} s$, and s is a sink state.

This roughly tells us that maximizer cannot win with the strategies $\left((a^{2n}b)\right)_n$, because against $a^{2n}b$ the minimizer plays the strategy $\tau_{\alpha_1\beta}$ for the first $2n-1$ turns and then plays the strategy $\tau_{\alpha_1\alpha}$ against the last a, making sure that after the b is played the we end up in the sink state s with at least $3/4$ probability. Continuing the computation we can verify that the belief monoid of the game in Figure 2 does not contain a reachability witness.

3.4 The Extended Markov and Belief Monoids

For defining leaktight half-blind games and in general for the proofs of correctness of the belief monoid algorithm we use the *extended* Markov and belief monoids. In simple words this means that we remember the transitions which were deleted by the iteration operation. This extension is necessary for detecting leaks which will be defined in Section .

The elements of the extended Markov monoid are pairs $(U, \widetilde{U})$ of $\mathbf{S_1} \times \mathbf{S_1}$ binary matrices where the right entry is not modified by the iteration operation and stores the edges that were deleted from the left entry by the iteration operation. Given two such pairs $(U, \widetilde{U})$ and $(V, \widetilde{V})$, define their product to be $(U, \widetilde{U}) \cdot (V, \widetilde{V}) = (UV, \widetilde{U}\widetilde{V})$. Given an idempotent $(E, \widetilde{E})$, define its iteration to be $(E, \widetilde{E})^{\#} = (E^{\#}, \widetilde{E})$.

Definition 6 (Extended Markov Monoid)**.** *The extended Markov monoid (denoted $\widetilde{\mathcal{M}}$) is the smallest set that is closed under product and iteration and contains $\{(B^{a,\tau}, B^{a,\tau}) \mid a \in \mathbf{A_1}, \tau \in \Sigma_2^p\} \cup \{(\mathbf{1}, \mathbf{1})\}$, where $\mathbf{1}$ is the unit matrix.*

The definition of the *extended* belief monoid (denoted $\widetilde{\mathcal{B}}$) remains the same as that of the belief monoid except that its elements are now subsets of $\widetilde{\mathcal{M}}$.

We give a few properties of the belief monoid that we use in the sequel and leave their proofs as an exercise.

Lemma 1. *Let $\mathbf{e} \in \widetilde{\mathcal{B}}$ be an idempotent element of the extended belief monoid. Then the following hold: (1) $\widetilde{\mathcal{B}}$ together with the unit element $\{\{\mathbf{1}\}\}$ is a monoid; (2) $\mathbf{e}^{\#}$ is idempotent; (3) $(\mathbf{e}^{\#})^{\#} = \mathbf{e}^{\#}$ and (4) $\mathbf{e}\mathbf{e}^{\#} = \mathbf{e}^{\#}\mathbf{e} = \mathbf{e}^{\#}$.*

The same properties also hold in the extended Markov monoid since it is a stabilisation monoid [13].

4 k-Decomposition Trees

The notion of k-decomposition trees was introduced in [9]. A k-decomposition tree is a data structure for factorizing finite words into factors that are iterated with respect to some finite monoid. In Section 6 we will use a variant of Simon's factorization forest theorem in order to bound the heights of k-decomposition trees, which in turn will be used to obtain upper and lower bounds on the probability of certain outcomes of the game.

Let A be a finite set, $(M, \cdot)$ a finite monoid and ϕ a morphism from the free monoid of A (i.e. A^*) to M. The set A^* is infinite while M is finite, so a pigeon-hole principle tells us that if we have a word w that is long enough it contains some factors $w_1, \ldots, w_n$ such that $\phi(w_1) = \cdots = \phi(w_n)$. Simon's forest factorization theorem is a very strong extension of this principle. It inductively factorizes the factors themselves in a tree whose height is bounded by a function of the size of the monoid independently of the length of the word w. Similarly to [21, 9, 13] we modify slightly this result to take into account the fact that $\mathcal{B}$ and $\mathcal{M}$ are not only monoids but they have some more structure.

First we define k-decomposition trees.

Definition 7 (k-decomposition Tree)**.** *Let A be a finite alphabet, $(M, \cdot)$ a finite monoid, equipped with a unary operation $\#$ that maps idempotents of M to themselves: $\# : E(M) \to E(M)$ and ϕ a morphism from A^* to M. The nodes of the k-decomposition tree are labeled by pairs (u, U), where $u \in A^*$ and $U \in M$. The right entry of the pair is called the* type *of the node. Let $k > 2$ and $w \in A^*$, then a k-decomposition tree of w with respect to M is a rooted and unranked tree whose root node is labeled by (w, W) for some $W \in M$ and every node is one of the following kinds: (1)* leaves *do not contain any children and are labeled by $(a, \phi(a))$ for $a \in A$; (2)* product nodes *have exactly two children, the left one labeled by (u, U) and right one by (v, V). The node itself is labeled by (uv, UV); (3)* idempotent nodes *have at most $k-1$ children labeled by $(u_1, E), \ldots, (u_j, E)$ where $E \in E(M)$ is idempotent and $j < k$. The node itself is labeled by $(u_1 \cdots u_j, E)$ and* iteration nodes *that have at least k children labeled by $(u_1, E), \ldots, (u_j, E)$ where E is idempotent and $j \geq k$. The node itself is labeled by $(u_1 \cdots u_j, E^{\#})$.*

The notion of a k-decomposition tree is introduced in [9], where it is shown that for all $w \in A^*$ and $k > 2$ there exists a k-decomposition tree whose height depends only on the size of M and not the length of the word — given that M is a stabilisation monoid. We provide a similar proof, for a slightly more general class of monoids that have the properties (1)-(4) given in Lemma 1 whereas the definition of a stabilisation monoid requires extra axioms. The proof was also given in [13] for the case $k = 3$.

Theorem 2 ([21, 9, 13])**.** *Let A be a finite alphabet, $(M, \cdot)$ a monoid equipped with a unary operator $\#$ that maps the idempotents of M to themselves and has the properties (1)-(4) given in Lemma 1, and ϕ a morphism from A^* to M. For all $w \in A^*$, $k > 3$ there exists a k-decomposition tree of w with respect to M whose height is at most $3 \cdot |M|^2$.*

We give a proof in the section that follows.

We will use k-decomposition trees in both the proof of soundness of the belief monoid algorithm in Section 6.1, and its completeness in Section 6.2. For soundness we construct k-decomposition trees for words over the alphabet whose letters are pairs, where the left component is a letter in $\mathbf{A_1}$ and the right component is a stationary strategy for the minimizer, with respect to the extended

Markov monoid $\widetilde{\mathcal{M}}$. On the other hand for completeness we use k-decomposition trees over the alphabet $\mathbf{A_1}$ with respect to the monoid $\widetilde{\mathcal{B}}$. The k-decomposition trees are used to prove lower and upper bounds on the probabilities of certain outcomes.

4.1 The Height of k-decomposition Trees

This section is devoted to proving Theorem 2.

We start with Simon's factorization forest theorem. A *Ramseyan decomposition tree* is the same as a k-decomposition tree except that it does not have iteration nodes, and there is no restriction on the number of children of idempotent nodes.

Let A be a finite alphabet, $(M, \cdot)$ a finite monoid, and ϕ a morphism from A^* to M. Then Simon's factorization forest theorem says:

Theorem 3 ([21]). *For all $w \in A^*$ there exists a Ramseyan decomposition tree of w with respect to M whose height is at most $3 \cdot |M|$.*

Let $\#$ be a mapping from the idempotent elements of M to themselves such that the properties (1) through (4) in Lemma 1 hold. We will prove Theorem 2. Let $w \in A^*$ and $k > 2$. We will prove that there exists a k-decomposition tree of height at most $3 \cdot J \cdot |M|$, where J is the number of $\mathcal{J}$-classes[1] which is smaller than $|M|$.

According to Simon's factorization theorem there exists a Ramseyan decomposition tree T of w of height at most $3 \cdot |M|$.

Let $\phi_0 = \phi$ and $A_0 = A$.

Call any idempotent node with children $(u_1, E), (u_2, E), \ldots, (u_j, E)$, a *primitive iteration node* if $E^\# \neq E$ and $j \geq k$. If T does not have any primitive iteration node, then $T_0 = T$ itself is a k-decomposition tree, and we are done. Otherwise *for all* primitive iteration nodes that are maximal in depth — i.e. there are no other primitive iteration nodes below — labeled (w, E) with children labeled $(w_1, E), \ldots, (w_j, E)$ where $w = w_1 \cdots w_j$ and $j \geq k$, add a new letter of the alphabet $A_1 = A_0 \cup \{a_w\}$, and change the morphism $\phi_1(a_w) = E^\#$ and $\phi_1(v) = \phi_0(v)$ for all other $v \in A_0^*$. The element $E^\#$ is in the monoid M since E is idempotent. Also transform the word u by replacing the factor w by the letter a_w and call this word u_1.

Now from Theorem 3 applied to M with alphabet A_1, morphism ϕ_1 and word u_1 there exists a Ramseyan decomposition tree T_1 of height at most $3 \cdot |M|$ where now the factor w in T_0 is replaced by the leaf $(a_w, E^\#)$. If T_1 does not contain any primitive iteration node then we are done, we can *unwrap* the leaf $(a_w, E^\#)$ by replacing it with the subtree of T_0 rooted in the primitive iteration node (w, E), except that it keeps the label $(w, E^\#)$. But if T_1 contains some primitive iteration node then we recurse the process described above which returns an new alphabet A_2, morphism ϕ_2 and Ramseyan decomposition tree T_2.

[1] $\mathcal{J}$-classes are an important notion in the study of finite semi-groups and monoids. We give precise definitions below.

Since we are removing more and more factors of the word u and adding them as new letters, repeating the procedure described above, must produce some T_k that does not contain any primitive iteration nodes. We claim that

Claim 1. *$k \leq J$ where J is the number of $\mathcal{J}$-classes of M.*

So the number of times that we recurse the procedure above to transform a Ramseyan decomposition tree to a k-decomposition tree whose height does not depend on the length of the word [2] but rather on the structure of M itself. In fact with Claim 1 the k-decomposition tree will have height at most $3 \cdot J \cdot |M| \leq 3 \cdot |M|^2$.

To prove Claim 1 we need some results in the theory of finite semigroups, in particular the Green's relations.

Let $U \in M$ an element of the monoid and define $UM = \{UU' \mid U' \in M\}$ and $MUM = \{VUV' \mid V, V' \in M\}$. Green's relations are four relations of equivalence on the elements of M, denoted $\mathcal{L}, \mathcal{R}, \mathcal{J}, \mathcal{H}$ and $\mathcal{D}$ defined as follows. For a more detailed account of the Green's relations and main theorems on finite semigroups see e.g. [7, 19] etc.

Definition 8 (Green's relations)**.** *Let $U, V \in M$,*

- $U \mathcal{L} V \iff MU = MV$,
- $U \mathcal{R} V \iff UM = VM$,
- $U \mathcal{J} V \iff MUM = MVM$,
- $U \mathcal{H} V \iff U \mathcal{L} V$ *and* $U \mathcal{R} V$,
- $U \mathcal{D} V \iff \exists W \in M, U \mathcal{R} W$ *and* $W \mathcal{L} V \iff \exists W \in M, U \mathcal{L} W$ *and* $W \mathcal{R} V$.

Where the last equivalence is because the relations $\mathcal{R}$ and $\mathcal{L}$ commute. Using these relations we can form partial orders $\leq_{\mathcal{L}}, \leq_{\mathcal{R}}, \leq_{\mathcal{J}}$, so that $U \leq_J V$ if and only if $MUM \subseteq MVM$ and so on.

Observe that $UE^{\#}V \leq_{\mathcal{J}} E^{\#}$, $UE^{\#} \leq_{\mathcal{J}} E^{\#}$, $E^{\#}V \leq_{\mathcal{J}} E^{\#}$ for any two elements $U, V \in M$, so taking the product of $E^{\#}$ with any other element, will produce another element of the monoid that is smaller with respect to the relation $\leq_{\mathcal{J}}$. Now we will show that for any idempotent $E \in M$ if $E^{\#} \neq E$ then $E^{\#} <_{\mathcal{J}} E$. This is Lemma 3 in [21]. Indeed the procedure above, when transforming primitive iteration nodes, it replaces the label from (w, E) to $(w, E^{\#})$, hence the number of times that this can be done is bounded by the number of $\mathcal{J}$-classes hence the Claim 1.

Before we continue with the proof we need two lemmata from the theory of finite monoids and semigroups.

Lemma 2. *No $\mathcal{H}$-class contains more than one idempotent element.*

Lemma 3. *Let $U, V \in M$,*

[2] Notice that for this to be true at each step we have transform *all the primitive iteration nodes of maximal depth and not one by one.*

- *If* $U \leq_{\mathcal{L}} V$ *and* $U\mathcal{J}V$ *then* $U\mathcal{L}V$.
- *If* $U \leq_{\mathcal{R}} V$ *and* $U\mathcal{J}V$ *then* $U\mathcal{R}V$.

Proofs of Lemma 2 and Lemma 3 can be found on any textbook on semigroup theory e.g. [7],[19].

Now we are ready to prove that when iterating we descend the $\mathcal{J}$-classes.

Lemma 4. *Let* $E \in M$ *an idempotent element such that* $E^\# \neq E$. *Then* $E^\# <_{\mathcal{J}} E$.

Proof. Since M fulfills the properties in Lemma 1, in particular property (4), $EE^\#E = E^\#$ hence it follows that $E^\# \leq_{\mathcal{J}} E$. We assume $E\mathcal{J}E^\#$ and get a contradiction. Regard that $E \leq_{\mathcal{L}} E^\# = EE^\#$, therefore — since M is finite — from Lemma 3, $E\mathcal{L}E^\#$. The argument that $E\mathcal{R}E^\#$ is dual. Consequently $E\mathcal{H}E^\#$. Since both E and $E^\#$ are idempotents in the same $\mathcal{H}$-class, Lemma 2 implies that $E = E^\#$ which is a contradiction. □

This concludes the proof of Theorem 2 and gives us a bound on the height of k-decomposition trees that depends only on the size of the monoid M.

5 Leaks

Leaks were first introduced in [13] to define a decidable class of instances for the value 1 problem for probabilistic automata on finite words. The decidable class of *leaktight automata* is general enough to encompass all known decidable classes for the value 1 problem [12] and is optimal in some sense [11]. We extend the notion of leak from probabilistic automata to half-blind games and prove that when a game does not contain any leak then the belief monoid algorithm decides the maxmin reachability problem.

We illustrate leaks in the simplified case of probabilistic finite automata.

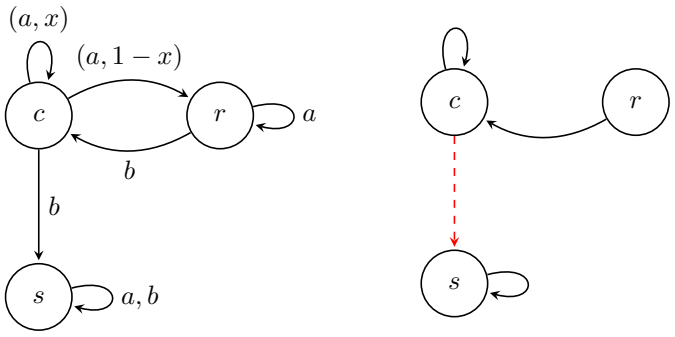

Figure 3: A probabilistic finite automaton exhibiting a leak.

Probabilistic automata (PA) can be seen as the degenerate case of half-blind games where the minimizer has no choice in any of the states that she

controls. Consider the PA (on the left) in Figure 3. When playing words from the sequence $\left(a^{f(n)}\right)_n$, the probability of staying in state c (if we start from state c) is $x^{f(n)}$. Given that $0 < x < 1$ and that f is an increasing function, we see that this probability can be made arbitrarily small by choosing n large enough. Similarly playing words from the sequence $\left(a^{f(n)}b\right)_n$, starting from the state c the probability to go to the sink state s is $x^{f(n)}$. The question is what can we say about the outcome if we play words from the sequence $\left((a^{f(n)}b)^{g(n)}\right)_n$ for some increasing function $g(n)$. For larger and larger n, is it the case that starting from the state c the probability of going to the sink state s is bounded away from 1? The answer depends on the value of x and the functions f, g. This behavior is illustrated in Figure 3 on the right side. Each time $a^{f(n)}b$ is played, the state c *leaks* some probability to the sink state s, denoted with the red dashed arrow. Having two or more leaks at the same time complicates the matters further, and this is the difficulty making the limit-sure decision problems undecidable in this setting.

Intuitively a leak happens when there is some communication between two recurrence classes with transitions that have a small probability of occurring. Whether this small probability builds up to render one of the recurrence classes transient is a computationally hard question to answer — and in fact impossible in general. Other examples of leaks can be found in [12] and the link between leaks and convergence rates are discussed further in [11].

We give a precise definition.

Definition 9 (Leaks)**.** *An element of the extended Markov monoid $(U, \widetilde{U}) \in \widetilde{\mathcal{M}}$ is a leak if it is idempotent and there exist $r, r' \in \mathbf{S_1}$, such that: (1) r, r' are U-recurrent, (2) $r \overset{U}{\not\to} r'$ and (3) $r \xrightarrow{\widetilde{U}} r'$.*

An element of the extended belief monoid $\mathbf{u} \in \widetilde{\mathcal{B}}$ is a leak if it contains $(U, \widetilde{U}) \in \mathbf{u}$ such that $(U, \widetilde{U})$ is a leak.

A game is leaktight if its extended belief monoid does not contain any leaks.

Note also that the question of whether a game is leaktight is decidable, since this information can be found in the belief monoid itself.

6 Correctness of the Belief Monoid Algorithm

This section contains the technical bulk of the paper since it is dedicated to proving that when the game is leaktight the belief monoid algorithm is both sound (a reachability witness is found implies $\underline{val}(s) = 1$) and complete (no reachability witness is found implies $\underline{val}(s) < 1$).

Theorem 4. *The belief monoid algorithm solves the maxmin reachability problem for half-blind leaktight games.*

Theorem 4 is a direct consequence of Theorem 5 and Theorem 6 which are given in the next two sections.

6.1 Soundness

In this section we give the main ideas to prove soundness of the belied monoid algorithm.

Theorem 5 (Soundness)**.** *Assume that the game is leaktight and that its extended belief monoid contains a reachability witness. Then the set of final states is maxmin reachable from the initial state.*

Theorem 5 is justifying the *yes* instances of the belief monoid algorithm, i.e. if the algorithm replies yes, then indeed $\underline{val}(s) = 1$. It is interesting to note that the equivalent soundness theorem for probabilistic automata in [13] does not make use of the leaktight hypothesis. Theorem 5 follows as a corollary of:

Lemma 5. *Given a game whose extended belief monoid is leaktight, with every element $\mathbf{u} \in \mathcal{B}$ of its belief monoid we can associate a sequence $(u_n)_n$, $u_n \in \Sigma_1$ such that for all $(\tau_n)_n$, $\tau_n \in \Sigma_2^p$ there exists $U \in \mathbf{u}$ and a subsequence $((u'_n, \tau'_n))_n \subset ((u_n, \tau_n))_n$ for which*

$$U(s,t) = 0 \implies \lim_n \mathbb{P}_s^{u'_n, \tau'_n}(t) = 0,$$

for all $s, t \in \mathbf{S_1}$.

We can prove Theorem 5 as follows. We are given a game that is leaktight and has a reachability witness $\mathbf{u} \in \mathcal{B}$, to whom we can associate a sequence of words $(u_n)_n$ according to Lemma 5. If on the contrary there exists $\epsilon > 0$ such that $\underline{val}(s) \leq 1 - \epsilon$ then there exists a sequence of strategies $(\tau_n)_n$ such that for all $n \in \mathbb{N}$, $\mathbb{P}_s^{u_n, \tau_n}(F) \leq 1 - \epsilon'$, for some $\epsilon' > 0$. This contradicts Lemma 5 because for the reachability witness we have by definition that for all $U \in \mathbf{u}$, $U(s,t) = 1$ implies $t \in F$.

We give a short sketch of the main ideas utilized into proving Lemma 5 before continuing with its proof in the section that follows.

To $\mathbf{a} \in \mathcal{B}$, $a \in \mathbf{A_1}$ we associate the constant sequence of words $(a)_n$. To the product of two elements in $\mathcal{B}$ we associate the concatenation of their respective sequences, and to $\mathbf{u}^{\#} \in \mathcal{B}$ the sequence $(u_n^n)_n$ is associated, given that $(u_n)_n$ is coupled with $\mathbf{u}$. Then we consider words whose letters are pairs (a, τ), where $a \in \mathbf{A_1}$ and τ is a strategy that maps $\mathbf{S_2}$ to $\mathbf{A_2}$, i.e. a pure and stationary strategy, and give a morphism from these words to the extended Markov monoid $\widetilde{\mathcal{M}}$. This allows us to construct k-decomposition trees of such words with respect to $\widetilde{\mathcal{M}}$. Then the k-decomposition trees are used to prove lower and upper bounds on the outcomes of the game under the strategy choices given by the word of pairs. The main idea is that we can construct for longer and longer words, k-decomposition trees for larger and larger k, thereby making sure that the iteration nodes have a large enough number of children which enables us to show that the probability of being in transient states is bounded above by a quantity that vanishes in the limit.

6.1.1 Proof of Lemma 5

Denote by Σ_2' the set of pure and stationary strategies for the minimizer, i.e. functions from $\mathbf{S_2}$ to $\mathbf{A_2}$. Let

$$\mathfrak{A} = \{(a, \alpha) \mid a \in \mathbf{A_1}, \alpha \in \Sigma_2'\}.$$

Note that $\mathfrak{A}$ is a finite set. Define ϕ the morphism from $\mathfrak{A}$ to $\widetilde{\mathcal{M}}$, that maps (a, α) to $(B^{a,\alpha}, B^{a,\alpha})$. Since a is a single letter, taking stationary strategies is the same as taking strategies from the set Σ_2^p since what the strategy plays after the first turn does not matter. Given a word $u \in \Sigma_1$ and a strategy $\tau \in \Sigma_2^p$, the pair (u, τ) can be seen as a word over the alphabet $\mathfrak{A}$. I.e. if $u = a_1 \cdots a_n$ and $\tau(n) = \alpha_n$, $n \in \mathbb{N}$, we see (u, τ) as $(a_1, \alpha_1)(a_2, \alpha_2) \cdots (a_n, \alpha_n)$.

Given $p \in A^*$, $k > 2$ and $h \in \mathbb{N}$, let $T_k^h(p)$ be the set of all k-decomposition trees of w with respect to $\widetilde{\mathcal{M}}$ whose height is at most h. Denote by $\mathfrak{T}_k^h(w) \subseteq \widetilde{\mathcal{M}}$ the set of types with which the root nodes of the trees in $T_k^h(w)$ are labeled. Note that $T_k^h(w)$ and consequently $\mathfrak{T}_k^h(w)$ can be empty, if h is too small.

We define the notion of *reification*, which intuitively makes precise what it means for a sequence of strategy choices (i.e. a sequence of words over the alphabet $\mathfrak{A}$) to realize the abstraction that is provided by a subset of $\widetilde{\mathcal{M}}$.

Definition 10 (Reification)**.** *Let $(p_n)_n$ be a sequence of words over the alphabet $\mathfrak{A}$, $h \in \mathbb{N}$ and $X \subseteq \widetilde{\mathcal{M}}$. We say that $(p_n)_n$* reifies *X with height h if there exists a subsequence $(p'_n)_n \subset (p_n)_n$ and $k \in \mathbb{N}$ such that*

$$\mathfrak{T}_k^h(p'_n) = X, \text{ for } n \in \mathbb{N},$$

moreover for infinitely many $i > k$, X appears infinitely often in the sequence $\left(\mathfrak{T}_i^h(p'_n)\right)_n$.

Reification is important because given that a sequence $(p_n)_n$ reifies some $X \in \widetilde{\mathcal{M}}$ with height h, we can prove lower and upper bounds on the outcomes of the game under $(p_n)_n$ that *agree* with some element in X.

First we show that any sequence of words over the alphabet $\mathfrak{A}$ reifies some $X \subseteq \widetilde{\mathcal{M}}$.

Lemma 6. *Let $(p_n)_n$ be a sequence of words over the alphabet $\mathfrak{A}$. There exists $h \in \mathbb{N}$ and $X \subseteq \widetilde{\mathcal{M}}$ such that $(p_n)_n$ reifies X with height h.*

Proof. Setting $h = 3 \cdot |\widetilde{\mathcal{M}}|^2$, for all $k > 2$ and $n \in \mathbb{N}$, we have $\mathfrak{T}_k^h((p_n)_n) \neq \emptyset$. This follows from Theorem 2.

Since $\widetilde{\mathcal{M}}$ is finite there exists $X_1 \subseteq \widetilde{\mathcal{M}}$ and a subsequence $(p'_n)_n \subset (p_n)_n$ such that $\mathfrak{T}_3^h(p'_n) = X_1$ for all $n \in \mathbb{N}$. If moreover there are infinitely many $i > 3$ such that X_1 appears infinitely often in the sequence $\left(\mathfrak{T}_i^h(p'_n)\right)_n$, then the lemma concludes. Otherwise there exists some $k_1 \in \mathbb{N}$ such that for all $i > k_1$, X_1 appears only finitely often in the sequence $\left(\mathfrak{T}_i^h(p'_n)\right)_n$. Now choose some other subsequence $(p''_n)_n \subset (p'_n)_n$ and $X_2 \subseteq \widetilde{\mathcal{M}}$ such that $\mathfrak{T}_{k_1}^h(p''_n) = X_2$ for

all $n \in \mathbb{N}$. Since $\widetilde{\mathcal{M}}$ is finite, the process above needs only a finite number of repetitions in order to find some $X \subseteq \widetilde{\mathcal{M}}$ such that $(p_n)_n$ reifies X with height h. □

Intuitively the lemma above means that under all sequences of strategy choices for the players it is possible to find a subsequence under which the outcome of the game is explained by an element of the extended Markov monoid. But this does not say anything about the belief monoid. We would like for all $\mathbf{u} \in \widetilde{\mathcal{B}}$ to have a sequence of words $(u_n)_n$ over the alphabet $\mathbf{A_1}$ such that for any sequence of strategies $(\tau_n)_n$, $(p_n)_n = ((u_n, \tau_n))_n$ reifies some $X \subseteq \widetilde{\mathcal{M}}$ and moreover X and $\mathbf{u}$ have at least one element in common. This is the purpose of the next lemma.

Lemma 7. *Let $\mathbf{u} \in \widetilde{\mathcal{B}}$, then there exists a sequence of words $(u_n)_n$ over the alphabet $\mathbf{A_1}$, $h \in \mathbb{N}$ and a function $N : \mathbb{N} \to \mathbb{N}$, such that for all sequence of strategies $(\tau_n)_n$ in Σ_2^p, $k > 2$ and $n > N(k)$,*

$$\mathfrak{T}_k^h\big((u_n, \tau_n)\big) \cap \mathbf{u} \neq \emptyset.$$

Proof. We proceed by induction on the elements of $\widetilde{\mathcal{B}}$.

- **Base case.** For elements $\mathbf{a} \in \widetilde{\mathcal{B}}$, where $a \in \mathbf{A_1}$, set the sequence of words to be the constant sequence $(a)_n$, set $h = 1$ and N to the constant function $N(k) := 0$ for all $k \in \mathbb{N}$. Then for all $\tau \in \Sigma_2^p$ and $k > 2$, the unique k-decomposition tree of (a, τ) is the single leaf node whose type is in $\mathbf{a}$ by definition of the morphism ϕ and the definition of $\mathbf{a}$ itself.

- **Product.** Assume that the lemma is true for the two elements $\mathbf{u}, \mathbf{v} \in \widetilde{\mathcal{B}}$, for $(u_n)_n, h_u, N_u$ and $(v_n)_n, h_v, N_v$ respectively. We will show that it also holds for the element $\mathbf{uv} \in \widetilde{\mathcal{B}}$, with the sequence of words $(u_n v_n)_n$, $h = \max\{h_u, h_v\} + 1$, and $N(k) = \max\{N_u(k), N_v(k)\}$.

  Let $(\tau_n)_n$ be a sequence of strategies, $k > 2$ and $n > N(k)$. Define $(\tau'_n)_n$ to be the sequence of strategies that are shifted by the lengths of u_n, i.e. $\tau'_n(i) = \tau_n(i + |u_n|)$ for $n, i \in \mathbb{N}$. Then by the induction hypothesis since $n > N_u(k)$ there exists a k-decomposition tree of length at most h_u for (u_n, τ_n) whose root node is labeled by some $(U, \widetilde{U}) \in \mathbf{u}$. Similarly there exists a k-decomposition tree of length at most h_v for (v_n, τ'_n) whose root node is labeled by some $(V, \widetilde{V}) \in \mathbf{v}$. Consequently we can construct a k-decomposition tree of length at most $\max\{h_u, h_v\} + 1$ of $(u_n v_n, \tau)$ whose root node is labeled by $(U, \widetilde{U}) \cdot (U, \widetilde{V}) = (UV, \widetilde{U}\widetilde{V}) \in \mathbf{uv}$, by making the root node a product node and add the two subtrees as children.

- **Iteration.** Assume that the lemma is true for some idempotent $\mathbf{u} \in \widetilde{\mathcal{B}}$. Then there exists a sequence $(u_n)_n$, $h_u \in \mathbb{N}$ and a function N_u for which the lemma holds. We will prove that it also holds for $\mathbf{u}^{\#} \in \widetilde{\mathcal{B}}$, the sequence $(u_n^n)_n$, $h = h_u + 3 \cdot |\widetilde{\mathcal{M}}|^2$ and the function N defined by $N(k) := N_u(k) + k^{3 \cdot |\widetilde{\mathcal{M}}|^2}$. Let $(\tau_n)_n$ be a sequence of strategies, $k > 2$ and

$n > N(k)$. Since $n > N_u(k)$ by the induction hypothesis we know that for all strategies τ, $\mathfrak{T}_k^{h_u}\big((u_n, \tau)\big) \cap \mathbf{u} \neq \emptyset$.

For $0 \leq i < n$ let τ_n^i be the shifted strategy by u_n^i, i.e. $\tau_n^i(j) = \tau_n(j + |u_n^i|)$, $j \in \mathbb{N}$.

For $0 \leq i < n$, pick some $(U_i, \widetilde{U}_i) \in \mathfrak{T}_k^{h_u}\big((u_n, \tau_n^i)\big) \cap \mathbf{u}$ and denote by T_i the associated k-decomposition tree. We modify the alphabet $\mathfrak{A}$ and add (u_n, τ_n^i), $0 \leq i < n$ as letters. At the same time modify the morphism ϕ by mapping (u_n, τ_n^i) to $(U_i, \widetilde{U}_i)$.Then applying Theorem 2 to the word (u_n^n, τ_n) we know that there exists a k-decomposition tree of height at most $3 \cdot |\widetilde{\mathcal{M}}|^2$, where the leaves are labeled by $\big((u_n, \tau_n^i), (U_i, \widetilde{U}_i)\big)$, $0 \leq i < n$. Plugging the trees T_i instead of the leaves we construct a k-decomposition tree for (u_n^n, τ_n) of height at most $h = h_u + 3 \cdot |\widetilde{\mathcal{M}}|^2$. Moreover since $n > N(k) \geq k^{3 \cdot |\widetilde{\mathcal{M}}|^2}$ there must exist at least one iteration node in this tree therefore the type of the root node can be written as a #-expression whose #-height is larger than 1. Consequently the type is in $\mathbf{u}^{\#}$.

□

Observe that the height h gets larger when we iterate, and the same for the function N, but since we do this only a finite number of times (it is induction on the finite monoid) we can give a uniform height h and function N such that the lemma above holds for all elements of the extended belief monoid with that height h and function N.

Combining the two lemmata above we have:

Lemma 8. *For all $\mathbf{u} \in \widetilde{\mathcal{B}}$ we have a sequence of words $(u_n)_n$ over the alphabet $\mathbf{A_1}$ such that for all $(\tau_n)_n$ there exists $X \subseteq \widetilde{\mathcal{M}}$ such that $(p_n)_n = ((u_n, \tau_n))_n$ reifies X with height h. Moreover $X \cap \mathbf{u} \neq \emptyset$.*

The *raison d'tre* of the k-decomposition trees, and their bounded height is because it allows us to give lower and upper bounds on certain outcomes of the game as in the following lemma. This is where the leaktight hypothesis is necessary. We start with the lower bound. The proof follows that of [13].

Lemma 9. *There exists a function $L : (\mathbb{N}, \mathbb{N}) \to \mathbb{R}$ mapping to the non-zero positive reals such that for all words $p = (w, \tau)$ over the alphabet $\mathfrak{A}$, $k > 2$ and T a k-decomposition tree of p of height at most h with the root node labeled by $(W, \widetilde{W}) \in \widetilde{\mathcal{M}}$, given that $\widetilde{\mathcal{M}}$ is leaktight then for all $s, t \in \mathbf{S_1}$,*

$$W(s,t) = 1 \implies \mathbb{P}_s^{w,\tau}(t) \geq L(h,k), \text{ and} \tag{1}$$

$$\widetilde{W}(s,t) = 1 \iff \mathbb{P}_s^{w,\tau}(t) > 0. \tag{2}$$

Proof. We proceed by induction on the structure of the k-decomposition tree T.

- **Leaves.** Leaves are labeled by some $\big((a, \alpha), (B^{a,\alpha}, B^{a,\alpha})\big)$ for $a \in \mathbf{A_1}$, $\alpha \in \Sigma_2'$. Then (2) is true by definition of $B^{a,\alpha}$, and (1) holds for the lower

bound ν, where ν is the smallest non-zero transition probability appearing in the transition table of the game.

- **Product nodes.** Assume that the lemma holds for the children that are labeled by $\big(u,(U,\widetilde{U})\big)$ and $\big(v,(V,\widetilde{V})\big)$ with with the lower bound L. It follows easily that the lemma holds for the parent node that is labeled by $\big(uv,(UV,\widetilde{U}\widetilde{V})\big)$ with the lower bound L^2. The words u and v above are words over the alphabet $\mathfrak{A}$.

- **Idempotent nodes.** Similarly to above if the lemma holds for the n children ($n < k$) that are labeled by $\big(u_1,(U,\widetilde{U})\big),\big(u_2,(U,\widetilde{U})\big),\ldots,\big(u_n,(U,\widetilde{U})\big)$ for some idempotent $(U,\widetilde{U}) \in \widetilde{\mathcal{M}}$ with lower bound L, it follows easily that it also holds for the parent node that is labeled by $\big(u_1u_2\cdots u_n,(U,\widetilde{U})\big)$ for the lower bound L^k.

- **Iteration nodes.** Assume that the lemma holds for the n children ($n \geq k$) that are labeled by $\big(u_1,(U,\widetilde{U})\big),\big(u_2,(U,\widetilde{U})\big),\ldots,\big(u_n,(U,\widetilde{U})\big)$ for some idempotent $(U,\widetilde{U}) \in \widetilde{\mathcal{M}}$ with lower bound L. Proving (2) is trivial, so we show only (1). Let $s,t \in \mathbf{S_1}$ such that $U^{\#}(s,t) = 1$, by definition t is U-recurrent. Then

$$\mathbb{P}_s^{u_1\cdots u_n}(t) \geq \mathbb{P}_s^{u_1}(t) \sum_{q\in\mathbf{S_1}} \mathbb{P}_t^{u_2\cdots u_{n-1}}(q)\mathbb{P}_q^{u_n}(t).$$

  Given that $(U,\widetilde{U})$ is leaktight it follows that for all $q \in \mathbf{S_1}$, $\mathbb{P}_t^{u_2\cdots u_{n-1}}(q) > 0$ implies $\mathbb{P}_q^{u_n}(t) \geq L$. Indeed, let $q \in \mathbf{S_1}$ such that $\mathbb{P}_t^{u_2\cdots u_{n-1}}(q) > 0$, then by the induction hypothesis we have $\widetilde{U}^{n-2}(t,q) = 1$ and since $\widetilde{U}$ is idempotent $\widetilde{U}(t,q) = 1$. The state t is U-recurrent, and $(U,\widetilde{U})$ is not a leak, therefore it follows from the definition of a leak that $U(q,t) = 1$. Using the induction hypothesis on the right most child, we have $\mathbb{P}_q^{u_n}(t) \geq L$. By the induction hypothesis for the left most child we have $\mathbb{P}_s^{u_1}(t) \geq L$. From here we conclude that

$$\mathbb{P}_s^{u_1\cdots u_n}(t) \geq L^2.$$

By induction we see that the lemma holds for the function $L(h,k) = \nu^{k^h}$. □

We now give a proof for the upper bound. Define $L = L(3\cdot|\widetilde{\mathcal{M}}|^2,3)$ to be the lower bound given from Lemma 9.

Lemma 10. *Let $h \in \mathbb{N}$, define and $K_h \in \mathbb{N}$ such that $h\cdot(1-L)^{K_h} < L$ and $K_h > |\mathbf{S_1}|$.*

For all words $p = (w,\tau)$ over the alphabet $\mathfrak{A}$, $h \in \mathbb{N}$, $k > K_h$ and T a k-decomposition tree of p of height at most h with the root node labeled by $(W,\widetilde{W}) \in \widetilde{\mathcal{M}}$, given that $\widetilde{\mathcal{M}}$ is leaktight then for all $s,t \in \mathbf{S_1}$

$$W(s,t) = 0 \implies \mathbb{P}_s^{w,\tau}(t) \leq h\cdot(1-L^{|\mathbf{S_1}|})^{\lfloor k/|\mathbf{S_1}|\rfloor}.$$

Proof. We proceed by induction on the structure of the k-decomposition tree T, while maintaining upper bounds F that are always smaller than $h \cdot (1 - L^{|\mathbf{S_1}|})^{\lfloor k/|\mathbf{S_1}| \rfloor}$.

- **Leaves.** The leaves are labeled by $\big((a,\alpha),(B^{a,\alpha},B^{a,\alpha})\big)$ for $a \in \mathbf{A_1}$ and $\alpha \in \Sigma_2'$, by definition we have an upper bound of 0.

- **Product nodes.** Assume that we have $(u,\big(U,\widetilde{U}\big))$, $(v,\big(V,\widetilde{V}\big))$ and the parent node labeled by $(uv,\big(UV,\widetilde{U}\widetilde{V}\big))$, where u,v are words over the alphabet $\mathfrak{A}$. Let $F \geq 0$ be the upper bound of the children, i.e. $U(s,t) = 0 \implies \mathbb{P}_s^u(t) \leq F$. Let $s,t \in \mathbf{S_1}$ be such that $UV(s,t) = 0$. Then the probability of all paths of length two, s,s',t such that $\mathbb{P}_s^u(s') > 0$ and $\mathbb{P}_{s'}^v(t) > 0$ is bounded above by F, therefore $\mathbb{P}_s^{uv}(t) \leq F$.

- **Idempotent nodes.** Assume that we have the children $p_1,\ldots,p_j, j < k$ each decorated by the same idempotent $(W,\widetilde{W})$, and let $s,t \in \mathbf{S_1}$ such that $W(s,t) = 0$. The words p_i are over the alphabet $\mathfrak{A}$. By the induction hypothesis the upper bound F holds for all the children.

  Denote by ρ the set of all paths $s_0 s_1 \cdots s_j$ such that $s_0 = s$, $s_j = t$ and $\widetilde{W}(s_i,s_{i+1}) = 1$for all $0 \leq i \leq j-1$. Since $W(s,t) = 0$ for all $\pi = s_0 \cdots s_j \in \rho$ there exists $0 \leq C(\pi) \leq j-1$ such that $W(s_{C(\pi)},s_{C(\pi)+1}) = 0$ and for all $0 \leq i \leq C(\pi)-1$, $W(s_i,s_{i+1}) = 1$. Define ρ' to be the set of such prefixes, i.e.

  $$\rho' = \{s_0 \cdots s_{C(\pi)} \mid \pi = s_0 \cdots s_j \in \rho\}.$$

  The set ρ' is nonempty because there exists some $r \in \mathbf{S_1}$ such that $W(s,r) = 1$ (this follows from the definition of the half-blind game, in every state we have some actions).

  Then we have

  $$\begin{aligned}\mathbb{P}_s^{p_1\cdots p_j}(t) &= \sum_{s_0\cdots s_j \in \rho} \mathbb{P}_{s_0}^{p_1}(s_1)\cdots\mathbb{P}_{s_{j-1}}^{p_j}(s_j)\\ &\leq \sum_{\pi = s_0\cdots s_{C(\pi)} \in \rho'} \mathbb{P}_{s_0}^{p_1}\cdots\mathbb{P}_{s_{C(\pi)-1}}^{p_{C(\pi)}}(s_{C(\pi)})\cdot F\\ &\leq F,\end{aligned}$$

  where the first inequality is because of the induction hypothesis and $W(s_{C(\pi)},s_{C(\pi)+1}) = 0$, whereas the second inequality is because for every path $\pi \in \rho$ there is exactly one path $\pi' \in \rho'$ such that π' is a prefix of π.

- **Iteration nodes.** Assume that we have the children $p_1,\ldots,p_j$, $j \geq k$ each decorated by the same idempotent $(W,\widetilde{W}) \in \widetilde{\mathcal{M}}$ and for whom the upper bound F holds. Let $s,t \in \mathbf{S_1}$ be such that $W^{\#}(s,t) = 0$. In case $W(s,t) = 0$ a proof like the one above for idempotent nodes gives F as the upper bound. Therefore we assume that $W(s,t) = 1$. Then by definition

t is W-transient and it communicates with some recurrence classes whose union we denote by $S_r \subseteq \mathbf{S_1}$. We will prove that for all $0 \leq i < j' \leq j$ such that $j' - i \geq |\mathbf{S_1}|$ there exists $i \leq i' \leq j'$ such that $i' - i \leq |\mathbf{S_1}|$ and

$$\mathbb{P}_t^{p_i \cdots p_{i'}}(S_r) \geq L^{|\mathbf{S_1}|}. \tag{3}$$

Let $i \in \{0, \dots j\}$, then there exists a 3-decomposition tree T_i for the word p_i, whose root node is labeled by the element $(W_i, \widetilde{W}) \in \widetilde{\mathcal{M}}$. It is possible that $W \neq W_i$, but for all $s', t' \in \mathbf{S_1}$, $W(s', t') = 0$ implies that $W_i(s', t') = 0$. This is because by the induction hypothesis, if $W(s', t') = 0$, we know that $\mathbb{P}_{s'}^{u_i}(t') \leq F$ whereas according to Lemma 9 for T_i, if $W_i(s', t') = 1$ we have $\mathbb{P}_{s'}^{u_i}(t') \geq L$, from $F \leq h \cdot (1 - L)^k$ and our choice of k, superior to K_h, this is a contradiction, hence $W_i(s', t') = 0$.

Let S_t be the set of states that are W-reachable from t (for all $t' \in S_t$, $W(t, t') = 1$) but not in S_r. These states are all W-transient and moreover for all $i \in \{0, \dots, j\}$, there exists a W_i path from t and any state in S_t to some element in S_r. This is because for all $t' \in S_t \cup \{t\}, r \in S_r$, $\widetilde{W}(t', r) = 1$, and there is no W_i path from S_r to $S_t \cup \{t\}$, if there was no W_i path from $S_t \cup \{t\}$ we could construct a leak, which contradicts the hypothesis that $\widetilde{\mathcal{M}}$ is leaktight. Similarly, for $0 \leq i < j' \leq j$ such that $i - j' \geq |\mathbf{S_1}|$, if $W_i \cdots W_{j'}(t, S_r) = 0$ we can construct a leak by repeating a factor of $W_i \cdots W_{j'}$, hence we can assume that there exists $i \leq i' \leq j'$, such that $i' - i \leq |\mathbf{S_1}|$ and $W_i \cdots W_{i'}(t, S_r) = 1$. Then it follows from Lemma 9 that $\mathbb{P}_t^{p_i \cdots p_{i'}}(S_r) \geq L^{|\mathbf{S_1}|}$ which concludes (3).

Let ρ be the set of all paths $s_0 \cdots s_j$ such that $s_0 = s$, $s_j = t$ and $\widetilde{W}(s_i, s_{i+1}) = 1$ for all $0 \leq i \leq j - 1$. We partition ρ into the set ρ_1 of all the paths that pass through S_r and ρ_2 the set of all paths that do not. Since t is W-transient, for all $r \in S_r$, $W(r, t) = 0$, consequently we can use the argument above for the idempotent nodes to give F as an upper bound for the probability of the event that constitutes the union of all the sets in ρ_1. As for ρ_2, because of transience of t and (3) the probability of the union of all the paths in ρ_2 can be bounded above by $(1 - L^{\mathbf{S_1}})^{\lfloor j/|\mathbf{S_1}| \rfloor}$.

We have shown that the upper bound grows only in the case of iteration nodes and it always is smaller than $h \cdot (1 - L^{|\mathbf{S_1}|})^{\lfloor k/|\mathbf{S_1}| \rfloor}$, since in ascending the tree, at each level we add at most a term of $(1 - L^{|\mathbf{S_1}|})^{\lfloor k/|\mathbf{S_1}| \rfloor}$. □

Now Lemma 5 follows as a corollary of Lemma 10 and Lemma 8. The main point is that the larger the k the smaller the lower bound we can prove.

6.2 Completeness

Before introducing the main theorem of this section let us give a definition.

Definition 11 (μ-faithful abstraction)**.** *Let $u \in \Sigma_1$ be a word, and $\mu > 0$ a strictly positive real number. We say that $\mathbf{u} \in \widetilde{\mathcal{B}}$ is a μ-faithful abstraction of the word u if for all $(U, \widetilde{U}) \in \mathbf{u}$ there exists $\tau \in \Sigma_2^p$ such that for all $s, t \in \mathbf{S_1}$,*

$$\widetilde{U}(s,t) = 1 \iff \mathbb{P}_s^{u,\tau}(t) > 0 \tag{4}$$

$$U(s,t) = 1 \implies \mathbb{P}_s^{u,\tau}(t) \geq \mu. \tag{5}$$

This section is devoted to giving the main ideas behind the proof and proving the following theorem.

Theorem 6. *Assume that the game is leaktight. Then there exists $\mu > 0$ such that for all words $u \in \Sigma_1$ there is some element $\mathbf{u} \in \widetilde{\mathcal{B}}$ that is a μ-faithful abstraction of u.*

The notion of μ-faithful abstraction is compatible with product in the following sense.

Lemma 11. *Let $\mathbf{u}, \mathbf{v} \in \widetilde{\mathcal{B}}$ be μ-faithful abstractions of $u \in \Sigma_1$ and $v \in \Sigma_1$ respectively. Then $\mathbf{uv}$ is a μ^2-faithful abstraction of $uv \in \Sigma_1$.*

A naïve use of Lemma 11 shows that any word w has a μ_w-faithful abstraction in $\widetilde{\mathcal{B}}$, where μ_w converges to 0 as the length of w increases. However we need μ_w to depend only on $\widetilde{\mathcal{B}}$, independently of $|w|$. For that we make use of k-decomposition trees. More precisely we build N-decomposition trees for words in Σ_1 where $N = 2^{3 \cdot |\widetilde{\mathcal{M}}|}$. We can construct N-decomposition trees for any word $u \in \Sigma_1$ whose height is at most $3 \cdot |\widetilde{\mathcal{B}}|^2$ and since N is fixed we will be able to propagate the constant μ, it only remains to take care that the constant does not shrink as a function of the number of children in iteration nodes, hence the following lemma.

Lemma 12. *Let $u \in \Sigma_1$ be a word factorized as $u = u_1 \cdots u_n$ where $n > 2^{3 \cdot |\widetilde{\mathcal{M}}|} = N$, and $\mathbf{u} \in \widetilde{\mathcal{B}}$ an idempotent element such that $\mathbf{u}$ is a μ-faithful abstraction of u_i, $1 \leq i \leq n$, for some $\mu > 0$. If $\mathbf{u}$ is not a leak then $\mathbf{u}^\#$ is a μ'-faithful abstraction of u, where $\mu' = \mu^{N+1}$.*

Theorem 6 is an easy consequence from the lemmata above, which can be shown as follows. We construct a N-decomposition tree for the word $u \in \Sigma_1$, and propagate the lower bound from the leaf nodes, for which we have the bound $\nu > 0$ (where ν is the smallest transition probability appearing in the game) up to the root node. If we know that a bound $\mu > 0$ holds for the children, for the parents we have the following lower bounds as a function of the kind of the node: (1) *product node*: μ^2; (2) *idempotent node* μ^N; (3) *iteration node* μ^{N+1}. Since the length of the tree is at most $h = 3 \cdot |\widetilde{\mathcal{B}}|^2$ we have the lower bound

$$\mu = \nu^{h(N+1)}$$

that holds for all $u \in \Sigma_1$.

Proof of Lemma 12. Let $(W, \widetilde{W}) \in \mathbf{u}^{\#}$ we want to build a strategy $\tau \in \Sigma_2^p$ such that (4) and (5) in Definition 11 hold, for $(W, \widetilde{W})$ the word u and the bound μ'. Let us first assume that $(W, \widetilde{W})$ is such that

$$W = F_1 G_1^{\#} \cdots F_k G_k^{\#} F_{k+1},$$
$$\widetilde{W} = \widetilde{F}_1 \widetilde{G}_1 \cdots \widetilde{F}_k \widetilde{G}_k \widetilde{F}_{k+1},$$

where $(F_i, \widetilde{F}_i) \in \mathbf{u}$,$(G_i, \widetilde{G}_i) \in \mathbf{u}$ and $(G_i, \widetilde{G}_i)$ are idempotent.

The set of #-expressions of $\mathbf{u} \subseteq \mathcal{M}$ denoted by $\mathfrak{E}(\mathbf{u})$ is a language defined by the grammar: $\mathfrak{E}(\mathbf{u}) := \mathbf{u} \mid \mathfrak{E}(\mathbf{u}) \cdot \mathfrak{E}(\mathbf{u}) \mid (\mathfrak{E}(\mathbf{u}))^{\#}$, so the terminal symbols are the elements of $\mathbf{u}$. There is $\gamma_{\mathbf{u}}$, a natural function mapping $\mathfrak{E}(\mathbf{u})$ to $\mathcal{M}$, i.e. the function that is the identity when restricted to the terminal symbols, otherwise $\gamma_{\mathbf{u}}(e \cdot e') = \gamma_{\mathbf{u}}(e)\gamma_{\mathbf{u}}(e')$, and $\gamma_{\mathbf{u}}(e^{\#}) = (\gamma_{\mathbf{u}}(e))^{\#}$. Given $e \in \mathfrak{E}(\mathbf{u})$ we define its *#-height* as the number of the deepest nesting of #. E.g. $\# - height(U^{\#}(VW^{\#})^{\#}) = 2$.

We can safely make this assumption because for all $(U, \widetilde{U}) \in \mathbf{u}$ we can find a #-expression $e \in \mathfrak{E}(\mathbf{u})$ whose #-height is 1, such that $\gamma_{\mathbf{u}}(e) = (U', \widetilde{U})$ and for all $s, t \in \mathbf{S_1}$ $U(s,t) = 1 \implies U'(s,t) = 1$. This is an easy exercise: when iterating we are removing edges.

Since $\mathbf{u}$ is a μ-faithful abstraction of u_i,$1 \leq i \leq n$, for all $(U, \widetilde{U})$ in $\mathbf{u}$ there is a strategy in Σ_2^p such that (4) and (5) hold. Let τ_1 be such a strategy for $(F_1, \widetilde{F}_1)$, τ_2 for $(G_1, \widetilde{G}_1)$ and so on until τ_{2k+1} for the selection $(F_{k+1}, \widetilde{F}_{k+1})$. We define the strategy τ by assigning one of the τ_i to some part of the word in the following way:

- against u_1 play τ_1,
- against u_2 play τ_2, play τ_2 also against $u_3, u_4, \ldots, u_{n-2k+1}$ each,
- against u_{n-2k+2} play τ_3, etc., in general against $u_{n-2k+1+i}$ play τ_{i+2}, $1 \leq i \leq 2k-1$.

One can visualize this in the following way.

$$\tau := \begin{array}{c|c|c|c|c|c|c} u_1 & u_2, u_3, \ldots, u_{n-2k+1} & u_{n-2k+2} & \cdots & u_{n-2k+2k-1} & u_n \\ \tau_1 & \tau_2 & \tau_3 & & \tau_{2k} & \tau_{2k+1} \\ F_1 & G_1 & F_2 & & G_k & F_{k+1} \end{array}.$$

This means that τ plays according to τ_2 against u_2 then it keeps playing according to τ_2 against u_3 and so on until u_{n-2k+1} is read. Note that it is well defined since we have assumed that $n > N$, and $N = 3 \cdot |\widetilde{\mathcal{M}}|^2 \geq 2k+1$ from Simon's forest factorization theorem.

Now we prove (4) for τ and u.

($\implies$) Let $s, t \in \mathbf{S_1}$ be such that $s \xrightarrow{\widetilde{W}} t$. Since $\widetilde{W} = \widetilde{F}_1 \widetilde{G}_1 \cdots \widetilde{G}_k \widetilde{F}_{k+1}$ and $\widetilde{G}_1$ is idempotent there exist $s_1, \ldots, s_{n-1} \in \mathbf{S_1}$ such that

$$s \xrightarrow{\widetilde{F}_1} s_1 \xrightarrow{\widetilde{G}_1} \cdots \xrightarrow{\widetilde{G}_1} s_{n-2k} \xrightarrow{\widetilde{F}_2} s_{n-2k+1} \xrightarrow{\widetilde{G}_2} s_{n-2k+2} \cdots \xrightarrow{\widetilde{G}_k} s_{n-1} \xrightarrow{\widetilde{F}_{k+1}} t. \quad (6)$$

Let $F(s_1, \ldots, s_{n-1})$ be equal to

$$\mathbb{P}_s^{u_1,\tau_1}(s_1)\mathbb{P}_{s_1}^{u_2,\tau_2}(s_2)\cdots\mathbb{P}_{s_{n-2k-1}}^{u_{n-2k+1},\tau_2}(s_{n-2k})\mathbb{P}_{s_{n-2k}}^{u_{n-2k+2},\tau_3}(s_{n-2k+1})\cdots\mathbb{P}_{s_{n-1}}^{u_n,\tau_{2k+1}}(t).$$

Then by the choice of τ we have $\mathbb{P}_s^{u,\tau}(t) \geq F(s_1, \ldots, s_{n-1})$. Since $\mathbf{u}$ is a μ-faithful abstraction of u_i, (6) implies that every factor of $F(s_1, \ldots, s_{n-1})$ is positive, hence $\mathbb{P}_s^{u,\tau}(t) > 0$.

($\Longleftarrow$) Let $s, t \in \mathbf{S_1}$ be such that $\mathbb{P}_s^{u,\tau}(t) > 0$, then similarly as above there must exist states $s_1, \ldots, s_{n-1}$ such that

$$\mathbb{P}_s^{u,\tau}(t) \geq F(s_1, \ldots, s_{n-1}) > 0.$$

This implies (6) since $\mathbf{u}$ is a μ-faithful abstraction of all u_i, and in turn, (6) implies that $s \xrightarrow{\widetilde{W}} t$, since $\widetilde{W} = \widetilde{F}_1\widetilde{G}_1\cdots\widetilde{G}_k\widetilde{F}_{k+1}$.

Now we prove (5) for τ and u and the bound $\mu' = \mu^{N+1}$. Let $s, t \in \mathbf{S_1}$ such that $s \xrightarrow{W} t$. Then there exists states $s_1, \ldots, s_{2k}$ such that

$$s \xrightarrow{F_1} s_1 \xrightarrow{G_1^{\#}} s_2 \xrightarrow{F_2} \cdots \xrightarrow{G_k^{\#}} s_{2k} \xrightarrow{F_{k+1}} t. \tag{7}$$

First we will show that

$$\mathbb{P}_{s_1}^{u_2,\ldots,u_{n-2k+1},\tau'}(s_2) \geq \mu^2, \tag{8}$$

where τ' is the strategy that plays τ_2 against u_2, and against u_3 and so on. This is exactly what the strategy τ does, after u_1 is read. Then we have

$$\mathbb{P}_{s_1}^{u_2\cdots u_{n-2k+1},\tau'}(s_2) \geq \mathbb{P}_{s_1}^{u_2,\tau_2}(s_2)\sum_{s'\in\mathbf{S_1}}\mathbb{P}_{s_2}^{u_3\cdots u_{n-2k},\tau''}(s')\mathbb{P}_{s'}^{u_{n-2k+1},\tau_2}(s_2),$$

where τ'' is the strategy that plays τ_2 against u_3 ,and against u_4 and so on. The strategy τ'' is the same as τ' just shifted by the first part u_2. From (7) $s_1 \xrightarrow{G_1^{\#}} s_2$ which implies that s_2 is G_1-recurrent, $s_1 \xrightarrow{\widetilde{G}_1} s_2$ and $s_1 \xrightarrow{G_1} s_2$. By the choice of τ_2 because $s_1 \xrightarrow{G_1} s_2$ we have

$$\mathbb{P}_{s_1}^{u_2\cdots u_{n-2k+1},\tau'}(s_2) \geq \mu\sum_{s'\in\mathbf{S_1}}\mathbb{P}_{s_2}^{u_3\cdots u_{n-2k},\tau''}(s')\mathbb{P}_{s'}^{u_{n-2k+1},\tau_2}(s_2). \tag{9}$$

Let s' be such that $\mathbb{P}_{s_2}^{u_3\cdots u_{n-2k},\tau''}(s') > 0$. Then from the definition of τ'', $s_2 \xrightarrow{\widetilde{G}_1^{n-2k-3}} s'$ and since $\widetilde{G}_1$ is idempotent $s_2 \xrightarrow{\widetilde{G}_1} s'$. We will prove that $s' \xrightarrow{G_1} s_2$. There are two cases:

- s' is G_1-recurrent: then both s' and s_2 are G_1-recurrent, and $s_2 \xrightarrow{\widetilde{G}_1} s'$. Since we have assumed that $\mathbf{u}$ is not a leak, then $s' \xrightarrow{G_1} s_2$.

- s' is G_1-transient: There exists some state r that is G_1-recurrent, such that $s' \xrightarrow{G_1} r$ and $r \not\xrightarrow{G_1} s'$. Now $s' \xrightarrow{G_1} r$ implies that $s' \xrightarrow{\widetilde{G}_1} r$, and from idempotency of $\widetilde{G}_1$, $s_2 \xrightarrow{\widetilde{G}_1} r$. Then from the argument for the case above $r \xrightarrow{G_1} s_2$, and finally from idempotency of G_1, $s' \xrightarrow{G_1} s_2$.

We have shown that for all s' such that $\mathbb{P}_{s_2}^{u_3 \cdots u_{n-2k}, \tau''}(s') > 0$, $s' \xrightarrow{G_1} s_2$. As a consequence, from the choice of τ_2 and (9) we have

$$\mathbb{P}_{s_1}^{u_2 \cdots u_{n-2k+1}, \tau'}(s_2) \geq \mu^2.$$

To finish up with the proof of (5), for all $s, s' \in \mathbf{S_1}$ and G_i, $s \xrightarrow{G_i^\#} s'$ implies that $s \xrightarrow{G_i} s'$, therefore from (7) we have

$$s \xrightarrow{F_1} s_1 \xrightarrow{G_1^\#} s_2 \xrightarrow{F_2} s_3 \xrightarrow{G_2} \cdots \xrightarrow{G_k} s_{2k} \xrightarrow{F_{k+1}} t, \tag{10}$$

so for all G_i, $2 \leq i \leq k$, we write G_i instead of $G_i^\#$. Then by the choice of the strategies τ_i and the definition of τ,

$$\mathbb{P}_s^{u,\tau}(t) \geq \mathbb{P}_s^{u_1,\tau_1}(s_1)\mathbb{P}_{s_1}^{u_2,\ldots,u_{n-2k+1},\tau'}(s_2) \cdots \mathbb{P}_{s_{2k}}^{u_n,\tau_{2k+1}}(t) \geq \mu \cdot \mu^2 \cdot \mu^{2k-1} = \mu^{2k+2},$$

where for the last inequality we have used (8) and (10). Since $2k+1 \leq N$, this concludes the proof of (5) for τ,u and the bound $\mu' = \mu^{N+1}$. □

7 Complexity of Optimal Strategies

The maxmin reachability problem solved by the belief monoid algorithm concerns games where the maximizer is restricted to pure strategies, and decides whether

$$\underline{val}(s) = \sup_{w \in \Sigma_1} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(F) = 1$$

where $\Sigma_1 = \mathbf{A_1}^*$. If we extend further the set Σ_1 of strategies of the maximizer and allow him to have mixed strategies too, then half-blind games have a value. Let $\Sigma_1^m = \Delta(\mathbf{A_1}^*)$ be the set of mixed words.

Theorem 7 ([17])**.** *Half-blind games where maximizer can use mixed strategies have a value:*

$$val(s) = \sup_{w \in \Sigma_1^m} \inf_{\tau \in \Sigma_2} \mathbb{P}_s^{w,\tau}(F) = \inf_{\tau \in \Sigma_2} \sup_{w \in \Sigma_1^m} \mathbb{P}_s^{w,\tau}(F).$$

Define Σ_2^f to be the set of finite-memory strategies for the minimizer. These are strategies that are stochastic finite-state probabilistic transducers reading histories and outputting elements of $\Delta(\mathbf{A_2})$, mixed actions. Let $val^f(s) = \inf_{\tau \in \Sigma_2^f} \sup_{w \in \Sigma_1} \mathbb{P}_s^{w,\tau}(F)$.

In general,

$$\underline{val}(s) \leq val(s) \leq val^f(s).$$

A natural question is whether the inequalities above are strict in general, i.e. whether mixed strategies are strictly more powerful for the maximizer and whether infinite-memory strategies are strictly more powerful for the minimizer.

The former question can be resolved easily. We can find examples where the maximizer wins more by mixing her strategy. In fact the example in Figure 2 suffices. For this example we have $\underline{val}(s) < val(s)$.

The latter question — whether there exists an example such that $val(s) < val^f(s)$ — is harder, and its answer is more counter-intuitive. When maximizer has full information, it is well-known that minimizer can play optimally with no memory (using a positional strategy). When maximizer is totally blind, one might believe that minimizer does not need any memory either because playing against an opponent that is totally blind to satisfy a safety objective seems rather easy. Surprisingly perhaps, minimizer requires infinite memory to play optimally against a blind maximizer and satisfy its safety objective. We show that there exists a game where $val(s) < val^f(s)$. This game is based on the following gadget.

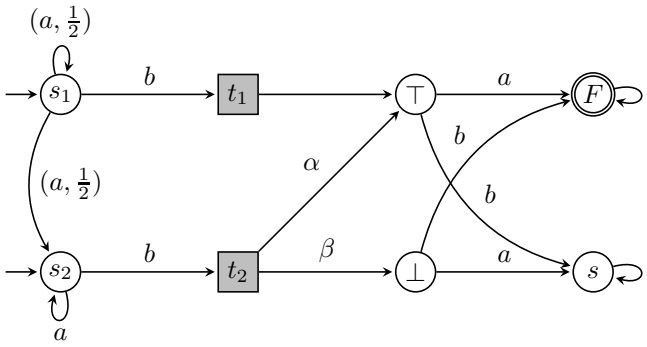

Figure 4: A gadget

We give the main idea behind the gadget. The maximizer wants to be able to ascertain whether she is in state top or bottom after playing his first b so that she can go to the final state. The objective of the minimizer is to make the probability of being in the top state equal to that of being in the bottom state, so that the maximizer cannot win more than 1/2. In order to do this, when it is his turn to make the choice between α and β (or a mixing of them) she has to know the exact probability distribution over t_1 and t_2. But this is impossible to keep track with a finite-memory strategy, i.e. the maximizer plays too many a's for the minimizer's small memory. Hence the maximizer can always win slightly more than 1/2. We then use this gadget in a game that emphasizes the importance of these winnings and prove that in that game

$1/2 = val(s) < val^f(s) = 1$. We prove this formally.

The game starts either at state s_1 or s_2 with equal probability. The maximizer can play a series of a's and eventually has to play a b if she wants to make progress. After which the minimizer observes whether the game is in the state t_1 or t_2. In case it is in t_1 the minimizer has no choice and proceeds to state $\top$. In case it is in t_2 the minimizer can choose between α and β to go either to state $\top$ or to state $\bot$. Then the maximizer has to guess which one it is. If the guess is right she wins if it is wrong she loses by going to the sink state. The goal of the minimizer is to keep track of the probability distribution on the states of the game such that when it is her time to make a decision she will play a mixed (between α and β) action such that the probability to be in $\top$ is equal to the probability to be in $\bot$ equal to $1/2$. Keeping track of the distribution will be impossible with a finite-memory strategy because the sequence of a's that the maximizer plays can be arbitrarily long.

Observe that $val(\gamma) = 1/2$, where γ is the initial distribution, i.e. $\gamma(s_1) = \gamma(s_2) = 1/2$, by giving the optimal strategies as follows. The maximizer can mix the two words ba and bb with equal probability. Call this mixed word w. Then for all strategies τ that the minimizer chooses we have $\mathbb{P}_\gamma^{w,\tau}(F) = 1/2$. On the other hand, after a b is played, the probability to be in the state t_2 is always larger than $1/2$, $\mathbb{P}_\gamma^{a^n b,\tau}(t_2) \geq 1/2$, and consequently the minimizer has an optimal action such that both $\top$ and $\bot$ are reached with equal probability and equal to $1/2$. Moreover this optimal action can be played by the minimizer by keeping track of the distribution on t_1 and t_2 by counting the number of as that are played before b. Albeit this requires unbounded memory. We give a proof of this in what follows.

Assume that the game stops just before the minimizer makes her action, then we have

$$\mathbb{P}_\gamma^{a^n b,\tau}(t_2) = 1 - \frac{1}{2} \cdot \frac{1}{2^n} = \frac{2^{n+1} - 1}{2^{n+1}}$$

Therefore if τ is optimal, after seeing $a^n b$ it would play the action β with the following probability,

$$\tau(a^n b)(\beta) = \frac{1}{2} \cdot \frac{2^{n+1}}{2^{n+1} - 1} = \frac{1}{2} \cdot \frac{1}{1 - \frac{1}{2^{n+1}}}.$$

With such a strategy it would ensure that $\mathbb{P}_\gamma^{a^n b,\tau}(\top) = \mathbb{P}_\gamma^{a^n b,\tau}(\bot) = 1/2$. We prove that this is impossible with a finite-memory strategy.

The proof is by contradiction. Assume that the minimizer has a finite-memory strategy with m states such that against the word $a^n b$ it plays the action β with probability $\frac{1}{2} \cdot \frac{1}{1 - \frac{1}{2^{n+1}}}$. From the definition of a finite-memory strategy, this implies that there exist two $m \times m$ stochastic matrices A and B, and $J \subset \{1, \ldots, m\}$ such that

$$\sum_{j \in J} (A^n B)_{i,j} = \frac{1}{2} \cdot \frac{1}{1 - \frac{1}{2^{n+1}}}, \tag{11}$$

where i is the initial memory location of the strategy, and for a matrix A we denote by $A_{i,j}$ the element on the ith row and jth column. We use the following well-known theorem. See e.g. [15].

Theorem 8. *Let A be a square $m \times m$ stochastic matrix and $\lambda_1, \lambda_1, \ldots, \lambda_r$ ($r \leq m$) its distinct eigenvalues. Then for all $n > m$*

$$(A^n)_{i,j} = \sum_{k=1}^{r} \lambda_k^n P_{ijk}(n),$$

where P_{ijk} are polynomials of smaller order than the multiplicity of λ_k.

Using Theorem 8 and doing a small calculation we see that indeed there exist polynomials $P_1, P_2, \ldots, P_r$ such that for all $n > m$

$$\sum_{j \in J} (A^n B)_{i,j} = \sum_{k=1}^{r} \lambda_k^n P_k(n).$$

On the other hand the Taylor expansion for $\frac{1}{1-\frac{1}{2^{n+1}}}$ give us

$$\sum_{j \in J} (A^n B)_{i,j} = \frac{1}{2} \cdot (1 + \frac{1}{2^{n+1}} + \frac{1}{2^{2(n+1)}} + \cdots).$$

Therefore

$$\sum_{k=1}^{r} \lambda_k^n P_k(n) = \frac{1}{2} \cdot (1 + \frac{1}{2^{n+1}} + \frac{1}{2^{2(n+1)}} + \cdots). \tag{12}$$

Now observe that for complex numbers $z_1, \ldots, z_m$, $m \geq 1$, with $|z_1| = |z_2| = \cdots = |z_m|$, real $c > 0$, and polynomials $f_1, \ldots, f_m$ on n of degree at most d,

$$\lim_{n \to \infty} \frac{c}{\sum_{i=1}^{m} z_i^n f_i(n)} = 1, \tag{13}$$

implies that $\sum_{i=1}^{m} z_i^n f_i(n) = \sum_{i=1}^{m} c_i z_i^n = c$ for some constants c_i. The reason being that (13) clearly cannot be true for $|z_i| < 1$, as for $|z_i| \geq 1$ assume that the dominating term of the denominator has the form $n^k \sum_{i=1}^{m} c_i z_i^n$ for constants c_i, then for (13) to hold we need $k = 0$. Hence $\sum_{i=1}^{m} z_i^n f_i(n) = \sum_{i=1}^{m} c_i z_i^n$, and similarly it is necessary that $|z_1| = |z_2| = \cdots = |z_m| = 1$. Finally because of (13) we have $\sum_{i=1}^{m} c_i z_i^n = c$.

Assume without loss of generality that $|\lambda_1| = |\lambda_2| = \cdots = |\lambda_{r_1}|$, for some $1 \leq r_1 \leq r$ and that $|\lambda_1| \geq |\lambda_i|$, $1 \leq i \leq r$. The expression on the left hand side of (14) is dominated by $\sum_{k=1}^{r_1} \lambda_k^n P_k(n)$ whereas the expression on the right hand side is dominated by the leading term $1/2$.

Consequently, because of the equality above, it holds that

$$\lim_{n \to \infty} \frac{\frac{1}{2}}{\sum_{k=1}^{r_1} \lambda_k^n P_k(n)} = 1.$$

Applying (13) we have $\sum_{k=1}^{r_1} \lambda_k^n P_k(n) = \sum_{k=1}^{r_1} \lambda_k^n c_k = \frac{1}{2}$. We substract both of these equal quantities from (12), to get

$$\sum_{k=r_1}^{r} \lambda_k^n P_k(n) = \frac{1}{2} \cdot (\frac{1}{2^{n+1}} + \frac{1}{2^{2(n+1)}} + \cdots). \tag{14}$$

Repeating the same argument for the leading terms of (14) we have

$$\lim_{n\to\infty} \frac{\frac{1}{2^{n+2}}}{\sum_{k=r_1}^{r_2} \lambda_k^n P_k(n)} = \lim_{n\to\infty} \frac{1}{4\sum_{k=r_1}^{r_2} (2\lambda_k)^n P_k(n)} = 1.$$

Again, applying (13) we get $\sum_{k=r_1}^{r_2} (2\lambda_k)^n P_k(n) = \sum_{k=r_1}^{r_2} c'_k 2^n \lambda_k^n = 1/4$. Hence we can subtract the quantity $\frac{1}{2^{n+2}}$ from both sides in (14). Repeating the same argument for the eigenvalues that are left we conclude that

$$0 = \frac{1}{2} \cdot (\frac{1}{2^{r(n+1)}} + \frac{1}{2^{(r+1)(n+1)}} + \cdots),$$

which is clearly a contradiction therefore there are no two finite stochastic matrices A, B such that (11) holds, and consequently the minimizer has no finite-strategy that is optimal in achieving the $1/2$ payoff. Nevertheless for all $\epsilon > 0$ the minimizer has ϵ-optimal strategies that have finite-memory. These strategies would constitute of counting the number of a's up to some length.

We have shown the following lemma.

Lemma 13. *In the game in Figure 4 for all finite-memory strategies τ for the minimizer there exists a word w such that*

$$\mathbb{P}_\gamma^{w,\tau}(F) > \frac{1}{2},$$

where γ is the initial distribution, $\gamma(s_1) = \gamma(s_2) = 1/2$.

Now we give an example that gives a stronger property. We will use the game in Figure 4 in another game as a gadget. We then demonstrate that for this larger game it also holds that $val(i) = 1/2$ where i is the initial state but $val^f(i) = 1$, i.e. for all finite-memory strategies τ and $\epsilon > 0$ there is a finite word that reaches the set of final states with probability larger than $1 - \epsilon$.

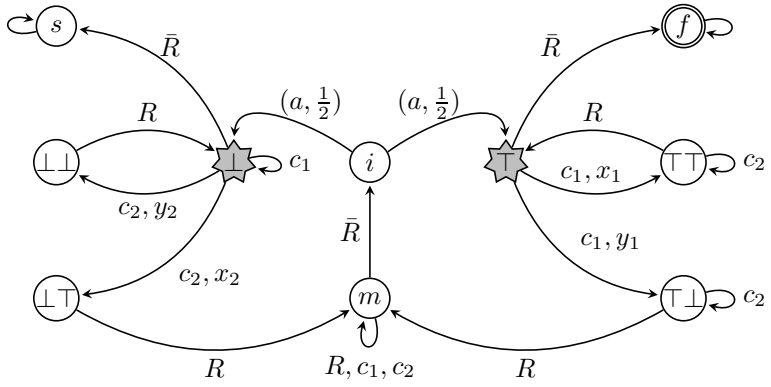

Figure 5: A game for which $val^{\infty}(i) = 1/2$

We give an informal description of the game in Fig. 5. The state i is the initial state. A fair coin is tossed at i and if it is heads then we move to state $\top$ otherwise we move to state $\bot$. Then, we toss a biased coin in $\top$ by playing c_1, if we happen to be in $\bot$, playing c_1 would not change anything. At this point another biased coin is tossed by playing c_2 as a result we are in one of the states $\bot\bot, \bot\top, \top\top, \top\bot$ after the two coin tosses. Repeating this process n times, i.e. by playing $a(c_1c_2R)^n$, we end up in state $\top$ if and only if we had $n+1$ heads and symmetrically we end up in state $\bot$ if and only if we have tossed $n+1$ tails. Now we play $\bar{R}$, and by doing so we win if we have tossed $n+1$ consecutive heads, we lose if we have tossed $n+1$ consecutive tails and otherwise we go to the state i. If we repeat this process k times, i.e. by playing the word

$$(a(c_1c_2R)^n\bar{R})^k,$$

then the probability to win the game will be arbitrarily close to 1 (for well chosen n and k) if and only if the coin tosses are biased towards heads, i.e. $x_1, x_2 > 1/2$. Then the idea is to embed the gadget in Fig.4 in place of the states $\bot$ and $\top$.

For all k let

$$\mu_k = \mathbb{P}_i^{(a(c_1c_2R)^n\bar{R})^k}(\neg\{f, s\}),$$

the probability to be in any state except the sink (s) or final (f) state after the word $(a(c_1c_2R)^n\bar{R})^k$ has been played. Then we have

$$\mu_0 = 1, \text{ and}$$
$$\mu_k = \mu_{k-1}(1 - \frac{1}{2}x_1^n - \frac{1}{2}y_2^n).$$

Hence

$$\mu_k = (1 - \frac{1}{2}x_1^n - \frac{1}{2}y_2^n)^k.$$

Observe that

$$\begin{aligned}\mathbb{P}_i^{(a(c_1c_2R)^n\bar{R})^k}(f) &= \frac{1}{2}x_1^n(\mu_0+\mu_1+\cdots+\mu_{k-1})\\ &= \frac{1}{2}x_1^n\frac{1-(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)^k}{1-(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)}\\ &= \frac{x_1^n}{x_1^n+y_2^n}\cdot(1-(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)^k).\end{aligned}$$

Then there exists some function g such that $\lim_{n\to\infty}(1-\frac{1}{2}x_1^n-\frac{1}{2}y_2^n)^{g(n)}=0$. Also, we have $x_1>y_2$ if and only if $\lim_{n\to\infty}\frac{x_1^n}{x_1^n+y_2^n}=1$.

If we embed the gadget in Fig. 4 in place of the states $\bot$ and $\top$ and replace the letter c_1 with the letters a_1, b_1 from the gadget and symmetrically c_2 with the letters a_2, b_2, and such that the final state of the gadget embedded on the right becomes $\top\top$, the sink state $\top\bot$ and symmetrically the final state of the gadget embedded on the left becomes $\bot\top$ and the sink state $\bot\bot$ together with Lemma 13 implies the following:

Theorem 9. *There exists a game with initial state i, such that $1/2 = val(i) < val^f(i) = 1$.*

The game in Fig. 5 is not leaktight. We conjecture that for leaktight games the finite-memory strategies are as powerful as the infinite-memory ones ($val(s) = val^f(s)$). One can prove that for all distributions on the states of the game there exists an optimal (mixed) action for the minimizer, and this way construct an optimal strategy. This strategy has in general unbounded memory. But intuitively for leaktight games the exact distribution is not important, only the support. We leave the veracity of this conjecture as an open problem.

Conclusion

We have defined a class of stochastic games with partial observation where the maxmin-reachability problem is decidable. This holds under the assumption that maximizer is restricted to deterministic strategies. The extension of this result to the value 1 problem where maximizer is allowed to use mixed strategies seems rather challenging.

References

[1] Alberto Bertoni. The solution of problems relative to probabilistic automata in the frame of the formal languages theory. In *Gl-4. Jahrestagung*, pages 107–112. Springer, 1975.

[2] Rohit Chadha, A Prasad Sistla, and Mahesh Viswanathan. Power of randomization in automata on infinite strings. In *CONCUR 2009-Concurrency Theory*, pages 229–243. Springer, 2009.

[3] Krishnendu Chatterjee and Laurent Doyen. Partial-observation stochastic games: How to win when belief fails. *ACM Transactions on Computational Logic (TOCL)*, 15(2):16, 2014.

[4] Krishnendu Chatterjee, Laurent Doyen, Sumit Nain, and Moshe Y. Vardi. The Complexity of Partial-Observation Stochastic Parity Games with Finite-Memory Strategies. In *FoSSaCS*, volume 8412 of *Lecture Notes in Computer Science*, pages 242–257. Springer, 2014.

[5] Krishnendu Chatterjee and Thomas A. Henzinger. Semiperfect-Information Games. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science*, FSTTCS '05, pages 1–18, Berlin, Heidelberg, 2005. Springer-Verlag.

[6] Krishnendu Chatterjee and Mathieu Tracol. Decidable problems for probabilistic automata on infinite words. In *Logic in Computer Science (LICS), 2012 27th Annual IEEE Symposium on*, pages 185–194. IEEE, 2012.

[7] A.H. Clifford. *The Algebraic Theory of Semigroups*, volume II. American Mathematical Ed., 2 edition, 1964.

[8] Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *Automata, languages and programming*, pages 139–150. Springer, 2009.

[9] Thomas Colcombet. Regular Cost Functions, Part I: Logic and Algebra over Words. *Logical Methods in Computer Science*, 9(3), 2013.

[10] Luca de Alfaro, Thomas A. Henzinger, and Orna Kupferman. Concurrent reachability games. *Theor. Comput. Sci.*, 386(3):188–217, 2007.

[11] Nathanaël Fijalkow. Profinite techniques for probabilistic automata and the optimality of the markov monoid algorithm. *CoRR*, abs/1501.02997, 2015.

[12] Nathanaël Fijalkow, Hugo Gimbert, Edon Kelmendi, and Youssouf Oualhadj. Deciding the value 1 problem for probabilistic leaktight automata. *Logical Methods in Computer Science*, 11(2), 2015.

[13] Nathanaël Fijalkow, Hugo Gimbert, and Youssouf Oualhadj. Deciding the Value 1 Problem for Probabilistic Leaktight Automata. In *LICS*, pages 295–304. IEEE Computer Society, 2012.

[14] Jerzy Filar and Koos Vrieze. *Competitive Markov decision processes*. Springer Science & Business Media, 2012.

[15] Feliks R Gantmacher. *The theory of matrices*, volume 1. Taylor & Francis, 1959.

[16] Hugo Gimbert and Youssouf Oualhadj. Probabilistic Automata on Finite Words: Decidable and Undecidable Problems. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul G. Spirakis, editors, *ICALP (2)*, volume 6199 of *Lecture Notes in Computer Science*, pages 527–538. Springer, 2010.

[17] Hugo Gimbert, Jérôme Renault, Sylvain Sorin, Xavier Venel, and Wieslaw Zielonka. On the values of repeated games with signals. *CoRR*, abs/1406.4248, 2014.

[18] Azaria Paz. Some aspects of probabilistic automata. *Information and Control*, 9(1):26–60, 1966.

[19] Jean Eric Pin and Raymond E Miller. *Varieties of formal languages*.

[20] Michael O. Rabin. Probabilistic Automata. *Information and Control*, 6(3):230–245, September 1963.

[21] Imre Simon. On Semigroups of Matrices over the Tropical Semiring. *ITA*, 28(3-4):277–294, 1994.