Qualitative Determinacy and Decidability of Stochastic Games with Signals *

Nathalie Bertrand[1], Blaise Genest[2], Hugo Gimbert[3]

[1] INRIA, IRISA
Rennes, France
nathalie.bertrand@irisa.fr

[2] CNRS, IRISA
Rennes, France
blaise.genest@irisa.fr

[3] CNRS, LaBRI
Bordeaux, France
hugo.gimbert@labri.fr

Abstract

We consider the standard model of finite two-person zero-sum stochastic games with signals. We are interested in the existence of almost-surely winning or positively winning strategies, under reachability, safety, Büchi or co-Büchi winning objectives. We prove two qualitative determinacy *results. First, in a reachability game either player* 1 *can achieve almost-surely the reachability objective, or player* 2 *can ensure surely the complementary safety objective, or both players have positively winning strategies. Second, in a Büchi game if player* 1 *cannot achieve almost-surely the Büchi objective, then player* 2 *can ensure positively the complementary co-Büchi objective. We prove that players only need strategies with* finite-memory, *whose sizes range from no memory at all to doubly-exponential number of states, with matching lower bounds. Together with the qualitative determinacy results, we also provide fix-point algorithms for deciding which player has an almost-surely winning or a positively winning strategy and for computing the finite memory strategy. Complexity ranges from* EXPTIME *to* 2EXPTIME *with matching lower bounds, and better complexity can be achieved for some special cases where one of the players is better informed than her opponent.*

Introduction

Numerous advances in algorithmics of stochastic games have recently been made [9, 8, 6, 4, 11, 13], motivated in part by application in controller synthesis and verification of open systems. Open systems can be viewed as two-players games between the system and its environment. At each round of the game, both players independently and simultaneously choose actions and the two choices together with the current state of the game determine transition probabilities to the next state of the game. Properties of open systems are modeled as objectives of the games [8, 12], and strategies in these games represent either controllers of the system or behaviors of the environment.

Most algorithms for stochastic games suffer from the same restriction: they are designed for games where players can fully observe the state of the system (e.g. concurrent games [9, 8] and stochastic games with perfect information [7, 13]). The full observation hypothesis can hinder interesting applications in controller synthesis because full monitoring of the system is hardly implementable in practice. Although this restriction is partly relaxed in [16, 5] where one of the players has partial observation and her opponent is fully informed, certain real-life distributed systems cannot be modeled without restricting observations of *both* players.

In the present paper, we consider *stochastic games with signals*, that are a standard tool in game theory to model partial observation [23, 20, 17]. When playing a stochastic game with signals, players cannot observe the actual state of the game, nor the actions played by their opponent, but are only informed via private signals they receive throughout the play. Stochastic games with signals subsume standard stochastic games [22], repeated games with incomplete information [1], games with imperfect monitoring [20], concurrent games [8] and deterministic games with imperfect information on one side [16, 5]. Players make their decisions based upon the sequence of signals they receive: a strategy is hence a mapping from finite sequences of private signals to probability distributions over actions.

From the algorithmic point of view, stochastic games with signals are considerably harder to deal with than stochastic games with full observation. While *values* of the latter games are computable [8, 4], simple questions like 'is there a strategy for player 1 which guarantees winning with probability more than $\frac{1}{2}$?' are *undecidable* even for restricted classes of stochastic games with signals [15]. For this reason, rather than *quantitative* properties (i.e. questions about values), we focus in the present paper on *qualitative* properties of stochastic games with signals.

We study the following qualitative questions about stochastic games with signals, equipped with reachability,

*This work is supported by ANR-06-SETI DOTS.

safety, Büchi or co-Büchi objectives:

(i) Does player 1 have an *almost-surely winning strategy*, i.e. a strategy which guarantees the objective to be achieved with probability 1, whatever the strategy of player 2?

(ii) Does player 2 have a *positively winning strategy*, i.e. a strategy which guarantees the opposite objective to be achieved with strictly positive probability, whatever the strategy of player 1?

Obviously, given an objective, properties (i) and (ii) cannot hold simultaneously. For games with a reachability, safety or Büchi objective, we obtain the following results:

(1) Either property (i) holds or property (ii) holds; in other words these games are *qualitatively determined*.

(2) Players only need strategies with *finite-memory*, whose memory sizes range from no memory at all to doubly-exponential number of states.

(3) Questions (i) and (ii) are decidable. We provide fix-point algorithms for computing uniformly all initial states that satisfy (i) or (ii), together with the corresponding finite-memory strategies. The complexity of the algorithms ranges from EXPTIME to 2EXPTIME.

These three results are detailed in Theorems 1, 2, 3 and 4. We prove that these results are tight and robust in several aspects. Games with co-Büchi objectives are absent from these results, since they are neither qualitatively determined (see Fig. 3) nor decidable (as proven in [2]).

Our *main result*, and the element of surprise, is that for winning positively a safety or co-Büchi objective, a player *needs* a memory with a doubly-exponential number of states, and the corresponding decision problem is 2EXPTIME-complete. This result departs from what was previously known [16, 5], where both the number of memory states and the complexity are simply exponential. These results also reveal a nice property of *reachability* games, that Büchi games do not enjoy: Every initial state is either *almost-surely winning* for player 1, *surely winning* for player 2 or *positively winning* for both.

Our results strengthen and generalize in several ways results that were previously known for concurrent games [9, 8] and deterministic games with imperfect information on one side [16, 5]. First, the framework of stochastic games with signals strictly emcompasses all the settings of [16, 9, 8, 5]. In concurrent games there is no signaling structure at all, and in deterministic games with imperfect information on one side [5] transitions are deterministic and player 2 observes everything that happens in the game, including results of random choices of her opponent.

No determinacy result was known for deterministic games with imperfect information on one side. In [16, 5], algorithms are given for deciding whether the imperfectly informed player has an almost-surely winning strategy for a Büchi (or reachability) objective but nothing can be inferred in case she has no such strategy. This open question is solved in the present paper, in the broader framework of stochastic games with signals.

Our qualitative determinacy result (1) is a radical generalization of the same result for concurrent games [8, Th.2], while proofs are very different. Interestingly, for concurrent games, qualitative determinacy holds for every omega-regular objectives [8], while for games with signals we show that it fails already for co-Büchi objectives. Interestingly also, stochastic games with signals and a reachability objective have a value [19] but this value is not computable [15], whereas it is computable for concurrent games with omega-regular objectives [10]. The use of randomized strategies is mandatory for achieving determinacy results, this also holds for stochastic games without signals [22, 9] and even matrix games [24], which contrasts with [3, 16] where only deterministic strategies are considered.

Our results about randomized finite-memory strategies (2), stated in Theorem 2, are either brand new or generalize previous work. It was shown in [5] that for deterministic games where player 2 is perfectly informed, strategies with a finite memory of exponential size are sufficient for player 1 to achieve a Büchi objective almost-surely. We prove the same result holds for the whole class of stochastic games with signals. Moreover we prove that for player 2 a doubly-exponential number of memory states is necessary and sufficient for achieving positively the complementary co-Büchi objective.

Concerning algorithmic results (3) (see details in Theorem 3 and 4) we show that our algorithms are optimal in the following meaning. First, we give a fix-point based algorithm for deciding whether a player has an almost-surely winning strategy for a Büchi objective. In general, this algorithm is 2EXPTIME. We show in Theorem 5 that this problem is indeed 2EXPTIME-hard. However, in the restricted setting of [5], it is already known that this problem is only EXPTIME-complete. We show that our algorithm is also optimal with an EXPTIME complexity not only in the setting of [5] where player 2 has perfect information but also under weaker hypothesis: it is sufficient that player 2 has *more information* than player 1. Our algorithm is also EXPTIME when player 1 has full information (Proposition 2). In both subcases, player 2 needs only exponential memory.

The paper is organized as follows. In Section 1 we introduce partial observation games, in Section 2 we define the notion of qualitative determinacy and we state our determinacy result, in Section 3 we discuss the memory needed by strategies. Section 4 is devoted to decidability questions and Section 5 investigates the precise complexity of the general problem as well as special cases.

1 Stochastic games with signals.

We consider the standard model of finite two-person zero-sum stochastic games with signals [23, 20, 17]. These are stochastic games where players cannot observe the actual state of the game, nor the actions played by their opponent, their only source of information are private signals they receive throughout the play. Stochastic games with signals subsume standard stochastic games [22], repeated games with incomplete information [1], games with imperfect monitoring [20] and games with imperfect information [5].

Notations. Given a finite set K, we denote by $\mathcal{D}(K) = \{\delta : K \to [0,1] \mid \sum_k \delta(k) = 1\}$ the set of probability distributions on K and for a distribution $\delta \in \mathcal{D}(K)$, we denote $\mathrm{supp}(\delta) = \{k \in K \mid \delta(k) > 0\}$ its support.

States, actions and signals. Two players called 1 and 2 have opposite goals and play for an infinite sequence of steps, choosing actions and receiving signals. Players observe their own actions and signals but they cannot observe the actual state of the game, nor the actions played and the signals received by their opponent. We borrow notations from [17]. Initially, the game is in a state chosen according to an initial distribution $\delta \in \mathcal{D}(K)$ known by both players; the initial state is k_0 with probability $\delta(k_0)$. At each step $n \in \mathbb{N}$, players 1 and 2 choose some actions $i_n \in I$ and $j_n \in J$. They respectively receive signals $c_n \in C$ and $d_n \in D$, and the game moves to a new state k_{n+1}. This happens with probability $p(k_{n+1}, c_n, d_n \mid k_n, i_n, j_n)$ given by fixed transition probabilities $p : K \times I \times J \to \mathcal{D}(K \times C \times D)$, known by both players.

Plays and strategies. Players observe their own actions and the signals they receive. It is convenient to assume that the action i player 1 plays is encoded in the signal c she receives, with the notation $i = i(c)$ (and symmetrically for player 2). This way, plays can be described by sequences of states and signals for both players, without mentioning which actions were played. A finite play is a sequence $p = (k_0, c_1, d_1, \ldots, c_n, d_n, k_n) \in (KCD)^*K$ such that for every $0 \leq m < n$, $p(k_{m+1}, c_{m+1}, d_{m+1} \mid k_m, i(c_{m+1}), j(d_{m+1})) > 0$. An infinite play is a sequence $p \in (KCD)^\omega$ whose prefixes are finite plays.

A (behavioral) strategy of player 1 is a mapping $\sigma : \mathcal{D}(K) \times C^* \to \mathcal{D}(I)$. If the initial distribution is δ and player 1 has seen signals $c_1, \ldots, c_n$ then she plays action i with probability $\sigma(\delta, c_1, \ldots, c_n)$. Strategies for player 2 are defined symmetrically. In the usual way, an initial distribution δ and two strategies σ and τ define a probability measure $\mathbb{P}_\delta^{\sigma,\tau}$ on the set of infinite plays, equipped with the σ-algebra generated by cylinders.

We use random variables K_n, I_n, J_n, C_n and D_n to denote respectively the n-th state, action of player 1, action of player 2, signal of player 1 and signal of player 2.

Winning conditions. The goal of player 1 is described by a measurable event Win called the *winning condition*. Motivated by applications in logic and controller synthesis [12], we are especially interested in *reachability, safety, Büchi and co-Büchi conditions*. These four winning conditions use a subset $T \subseteq K$ of *target states* in their definition. The reachability condition stipulates that T should be visited at least once, $\mathrm{Win} = \{\exists n \in \mathbb{N}, K_n \in T\}$, the safety condition is complementary $\mathrm{Win} = \{\forall n \in \mathbb{N}, K_n \notin T\}$. For the Büchi condition the est of target states has to be visited infinitely often, $\mathrm{Win} = \{\exists A \subseteq \mathbb{N}, |A| = \infty, \forall n \in A, K_n \in T\}$, and the co-Büchi condition is complementary $\mathrm{Win} = \{\exists m \in \mathbb{N}, \forall n \geq m, K_n \notin T\}$.

Almost-surely and positively winning strategies. When player 1 and 2 use strategies σ and τ and the initial distribution is δ, then player 1 wins the game with probability:

$$\mathbb{P}_\delta^{\sigma,\tau}(\mathrm{Win}) \quad .$$

Player 1 wants to maximize this probability, while player 2 wants to minimize it. The best situation for player 1 is when she has an almost-surely winning strategy.

Definition 1 (Almost-surely winning strategy)**.** *A strategy σ for player 1 is* almost-surely winning *from an initial distribution δ if*

$$\forall \tau, \mathbb{P}_\delta^{\sigma,\tau}(\mathrm{Win}) = 1 \quad . \tag{1}$$

When such a strategy σ exists, both δ and its support $\mathrm{supp}(\delta)$ are said to be almost-surely winning as well.

A less enjoyable situation for player 1 is when she only has a positively winning strategy.

Definition 2 (Positively winning strategy)**.** *A strategy σ for player 1 is* positively winning *from an initial distribution δ if*

$$\forall \tau, \mathbb{P}_\delta^{\sigma,\tau}(\mathrm{Win}) > 0 \quad . \tag{2}$$

When such a strategy σ exists, both δ and its support $\mathrm{supp}(\delta)$ are said to be positively winning as well.

The worst situation for player 1 is when her opponent has an almost-surely winning strategy τ, which ensures $\mathbb{P}_\delta^{\sigma,\tau}(\mathrm{Win}) = 0$ for all strategies σ chosen by player 1. Symmetrically, a strategy τ for player 2 is positively winning if it guarantees $\forall \sigma, \mathbb{P}_\delta^{\sigma,\tau}(\mathrm{Win}) < 1$. These notions only depend on the support of δ since $\mathbb{P}_\delta^{\sigma,\tau}(\mathrm{Win}) = \sum_{k \in K} \delta(k) \cdot \mathbb{P}_{\mathbf{1}_k}^{\sigma,\tau}(\mathrm{Win})$.

Consider the one-player game depicted on Fig. 1. The objective of player 1 is to reach state t. The initial distribution is $\delta(1) = \delta(2) = \frac{1}{2}$ and $\delta(t) = \delta(s) = 0$. Player 1

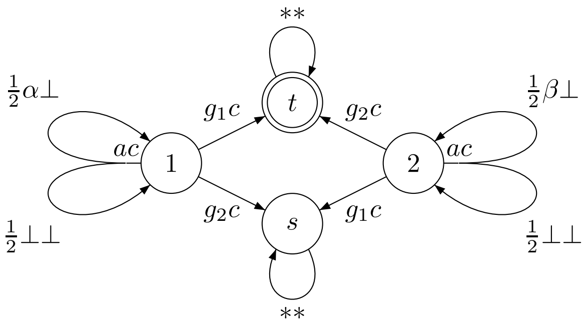

Figure 1. When the initial state is chosen at random between states 1 and 2, player 1 has a strategy to reach t almost surely.

plays with actions $I = \{a, g_1, g_2\}$, where g_1 and g_2 mean respectively 'guess 1' and 'guess 2', while player 2 plays with actions $J = \{c\}$ (that is, player 2 has no choice). Player 1 receives signals $C = \{\alpha, \beta, \bot\}$ and player 2 is 'blind', she always receives the same signal $D = \{\bot\}$. Transitions probabilities are represented in a quite natural way. When the game is in state 1 and both players play a, then player 1 receives signal α or $\bot$ with probability $\frac{1}{2}$, player 2 receives signal $\bot$ and the game stays in state 1. In state 2 when both actions are a's, player 1 cannot receive signal α but instead she may receive signal β. When 'guessing the state' i.e. playing action g_i in state $j \in \{1, 2\}$, player 1 wins the game if $i = j$ (she guesses the correct state) and loses the game if $i \neq j$. The star symbol $*$ stands for any action. In this game, player 1 has a strategy to reach t almost surely. Her strategy is to keep playing action a as long as she keeps receiving signal $\bot$. The day player 1 receives signal α or β, she plays respectively action g_1 or g_2. This strategy is almost-surely winning because the probability for player 1 to receive signal $\bot$ forever is 0.

2 Qualitative Determinacy.

If an initial distribution is positively winning for player 1 then by definition it is *not* almost-surely winning for his opponent player 2. A natural question is whether the converse implication holds.

Definition 3 (Qualitative determinacy)**.** *A winning condition* Win *is* qualitatively determined *if for every game equipped with* Win*, every initial distribution is either almost-surely winning for player* 1 *or positively winning for player* 2*.*

Comparison with value determinacy. *Qualitative determinacy* is similar to but different from the usual notion of *(value) determinacy* which refers to the existence of a *value*. Actually both qualitative determinacy and value determinacy are formally expressed by a quantifier inversion. On one hand, qualitative determinacy rewrites as:

$$(\forall\sigma\ \exists\tau\ \mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) < 1) \implies (\exists\tau\ \forall\sigma\ \mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) < 1) \enspace .$$

On the other hand, the game has a value if:

$$\sup_\sigma \inf_\tau \mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) \geq \inf_\tau \sup_\sigma \mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) \enspace .$$

Both the converse implication of the first equation and the converse inequality of the second equation are obvious.

While *value determinacy* is a classical notion in game theory [14], to our knowledge the notion of *qualitative determinacy* appeared only in the context of omega-regular concurrent games [9, 8] and stochastic games with perfect information [13].

Existence of an almost-surely winning strategy ensures that the value of the game is 1, but the converse is not true. Actually it can even hold that player 2 has a positively winning strategy while at the same time the value of the game is 1. For example, consider the game depicted on Fig. 2, which is a slight modification of Fig. 1 (only signals of player 1 and transitions probabilities differ). Player 1 has signals $\{\alpha, \beta\}$ and similarly to the game on Fig 1, her goal is to reach the target state t by guessing correctly whether the initial state is 1 or 2. On one hand, player 1 can guarantee a winning probability as close to 1 as she wants: she plays a for a long time and compares how often she received signals α and β. If signals α were more frequent, then she plays action g_1, otherwise she plays action g_2. Of course, the longer player 1 plays a's the more accurate the prediction will be. On the other hand, the only strategy available to player 2 (always playing c) is positively winning, because any sequence of signals in $\{\alpha, \beta\}^*$ can be generated with positive probability from both states 1 and 2.

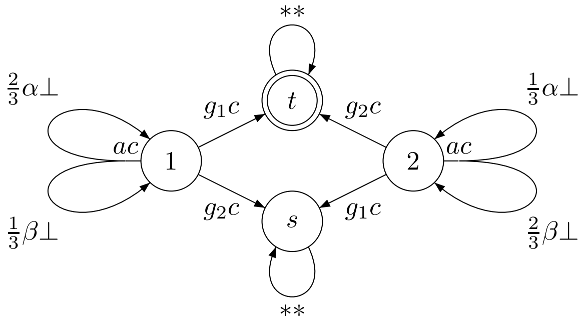

Figure 2. A reachability game with value 1 where player 2 has a positively winning strategy.

Qualitative determinacy results. The first main result of this paper is the qualitative determinacy of stochastic games with signals for the following winning objectives.

Theorem 1. *Reachability, safety and Büchi games are qualitatively determined.*

While qualitative determinacy of safety games is not too hard to establish, proving determinacy of Büchi games is harder. Notice that the qualitative determinacy of Büchi games implies the qualitative determinacy of reachability games, since any reachability game can be turned into an equivalent Büchi one by making all target states absorbing.

The proof of Theorem 1 is postponed to Section 4, where the determinacy result will be completed by a decidability result: there are algorithms for computing which initial distributions are almost-surely winning for player 1 or positively winning for player 2. This is stated precisely in Theorems 3 and 4.

A consequence of Theorem 1 is that in a reachability game, every initial distribution is either almost-surely winning for player 1, *surely* winning for player 2, or positively winning for both players. Surely winning means that player 2 has a strategy τ for preventing every finite play consistent with τ from visiting target states.

Büchi games do not share this nice feature because co-Büchi games are not qualitatively determined. An example of a co-Büchi game which is not determined is represented in Fig. 3. In this game, player 1 observes everything, player 2 is blind (she only observes her own actions), and player 1's objective is to avoid state t from some moment on. The initial state is t.

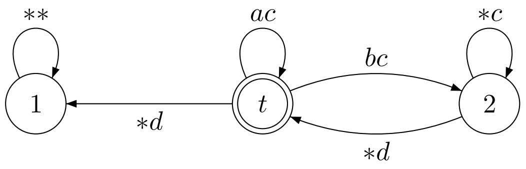

Figure 3. Co-Büchi games are not qualitatively determined.

On one hand, player 1 does not have an almost-surely winning strategy for the co-Büchi objective. Fix a strategy σ for player 1 and suppose it is almost-surely winning. To win against the strategy where player 2 plays c forever, σ should eventually play a b with probability 1. Otherwise, the probability that the play stays in state t is positive, and σ is not almost-surely winning, a contradiction. Since σ is fixed there exists a date after which player 1 has played b with probability arbitrarily close to 1. Consider the strategy of player 2 which plays d at that date. Although player 2 is blind, obviously she can play such a strategy which requires only counting time elapsed since the beginning of the play. With probability arbitrarily close to 1, the game is in state 2 and playing a d puts the game back in state t. Playing long sequences of c's followed by a d, player 2 can ensure with probability arbitrarily close to 1 that if player 1 plays according to σ, the play will visit states t and 2 infinitely often, hence will be lost by player 1. This contradicts the existence of an almost-surely winning strategy for player 1.

On the other hand, player 2 does not have a positively winning strategy either. Fix a strategy τ for player 2 and suppose it is positively winning. Once τ is fixed, player 1 knows how long she should wait so that if action d was never played by player 2 then there is arbitrarily small probability that player 2 will play d in the future. Player 1 plays a for that duration. If player 2 plays a d then the play reaches state 1 and player 1 wins, otherwise the play stays in state t. In the latter case, player 1 plays action b. Player 1 knows that with very high probability player 2 will play c forever in the future, in that case the play stays in state 2 and player 1 wins. If player 1 is very unlucky then player 2 will play d again, but this occurs with small probability and then player 1 can repeat the same process again and again. Similar examples can be used to prove that stochastic Büchi games with signals do not have a value [18].

3 Memory needed by strategies.

3.1 Finite-memory strategies.

Since our ultimate goal are algorithmic results and controller synthesis, we are especially interested in strategies that can be finitely described, like finite-memory strategies.

Definition 4 (Finite-memory strategy)**.** *A* finite-memory strategy *for player* 1 *is given by a finite set* M *called the memory together with a strategic function* $\sigma_M : M \to \mathcal{D}(I)$*, an update function* $\mathrm{upd}_M : M \times C \to \mathcal{D}(M)$*, and an initialization function* $\mathrm{init}_M : \mathcal{P}(K) \to \mathcal{D}(M)$*. The* memory size *is the cardinal of* M*.*

In order to play with a finite-memory strategy, a player proceeds as follows. She initializes the memory of σ to $\mathrm{init}_M(L)$, where $L = \mathrm{supp}(\delta)$ is the support of the initial distribution δ. When the memory is in state $m \in M$, she plays action i with probability $\sigma_M(m)(i)$ and after receiving signal c, the new memory state is m' with probability $\mathrm{upd}_M(m,c)(m')$.

On one hand it is intuitively clear how to play with a finite-memory strategy, on the other hand the behavioral strategy associated with a finite-memory strategy[1] can be

[1] precisely defined in the Appendix.

quite complicated and requires the player to use infinitely many different probability distributions to make random choices (see discussions in [9, 8, 13]).

In the games we consider, the construction of finite-memory strategies is often based on the notion of *belief*. The belief of a player at some moment of the play is the set of states she thinks the game could possibly be in, according to the signals she received so far.

Definition 5 (Belief)**.** *From an initial set of states $L \subseteq K$, the belief of player 1 after receiving signal c (hence playing action $i(c)$), is the set of states k such that there exists a state l in L and a signal $d \in D$ with $p(k, c, d \mid l, i(c), j(d)) > 0$. The belief of player 1 after receiving a sequence of signals $c_1, \ldots, c_n$ is defined inductively by:*

$$\mathcal{B}_1(L, c_1, \ldots, c_n) = \mathcal{B}_1(\mathcal{B}_1(L, c_1, \ldots, c_{n-1}), c_n).$$

Beliefs of player 2 are defined similarly.

Our second main result is that for the qualitatively determined games of Theorem 1, finite-memory strategies are sufficient for both players. The amount of memory needed by these finite-memory strategies is summarized in Table 1 and detailed in Theorem 2.

	Almost-surely	**Positively**
Reachability	belief	memoryless
Safety	belief	doubly-exp
Büchi	belief	
Co-Büchi		doubly-exp

Table 1. Memory required by strategies.

Theorem 2 (Finite-memory is sufficient)**.** *Every reachability game is either won positively by player 1 or won surely by player 2. In the first case playing randomly any action is a positively winning strategy for player 1 and in the second case player 2 has a surely winning strategy with finite-memory $\mathcal{P}(K)$ and update function $\mathcal{B}_2$.*

Every Büchi game is either won almost-surely by player 1 or won positively by player 2. In the first case player 1 has an almost-surely winning strategy with finite-memory $\mathcal{P}(K)$ and update function $\mathcal{B}_1$. In the second case player 2 has a positively winning strategy with finite-memory $\mathcal{P}(\mathcal{P}(K) \times K)$.

The situation where a player needs the least memory is when she wants to win positively a reachability game. To do so, she uses a memoryless strategy consisting in playing randomly any action.

To win almost-surely games with reachability, safety and Büchi objectives, it is sufficient for a player to remember her belief. A canonical almost-surely winning strategy consists in playing randomly any action which ensures the next belief to be almost-surely winning [2]. Similar strategies were used in [5]. These two results are not very surprising: although they were not stated before as such, they can be proved using techniques similar to those used in [16, 5].

The element of surprise is the amount of memory needed for winning positively co-Büchi and safety games. In these situations, it is still enough for player 1 to use a strategy with finite-memory but, surprisingly perhaps, an exponential size memory is not enough. Instead doubly-exponential memory is necessary as will be proved in the next subsection.

Doubly-exponential size memory is also sufficient. Actually for winning positively, it is enough for player 1 to make hypothesis about beliefs of player 2, and to store in her memory all pairs (k, L) of possible current state and belief of her opponent. The update operator of the corresponding finite-memory strategy uses numerous random choices so that the opponent is unable to predict future moves. More details are available in the proof of Theorem 4.

3.2 Doubly-exponential memory is necessary to win positively safety games.

We now show that a doubly-exponential memory is necessary to win positively safety (and hence co-Büchi) games. We construct, for each integer n, a reachability game, whose number of state is polynomial in n and such that player 2 has a positively winning strategy for her safety objective. This game, called $\mathsf{guess\_my\_set}_n$, is described on Fig. 4. The objective of player 2 is to stay away from t, while player 1 tries to reach t.

We prove that whenever player 2 uses a finite-memory strategy in the game $\mathsf{guess\_my\_set}_n$ then the size of the memory has to be doubly-exponential in n, otherwise the safety objective of player 2 may not be achieved with positive probability. This is stated precisely later in Proposition 1. Prior to that, we briefly describe the game $\mathsf{guess\_my\_set}_n$ for fixed $n \in \mathbb{N}$.

Idea of the game. The game $\mathsf{guess\_my\_set}_n$ is divided into three parts. In the first part, player 1 generates a set $X \subsetneq \{1, \ldots, n\}$ of size $|X| = n/2$. There are $\binom{n}{n/2}$ possibilities of such sets X. Player 2 is blind in this part and has no action to play.

In the second part, player 1 announces by her actions $\frac{1}{2}\binom{n}{n/2}$ (pairwise different) sets of size $n/2$ which are different from X. Player 2 has no action to play in that part, but she observes the actions of player 1 (and hence the sets announced by player 1).

In the third part, player 2 can announce by her action up to $\frac{1}{2}\binom{n}{n/2}$ sets of size $n/2$. Player 1 observes actions of

[2] for reachability and safety games, we suppose without loss of generality that target states are absorbing.

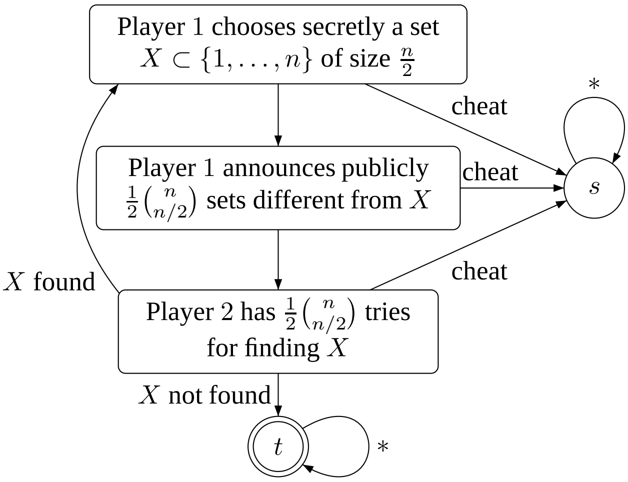
Player 1 chooses secretly a set
$X \subset \{1,\dots,n\}$ of size $\frac{n}{2}$
Player 1 announces publicly
$\frac{1}{2}\binom{n}{n/2}$ sets different from X
Player 2 has $\frac{1}{2}\binom{n}{n/2}$ tries
for finding X
cheat
cheat
cheat
$*$
s
X found
X not found
t
$*$

Figure 4. A game where player 2 needs a lot of memory to stay away from target state t.

player 2. If player 2 succeeds in finding the set X, the game restarts from scratch. Otherwise, the game goes to state t and player 1 wins.

It is worth noticing that in order to implement the game $\mathsf{guess\_my\_set}_n$ in a compact way, we allow player 1 to cheat, and rely on probabilities to always have a chance to catch player 1 cheating, in which case the game is sent to the sink state s, and player 1 loses. That is, player 1 has to play following the rules without cheating else she cannot win almost-surely her reachability objective. Notice also that player 1 is better informed than player 2 in this game.

Concise encoding. We now turn to a more formal description of the game $\mathsf{guess\_my\_set}_n$, to prove that it can be encoded with a number of states polynomial in n. There are three problems to be solved, that we sketch here. First, remembering set X in the state of the game would ask for an exponential number of states. Instead, we use a fairly standard technique: recall at random a single element $x \in X$. In order to check that a set Y of size $n/2$ is different from the set X of size $n/2$, we challenge player 1 to point out some element $y \in Y \setminus X$. We ensure by construction that $y \in Y$, for instance by asking it when Y is given. This way, if player 1 cheats, then she will give $y \in X$, leaving a positive probability that $y = x$, in which case the game is sure that player 1 is cheating and punishes player 1 by sending her to state s where she loses.

The second problem is to make sure that player 1 generates an exponential number of pairwise different sets $X_1, X_2, \dots, X_{\frac{1}{2}\binom{n}{n/2}}$. Notice that the game cannot recall even one set. Instead, player 1 generates the sets in some total order, denoted $<$, and thus it suffices to check only one inequality each time a set X_{i+1} is given, namely $X_i < X_{i+1}$. It is done in a similar but more involved way as before, by remembering randomly two elements of X_i instead of one.

The last problem is to count up to $\frac{1}{2} \cdot \binom{n}{n/2}$ with a logarithmic number of bits. Again, we ask player 1 to increment a counter, while remembering only one of the bits and punishing her if she increments the counter wrongly.

Proposition 1. *Player 2 has a finite-memory strategy with $3 \times 2^{\frac{1}{2}\cdot\binom{n}{n/2}}$ different memory states to win positively* $\mathsf{guess\_my\_set}_n$*.*

No finite-memory strategy of player 2 with less than $2^{\frac{1}{2}\cdot\binom{n}{n/2}}$ memory states wins positively $\mathsf{guess\_my\_set}_n$*.*

Proof. The first claim is quite straightforward. Player 2 remembers in which part she is (3 different possibilities). In part 2, player 2 remembers all the sets proposed by player 1 ($2^{\frac{1}{2}\cdot\binom{n}{n/2}}$ possibilities). Between part 2 and part 3, player 2 inverses her memory to remember the sets player 1 did not propose (still $2^{\frac{1}{2}\cdot\binom{n}{n/2}}$ possibilities). Then she proposes each of these sets, one by one, in part 3, deleting the set from her memory after she proposed it. Let us assume first that player 1 does not cheat and plays fair. Then all the sets of size $n/2$ are proposed (since there are $2 \cdot \frac{1}{2} \cdot \binom{n}{n/2}$ such sets), that is X has been found and the game starts another round without entering state t. Else, if player 1 cheats at some point, then the probability to reach the sink state s is non zero, and player 2 also wins *positively* her safety objective.

The second claim is not hard to show either. The strategy of player 1 is to never cheat, which prevents the game from entering the sink state. In part 2, player 1 proposes the sets X in a lexicographical way and uniformly at random. Assume by contradiction that player 2 has a counter strategy with strictly less than $2^{\frac{1}{2}\cdot\binom{n}{n/2}}$ states of memory that wins positively the safety objective. Consider the end of part 2, when player 1 has proposed $\frac{1}{2} \cdot \binom{n}{n/2}$ sets. If there are less than $2^{\frac{1}{2}\cdot\binom{n}{n/2}}$ states the memory of player 2 can be in, then there exists a memory state m_* of player 2 and at least two sets A, B among the $\frac{1}{2} \cdot \binom{n}{n/2}$ sets proposed by player 1 such that the memory of player 2 after A is m_* with non zero probability and the memory of player 2 after B is m_* with non zero probability. Now, $\overline{A} \cup \overline{B}$ has strictly more than $\frac{1}{2} \cdot \binom{n}{n/2}$ sets of $n/2$ elements. Hence, there is a set $X \in \overline{A} \cup \overline{B}$ with a positive probability not to be proposed by player 2 after memory state m_*. Without loss of generality, we can assume that $X \notin A$ (the other case $X \notin B$ is symmetrical). Now, for each round of the game, there is a positive probability that X is the set in the memory of player 1, that player 1 proposed sets A, in which case player 2 has a (small) probability not to propose X and then the game goes to t, where player 1 wins. Player 1 will thus eventually reach the target state with probability 1, hence a contradiction. This achieves the proof that no finite-memory strategy of player 2 with less than $2^{\frac{1}{2}\cdot\binom{n}{n/2}}$ states of memory is positively winning. □

4 Decidability.

We turn now to the algorithms which compute the set of supports that are almost-surely or positively winning for various objectives.

Theorem 3 (Deciding positive winning in reachability games)**.** *In a reachability game each initial distribution δ is either positively winning for player 1 or surely winning for player 2, and this depends only on $\operatorname{supp}(\delta) \subseteq K$. The corresponding partition of $\mathcal{P}(K)$ is computable in time $\mathcal{O}\left(G \cdot 2^K\right)$, where G denotes the size of the description of the game. The algorithm computes at the same time the finite-memory strategies described in Theorem 2.*

As often in algorithmics of game theory, the computation is achieved by a fix-point algorithm.

Sketch of proof. The set of supports $\mathcal{L} \subseteq \mathcal{P}(K)$ surely-winning for player 2 are characterized as the largest fix-point of some monotonic operator $\Phi : \mathcal{P}(\mathcal{P}(K)) \to \mathcal{P}(\mathcal{P}(K))$. The operator Φ associates with $\mathcal{L} \subseteq \mathcal{P}(K)$ the set of supports $L \in \mathcal{L}$ that do not intersect target states and such that player 2 has an action which ensures that her next belief is in $\mathcal{L}$ as well, whatever action is chosen by player 1 and whatever signal player 2 receives. For $\mathcal{L} \subseteq \mathcal{P}(K)$, the value of $\Phi(\mathcal{L})$ is computable in time linear in $\mathcal{L}$ and in the description of the game, yielding the exponential complexity bound. □

To decide whether player 1 wins almost-surely a Büchi game, we provide an algorithm which runs in doubly-exponential time and uses the algorithm of Theorem 3 as a sub-procedure.

Theorem 4 (Deciding almost-sure winning in Büchi games)**.** *In a Büchi game each initial distribution δ is either almost-surely winning for player 1 or positively winning for player 2, and this depends only on $\operatorname{supp}(\delta) \subseteq K$. The corresponding partition of $\mathcal{P}(K)$ is computable in time $\mathcal{O}(2^{2^G})$, where G denotes the size of the description of the game. The algorithm computes at the same time the finite-memory strategies described in Theorem 2.*

Sketch of proof. The proof of Theorem 4 is based on the following ideas.

First, suppose that from *every* initial support player 1 can win the *reachability objective* with positive probability. Then, repeating the same strategy, Player 1 can guarantee the Büchi condition to hold with probability 1. According to Theorem 3, in the remaining case there exists a support L surely winning for player 2 for her co-Büchi objective.

We prove that in case player 2 can *force the belief of player* 1 to be L someday with positive probability from another support L', then L' is positively winning as well for player 2. This is not completely obvious because in general player 2 cannot know exactly *when* the belief of player 1 is L. For winning positively from L', player 2 plays totally randomly until she guesses randomly that the belief of player 1 is L, at that moment she switches to a strategy surely winning from L. Such a strategy is far from being optimal, because player 2 plays randomly and in most cases she makes a wrong guess about the belief of player 1. However player 2 wins positively because there is a chance she is lucky and guesses correctly at the right moment the belief of player 1.

Player 1 should surely avoid her belief to be L or L' if she wants to win almost-surely. However, doing so player 1 may prevent the play from reaching target states, which may create another positively winning support for player 2, and so on...

Using these ideas, we prove that the set $\mathcal{L}_\infty \subseteq \mathcal{P}(K)$ of supports almost-surely winning for player 1 for the Büchi objective is the largest set of initial supports from where

(†) player 1 has a strategy for winning positively the reachability game while ensuring at the same time her belief to stay in $\mathcal{L}_\infty$.

Property (†) can be reformulated as a reachability condition in a new game whose states are states of the original game augmented with beliefs of player 1, kept hidden to player 2.

The fix-point characterization suggests the following algorithm for computing the set of supports positively winning for player 2: $\mathcal{P}(K)\backslash\mathcal{L}_\infty$ is the limit of the sequence $\emptyset = \mathcal{L}'_0 \subsetneq \mathcal{L}'_0 \cup \mathcal{L}''_1 \subsetneq \mathcal{L}'_0 \cup \mathcal{L}'_1 \subsetneq \mathcal{L}'_0 \cup \mathcal{L}'_1 \cup \mathcal{L}''_2 \subsetneq \ldots \subsetneq \mathcal{L}'_0 \cup \cdots \cup \mathcal{L}'_m = \mathcal{P}(K)\backslash\mathcal{L}_\infty$, where

(a) from supports in $\mathcal{L}''_{i+1}$ player 2 can surely guarantee the safety objective, under the hypothesis that player 1 beliefs stay outside $\mathcal{L}'_i$,

(b) from supports in $\mathcal{L}'_{i+1}$ player 2 can ensure with positive probability the belief of player 1 to be in $\mathcal{L}''_{i+1}$ someday, under the same hypothesis.

The overall strategy of player 2 positively winning for the co-Büchi objective consists in playing randomly for some time until she decides to pick up randomly a belief L of player 1 in some $\mathcal{L}''_i$. She forgets the signals she has received up to that moment and switches definitively to a strategy which guarantees (a). With positive probability, player 2 is lucky enough to guess correctly the belief of player 1 at the right moment, and future beliefs of player 1 will stay in $\mathcal{L}'_i$, in which case the co-Büchi condition holds. □

Property † can be formulated by mean of a fix-point according to Theorem 3, hence the set of supports positively winning for player 2 can be expressed using two embedded fix-points. This should be useful for actually implementing the algorithm and for computing symbolic representations of winning sets.

5 Complexity and special cases.

In this section we show that our algorithms are optimal regarding complexity. Furthermore, we show that these algorithms enjoy better complexity in restricted cases, generalizing some known algorithms [16, 5] to more general subcases, while keeping the same complexity.

The special cases that we consider regard inclusion between knowledges of players. To this end, we define the following notion. If at each moment of the game the belief of player x is included in the one of player y, then player x is said to have more information (or to be better informed) than player y. It is in particular the case when for every transition, the signal of player 1 contains the signal of player 2.

5.1 Lower bound.

We prove here that the problem of knowing whether the initial support of a reachability game is almost-surely winning for player 1 is 2EXPTIME-complete. The lower bound even holds when player 1 is more informed than player 2.

Theorem 5. *In a reachability game, deciding whether player 1 has an almost-surely winning strategy is 2EXPTIME-hard, even if player 1 is more informed than player 2.*

Sketch of proof. To prove the 2EXPTIME-hardness we do a reduction from the membership problem for alternating EXPSPACE Turing machines. Let $\mathcal{M}$ be such a Turing machine and w be an input word of length n. Player 1 is responsible for choosing the successor configuration in existential states while player 2 owns universal states. The role of player 2 is to simulate an execution of $\mathcal{M}$ on w according to the rules she and player 1 choose. For each configuration she thus enumerates the tape contents. Player 1 aims at reaching target states, which are configurations where the state is the final state of the Turing machine. Hence, if player 2 does not cheat in her task, player 1 has a surely winning strategy to reach her target if and only if w is accepted by $\mathcal{M}$. However player 2 could cheat while describing the tape contents, that is she could give a configuration not consistent with the previous configuration and the chosen rule. To be able to detect the cheating and punish player 2, one has to remember a position of the tape. Unfortunately, the polynomial-size game cannot remember this position directly, as there are exponentially many possibilities. Instead, we use player 1 to detect the cheating of player 2. She will randomly choose a position and the corresponding letter to remember, and check at the next step that player 2 did not cheat on this position. To prevent player 1 from cheating, that is saying player 2 cheats although she did not, some information is remembered in the states of the game (but hidden to both players). Here again, the game cannot remember the precise position of the letter chosen by player 1, since it could be exponential in n, so she randomly remembers a bit of the binary encoding of the letter's position. This way, both players can be caught if they cheat. If the play reaches a final configuration of $\mathcal{M}$, player 1 wins. If player 2 cheats and player 1 delates her, the play is won by player 1. Player 1 has a reset action in case she witnesses player 2 has cheated, but she was not caught. If player 1 cheats and is caught by the game, the play is won by player 2. This construction ensures that player 1 has an almost sure winning strategy if and only if w is accepted by the alternating Turing machine $\mathcal{M}$. Indeed, on the one hand, if w is accepted, player 2 needs to cheat infinitely often (after each reset), so that the final state of $\mathcal{M}$ is not reached. Player 1 has no interest in cheating, and at each step, she has a positive probability (uniformely bounded by below) to catch player 2 cheating, and thus to win the play. Hence, player 1 wins almost-surely. On the other hand, if w is not accepted by $\mathcal{M}$, player 2 shouldn't cheat. The only way for player 1 to win, is to cheat, by denonciating player 2 even if she didn't cheat. Here, there is a positive probability that the game remembered the correct bit, that testifies that player 1 cheated, and this causes the loss of player 1. Hence, player 1 does not have an almost-sure strategy. □

5.2 Special cases.

A first straightforward result is that in a safety game where player 1 has full information, deciding whether she has an almost-surely winning strategy is in PTIME.

Now, consider a Büchi game. In general, as shown in the previous section, deciding whether the game is almost-surely winning for player 1 is 2EXPTIME-complete. However, it is already known that when player 2 has a full observation of the game the problem is EXPTIME-complete only [5]. We show that our algorithm keeps the same EXPTIME upper-bound even in the more general case where player 2 is more informed than player 1, as well as in the case where player 1 fully observes the state of the game.

Proposition 2. *In a Büchi game where either player 2 has more information than player 1 or player 1 has complete observation, deciding whether player 1 has an almost-surely winning strategy or not (in which case player 2 has* *a positively winning strategy) can be done in exponential time.*

Sketch of proof. In both cases, player 2 needs only exponential memory because if player 2 has more information, there is always a *unique* belief of player 1 compatible with her signals, and in case player 1 has complete observation her belief is always a singleton set. □

Note that the latter proposition does not hold when player 1 has more information than player 2. Indeed in the game from the proof of Theorem 5, player 1 does have more information than player 2 (but she does not have full information).

6 Conclusion.

We considered stochastic games with signals and established two determinacy results. First, a reachability game is either almost-surely winning for player 1, surely winning for player 2 or positively winning for both players. Second, a Büchi game is either almost-surely winning for player 1 or positively winning for player 2. We gave algorithms for deciding in doubly-exponential time which case holds and for computing winning strategies with finite memory.

The question 'does player 1 have a strategy for winning positively a Büchi game?' is undecidable [2], even when player 1 is blind and alone. An interesting research direction is to design subclasses of stochastic games with signals for which the problem is decidable, for example it should hold for deterministic games of [5] with complete observation on one side [21].

References

[1] R. J. Aumann. *Repeated Games with Incomplete Information.* MIT Press, 1995.

[2] C. Baier, N. Bertrand, and M. Größer. On decision problems for probabilistic Büchi automata. In *Proc. of FOSSACS'08*, vol. 4972 of *LNCS*, pp. 287–301. Springer, 2008.

[3] D. Berwanger, K. Chatterjee, L. Doyen, T. A. Henzinger, and S. Raje. Strategy construction for parity games with imperfect information. In *Proc. of CONCUR'08*, vol. 5201 of *LNCS*, pp. 325–339. Springer, 2008.

[4] K. Chatterjee, L. de Alfaro, and T. A. Henzinger. The complexity of stochastic Rabin and Streett games. In *Proc. of ICALP'05*, vol. 3580 of *LNCS*, pp. 878–890. Springer, 2005.

[5] K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Algorithms for omega-regular games of incomplete information. *Logical Methods in Computer Science*, 3(3), 2007.

[6] K. Chatterjee, M. Jurdzinski, and T. A. Henzinger. Quantitative stochastic parity games. In *Proc. of SODA'04*, pp. 121–130. SIAM, 2004.

[7] A. Condon. The complexity of stochastic games. *Information and Computation*, 96:203–224, 1992.

[8] L. de Alfaro and T. A. Henzinger. Concurrent omega-regular games. In *Proc. of LICS'00*, pp. 141–154. IEEE, 2000.

[9] L. de Alfaro, T. A. Henzinger, and O. Kupferman. Concurrent reachability games. *Theoretical Computer Science*, 386(3):188–217, 2007.

[10] L. de Alfaro and R. Majumdar. Quantitative solution of omega-regular games. In *Proc. of STOC'01*, pp. 675–683. ACM, 2001.

[11] H. Gimbert and F. Horn. Simple stochastic games with few random vertices are easy to solve. In *Proc. of FOSSACS'08*, vol. 4972 of *LNCS*, pp. 5–19. Springer, 2008.

[12] E. Grädel, W. Thomas, and T. Wilke. *Automata, Logics and Infinite Games*, vol. 2500 of *LNCS*. Springer, 2002.

[13] F. Horn. *Random Games*. PhD thesis, Université Denis-Diderot, 2008.

[14] J.-F. Mertens and A. Neyman. Stochastic games have a value. In *Proc. of the National Academy of Sciences USA*, vol. 79, pp. 2145–2146, 1982.

[15] A. Paz. *Introduction to probabilistic automata*. Academic Press, 1971.

[16] J. H. Reif. Universal games of incomplete information. In *Proc. of STOC'79*, pp. 288–308. ACM, 1979.

[17] J. Renault. The value of repeated games with an informed controller. Technical report, CEREMADE, Paris, Jan. 2007.

[18] J. Renault. Personal Communication, July 2008.

[19] J. Renault and S. Sorin. Personal Communication, June 2008.

[20] D. Rosenberg, E. Solan, and N. Vieille. Stochastic games with imperfect monitoring. Technical Report 1376, Northwestern University, July 2003.

[21] O. Serre. Personal Communication, January 2009.

[22] L. S. Shapley. Stochastic games. In *Proc. of the National Academy of Sciences USA*, vol. 39, pp. 1095–1100, 1953.

[23] S. Sorin. *A first course on zero-sum repeated games*. Springer, 2002.

[24] J. von Neumann and O. Morgenstern. *Theory of games and economic behavior*. Princeton University Press, 1944.

Technical Appendix

A Details for Section 3

We give here all the details for encoding the game $\mathsf{guess\_my\_set}_n$ with a game of polynomial size. First, we describe how to ensure that a player does exponentially many steps. We show this for a game with one and a half player, that is one of the player has no move available. This game can thus be applied to any player.

A.1 Exponential number of steps

Let $y_1 \cdots y_n$ be the binary encoding of a number y exponential in n (y_n being the parity of x). Here is a reachability game that the player needs to play for ny steps to surely win. Intuitively, the player needs to enumerate one by one the successors of 0 until reaching $y_1 \cdots y_n$ in order to win. Let say $x'_1 \cdots x'_n$ is the binary encoding of the successor counter x' of counter x. In order to check that the player does not cheat, the bit x'_i for a random i is secretly remembered. It can be easily computed on the fly reading $x_i \ldots x_n$. Indeed, $x'_i = x_i$ iff there exists some $k > i$ with $x_k = 0$.

Action a and signal coincide, and $a \in \{0,1,2\}$, $a \in \{0,1\}$ standing for the current bit x_i, and $a = 2$ standing for the fact that the player claims having reached x.

The state space is basically the following: $(i,b,j,b',j',c)_{i,j,j' \leq n, x, x' \in \{0,1\}}$. The signification of such a state is that the player will give bit x_i, b, j are the check to make to the current number (checking that $x_j = b$), b', j' are the check to make to the successor of x ($x'_{j'} = b'$) , and c indicates whether there is a carry (correcting b' in case $c = 1$ at the end of the current number ($i = n$)). The initial distribution is the uniform distribution on $(0,0,k,0,1)$ (checking that the initial number generated is indeed 0). If the player plays 2, then if $y_j = b$ the game goes to the goal state, else it goes to a sink state s.

We have $P((i,b,j,b',j',c),a,\mathrm{s}) = 1$ if $i = j$ and $a \neq b$. Else, if $i \neq n$, $P((i,b,j,b',j',c),a,(i+1,b,j,b',j',c \wedge a)) = \frac{1}{2}$ (the current bit will not be checked, and the carry is 1 if both c and a are 1), and $P((i,b,j,b',j',c),a,(i+1,b,j,a,i,1)) = 1/2$. At last, for $i = n$, we have $P((i,b,j,b',j',c),a,(1,b' \wedge c,j',a,1,1)) = 1$ (the bit of the next number becomes the bit for the current configuration, taking care of the carry c). Clearly, if the player does not play yn steps of the game, then it means she did not compute accurately the successor at one step, hence it has a chance to get caught and lose. That is, the probability to reach the goal state is not 1.

A.2 Implementing $\mathsf{guess\_my\_set}_n$ with a polynomial size game.

We now turn to the formal definition of $\mathsf{guess\_my\_set}_n$, with a number of states polynomial in n. At each time (but in state s), player 1 can restart the game from the begining (but from the sink state), we will say that it performs another round of the game.

The first part of the game is fairly standard, it consists in asking player 1 (who wants to reach some goal) for a set X of $n/2$ numbers below n. The states of the game are of the form (x,i), where x is the number remembered by the system (hidden for both players), and $i \leq n-2$ is the size of X so far. Player 1 actions and signals are the same, equal to $\{0,\ldots,n\}$, There is no action nor signal for player 2. We have $P((x,i),x,s) = 1$ (player 1 is caught cheating by proposing again the same number remembered by the system). For all $y \neq x$, we have $P((x,i),y,(x,i+1)) = 1/2$ (the number y is accepted as new and the memory x is not updated), $P((x,i),y,(y,i+1) = 1/2$ (the number y is accepted as new and the memory is updated $x := y$). If player 1 plays 0, it means that she has given $n/2$ number, the system checks that the current state is indeed $(x,n/2)$ and goes to the next part. If the current state is not $(x,n/2)$, then it goes to s and player 1 looses.

The number x in the memory of the system at the end of part 1 will be used and remembered all along this round of the game in the other parts. We turn now to the second part, where player 1 gives $\frac{1}{2} \cdot \binom{n}{n/2}$ sets Y different to X. First, in order to be sure that every set Y she proposes is never X, player 1 is asked to give one number in $Y \setminus \{x\}$ (this number is not observed by player 2). Giving x sends the game into the sink state s from which player 1 loses. Since player 1 does not know what x is, playing any number in X is dangerous and ensures that the probability of the play reaching the sink state s is stricly positive, hence it cannot reach its goal almost surely. The way the sets are announced by player 1 is the following. First, player 1 is asked whether number 1 belongs to the set it is annoucing (she plays a if yes, a' if not, and a'' if it is and furthermore it is the biggest number which will change compared to the following set). Player 2 has no choice of action to play. The observation of player 1 and 2 is the same as the action of player 1, that is player 2 is informed of the sets announced by player 1.

Second, the game needs to ensure that each set is different. For that, it asks player 1 to generate the sets in lexicographic order (if Y is given before Y', then there exists $i, j \in Y \times Y'$ such that $i < j$ and for all $k \in X$ with $k > i$, $k \in X'$ and $k \neq j$), and to announce in its action what is the biggest number i of current set Y which will be changed next time. The game remembers i, plus one number $j \in Y$ with $j > i$ (if any) (it can be done with polynomial number of states). The game checks whether the next set Y' contains j, plus a number $i' \in Y'$ with $i < i'$ and $i' \neq j$. Again, since player 1 does not know the number j chosen, if player 1 cheats and changes a number $k > i$ of Y, then there is always a chance that the game remembers that number and catches player 1 cheating, in which case the game goes to the sink state s. To be sure that player 1 gives $\frac{1}{2} \cdot \binom{n}{n/2}$ sets, she plays the game of section A.1 step by step, advancing to the successor of the current counter only when a set Y is proposed. Furthermore, when she has finished giving $\frac{1}{2} \cdot \binom{n}{n/2}$, she goes to the third part.

The third part ressembles the second part: player 2 proposes $\frac{1}{2} \cdot \binom{n}{n/2}$ sets instead of player 1, and player 1 observes these sets. For each set Y proposed by player 2, player 1 has to give an event in $X \setminus Y$ (this is not observed by player 2). This is ensured in the same way as in part 2. Recall that Player 1 has always a reset action to restart the game from step 1, but in the sink state s. That is, if $Y = X$, player 1 can ends the round, and restart the game with a new set X in the following round.

After each set proposed by player 2, the game of section A.1 advances to its next step. Once there has been $\frac{1}{2} \cdot \binom{n}{n/2}$ sets Y proposed with the proof by Player 1 that $X \neq Y$, then Player 1 goes to the goal state t and wins.

B Details for Section 4

B.1 Strategies with finite memory

Definition 6 (Behavioral strategy associated with a finite memory strategy)**.** *A strategy with finite memory is described by a finite set M called the memory, a strategic function $\sigma_M : M \to \mathcal{D}(I)$, an update function $\mathrm{upd}_M : M \times C \to M$, an initialization function $\mathrm{init}_M : \mathcal{P}(K) \to M$. The associated behavioral strategy is defined by*

$$\sigma(\delta)(c_1 \cdots c_n)(i) = \sum_{m_0 \cdots m_n \in M^{n+1}} \mathrm{init}_M(\mathrm{supp}(\delta))(m_0) \cdot \mathrm{upd}_M(m_0, c_1)(m_1) \cdots \mathrm{upd}_M(m_{n-1}, c_n)(m_n) \cdot \sigma_M(m_n)(i)$$

B.2 Beliefs and the shifting lemma

When ”shifting time” in proofs, we will use the following *shifting lemma*, either explicitely or implicitely.

Lemma 1 (Shifting lemma)**.** *Let $f : S^\omega \to \{0,1\}$ be the indicator function of a measurable event, δ be an initial distribution and σ and τ two strategies. Then:*

$$\mathbb{P}_\delta^{\sigma,\tau}\left(f(K_1, K_2, \ldots) = 1 \mid C_1 = c, D_1 = d\right) = \mathbb{P}_{\delta_{cd}}^{\sigma_c,\tau_d}\left(f(K_0, K_1, \ldots) = 1\right),$$

where $\forall k \in K, \delta_{cd}(k) = \mathbb{P}_\delta^{\sigma,\tau}\left(K_1 = k \mid C_1 = c, D_1 = d\right)$, $\sigma_c(c_2 c_3 \cdots c_n) = \sigma(c c_2 c_3 \cdots c_n)$ and $\tau_d(d_2 d_3 \cdots d_n) = \sigma(d d_2 d_3 \cdots d_n)$.

Proof. Using basic definitions, this holds when f is the indicator function of a union of cylinders, and the class of events that satisfy this property is a monotone class. □

We will use heavily the following properties of beliefs.

Proposition 3. *Let σ, τ be strategies for player 1 and 2 and δ an initial distribution with support L. Then for $n \in \mathbb{N}$,*

$$\mathbb{P}_\delta^{\sigma,\tau}\left(K_{n+1} \in \mathcal{B}_1(L, C_1, \ldots, C_n)\right) = 1 \ . \tag{3}$$

Moreover, let τ_U be the strategy for player 2 which plays every action uniformly at random. Then for every $n \in \mathbb{N}$ and $c_1 \cdots c_n \in C^*$, if $\mathbb{P}_\delta^{\sigma,\tau_U}\left(C_1 = c_1, \ldots, C_n = c_n\right) > 0$ then for every state $k \in K$,

$$(k \in \mathcal{B}_1(L, c_1, \ldots, c_n)) \iff \left(\mathbb{P}_\delta^{\sigma,\tau_U}\left(K_{n+1} = k, C_1 = c_1, \ldots, C_n = c_n\right) > 0\right) \ . \tag{4}$$

Consider the reachability, safety, Büchi or co-Büchi condition, and suppose σ and δ are almost-surely winning for player 1. Then for every $n \in \mathbb{N}$ and strategy τ,

$$\mathbb{P}_\delta^{\sigma,\tau}\left(\mathcal{B}_1(L, D_1, \ldots, D_n) \text{ is a.s.w. for player } 1\right) = 1 \ . \tag{5}$$

Proof. Easy from the definitions using the shifting lemma. Recall for reachability and safety games, we suppose without loss of generality that target states are absorbing. The first statement says that the current state is always in the belief of player 1. The second statement says that in case player 2 plays every action, then every state in the belief of player 1 is a possible current state. The third statement says in case player 1 plays with an almost-surely winning strategy, his belief should stay almost-surely winning. This is because σ should be almost-surely winning against τ_U as well. □

B.3 Proof of Theorem 3

Theorem 3 (Deciding positive winning in reachability games)**.** *In a reachability game each initial distribution δ is either positively winning for player 1 or surely winning for player 2, and this depends only on $\operatorname{supp}(\delta) \subseteq K$. The corresponding partition of $\mathcal{P}(K)$ is computable in time $\mathcal{O}\left(G \cdot 2^K\right)$, where G denotes the size of the description of the game. The algorithm computes at the same time the finite-memory strategies described in Theorem 2.*

The proof is elementary. By inspection of the proof, one can obtain bounds on time and probabilities before reaching a target state, using the uniform memoryless strategy σ_U. From an initial distribution positively winning for the reachability objective, for every strategy τ,

$$\mathbb{P}_\delta^{\sigma_U,\tau}\left(\exists n \leq 2^K, K_n \in T\right) \geq \left(\frac{1}{p \mid I \mid}\right)^{2^K} \quad , \tag{6}$$

where p is the smallest non-zero transition probability.

Proof. Let $\mathcal{L}_\infty \subseteq \mathcal{P}(K\backslash T)$ be the greatest fix-point of the monotonic operator $\Phi : \mathcal{P}(\mathcal{P}(K\backslash T)) \to \mathcal{P}(\mathcal{P}(K\backslash T))$ defined by:

$$\Phi(\mathcal{L}) = \{L \in \mathcal{L} \mid \exists j_L \in J, \forall d \in D, (j_L = j(d)) \implies \mathcal{B}_2(L,d) \in \mathcal{L}\} \enspace ,$$

in other words $\Phi(\mathcal{L})$ is the set of supports such that player 2 has an action j, such that whatever signal d she might receive (coherent with j of course) her new belief will still be in $\mathcal{L}$. Let σ_R be the strategy for player 1 that plays randomly any action.

We are going to prove that:

(A) every support in $\mathcal{L}_\infty$ is surely winning for player 2,

(B) and σ_R is positively winning from any support $L \subseteq K$ which is not in $\mathcal{L}_\infty$.

We start with proving (A). For winning surely from any support $L \in \mathcal{L}_\infty$, player 2 uses the following finite-memory strategy τ: if the current belief of player 2 is $L \in \mathcal{L}_\infty$ then player 2 chooses an action j_L such that whatever signal d player 2 receives (with $j(d) = j_L$), her next belief $\mathcal{B}_2(L,d)$ will be in $\mathcal{L}_\infty$ as well. By definition of Φ there always exists such an action j_L, and this defines a finite memory strategy with memory $\mathcal{P}(K\backslash T)$ and update operator $\mathcal{B}_2$.

When playing with strategy τ, starting from a support in $\mathcal{L}_\infty$, beliefs of player 2 never intersect T. According to 3 of Proposition 3, this guarantees the play never visits T, whatever strategy is used by player 1.

Conversely, we prove (B). Once the memoryless strategy σ_R for player 1 is fixed, the game is a one-player game where only player 2 has choices to make: it is enough to prove (B) in the special case where the set of actions of player 1 is a singleton $I = \{i\}$. Let $\mathcal{L}_0 = \mathcal{P}(K\backslash T) \supseteq \mathcal{L}_1 = \Phi(\mathcal{L}_0) \supseteq \mathcal{L}_2 = \Phi(\mathcal{L}_1) \ldots$ and $\mathcal{L}_\infty$ be the limit of this sequence, the greatest fixpoint of Φ. We prove that for any support $L \in \mathcal{P}(K)$, if $L \notin \mathcal{L}_\infty$ then:

$$L \text{ is positively winning for player 1} \enspace . \tag{7}$$

If $L \cap T \neq \emptyset$, (7) is obvious. For dealing with the case where $L \in \mathcal{P}(K\backslash T)$, we define for every $n \in \mathbb{N}$, $\mathcal{K}_n = \mathcal{P}(K\backslash T)\backslash\mathcal{L}_n$, and we prove by induction on $n \in \mathbb{N}$ that for every $L \in \mathcal{K}_n$, for every initial distribution δ_L with support L, for every strategy τ,

$$\mathbb{P}^{\tau}_{\delta_L}(\exists m, 2 \leq m \leq n+1, K_m \in T) > 0 \enspace . \tag{8}$$

For $n = 0$, (8) is obvious because $\mathcal{K}_0 = \emptyset$. Suppose that for some $n \in \mathbb{N}$, (8) holds for every $L \in \mathcal{K}_n$, and let $L \in \mathcal{K}_{n+1}$. If $L \in \mathcal{K}_n$ then by inductive hypothesis, (8) holds. Otherwise $L \in \mathcal{K}_{n+1}\backslash\mathcal{K}_n$ and by definition of $\mathcal{K}_{n+1}$,

$$L \in \mathcal{L}_n\backslash\Phi(\mathcal{L}_n) \enspace . \tag{9}$$

Let δ_L be an initial distribution with support L and τ a strategy for player 2. Let j be an action such that $\tau(\delta_L)(j) > 0$. According to (9), by definition of Φ, there exists a signal $d \in D$ such that $j = j(d)$ and $\mathcal{B}_2(L,d) \notin \mathcal{L}_n$. If $\mathcal{B}_2(L,d) \cap T \neq \emptyset$ then according to Proposition 3, $\mathbb{P}^{\tau}_{\delta_L}(K_2 \in T) > 0$. Otherwise $\mathcal{B}_2(L,d) \in \mathcal{P}(K\backslash T)\backslash\mathcal{L}_n = \mathcal{K}_n$ hence distribution $\delta_d(k) = \mathbb{P}^{\tau}_{\delta_L}(K_2 = k \mid D_1 = d)$ has its support in $\mathcal{K}_n$. By inductive hypothesis,for every strategy τ', $\mathbb{P}^{\tau'}_{\delta_d}(\exists m \in \mathbb{N}, 2 \leq m \leq n+1, K_m \in T) > 0$ hence according to the shifting lemma and the definition of δ_d, $\mathbb{P}^{\tau}_{\delta}(\exists m \in \mathbb{N}, 3 \leq m \leq n+2, K_m \in T) > 0$, which achieves the inductive step.

For computing the partition of supports between those positively winning for player 1 and those surely winning for player 2, it is enough to compute the largest fixpoint of Φ. Since Φ is monotonic, and each application of the operator can be computed in time linear in the size of the game (G) and the number of supports (2^K) the overall computation can be achieved in time $G2^K$. For computing strategy τ, it is enough to compute for each $L \in \mathcal{L}_\infty$ an action j_L which ensures $\mathcal{B}_2(L,d) \in \mathcal{L}_\infty$. □

B.4 Proof of Theorem 4

Theorem 4 (Deciding almost-sure winning in Büchi games)**.** *In a Büchi game each initial distribution δ is either almost-surely winning for player 1 or positively winning for player 2, and this depends only on $\mathrm{supp}(\delta) \subseteq K$. The corresponding partition of $\mathcal{P}(K)$ is computable in time $\mathcal{O}(2^{2^G})$, where G denotes the size of the description of the game. The algorithm computes at the same time the finite-memory strategies described in Theorem 2.*

We start with formalizing what it means for player 1 to force her pessimistic beliefs to stay in a certain set.

Definition 7. *Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of supports. We say that player 1* can enforce her beliefs to stay outside $\mathcal{L}$ *if player 1 has a strategy σ such that for every strategy τ of player 2 and every initial distribution δ whose support is not in $\mathcal{L}$,*

$$\mathbb{P}_{\delta}^{\sigma,\tau}\left(\forall n \in \mathbb{N}, \mathcal{B}_1(L, C_1, \ldots, C_n) \notin \mathcal{L}\right) = 1 \ . \tag{10}$$

Equivalently, for every $L \notin \mathcal{L}$, the set:

$$I(L) = \{i \in I \text{ such that } \forall c \in C, \text{ if } i = i(c) \text{ then } \mathcal{B}_1(L, c) \notin \mathcal{L}\} \ ,$$

is not empty.

Proof. The equivalence is straightforward. In one direction, let σ be a strategy with the property above, $L \notin \mathcal{L}$, δ_L a distribution with support L and i an action such that $\sigma(\delta_L)(i) > 0$. Then according to (10), $i \in I_L$ hence I_L is not empty. In the other direction, if I_L is not empty for every $L \notin \mathcal{L}$ then consider the finite-memory strategy σ which consists in playing any action in I_L when the belief is L. Then by definition of beliefs (10) holds. □

We need the notion of $\mathcal{L}$-games.

Definition 8 ($\mathcal{L}$-games)**.** *Let $\mathcal{L}$ be a set of supports such that player 1 can enforce her beliefs to stay outside $\mathcal{L}$. For every support $L \notin \mathcal{L}$, let $I(L)$ be the set of actions given by Definition 7. The $\mathcal{L}$-game has same actions, transitions and signals than the original partial observation game, only the winning condition changes: player 1 wins if the play reaches a target state and moreover player 1 does not use actions other than I_L whenever her pessimistic belief is L. Formally given an initial distribution δ with support L and two strategies σ and τ the winning probability of player 1 is:*

$$\mathbb{P}_{\delta}^{\sigma,\tau}\left(\exists n, K_n \in T \text{ and } \forall n, I_n \in I(\mathcal{B}_1(L, C_1, \ldots, C_n))\right).$$

Actually, winning positively an $\mathcal{L}$-game amounts to winning positively a reachability game with state space $\mathcal{P}(K) \times K$, as shown by the following lemma and its proof.

Proposition 4 ($\mathcal{L}$-games)**.** *Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of supports such that $\mathcal{L}$ is upward-closed and player 1 can enforce her pessimistic beliefs to stay outside $\mathcal{L}$.*

(i) In the $\mathcal{L}$-game, every support is either positively winning for player 1 or surely winning for player 2. We denote $\mathcal{L}''$ the set of supports that are not in $\mathcal{L}$ and are surely winning for player 2 in the $\mathcal{L}$-game.

(ii) Suppose $\mathcal{L}''$ is empty i.e. every support not in $\mathcal{L}$ is positively for player 1 in the $\mathcal{L}$-game. Then every support not in $\mathcal{L}$ is almost-surely winning for player 1, both in the $\mathcal{L}$-game and also for the Büchi objective. Moreover, the strategy $\sigma_{\mathcal{L}}$ for player 1 which consists in chosing randomly any action in $I(L)$ when her belief is L is almost-surely winning in the $\mathcal{L}$-game.

(iii) *Suppose $\mathcal{L}''$ is* not *empty. Then player 2 has a strategy τ for winning surely the $\mathcal{L}$-game from any support in $\mathcal{L}''$, and τ has finite memory $\mathcal{P}((\mathcal{P}(K)\backslash\mathcal{L}) \times K)$.*

(iv) *There is an algorithm running in time doubly-exponential time in the size of G for computing $\mathcal{L}''$ and, in case (iii) holds, strategy τ.*

The proof is based on Theorem 3.

Proof. We define a reachability game which is a synchronized product of the original game G with beliefs of player 1, with a few modifications. this new reachability game is denoted $G_{\mathcal{L}}$. The state space is $K \times (\mathcal{P}(K)\backslash\mathcal{L}) \cup \{\perp\}$, where $\{\perp\}$ is a sink state, used for punishing player 1 whenever he uses an action not in $I(L)$. Target states of $G_{\mathcal{L}}$ are those whose first component is a target state of the initial game G. Actions and signals of both players are the same as in G. The transition function is the product of the transition function of G (for the first component), together with the belief operator $\mathcal{B}_1$ (for the second component), with one modification: whenever the current state is (l, L) and player 1 plays an action i which is not in $I(L)$, the next state is $\{\perp\}$, and remains $\{\perp\}$ forever.

Applying Theorem 3 to the reachability game $G_{\mathcal{L}}$, we get (i) and (iii). Property (i) holds because a strategy for player 1 is positively winning in the $\mathcal{L}$-game if and only if it is positively winning in $G_{\mathcal{L}}$ and a strategy for player 2 is surely winning in the $\mathcal{L}$-game if and only if it is surely winning in $G_{\mathcal{L}}$. Property (iii) holds according to Theorem 3, because the state space of $G_{\mathcal{L}}$ is $K \times (\mathcal{P}(K)\backslash\mathcal{L}) \cup \{\perp\}$ and player 2 can forget about state $\perp$ because it is a sink state.

Computability of $\mathcal{L}''$ and σ and τ stated in (iv) is straightforward from Theorem 3 applied to $G_{\mathcal{L}}$.

Now we suppose $\mathcal{L}''$ is empty and prove (ii). According to Theorem 3, any support not in $\mathcal{L}$ is positively winning for player 1 in $G_{\mathcal{L}}$ and moreover the strategy σ_R which consists in playing randomly any action is positively winning for player 1. When the belief of player 1 is L, playing an action i which is not in $I(L)$ leads immediatly to a non-accepting sink state, hence strategy $\sigma_{\mathcal{L}}$ which consists in playing randomly any action in $I(L)$ is positively winning as well, from any initial distribution whose support is not in $\mathcal{L}$.

To prove (ii) it is enough to show that for every initial distribution δ whose support is not in $\mathcal{L}$,

$$\sigma_{\mathcal{L}} \text{ is almost-surely winning for player 1 from } \delta \ . \tag{11}$$

Note this is a consequence of (6), but we quickly reprove it. For proving (11), we need to give an upper bound on the time to wait before seeing a target state. We start with proving that for each $L \notin \mathcal{L}$ there exists $N_L \in \mathbb{N}$ such that for every strategy τ, for every distribution δ with support L,

$$\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n \leq N_L, K_n \in T) \geq \frac{1}{N_L} \ . \tag{12}$$

We suppose such an N_L does not exist and seek for a contradiction. Suppose for every N there exists τ_N and δ_N with support L such that (12) does not hold. Without loss of generality, since σ is fixed and property (12) only concerns the first N steps of the game, we can "de-randomize" strategy τ_N and suppose τ_N is deterministic i.e. $\tau_N : D^* \to J$. Without loss of generality, we can assume as well that δ_N converges to some distribution δ, whose support is necessarily included in L. Using Koenig's lemma, it is easy to build a strategy $\tau : D^* \to J$ such that for infinitely many N,

$$\mathbb{P}_{\delta_N}^{\sigma,\tau}(\exists n \leq N, K_n \in T) < \frac{1}{N} \ .$$

Taking the limit when $N \to \infty$, we get:

$$\mathbb{P}_{\delta}^{\sigma,\tau}(\exists n, K_n \in T) = 0 \ .$$

this contradicts the fact that σ is positively winning from L, because the support of δ is included in $L \notin \mathcal{L}$ and by hypothesis $\mathcal{L}$ is upward closed hence $\mathrm{supp}(\delta) \notin \mathcal{L}$ as well. This proves the existence of N_L such that (12) holds.

Now we can achieve the proof of (ii). Let $N = \max\{N_L \mid L \notin \mathcal{L}\}$. Then for every strategy τ and every distribution δ whose support is not in $\mathcal{L}$,

$$\mathbb{P}_{\delta}^{\sigma,\tau}\left(\exists n \leq N, K_n \in T\right) \geq \frac{1}{N} \ . \tag{13}$$

Since $\sigma_{\mathcal{L}}$ guarantees the belief of player 1 to stay outside $\mathcal{L}$, we can apply the shifting lemma and get:

$$\mathbb{P}_{\delta}^{\sigma_{\mathcal{L}},\tau}\left(\forall n \leq 2N, K_n \notin T\right) \leq (1 - \frac{1}{N})^2 \ ,$$

and by induction,

$$\mathbb{P}_{\delta}^{\sigma_{\mathcal{L}},\tau}\left(\exists n \in \mathbb{N}, K_n \in T\right) = 1 \ .$$

This holds for every strategy τ and every distribution δ whose support is not in $\mathcal{L}$. Since $\sigma_{\mathcal{L}}$ guarantees the belief of player 1 to stay outside $\mathcal{L}$, by induction using (4) we obtain

$$\mathbb{P}_{\delta}^{\sigma_{\mathcal{L}},\tau}\left(\exists^{\infty} n, K_n \in T\right) = 1 \ .$$

This achieves to prove that $\sigma_{\mathcal{L}}$ is almost-surely winning from any support $L \notin \mathcal{L}$ for the Büchi condition. This proves (11) hence (ii). □

The following proposition provides a fix-point characterization of almost-surely winning supports for player 1.

Proposition 5 (Fix-point characterization of almost-surely winning supports)**.** *Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of supports. Suppose player 1 can enforce her beliefs to stay outside $\mathcal{L}$. Then,*

(i) either every support $L \notin \mathcal{L}$ is almost-surely winning for player 1 and her Büchi objective,

(ii) or there exists a set of supports $\mathcal{L}' \subseteq \mathcal{P}(K)$ and a strategy τ^* for player 2 such that:

(a) $\mathcal{L}'$ is not empty and does not intersect $\mathcal{L}$,

(b) player 1 can enforce her beliefs to stay outside $\mathcal{L} \cup \mathcal{L}'$,

(c) for every strategy σ and initial distribution δ with support in $\mathcal{L}'$,

$$\mathbb{P}_{\delta}^{\sigma,\tau^*}\left(\forall n \geq 2^K, K_n \notin T \mid \forall n, \mathcal{B}_1(L, C_1, \ldots, C_n) \notin \mathcal{L}\right) > 0 \ . \tag{14}$$

There exists an algorithm running in time doubly-exponential in the size of G for deciding which of cases (i) or (ii) holds. In case (i) holds, the strategy $\sigma_{\mathcal{L}}$ for player 1 which consists in playing randomly any action in $I(L)$ when her belief is L is almost-surely winning for the Büchi objective. In case (ii) holds, the algorithm computes at the same time $\mathcal{L}'$ and a finite memory strategy τ^* with memory $\mathcal{P}(\mathcal{L}' \times K)\backslash\{\emptyset\}$ such that (14) *holds.*

Proof. Let $\mathcal{L}''$ be the set of supports surely winning for player 2 in the $\mathcal{L}$-game. Let τ_U be the memoryless strategy for player 2 playing randomly any action. Let $\mathcal{L}'$ be the set of supports L such that $L \notin \mathcal{L}$ and,

$$\forall \sigma, \mathbb{P}_{\delta_L}^{\sigma,\tau_U}\left(\exists n \leq 2^K, \mathcal{B}_1(L, C_1, \ldots, C_n) \in \mathcal{L}'' \cup \mathcal{L}\right) > 0 \ , \tag{15}$$

where δ_L is the uniform distribution on L.

We start with proving that if $\mathcal{L}''$ is empty then case (i) of Proposition 5 holds. Since player 1 can enforce her beliefs to stay outside $\mathcal{L}$, then $\mathcal{L}'$ is empty as well. Moreover, according to (ii) of Proposition 4, every support not in $\mathcal{L}$ is almost-surely winning for player 1 for the Büchi condition, hence we are in case (i) of Proposition 5.

Suppose now that $\mathcal{L}''$ is *not* empty, Then we prove (ii)(a), (ii)(b) and (ii)(c) of Proposition 5.

First (ii)(a) is obvious because since $\mathcal{L}'' \subseteq \mathcal{L}'$, then $\mathcal{L}'$ is not empty either

Now we prove property (ii)(b) holds: player 1 can enforce his beliefs to stay outside $\mathcal{L} \cup \mathcal{L}'$. There exists σ such that (15) does not hold, and we can even suppose σ deterministic, i.e. $\sigma : \mathcal{P}(K) \times C^* \to I$. This strategy σ guarantees the belief of player 1 to stay outside $\mathcal{L}'' \cup \mathcal{L}$ for the first 2^K steps os the game. We can modify σ such that this holds for all steps of the game. For that, player 1 can use strategy σ' which plays like σ, and as soon as player 1 has twice the same belief L, she forgets every signal she received between the two occurences of L and keep playing with σ. Using (4) and the shifting lemma, one proves that if playing σ' there is positive probability that the belief of player 1 is in $\mathcal{L}" \cup \mathcal{L}$ someday then there is positive probability that the belief of player 1 is in $\mathcal{L}" \cup \mathcal{L}$ someday and moreover all beliefs of player 1 are different up to that moment. Since there are at most 2^K different beliefs, this contradicts the definition of σ. Hence σ' guarantees the belief of player 1 to stay outside $\mathcal{L}'' \cup \mathcal{L}$ forever. As a consequence, σ' guarantees the belief of player 1 to stay outside $\mathcal{L}'$ as well forever, again this is an application of (4) and the shifting lemma.

Description of the positively winning strategy τ^* for player 2. It remains to prove (ii)(c). According to (iii) of Proposition 4, there exists a strategy τ' for player 2 which is surely winning in the $\mathcal{L}$-game from any support in $\mathcal{L}''$.

We define a strategy τ^* for player 2 which guarantees (14) to hold. At each step, player 2 throws a coin. As long as the result is "tail", then player 2 plays randomly any action: she keeps playing with τ_U. If the result is "head" then player 2 picks randomly a support $L \in \mathcal{L}''$ (actually she guesses the belief of player 1), forgets all her signals up to now and switches definitively to strategy τ' with initial support L.

Intuitively, what matters with strategy τ^* is that the opponent player 1 does not know whether he faces strategy τ' or strategy τ_U, because everything is possible with strategy τ_U. Formalizing this very simple idea is a bit painful.

Let us prove that τ^* guarantees property (14) to hold.

We start with proving for every strategy σ of player 1 and δ an initial distribution whose support is in $L \in \mathcal{L}'$, there exists a support $L'' \in \mathcal{L}''$, $N \leq 2^K$ and $c_1 \cdots c_N \in C^*$ such that:

$$\forall l \in L'', \delta''(l) = \mathbb{P}_{\delta}^{\sigma,\tau^*}(K_n = l, C_1 = c_1, \ldots, C_N = c_N) > 0 \ . \tag{16}$$

By definition of $\mathcal{L}'$ and τ_U, there exists $c_1, \ldots, c_N$ and a support $L'' \in \mathcal{L}''$ such that $L'' = \mathcal{B}_1(L, c_1, \ldots, c_N)$, $N \leq 2^K$ and $\mathbb{P}_{\delta}^{\sigma,\tau_U}(C_1 = c_1, \ldots, C_n = c_N) > 0$. Let, Then, according to (4), $\forall l \in L'', \mathbb{P}_{\delta}^{\sigma,\tau_U}(K_n = l, C_1 = c_1, \ldots, C_N = c_N) > 0$. Since by definition of τ^*, there is positive probability that τ plays like τ_U up to stage N, then we get (16).

Now we can achieve the proof of (14). Since τ' is surely winning in the $\mathcal{L}$-game from $L'' \in \mathcal{L}''$, it guarantees that:

$$\forall \sigma, \mathbb{P}_{\delta''}^{\sigma,\tau'}(\forall n \in \mathbb{N}, K_n \notin T \mid \forall n \in \mathbb{N}, I_n \in I(\mathcal{B}_1(L'', C_1, \ldots, C_n))) = 1 \ .$$

There is positive probability that at stage n, τ^* switches to strategy τ' in initial state L''. By definition of beliefs, $\mathcal{B}_1(L'', C_1, \ldots, C_n) = \mathcal{B}_1(L, c_1, \ldots, c_N, C_1, \ldots, C_n)$, hence according to (16) and the shifting lemma,

$$\forall \sigma, \mathbb{P}_{\delta}^{\sigma,\tau^*}(\forall n \geq N, K_n \notin T, C_1 \cdots C_N = c_1 \cdots c_N \mid \forall n \geq N, I_n \in I(\mathcal{B}_1(L, C_1, \ldots, C_n))) > 0 \ . \tag{17}$$

According to the definition of $I(L)$, for every σ and $n \in \mathbb{N}$, $\mathbb{P}_{\delta}^{\sigma,\tau_U}\left(\mathcal{B}_1(L,C_1,\ldots,C_n,C_{n+1}) \in \mathcal{L} \mid I_n \notin I(\mathcal{B}_1(L,C_1,\ldots,C_n))\right) > 0$ and since there is positive probability that τ plays like τ_U up to stage n, the same holds for τ, hence:

$$\mathbb{P}_{\delta}^{\sigma,\tau^*}\left(\forall n \in \mathbb{N}, I_n \in I(\mathcal{B}_1(L,C_1,\ldots,C_n)) \mid \forall n \in \mathbb{N}, \mathcal{B}_1(L,C_1,\ldots,C_n) \notin \mathcal{L}\right) > 0 \ .$$

This last equation together with (17) proves (14), which achieves to prove (ii)(c) of Proposition 5.

Description of the algorithm. To achieve the proof of Proposition 5, we have to describe the doubly-exponential algorithm. This algorithm is a fix-point algorithm, actually there are two embedded fix-points, since this algorithm uses twice as sub-procedures the algorithm provided by Theorem 3 on game $G_{\mathcal{L}}$ defined in the proof of Proposition 4.

The algorithm of Proposition 4, property (iv) is used for computing $\mathcal{L}''$, and σ or τ'.

In case $\mathcal{L}''$ is empty, the algorithm simply outputs strategy $\sigma_{\mathcal{L}}$ described in (ii) of Proposition 5. In case $\mathcal{L}''$ is not empty, the algorithm computes the set of supports $\mathcal{L}'$ defined by (15), from which player 2 can force the belief of player 1 to be in $\mathcal{L}'' \cup \mathcal{L}$ someday with positive probability. For computing $\mathcal{L}'$, we have to fix strategy τ_U in the game $G_{\mathcal{L}}$ and check whether player 1 has a strategy for avoiding surely his beliefs to be in $\mathcal{L}' \cup \mathcal{L}$, which can be done running the algorithm of Proposition 5 to the game $G_{\mathcal{L}}$. Remark we prove the bound 2^K can be replaced by $\mathbb{N}$ in (15).

Once $\mathcal{L}'$ has been computed, the algorithm outputs strategy τ^* described above. □

The proof of Theorem 4 illustrates how to compose the various finite memory strategies of Proposition 5 to obtain a strategy for player 2 which is positively winning and has finite memory $\mathcal{P}(\mathcal{P}(K) \times K)$.

Proof of Theorem 4. According to Proposition 5, starting with $\mathcal{L}_0 = \emptyset$, there exists a sequence $\mathcal{L}'_0, \mathcal{L}'_1, \ldots, \mathcal{L}'_n$ of disjoint non-empty sets of supports such that for every $m \leq n$,

- if $0 \leq m < M$ then $\mathcal{L}_m = \mathcal{L}'_0 \cup \cdots \cup \mathcal{L}'_{m-1}$, matches case (ii) of Proposition 5. We denote τ_m the corresponding finite memory strategy.
- $\mathcal{L}_M$ matches case (i) of Proposition 5.

Then according to Proposition 5, the set of supports positively winning for player 2 is exactly $\mathcal{L}_M$, and supports that are not in $\mathcal{L}_M$ are almost-surely winning for player 1. This proves qualitative determinacy.

The sequence $\mathcal{L}'_0, \mathcal{L}'_1, \ldots, \mathcal{L}'_n$ is computable in doubly-exponential time, because each application of Proposition 5 involves running the doubly exponential-time algorithm, and the length of the sequence is at most doubly-exponential in the size of the game.

The only thing that remains to prove is the existence and computability of a positively winning strategy τ^+ for player 2, with finite memory $\mathcal{P}(\mathcal{P}(K) \times K)$. Strategy τ consists in playing randomly any action as long as a coin gives result ”head”. When the coin gives result ”tail”, then strategy τ^+ chooses randomly an integer $0 \leq m < M$ and a support $L \in \mathcal{L}'_m$ and switches to strategy τ_m. Since each strategy τ_m has memory $\mathcal{P}(\mathcal{L}'_m \times K) \backslash \{\emptyset\}$ and the $\mathcal{L}'_m$ are distincts, strategy τ^+ has memory $\mathcal{P}(\mathcal{P}(K) \times K)$ with $\emptyset$ used as the initial memory state.

We prove that τ^+ is positively winning for player 2 from $\mathcal{L}_M$. Let σ be a strategy for player 1, $L \in \mathcal{L}_M$ and δ an initial distribution with support L. Let m_0 be the smallest index m such that

$$\mathbb{P}_{\delta}^{\sigma,\tau^+}\left(\exists n \in \mathbb{N}, \mathcal{B}_1(L,C_1,\ldots,C_n) \in \mathcal{L}'_m\right) > 0 \ .$$

Since $L \in \mathcal{L}_M$ and $\mathcal{L}_M = \bigcup_{m<M} \mathcal{L}'_m$, the set in the definition of m_0 is non-empty and m_0 is well defined. Let $n_0 \in \mathbb{N}$ and $c_1, c_2, \ldots, c_{n_0} \in C^{n_0}$ such that $\mathcal{B}_1(L, c_1, \ldots, c_{n_0}) \in \mathcal{L}'_{m_0}$ and

$$\mathbb{P}_\delta^{\sigma,\tau^+}(C_1 = c_1, \ldots, C_{n_0} = c_{n_0}) > 0 \ .$$

According to the definition of τ^+, there is positive probability that τ^+ plays randomly until step n_0 hence according to (4), for every state $l \in \mathcal{B}_1(L, c_1, \ldots, c_{n_0})$,

$$\mathbb{P}_\delta^{\sigma,\tau^+}(C_1 = c_1, \ldots, C_{n_0} = c_{n_0} \text{ and } K_n = l) > 0 \ . \tag{18}$$

According to the definition of τ^+ again, there is positive probability that τ^+ switches to strategy τ_{m_0} at instant n_0. Since $\mathcal{B}_1(L, c_1, \ldots, c_{n_0}) \in \mathcal{L}'_{n_0}$ hence according to (18) and to (14) of Proposition 5,

$$\mathbb{P}_\delta^{\sigma,\tau^+}\left(\forall n \geq 2^K, K_n \notin T \mid \forall n \geq n_0, \mathcal{B}_1(L, C_1, \ldots, C_n) \notin \mathcal{L}_{m_0}\right) > 0 \ . \tag{19}$$

By definition of m_0 and since $\mathcal{L}_{m_0} = \mathcal{L}'_0 \cup \cdots \cup \mathcal{L}'_{m_0-1}$,

$$\mathbb{P}_\delta^{\sigma,\tau^+}(\forall n \in \mathbb{N}, \mathcal{B}_1(L, C_1, \ldots, C_n) \notin \mathcal{L}_{m_0}) = 1 \ ,$$

then together with (19),

$$\mathbb{P}_\delta^{\sigma,\tau^+}\left(\forall n \geq 2^K, K_n \notin T\right) > 0 \ ,$$

which proves that τ^+ is positively winning for the co-Büchi condition. □

C Details for Section 5

Proof of 2EXPTIME-hardness

We give here a more detailed proof for the 2EXPTIME-hardness of the problem of deciding whether player 1 has an almost-surely winning strategy in a reachability game.

Theorem 5. *In a reachability game, deciding whether player* 1 *has an almost-surely winning strategy is 2*EXPTIME*-hard, even if player* 1 *is more informed than player* 2*.*

Proof. We reduce the membership problem for alternating EXPSPACE Turing machines. Let $\mathcal{M}$ be an EXPSPACE alternating Turing machine, amd w be an input word of length n. From $\mathcal{M}$ we build a stochastic game with partial observation such that player 1 can achieve almost-surely a reachability objective if and only if w is accepted by $\mathcal{M}$. The idea of the game is that player 2 describes an execution of $\mathcal{M}$ on w, that is, she enumerates the tape contents of successive configurations. Moreover she chooses the rule to apply when the state of $\mathcal{M}$ is universal, whereas player 1 is responsible for choosing the rule in existential states. When the Turing machine reaches its final state, the play is won by player 1. In this simple deterministic game, if player 2 really implements some execution of $\mathcal{M}$ on w, player 1 has a surely winning strategy if and only if w is accepted by $\mathcal{M}$. Indeed, if all executions on w reach the final state of $\mathcal{M}$, then whatever the choices player 2 makes in universal states, player 1 can properly choose rules to apply in existential states in order to reach a final configuration of the Turing machine. On the other hand, if some execution on w does not lead to the final state of $\mathcal{M}$, player 1 is not sure to reach a final configuration and win the game.

This reasoning holds under the assumption that player 2 effectively describes the execution of $\mathcal{M}$ on w consistent with the rules chosen by both players. However, player 2 could cheat when enumerating successive configurations of the execution. She would for instance do so, if w is indeed accepted by $\mathcal{M}$, in order to have a chance not to lose the game. To prevent player 2 from cheating (or at least to prevent her from cheating too often), it would be convenient for the game to remember the tape contents, and check that in the next configuration, player 2 indeed applied the chosen rule. However, the game can remember only a logarithmic number of bits, while the configurations have a number of bits exponential in n. Instead, we ask player 1 to pick any position k of the tape, and to announce it to the game (player 2 does not know k), which is described by a linear number of bits. The game keeps the the letter at this position together with the previous and next letter on the tape. This allows the game to compute the letter a at position k of the *next* configuration. As player 2 describes the next configuration, player 1 will annouce to the game that position k has been reached again. The game will thus check that the letter player 2 gives is indeed a. This way, the game has a positive probability to detect that player 2 is cheating. If so, the game goes to a sink state which is winning for player 1. To increase the probability for player 1 of observing player 2 cheating, player 1 has the possibility to restart the whole execution from the beginning whenever she wants. In particular, she will do so when an execution lasts longer than 2^{2^n} steps. This way, if player 2 cheats infinitely often, player 1 will detect it with probability one, and will win the game almost-surely. So far, we described a deterministic game satisfying that if w is accepted by $\mathcal{M}$, player 1 has a mixed strategy to reach her winning state almost surely, and without cheating (that is, denonciating player 2 only if she was cheating).

We now have to take into account that player 1 could cheat: she could point a certain position of the tape contents at a given step, and point somewhere else in the next step. To avoid this kind of behaviour, or at least refrain it, a piece of information about the position pointed by player 1 is kept secret (to both players) in the state of the game. More precisely, a bit of the binary encoding of the letter position on the tape, and the position of this bit itself is randomly chosen among the at most n possible positions. If player 1 is caught cheating (that is, if the bits at the position remembered differ between both step), the game goes to a sink state losing for player 1. This way, when player 1 decides to cheat, there is a positive probability that she loses the game. At this stage, the game is stochastic (a bit and a position are remembered randomly in states of the game), player 1 does not have full information (she does not know which bit is remembered in the state), but she has more information than player 2 (the latter does not know what letter player 1 decided to memorize). Moreover, the game satisfies the following: w is accepted by $\mathcal{M}$ if and only if player 1 has mixed winning strategy which ensures reaching a goal state almost surely.

□