Qualitative Determinacy and Decidability of Stochastic Games with Signals

NATHALIE BERTRAND, INRIA, Centre Bretagne Atlantique, IRISA
BLAISE GENEST, CNRS, IRISA
HUGO GIMBERT, CNRS, Université de Bordeaux, LaBRI

We consider two-person zero-sum stochastic games with signals, a standard model of stochastic games with imperfect information. The only source of information for the players consists of the signals they receive; they cannot directly observe the state of the game, nor the actions played by their opponent, nor their own actions.

We are interested in the existence of almost-surely winning or positively winning strategies, under reachability, safety, Büchi, or co-Büchi winning objectives, and the computation of these strategies when the game has finitely many states and actions. 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 achieve surely the dual 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 dual co-Büchi objective. We prove that players only need strategies with *finite memory*. The number of memory states needed to win with finite-memory strategies ranges from one (corresponding to memoryless strategies) to doubly exponential, with matching upper and 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 an associated finite-memory strategy. Complexity ranges from EXPTIME to 2EXPTIME, with matching lower bounds. Our fix-point algorithms also enjoy a better complexity in the cases where one of the players is better informed than their opponent.

Our results hold even when players do not necessarily observe their own actions. The adequate class of strategies, in this case, is mixed or general strategies (they are equivalent). Behavioral strategies are too restrictive to guarantee determinacy: it may happen that one of the players has a winning general strategy but none of them has a winning behavioral strategy. On the other hand, if a player can observe their actions, then general, mixed, and behavioral strategies are equivalent. Finite-memory strategies are sufficient for determinacy to hold, provided that randomized memory updates are allowed.

CCS Concepts: • **Security and privacy** → **Logic and verification**; • **Mathematics of computing** → *Markov processes*; • **Computing methodologies** → *Multi-agent systems*; • **Applied computing** → Economics;

Additional Key Words and Phrases: Controller synthesis, stochastic games, imperfect information, algorithmic game theory

33

The work is supported by the ANR projet STOCH-MC (ANR-13-BS02-0011-01) and the ESF Research Networking Programme project "GAMES2."
Authors' addresses: N. Bertrand, B. Genest, and H. Gimbert; emails: {nathalie.bertrand, blaise.genest}@inriai.fr, hugo.gimbert@cnrs.fr.

https://doi.org/10.1145/3107926

ACM Reference format:
Nathalie Bertrand, Blaise Genest, and Hugo Gimbert. 2017. Qualitative Determinacy and Decidability of Stochastic Games with Signals. *J. ACM* 64, 5, Article 33 (September 2017), 48 pages.
https://doi.org/10.1145/3107926

1 INTRODUCTION

Numerous advances in algorithmics of stochastic games have recently been made [7, 12, 15, 16, 19, 25], motivated in part by applications in controller synthesis and verification of open systems. Open systems can be viewed as two-player 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 [15, 22], 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 [15, 16] and stochastic games with perfect information [13, 25]). The full observation hypothesis can hinder interesting applications in controller synthesis; actually, in most controllable open systems, full monitoring for the controller is not implementable in practice. For example, the controller of an autonomous driverless subway system cannot directly observe a hardware failure and is only informed about failures detected by the monitoring system, including false alarms due to sensors failures. Moreover, giving full information to the environment is not realistic either. Consider the following example inspired from collision regulation in ethernet protocols: the controller has to share the ethernet layer with the environment, and both of them are trying to send a data packet. For that the controller selects a date in microseconds between 1 and 512 and then the environment does the same, and then both of them try to send their data packet at the date they chose. Choosing the same date results in a data collision, and the process is repeated until there is no collision, and at that time the data can be sent. If the environment has full observation, one knows which date has been chosen by the controller and is able to create collisions on purpose ad infinitum, which prevents the controller from sending his or her data. However, if the date chosen by the controller is kept secret, then the environment cannot prevent the data from being sent eventually almost surely.

In the present article, we consider *stochastic games with signals*, which are a standard tool in game theory to model imperfect information in stochastic games [31, 33, 35]. When playing a stochastic game with signals, players cannot observe the actual state of the game, nor the actions played by themselves or their opponent: the only source of information of a player consists of private signals they receive throughout the play. Stochastic games with signals subsume standard stochastic games [34], repeated games with incomplete information [2], games with imperfect monitoring [33], concurrent games [15], and deterministic games with imperfect information on one side [11, 30].

Intuitively, players make their decisions based on the sequence of signals they receive, which is formalized with strategies. As explained in [14], some care has to be given to the way strategies are formalized as mathematical objects. Players may play using *behavioral strategies*, which are mappings from sequences of signals to probability distributions over actions [1]. This includes in particular the case of *pure strategies* where the actions are chosen deterministically; that is, the distributions are Dirac over a single signal.

A more general class of strategies are *mixed strategies,* which are probability measures over the set of pure strategies. When each player observes their own actions, Kuhn's theorem states that behavioral strategies have the same strategic power as mixed strategies [1]. However, Kuhn's theorem does not apply when actions are nonobservable [1, 14]. Intuitively, in stochastic games with signals, it may be necessary for the players to base their strategies on the outputs of random generators, kept secret from their adversary, which is not always possible to do with behavioral strategies. For this reason, in the present article, players are playing with *general strategies*, which are probability measures over the set of randomized behavioral strategies.

We show that general and mixed strategies are equivalent and, essentially, games with general strategies and nonobservable actions are strategically and algorithmically equivalent to games with behavioral strategies and observable actions. Precisely, in a game with nonobservable actions, a player has a winning general strategy if and only if the player has a winning behavioral strategy when he or she is allowed to observe his or her own actions (Theorem 4.9). This holds not only for Büchi games but also for every game with a Borel winning condition. Moreover, if there exist winning finite-memory strategies in the game with observable actions, they can be transformed to winning finite-memory strategies in the game with nonobservable actions with very limited impact on the size of the memory.

From the algorithmic point of view, focusing on games with ω-regular winning conditions, stochastic games with signals are considerably harder to deal with than stochastic games with full observation. While *values* of the latter games are computable [7, 15], simple questions like "is there a strategy for player 1 that guarantees winning with probability more than $\frac{1}{2}$?" are *undecidable* even for the restricted class of stochastic reachability games with a single signal and a single player [28]. Also, for this restricted class corresponding to Rabin's probabilistic automata [29], the value 1 problem is undecidable [20]. In the present article, rather than *quantitative* properties (i.e., questions about values), we focus on *qualitative* properties of stochastic games with signals.

We study the following qualitative questions about stochastic games with signals, equipped with reachability, safety, or Büchi objectives:

(i) Does player 1 have an *almost-surely winning strategy*, that is, a strategy that guarantees the objective to be achieved with probability 1, whatever the strategy of player 2?
(ii) Does player 2 have a *positively winning strategy*, that is, a strategy that 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. 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*. Depending on the class of objective, the number of memory states needed ranges from one (memoryless) to doubly exponential.
(3) Questions (i) and (ii) are decidable. We provide fix-point algorithms for computing 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.
(4) The general case of games with nonobservable actions and general strategies is reducible to the case of games with observable actions and behavioral strategies.

The first three results are detailed in Theorems 6.1, 6.6, 8.2, and 8.3. We prove that these results are tight and robust in several aspects. Games with co-Büchi objectives are absent from this picture, since they are neither qualitatively determined (see Section 7.2) nor decidable (as shown in [3, 9]).

Another surprising fact is that for winning positively a game with 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 contrasts with previous results about stochastic games with imperfect information [11, 30], where both the number of memory states and the complexity are simply exponential. Our contributions also reveal a nice property of *reachability* games: every initial state is either *almost-surely winning* for player 1, *surely winning* for player 2, or *positively winning* for both players.

Our results strengthen and generalize in several ways results that were previously known for concurrent games [15, 16] and deterministic games with imperfect information on one side [11, 30].

First, the framework of stochastic games with signals strictly encompasses all the settings of [11, 15, 16, 30]. In concurrent games, there is no signaling structure at all, and in deterministic games with imperfect information on one side [11], transitions are deterministic and player 2 observes everything that happens in the game, including the actions played by his or her opponent.

We believe that the extension of results of [11] to games with imperfect information on both sides is necessary to perform controller synthesis on real-life systems. The collision protocol example described previously suggests that simple robust protocols may not be robust against attacks of an omniscient environment, unless they are allowed to hide information from the environment.

Second, we prove that Büchi games are qualitatively determined: when player 1 cannot win almost surely a Büchi game, then his or her opponent can win positively. This was not known previously, even for games with imperfect information on one side: in [11, 30], algorithms are given for deciding whether the imperfectly informed player has an almost-surely winning strategy for a Büchi (or reachability) objective; however, no results (e.g., strategy for the opponent) are given in case this player has no such strategy. Our qualitative determinacy result (1) is a radical generalization of the same result for concurrent games [15, Theorem 2], using different techniques. Interestingly, for concurrent games, qualitative determinacy holds for every omega-regular objectives [15], 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 [32], but this value is not computable [28], whereas it is computable for concurrent games with omega-regular objectives [17]. The use of randomized strategies is mandatory for achieving determinacy results; this also holds for stochastic games without signals [16, 34] and even matrix games [37], which contrasts with [5, 30], where only deterministic strategies are considered.

Qualitative determinacy is a crucial property of stochastic games when used for controller synthesis, because it allows for incremental design and refinement of systems models and controllers. In case a model of a system (say, the door control system of the driverless subway in Paris) does not have a correct controller, qualitative determinacy implies that the environment has a strategy to beat the controller. Such an environment strategy can be used to perform simulation and get error traces, and we believe this can be of great help for the system designers. In case where the environment strategy is not implementable on the actual system, then the corresponding restrictions on the environment behavior should be added to the model of the system. Otherwise, the system itself should be modified in order to defeat this particular environment strategy. Without qualitative determinacy, the designer is left with no feedback when the algorithm answers that there is no winning strategy for the system, and this is a serious limitation to the industrial use of automatic controller synthesis.

Qualitative determinacy also has a strong theoretical interest. The study of zero-sum stochastic games is usually focused on the existence of the *value* of games: the value is the threshold payoff that is a minimal income for player 1 and a maximal loss for player 2, when playing with optimal strategies. The existence of a value is a clue that the strategy sets of the players are rich enough to let them play efficiently; for example, deterministic strategies that do not use random coin tosses are too restrictive to play a rock-paper-scissors game. The synthesis of almost-surely winning strategies is not related to the notion of value since there are games with value 1 but no almost-surely winning strategies. In our opinion, qualitative determinacy is the key notion of determinacy for almost-surely winning strategies and the key criterion to check that the players are given sets of strategies that are not too restricted.

From this perspective, our qualitative determinacy result shows that general strategies (or equivalently mixed strategies) and finite-memory strategies with *randomized* memory updates are the right class of strategies to play stochastic games with signals. Indeed, if players are restricted to use behavioral strategies or finite-memory strategies with *deterministic* memory updates, then qualitative determinacy does not hold anymore, as demonstrated by the counterexample in Section 2.6.

Our results about winning finite-memory strategies (2), stated in Theorem 6.6, are either brand new or generalize previous work. It was shown in [11] 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 extend these results to the case where player 2 has partial observation too. Moreover, we prove that for player 2, a doubly exponential number of memory states is necessary and sufficient to achieve positively the dual co-Büchi objective.

Concerning algorithmic results (3) (see details in Theorem 8.2 and 8.3), we give a fix-point-based algorithm for deciding whether a player has an almost-surely winning strategy for a Büchi objective. If this is the case, a strategy for achieving almost surely the Büchi objective (with an exponential number of memory states) can be derived easily. If it is not the case, a strategy (with a doubly exponential number of memory states) for player 2 to prevent the Büchi objective with positive probability can be derived easily. Our algorithm with 2EXPTIME complexity is optimal since the problem is indeed 2EXPTIME-hard (see Theorem 10.1). The same algorithm is also optimal, and with an EXPTIME complexity, under the hypothesis that player 2 has *more information* than player 1. This generalizes the EXPTIME-completeness result of [11], in the case where player 2 has perfect information. Last, our algorithm also runs in EXPTIME when player 1 has full information. In both subcases, player 2 needs only exponential memory (see Proposition 10.2).

A refined version of Büchi objectives has been introduced in [36]: instead of requiring infinitely many visits to accepting states, it asks that the limit average of visits to accepting states be positive. Considering this winning condition for the restricted class of probabilistic automata (which correspond to single-player stochastic games in which the player is blind) makes the positively winning set of states computable, contrary to probabilistic automata equipped with a standard Büchi condition. However, whether such a condition can be ensured almost surely is still undecidable.

An algorithm for deciding whether player 1 wins almost surely a Büchi game with imperfect information has been obtained in [23, 24], concurrently to our own work. We go one step further since we show qualitative determinacy and we compute not only the almost-surely winning strategies of player 1 but also the positively winning strategies of player 2.

Moreover, in the present article, we do not assume a priori that a player observes his or her own actions. This requires using the most general class of finite-memory strategies where the memory updates are randomized, by contrast with finite-memory strategies with deterministic updates used in [23]. In a nutshell, we prefer general finite-memory strategies because mimicking randomness with deterministic transitions can be very costly, or even not possible. First, finite-memory strategies with randomized updates are strictly more expressive than those with deterministic updates: qualitative determinacy does not hold anymore if players are restricted to finite-memory strategies with deterministic updates [14]. Second, general finite-memory strategies are more compact: a memory of nonelementary size is needed in general to win stochastic games with finite-memory strategies with deterministic updates [8], while in the present article we obtain doubly exponential memory upper bounds when using general finite-memory strategies.

The article is organized as follows. In Section 2, we introduce stochastic games with signals and we define the notion of qualitative determinacy. In Section 3, we give examples. In Section 4, we show that games with general strategies and nonobservable actions are essentially the same as games with behavioral strategies and observable actions. In Section 5, we introduce belief strategies in games with observable actions. The main results (qualitative determinacy, memory complexity, and algorithmic complexity) are stated in Section 6 and proved in the next sections, Section 7 for determinacy, Section 8 for algorithmic results and upper bound on the memory and the complexity, Section 9 for the lower bounds on the memory, and Section 10 for the lower-complexity bounds.

This article is an extended version of [4]. In particular, there are three novelties: we present an extended comparison between behavioral and general strategies, including a reduction from games with nonobservable actions and general strategies to games with observable actions and behavioral strategies; we provide a direct proof of qualitative determinacy; and our results hold in the general case where the players cannot observe their actions. Moreover, complete proofs are provided.

2 STOCHASTIC GAMES WITH SIGNALS

We consider the standard model of finite two-person zero-sum stochastic games with signals [31, 33, 35]. These are stochastic games where players cannot observe the actual state of the game, nor the actions played by themselves and their opponent; their only source of information consists of private signals they receive throughout the play. However, since the players know the transitions and in particular the signaling structure of the game, their private signals give them some clues about the information hidden from them. Stochastic games with signals subsume standard stochastic games [34], repeated games with incomplete information [2], games with imperfect monitoring [33], games with imperfect information [11, 23], and partial-observation stochastic games [10].

Notations. Given a finite or countable set K, we denote by $\Delta(K) = \{\delta : K \to [0,1] \mid \sum_k \delta(k) = 1\}$ the set of probability distributions on K. For every distribution $\delta \in \Delta(K)$, we denote by $\mathrm{supp}(\delta) = \{k \in K \mid \delta(k) > 0\}$ its support. For every state $k \in K$, we denote by $\mathbf{1}_k$ the unique distribution whose support is the singleton $\{k\}$. In general, when a set S is equipped with a σ-algebra, we denote by $\Delta(S)$ the set of probability measures on S.

States, actions, signals, and arenas. Two players called 1 and 2 have opposite goals and play for an infinite sequence of steps, choosing actions and receiving signals. Players observe the signals they receive, but they cannot observe the actual state of the game, nor the actions that are played, nor the signals received by their opponent. We assume player 1 to be female and player 2 to be male.

An *arena* is a tuple (K, I, J, C, D, p), where K is the set of states, I and J are the sets of *actions* of player 1 and player 2, C and D are the sets of *signals* of player 1 and player 2, and $p : K \times I \times J \to \Delta(K \times C \times D)$ are the *transition probabilities.* Notations are borrowed from [31].

Initially, the game is in a state $k_0 \in K$ chosen according to an initial distribution $\delta \in \Delta(K)$ 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 $p(k_{n+1}, c_{n+1}, d_{n+1} \mid k_n, i_n, j_n)$. This fixed probability is known by both players, as well as the whole description of the game.

We provide two examples of stochastic games with signals.

Example 2.1. The first example is a one-player game. It is depicted in Figure 1.

Actions of player 1 are $I = \{a, g_1, g_2\}$, and her signals are $C = \{\alpha, \beta, \perp\}$. Player 2 has a single action and a single signal that are not represented. Transition probabilities represented in Figure 1 are interpreted in the following way. When the game is in state 1 and player 1 plays a, then player

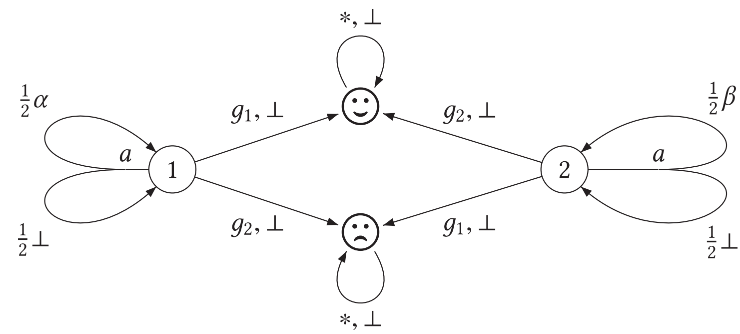

Fig. 1. A one-player stochastic game with signals.

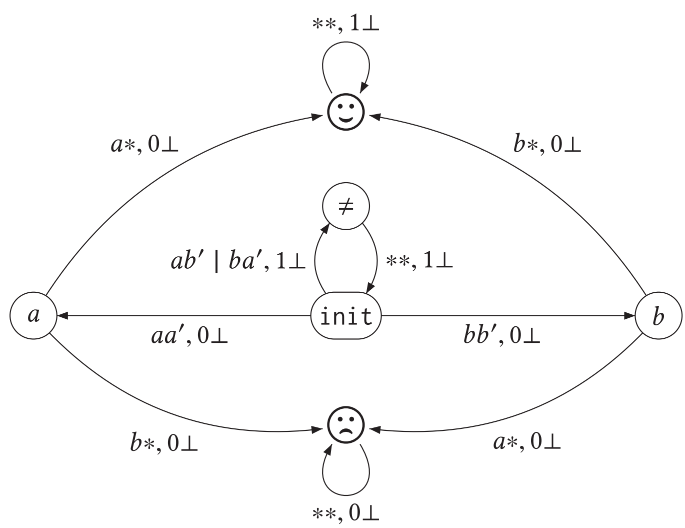

Fig. 2. A two-player stochastic game with signals.

1 receives signal α or $\perp$, each with probability $\frac{1}{2}$, and the game stays in state 1. In state 2, when the action of player 1 is a, then player 1 cannot receive signal α but instead she may receive signal β. The star symbol $*$ stands for any action: states ☺ and ☹ are absorbing.

The objective of player 1 is to reach the ☺-state. The initial distribution is $\delta(1) = \delta(2) = \frac{1}{2}$ and $\delta(☺) = \delta(☹) = 0$.

In order to reach the state ☺, player 1 has to correctly "guess the state"; that is, player 1 should play action g_1 in state 1 and action g_2 in state 2. Otherwise, the game gets stuck in the state ☹ from where there is no way to ever reach ☺.

Example 2.2. The second example is depicted in Figure 2. The initial state is `init`. Player 1 has actions $I = \{a, b\}$ and receives two signals $C = \{0, 1\}$, while player 2 has actions $J = \{a', b'\}$ and receives only one signal $D = \{\perp\}$. Again, the symbol $*$ stands for "any action." For example, from state a, whenever player 1 plays action a, then whatever action is chosen by player 2, the next state is ☺ and player 1 receives signal $\perp$.

Again, the objective of player 1 is to reach the ☺-state. For that she should do two things. First exit the set of states $\{\texttt{init}, \neq\}$. For that player 1 should match the action of player 2, at an even step, by playing a at the same time player 2 plays a' or by playing b at the same time player 2 plays b'. Then player 1 should play again the same action in order to reach ☺.

Plays. A finite play is a sequence $\pi = (k_0, i_0, j_0, c_1, d_1, k_1, \ldots, c_n, d_n, k_n) \in (KIJCD)^*K$ such that for every $0 \leq m < n$, $p(k_{m+1}, c_{m+1}, d_{m+1} \mid k_m, i_m, j_m) > 0$. An infinite play is a sequence in $(KIJCD)^\omega$ such that each prefix in $(KIJCD)^*K$ is a finite play.

Strategies. At each step of a game, both players face a choice: they have to select an action. The way players select those actions is represented by a mathematical object called a *strategy*. In the sequel, we introduce several classes of strategies, defined by the resources available to the players, and discuss how they relate.

2.1 Finite-Memory Strategies

Since our target application is controller synthesis, we seek strategies that are easily representable and implementable, which is why we are especially interested in *finite-memory strategies*. In the present article, we study several algorithmic game problems and provide solutions to determine the winner of the game and at the same time compute a finite-memory winning strategy (see Section 6.3).

There are various definitions of finite-memory strategies in the literature. A quite complete presentation is given in [14]. We provide now the most general definition of strategies with finite memory, called *general finite-memory strategies* in [14]. This is the notion of finite-memory strategy we use throughout the present article.

A strategy with finite memory set M for player 1 with set of signals C and set of actions I is a tuple $\sigma = (\mathrm{init}, \mathrm{upd}, \sigma_M)$, with

- $\mathrm{init} \in \Delta(M)$ the initial distribution of the memory,
- $\mathrm{upd} : C \times M \to \Delta(M)$ the memory update function, and
- $\sigma_M : M \to \Delta(I)$ the action choice.

Note that the memory initialization, the memory update, and the action choice are randomized. Finite-memory strategies for player 2, with set of signals D and set of actions J, are defined in a similar way.

A play according to a finite-memory strategy is as follows: the memory is initialized to a memory state $m_0 \in M$ chosen randomly, according to the probability distribution init. When the memory is in state $m \in M$, the player plays an action according to the distribution $\sigma_M(m)$, the transition occurs, and the player receives a signal c. Then the new memory state is chosen according to the distribution $\mathrm{upd}(c, m)$. Note that the memory state of the strategy of a player is not observable by his or her opponent, the only source of information of a player is the private signals he or she receives, and the player has no clue about the strategy structure of his or her opponent.

The formal definition of the probability measure on plays generated by a strategy profile (i.e., a strategy for each of the players and an initial distribution) is postponed to Section 2.3. First we discuss several notions of finite-memory strategies and compare them in terms of expressivity and succinctness.

2.2 Finite-Memory Strategies: Deterministic or Randomised Updates?

We motivate our preference of finite-memory strategies with a randomized update function

$$\mathrm{upd} : C \times M \to \Delta(M),$$

which allows the player to perform and store private coin tosses. Another option is to use deterministic update functions:

$$\mathrm{upd} : C \times M \to M,$$

as in [8, 16, 23]. We refer to such a strategy as a finite-memory strategy *with deterministic updates*.

Note that in [14], finite-memory strategies with deterministic updates are called *behavioral (finite-memory) strategies*. However, we prefer to avoid using this terminology because in the context of this article it may be misleading: the adjective *behavioral* is traditionally used to qualify strategies with arbitrary memory [1]. Moreover, in the next subsection, we give an example showing that there are behavioral strategies that can be implemented with finite-memory with randomized updates but which cannot be implemented with a finite-memory strategy with deterministic updates.

The conclusion of [14] states that both classes of strategies, with deterministic or randomized updates, have "strengths and weaknesses" and the authors "do not favour one over the others." However, in the case of stochastic games with signals and Büchi conditions, it seems to us that randomized updates are the right choice, for two reasons: *expressivity* and *succinctness*.

Finite-memory strategies with randomized updates are much more succinct. Of course, with controller synthesis in mind, the fewer memory states there are, the better: a strategy with a small description is easier to compute and implement as a controller. From this point of view, a very strong point in favor of finite-memory strategies with randomized updates is given in [8]. Namely, when a player is restricted to deterministic updates, this may cause in the worst case a dramatic blowup of the memory size required to win almost surely or positively a reachability game.

Actually, the fact that general finite-memory strategies are more expressive than the ones with deterministic updates is related to the observability of actions, which may look like a tiny detail in the first place but requires cautious attention, as demonstrated in [8, 14]. When a player chooses his or her next action with respect to a probability distribution over his or her set of actions, should we assume that the player observes the action a actually selected by this lottery?

When players are restricted to finite-memory strategies with deterministic updates, letting players observe or not their actions is a game-changer. First, [14] shows that it may change the winner of the game. Second, Corollary 4.8 and Lemma 6.7 in [8] show that *nonelementary* many memory states may be necessary for player 1 to win almost surely a reachability game using deterministic updates. This lower bound holds for games when players cannot observe their actions. However, in the present article, we demonstrate that exponential memory is sufficient when actions are observable (Proposition 6.5), and according to Theorem 4.9, the same results holds when actions are not observable.

In contrast, when randomized updates are allowed, observing actions makes no difference: we can assume that players do not observe their actions without changing the winner of the game and with very little impact on the memory size of strategies. The reason is given in Lemma 4.8: any strategy σ with finite memory M can be easily transformed into an *equivalent* strategy σ' with finite memory $M \times I$ (and randomized updates), where the action choice is *deterministic* (and actually very simple: in state (m, i) the player plays action i). Since the action choice is *deterministic*, the player knows exactly which action was played, independently of the signals they receive.

2.3 General, Mixed, and Behavioral Strategies

There are many stochastic games with signals where finite-memory strategies are not sufficient: [3] gives an example of a one-player Büchi game where the player is blind (i.e., always receives the same signal); the player can win the game with positive probability but no finite-memory strategy ensures this.

Again, as in the case of finite-memory strategies, there are several notions of strategies with arbitrary memory in the literature, and we use the most general one in the present article.

The three natural classes of strategies for player 1 in a stochastic game where she receives signals in C are defined as follows:

- A *behavioral strategy* associates with each finite sequence of signals of player 1 a probability distribution over her actions:
$$\sigma : C^* \to \Delta(I).$$
In case σ is not randomized (i.e., when the image of σ is always a Dirac distribution), the strategy is said to be *pure*.
- A *mixed strategy* is a probability measure over pure strategies:
$$\sigma \in \Delta(C^* \to I).$$
- A *general strategy* is a probability measure over behavioral strategies:
$$\sigma \in \Delta(C^* \to \Delta(I)),$$

where $C^* \to I$ denotes the set of functions from C^* to I equipped with the product topology of the copies of the discrete set I and $C^* \to \Delta(I)$ is the set of functions from C^* to $\Delta(I)$ equipped with the product topology of the copies of the metric space $\Delta(I)$. Clearly behavioral and mixed strategies are contained in the class of general strategies.

To our knowledge, the class of *general* strategies was introduced in [14]. The notions of mixed strategies and behavioral strategies are classic. Kuhn's theorem states that these classes of strategies are equivalent when players have perfect recall [2].

We use K_n, I_n, J_n, C_{n+1} and D_{n+1} to denote the random variables corresponding respectively to the nth state, action of player 1, action of player 2, signal of player 1, and signal of player 2, and we denote by P_n the finite play $P_n = K_0, I_0, J_0, C_1, D_1, K_1, \ldots, C_n, D_n, K_n$.

In the usual way, an initial distribution δ and two behavioral 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, that is, sets of infinite plays that extend a common prefix finite play. The probability measure $\mathbb{P}_\delta^{\sigma,\tau}$ is the only probability measure over $(KIJCD)^\omega$ such that for every $k \in K$, $\mathbb{P}_\delta^{\sigma,\tau}(K_0 = k) = \delta(k)$, and for every $n \in \mathbb{N}$,

$$\begin{aligned}
&\mathbb{P}_\delta^{\sigma,\tau}(K_{n+1}, C_{n+1}, D_{n+1} \mid P_n) \\
&\qquad = \sigma(P_n)(C_{n+1}) \cdot \tau(P_n)(D_{n+1}) \cdot p(K_{n+1}, C_{n+1}, D_{n+1} \mid K_n, I_n, J_n), \qquad (1)
\end{aligned}$$

where we use standard notations for conditional probability measures.

A pair of general strategies $\Sigma \in \Delta(C^* \to \Delta(I))$ for player 1 and $T \in \Delta(D^* \to \Delta(J))$ for player 2 and an initial probability distribution δ define altogether a probability measure $\mathbb{P}_\delta^{\Sigma,T}$ over the set of infinite plays, for $E \subseteq (KIJCD)^\omega$ measurable,

$$\mathbb{P}_\delta^{\Sigma,T}(E) = \int_{\sigma:C^*\to\Delta(I)} \int_{\tau:D^*\to\Delta(J)} \mathbb{P}_\delta^{\sigma,\tau}(E)\, d\Sigma(\sigma) dT(\tau).$$

This is well defined since the collection $\mathcal{E}$ of events $E \subseteq (KIJCD)^\omega$ such that the function $(\sigma, \tau) \to \mathbb{P}_\delta^{\sigma,\tau}(E)$ is measurable contains all measurable E. This is because $\mathcal{E}$ clearly contains cylinders and is stable by complement and countable union.

Actually, there is an equivalent way to define $\mathbb{P}_\delta^{\Sigma,T}$.

LEMMA 2.3. *For every general strategy Σ of player 1, define $\mathbb{E}_\Sigma : I(CI)^* \to [0,1]$ by*

$$\mathbb{E}_\Sigma(i_0, c_1, i_1, \ldots, c_n, i_n) = \int_{\sigma:C^*\to\Delta(I)} \sigma(\varepsilon)(i_0) \cdot \sigma(c_0)(i_1) \cdots \sigma(c_0 \cdots c_n)(i_n) d\Sigma(\sigma).$$

Then $\mathbb{P}_\delta^{\Sigma,T}$ is the only probability measure on the set of infinite plays such that for every finite play $\pi = k_0, i_0, j_0, c_1, d_1, k_1, \ldots k_n$,

$$\mathbb{P}_\delta^{\Sigma,T}(P_n = \pi) = \delta(k_0) \cdot \mathbb{E}_\Sigma(i_0, c_1, i_1, \ldots, c_n, i_n) \cdot \mathbb{E}_T(j_0, d_1, j_1, \ldots, d_n, j_n). \qquad (2)$$

Proof. A simple computation shows that the condition is necessary. And this defines a unique probability measure since the events $\{P_n = \pi\}$ are exactly the cylinders of $K(IJCD)^\omega$, and these cylinders generate the whole σ-algebra. □

2.4 From Finite-Memory to General Strategies

Of course, a finite-memory strategy can be seen as a general strategy: intuitively, the state space of the game is enlarged by including the memory state, which is observable only by the player playing the finite-memory strategy, and the memory state is updated upon each transition of the game.

The formal definition of the general strategy Σ_M associated with a finite-memory strategy $\sigma = (M, \text{init}, \text{upd}, \sigma_M)$ for player 1 requires some care. We use an intermediate object μ_M, which describes how memory updates are performed in σ. Let μ_M be the unique probability measure on the set of functions $f : C^* \times M \to M$ equipped with the Borel algebra generated by the product topology and such that

$$\mu_M(\{f \mid f(c_0 \cdots c_n, m) = m'\}) = \text{upd}(m, c_n)(m').$$

A fixed $m_0 \in M$ and $f : C^* \times M \to M$ naturally define a behavioral strategy $\sigma_{M,m_0,f} : C^* \to \Delta(I)$ by

$$\sigma_{M,m_0,f}(c_0 \cdots c_n) = \sigma_M(m_n(c_0 \cdots c_n)),$$

$$\text{where } m_n(c_0 \cdots c_n) = \begin{cases} m_0 & \text{if } n = 0 \\ f(c_0 \cdots c_n, m_{n-1}) & \text{otherwise.} \end{cases}$$

Then the general strategy Σ_M associated with σ_M is defined by

$$\Sigma_M(E) = \sum_{m_0 \in M} \text{init}(m_0) \cdot \mu_M(\{f \mid \sigma_{M,m_0,f} \in E\}).$$

Remark 1. *The class of* behavioral finite-memory strategies *defined in [14], which we call* finite memory with deterministic updates *in the present article, does* not *coincide with the intersection of the set of* finite-memory *strategies and the set of* behavioral *strategies. A counterexample is the behavioral strategy $\sigma : C^* \to \Delta(\{a, b\})$ defined by $\sigma(c_1, \ldots, c_n)(a) = \sum_{i=1}^{n} \frac{1}{2^i}$. This strategy is behavioral by definition and can easily be implemented by a finite-memory strategy with randomized updates and two memory states $\{A, B\}$ as follows:* $\text{init}(B) = 1$, $\sigma_M(A)(a) = 1$, *and* $\sigma_M(B)(b) = 1$, *and for every $c \in C$,* $\text{upd}(c, B)(A) = \text{upd}(c, B)(B) = \frac{1}{2}$, *and* $\text{upd}(c, A)(A) = 1$. *However, since $\sigma(c_1, \ldots, c_n)(a)$ can take infinitely many different values, no finite-memory strategy with deterministic updates can implement σ.*

2.5 Winning Conditions and Winning Strategies

The goal of player 1 is described by a measurable set of infinite plays Win called the *winning condition*. Formally, a game is a pair made of an arena and a winning condition on the arena.

Motivated by applications in logic and controller synthesis [22], 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:

$$\text{Reach} = \{\exists n \in \mathbb{N}, K_n \in T\}.$$

The safety condition is dual:

$$\text{Safe} = \{\forall n \in \mathbb{N}, K_n \notin T\}.$$

For the Büchi condition, the set of target states has to be visited infinitely often:

$$\text{Büchi} = \{\forall m \in \mathbb{N}, \exists n \geq m, K_n \in T\}.$$

And the co-Büchi condition is dual:

$$\text{CoBüchi} = \{\exists m \in \mathbb{N}, \forall n \geq m, K_n \notin T\}.$$

When player 1 and 2 use strategies σ and τ and the initial distribution is δ, then player 1 wins the game with probability

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

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

Definition 2.4 (Almost-Surely Winning Strategy). A strategy σ for player 1 is *almost-surely winning* from an initial distribution δ if

$$\forall \tau, \mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) = 1. \tag{3}$$

When such an almost-surely strategy σ exists, the initial distribution δ is said to be almost-surely winning (for player 1).

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

Definition 2.5 (Positively Winning Strategy). A strategy σ for player 1 is *positively winning* from an initial distribution δ if

$$\forall \tau, \mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) > 0. \tag{4}$$

When such a strategy σ exists, the initial distribution δ is said to be positively winning (for player 1).

Symmetrically, a strategy τ for player 2 is positively winning if it guarantees $\forall \sigma, \mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) < 1$.

The worst situation for player 1 is when her opponent has an almost-surely winning strategy τ, which thus ensures $\mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) = 0$ for all strategies σ chosen by player 1.

Note that whether a distribution δ is almost-surely or positively winning depends only on its support, because $\mathbb{P}_\delta^{\sigma,\tau}(\text{Win}) = \sum_{k \in K} \delta(k) \cdot \mathbb{P}_\delta^{\sigma,\tau}(\text{Win} \mid K_0 = k)$. As a consequence, we will say that a support $L \subseteq K$ is almost-surely or positively winning for a player if there exists a distribution with support L that has the same property.

Example 2.6. Consider the one-player game depicted on Figure 1. The objective of player 1 is to reach the ☺-state. The initial distribution is $\delta(1) = \delta(2) = \frac{1}{2}$ and $\delta(☺) = \delta(☹) = 0$.

In this game, player 1 has a strategy to reach ☺ almost surely. Her strategy is to keep playing action a as long as she keeps receiving signal $\perp$. 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 $\perp$ forever is 0. This almost-surely winning strategy can be represented by a finite-memory strategy with three memory states $M = \{m_a, m_1, m_2\}$ whose initial mapping is constant equal to the Dirac distribution on m_a and whose (deterministic) transitions are depicted in Figure 3.

2.6 Qualitative Determinacy Versus Value Determinacy

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

Definition 2.7 (Qualitative Determinacy). A winning condition Win is *qualitatively determined* if for every stochastic game with signals equipped with Win, every initial distribution is either almost-surely winning for player 1 or positively winning for player 2.

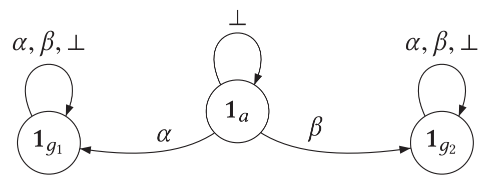

Fig. 3. A three-state almost-surely winning finite-memory strategy for the game of Figure 1. The initial distribution is the Dirac distribution on the middle state. States are labeled by the distribution to be played; all of them are Dirac distributions in this example.

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

$$\left(\forall\sigma\,\exists\tau\,\mathbb{P}_{\delta}^{\sigma,\tau}\,(\mathrm{Win}) < 1\right) \Rightarrow \left(\exists\tau\,\forall\sigma\,\mathbb{P}_{\delta}^{\sigma,\tau}\,(\mathrm{Win}) < 1\right.$$

On the other hand, the game has a value if

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

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 [26, 27, 34], to our knowledge the notion of *qualitative determinacy* appeared only recently in the context of omega-regular concurrent games [15, 16], BPA games [6], and stochastic games with perfect information [25]. Note that qualitative determinacy for two-player stochastic full-information parity finitely branching games is currently an open question [6].

The existence of an almost-surely winning strategy ensures that the value of the game is 1, but the converse is not true, even in one-player games. This is shown in Section 3 by the counterexample in Figure 6. As a consequence, player 2 may have a positively winning strategy in a game with value 1.

A difference between qualitative determinacy and value determinacy is that qualitative determinacy may hold for a winning condition but not for the complementary condition. The present article provides such an example: Büchi games are qualitatively determined but co-Büchi games are not.

Whether a game is qualitatively determined or not depends on the class of strategies used by the players. With general strategies, or equivalently with mixed strategies, Büchi games are qualitatively determined (Theorem 6.1). However, if players are restricted to play behavioral strategies or finite-memory strategies with deterministic updates, then in general qualitative determinacy does not hold anymore, as shown by the following example.

Example 2.8. Consider the example in Figure 4 taken from [14], where the aim of player 1 is to reach state ☺. This example is similar to the one in Figure 2.

Both players are blind: whatever happens, they always receive the same signal $\perp$. Starting in the initial state `init`, player 1 wishes to reach state ☺. For that she has to exit first the set of states $\{\texttt{init}, \neq\}$ by matching the action of her opponent at an even date: a for a' and b for b'. Then she should repeat again the same action in order to reach ☺.

Using a behavioral strategy $\sigma : C^* \to \Delta(I)$, player 1 cannot win almost surely. Since player 1 is blind, $C = \{\perp\}$ and the way player 1 chooses actions only depends on the time elapsed. There are two cases. First, assume that σ plays deterministically at every even time step $2n$ for every

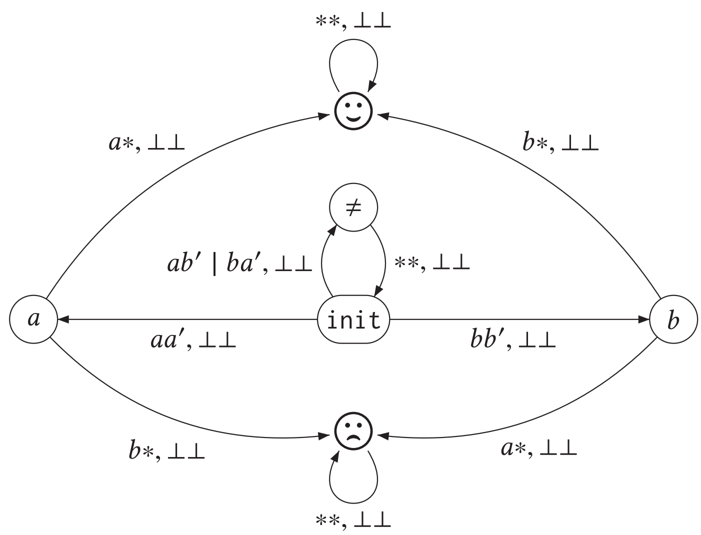

Fig. 4. Example where behavioral strategies are not sufficient.

$n \in \mathbb{N}$ (the first step has index 0). Then a pure strategy τ for player 2 beats σ. It suffices for τ to play letter b' (a', respectively) at step $2n$ when strategy σ plays letter a (b, respectively) at step $2n$. Then at each odd time point $2n + 1$, the plays is in state $\neq$, and it never reaches state ☺. Notice that the actions played at odd time steps are irrelevant. In the second case, consider the first even time point $2i$ such that both actions a and b are proposed by σ, with nonzero probability (x for a and $1 - x$ for b). Then consider what is proposed by σ at step $2i + 1$: by symmetry, assume that b is proposed with nonzero probability y (it is possible that $y = 1$, but if $y = 0$, we consider letter a instead). Again, because of behavioral strategies, player 1 does not remember her previous actions, nor the actions of player 2 up to now, and hence both a, b are proposed whether player 1 played a or played b at step $2i + 1$.

The strategy τ of player 2 beating σ is the following. It does the opposite of σ for the first i even steps (and anything for the first i odd steps—it is irrelevant). Then it plays deterministically a' at both steps $2i$ and $2i + 1$. At step $2i$, the play according to σ, τ is in state `init` with probability 1. Then with probability x, it goes to state a, and thus goes to the sink with probability xy after $2i + 1$ steps, and stays there. Hence, the probability to reach ☺ under this strategy is at most $1 - xy$. That is, behavioral strategies are not sufficient for player 1 to win this game almost surely.

On the other hand, player 1 has a finite memory strategy $\sigma = (\text{init}, \text{upd}, \sigma_M)$, which is almost-surely winning. The strategy is depicted in Figure 5. M has 4 states A, AA, B, BB, and the initial memory is given by $\text{init}(A) = \text{init}(B) = \frac{1}{2}$. The action choice is deterministic: $\sigma_M(A)(a) = \sigma_M(AA)(a) = 1$, $\sigma_M(B)(b) = \sigma_M(BB)(b) = 1$. The update function is randomized, defined by $\text{upd}(A)(AA) = 1$, $\text{upd}(B)(BB) = 1$, and $\text{upd}(AA)(A) = \text{upd}(AA)(B) = \text{upd}(BB)(A) = \text{upd}(BB)(B) = \frac{1}{2}$. It ensures that at odd times, $\{a, b\}$ are played uniformly. Moreover, player 1 knows at every even time point thanks to her memory state what she played at the previous odd time point. Thus, player 1 can play deterministically the same letter. No matter the strategy τ played by player 2, the plays following (σ, τ) reach ☺ with probability 1.

Finally, player 1 can win almost surely with a finite-memory strategy; however, no behavioral strategy is almost-surely winning for her: general strategies are more powerful than behavioral strategies.

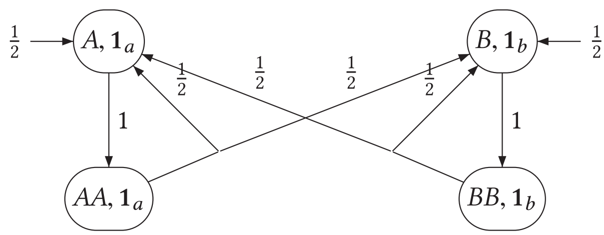

Fig. 5. A four-state almost-surely winning finite-memory strategy for player 1 in the game of Figure 4. The initial distribution is $\frac{1}{2}A + \frac{1}{2}B$. States are labeled by the distribution to be played; all of them are Dirac distributions in this example: action a in states A and AA and action b in states B and BB. There is only one signal $\perp$ for player 1, which is not represented. The memory updates from A and B are deterministic; those from AA and BB are not.

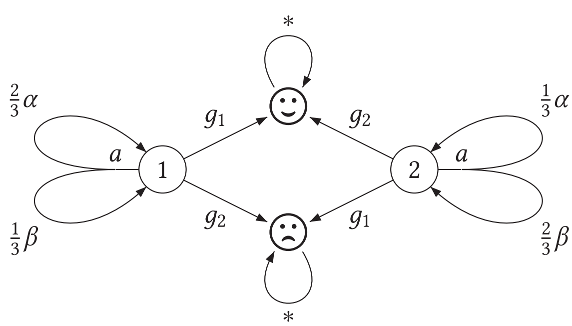

Fig. 6. A one-player reachability game with value 1 where player 1 does not win almost surely.

3 EXAMPLES

3.1 A One-Player Reachability Game with Value 1 but Player 1 Does Not Win Almost Surely

Consider the one-player game depicted in Figure 6, which is a slight modification of the one from Figure 1 (only signals of player 1 and transition probabilities differ). Player 1 has signals $\{\alpha, \beta\}$, and similarly to the game in Figure 1, her goal is to reach the target state ☺ 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 signal α was more frequent, then she plays action g_1; otherwise, she plays action g_2. Of course, the longer player 1 plays as, the more accurate the prediction will be. On the other hand, the only strategy available to player 2 is positively winning, because any sequence of signals in $\{\alpha, \beta\}^*$ can be generated with positive probability from both states 1 and 2.

3.2 A Game Where the Signaling Structure Matters

We give a second example in Figure 7 where the signaling structure matters; whether player 1 can win positively or not depends not only on her own signaling structure but also on the signaling structure of her opponent.

The game starts in state 0, and both players choose heads (h) or tails (t). If they agree, the game moves to state =, otherwise to state ≠. The behavior is similar from state 1, but the signals received by player 2 might be different. Player 1 is blind and can only count the number of steps so far. The objective for player 1 is to reach the ☺-state, and she succeeds if player 2 makes a wrong guess:

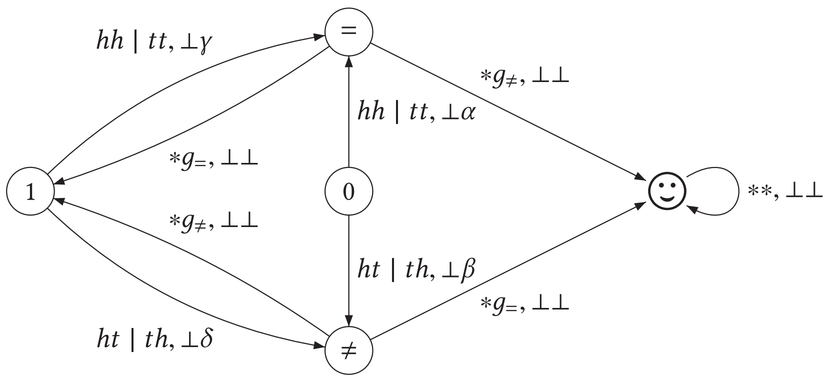

Fig. 7. Player 1 wins almost surely, positively or not, depending on the signals for player 2.

either he plays $g_{\neq}$ from state *eq* or he plays $g_{=}$ from states $\neq$. Depending on the signals α, β, γ, and δ received by player 2, the game will be almost-surely winning, positively winning, or winning with probability zero for player 1.

Assume first that all signals α, β, γ, and δ are distinct. Then, player 2 always knows when the play enters states *eq* and *neq* and can play accordingly, in order to avoid the ☺-state. Therefore, player 2 has a surely winning strategy (i.e., a strategy such that every play consistent with the strategy is winning for player 2) for her safety objective, and player 1 wins with probability 0.

Assume now that $\alpha = \beta$, but γ and δ are distinct. Informally, after the first move, player 2 cannot distinguish if the play is in state = or $\neq$. His best choice is then to play uniformly at random $g_{=}$ and $g_{\neq}$. Later, if the game reaches state 1, since $\gamma \neq \delta$, player 2 will be able to avoid the ☺-state, whatever player 1 does. For both players, in the first move, the best choice is to play uniformly at random heads or tails, so that in this case, player 1 wins with probability 1/2.

Last, assume that $\alpha = \beta$ and $\gamma = \delta$, so that player 2 can never distinguish between states = or $\neq$. The best strategy for player 1 is to always choose uniformly at random heads or tails. Against this strategy, and whatever player 2 does, every other move, the probability is half to move to the ☺-state, so that player 1 wins almost surely.

4 GAMES WITH GENERAL STRATEGIES AND NONOBSERVABLE ACTIONS ARE ALGORITHMICALLY EQUIVALENT TO GAMES WITH BEHAVIORAL STRATEGIES AND OBSERVABLE ACTIONS

In this section, we show some connections between general and behavioral strategies, and games with observable and nonobservable actions. We show that games with general strategies and nonobservable actions are essentially the same as games with behavioral strategies and observable actions. As a consequence, solving games with general strategies and nonobservable actions is of the same algorithmic complexity up to linear time reductions as solving games with behavioral strategies and observable actions.

4.1 Arenas with Observable Actions

In general, players may ignore what actions they exactly played at the previous steps, because the signals they receive may not contain this information. Otherwise, the arena they play in is said to have observable actions, in the following sense.

Definition 4.1 (Observable Actions). An arena $\mathcal{A} = (K, I, J, C, D, p)$ has observable actions if there exist two mappings $\text{Act}_1 : C \to I$ and $\text{Act}_2 : D \to J$ such that

$$p(t, c, d \mid s, i, j) > 0 \iff (i = \text{Act}_1(c) \land j = \text{Act}_2(d)).$$

The action-observable arena associated with an arena $\mathcal{A} = (K, I, J, C, D, p)$ is the arena where actions are added to signals. Formally, this is the arena $\text{Obs}(\mathcal{A}) = (K, I, J, C \times I, D \times J, p')$ such that $p'(t, (c, i'), (d, j') \mid s, i, j) = 0$ whenever $i \neq i'$ or $j \neq j'$ and $p'(t, (c, i), (d, j) \mid s, i, j) = p(t, c, d \mid s, i, j)$. A strategy σ in $\mathcal{A}$ can be naturally seen as a strategy $\text{Obs}(\sigma)$ in $\text{Obs}(\mathcal{A})$ as well, by composition with the projection from $(C \times I)^*$ to C^*. In the same way, a finite or infinite play $\pi = k_0, i_0, j_0, c_1, d_1, k_1 \ldots$ in $\mathcal{A}$ can be naturally transformed into the play $\text{Obs}(\pi) = k_0, i_0, j_0, (c_1, i_0), (d_1, j_0), k_1 \ldots$ in $\text{Obs}(\mathcal{A})$ by adding actions to signals. This defines also a transformation of a winning condition Win in $\mathcal{A}$ to the winning condition Obs(Win) in $\text{Obs}(\mathcal{A})$. These transformations preserve winning probabilities.

Lemma 4.2. *Let $\mathcal{A}$ be an arena. For every general strategy Σ and T in $\mathcal{A}$, $\mathbb{P}_\delta^{\Sigma,T}(\text{Win}) = \mathbb{P}_\delta^{\text{Obs}(\Sigma),\text{Obs}(T)}(\text{Obs}(\text{Win}))$.*

Proof. Since $\text{Obs}(\Sigma)$ and $\text{Obs}(T)$ do not take into account the actions added to signals in $\text{Obs}(G)$, a finite play π has exactly the same probability to occur in G with strategies Σ and T as the corresponding finite play $\text{Obs}(\pi)$ in $\text{Obs}(G)$ with strategies $\text{Obs}(\Sigma)$ and $\text{Obs}(T)$. Let $\mathcal{E}$ be the collection of measurable sets of infinite plays E such that $\text{Obs}(E)$ is measurable and $\mathbb{P}_\delta^{\Sigma,T}(E) = \mathbb{P}_\delta^{\text{Obs}(\Sigma),\text{Obs}(T)}(\text{Obs}(E))$. Then, according to supra, $\mathcal{E}$ contains all cylinders. Since $\mathcal{E}$ is closed under complementation and countable union, it contains also all measurable sets, including Win in particular. □

4.2 Preliminary Lemmas

The technical core of our reduction from games with nonobservable actions and general strategies to games with observable actions and behavioral strategies is a series of lemmas. Most of these results rely on the notion of equivalent strategies.

Definition 4.3 (Equivalent Strategies). In an arena $\mathcal{A}$, two general strategies Σ_1, Σ_2 for player 1 are equivalent, denoted $\Sigma_1 \equiv \Sigma_2$, if for every general strategy T of player 2 and every initial distribution δ, the probability measures $\mathbb{P}_\delta^{\Sigma_1,T}$ and $\mathbb{P}_\delta^{\Sigma_2,T}$ coincide.

A sufficient condition for two general strategies to be equivalent is given by the following corollary of Lemma 2.3.

Lemma 4.4. *Two general strategies Σ_1, Σ_2 are equivalent whenever $\mathbb{E}_{\Sigma_1} = \mathbb{E}_{\Sigma_2}$.*

First, we show that mixed strategies are as powerful as general strategies (Lemma 4.5). Then, in case actions are observable, Lemma 4.6 shows that behavioral strategies are as powerful as general strategies. Finally, Lemma 4.7 and Lemma 4.8 show that whether actions are observable or not does not matter when playing with general strategies and finite-memory strategies.

Lemma 4.5. *In every arena, every general strategy has an equivalent mixed strategy.*

Proof. The idea behind the proof of Lemma 4.5 is very natural. The main difference between a general strategy and a mixed one is that a mixed strategy performs all the randomization it needs once for all before the play begins: once a pure strategy $\sigma : C^* \to I$ is selected, the player can play deterministically. By contrast, a general strategy selects a behavioral strategy $\sigma : C^* \to \Delta(I)$ and the player has to use extra random generators during the play in order to play σ.

Intuitively, it is quite easy for a mixed strategy Σ' to mimic a general strategy Σ. Before the play begins, the mixed strategy Σ' selects a behavioral strategy $\sigma : C^* \to \Delta(I)$ using the lottery Σ. Moreover, Σ' resolves every possible future random choice of σ by picking uniformly at random a sample ω in the sample space:

$$\Omega = \{C^* \to [0,1]\}.$$

For that we equip Ω with the uniform probability measure μ obtained as the product of copies of the uniform measure λ on $[0, 1]$.

Knowing the sample $\omega \in \Omega$, all future choices can be made deterministically, thanks to a transformation that turns a behavioral strategy $\sigma \in C^* \to \Delta(I)$ and a sample $\omega \in \Omega$ into a pure strategy $\sigma_\omega \in C^* \to I$ called the ω-determinization of σ. To play σ_ω, when a sequence of signals $c_0 c_1 \cdots c_n$ has occurred, player 1 does not use the lottery $\sigma(c_0 c_1 \cdots c_n) \in \Delta(I)$ to choose her next action. Instead, she uses the value of the random sample $\omega(c_0 c_1 \cdots c_n)$ to determine this action. This should be done in a way that guarantees that the probability to choose action i is equal to $\sigma(c_0 c_1 \cdots c_n)(i)$; that is, we want the transformation $(\sigma, \omega) \to \sigma_\omega$ to guarantee

$$\sigma(c_0 \cdots c_n)(i) = \mu(\{\omega \mid \sigma_\omega(c_0 \cdots c_n) = i\}). \tag{5}$$

For that, we enumerate I as $I = \{i_0, i_1, \ldots, i_m\}$, and for every $c_0 c_1 \cdots c_n \in C^*$ we partition $[0, 1]$ into $m + 1$ intervals $[0 = x_0, x_1[, [x_1, x_2[, \ldots, [x_{m-1}, x_m], [x_m, x_{m+1} = 1]$ such that the width of $[x_k, x_{k+1}[$ is proportional to the probability that $\sigma(c_0 \cdots c_n)$ chooses i_k, that is, $x_{k+1} = x_k + \sigma(c_0 \cdots c_n)(i_k)$. This way Equation (5) holds because

$$\mu(\{\omega \mid \sigma_\omega(c_0 \cdots c_n) = i_k\}) = \lambda([x_k, x_{k+1}[) = \sigma(c_0 \cdots c_n)(i_k).$$

Then, for every general strategy $\Sigma \in \Delta(C^* \to \Delta(I))$ we define a mixed strategy $\Sigma' \in \Delta(C^* \to I)$ as

$$\Sigma'(E) = \int_{\omega \in \Omega} \Sigma(\{\sigma \mid \sigma_\omega \in E\}) d\mu(\omega).$$

To prove that Σ' is well defined, we have to establish that the function $\Psi_E : \omega \to \Sigma(\{\sigma \mid \sigma_\omega \in E\})$ is measurable whenever E is. Remark that for every $c_0 \cdots c_n \in C^*$ and $i \in I$ the function from $\phi : [0, 1] \to [0, 1]$ defined by $x \to \Sigma(\sigma \mid \sigma(c_0 \cdots c_n)(i) \geq x)$ is monotonic and thus is Lebesgue-measurable. If $E = \{\sigma \mid \sigma(c_0 \cdots c_n) = i\}$, then

$$\Psi_E(\omega) = \Sigma(\{\sigma \mid \sigma_\omega \in E\}) = \Sigma(\{\sigma \mid \sigma(c_0 \cdots c_n)(i) \geq \omega(c_0 \cdots c_n)\}) = \phi(\omega(c_0 \cdots c_n));$$

thus, Ψ_E is Lebesgue-measurable whenever E is a cylinder. Moreover, the class of E such that Ψ_E is measurable is stable by complement and countable unions. Thus, Ψ_E is well defined.

To show that Σ and Σ' are equivalent, we rely on Lemma 4.4:

$$\begin{aligned}
\mathbb{E}_\Sigma(i_0, c_1, \ldots, c_n, i_n) &= \int_{\sigma : C^* \to \Delta(I)} \sigma(\varepsilon)(i_0) \cdot \sigma(c_0)(i_1) \cdots \sigma(c_0 \cdots c_n)(i_n) d\Sigma(\sigma) \\
&= \int_{\sigma : C^* \to I} \int_{\omega \in \Omega} \mu(\{\omega \mid \sigma_\omega(\varepsilon) = i_0, \sigma_\omega(c_1) = i_0, \ldots, \sigma_\omega(c_1 \cdots c_n) = i_n\}) d\mu(\omega) d\Sigma(\sigma) \\
&= \int_{\omega \in \Omega} \Sigma(\{\sigma : C^* \to I \mid \sigma_\omega(\varepsilon) = i_0, \sigma_\omega(c_1) = i_0, \ldots, \sigma_\omega(c_1 \cdots c_n) = i_n\}) d\mu(\omega) \\
&= \Sigma'(\{\sigma : C^* \to I \mid \sigma(\varepsilon) = i_0, \sigma(c_1) = i_0, \ldots, \sigma(c_1 \cdots c_n) = i_n\}) \\
&= \mathbb{E}_{\Sigma'}(i_0, c_1, \ldots, c_n, i_n),
\end{aligned}$$

where the first and last inequalities are by definition of $\mathbb{E}_\Sigma$ and $\mathbb{E}_{\Sigma'}$, the second equality is a consequence of Equation (5), the third is Fubini's theorem, and the fourth is the definition of Σ'. □

The following result is a corollary of the generalization of Kuhn's theorem proved in [2]: whenever players have perfect recall, mixed strategies and behavioral strategies are equivalent. In order for this section to be self-contained, we provide a proof along the same lines.

Lemma 4.6. *In every arena with observable actions, every general strategy has an equivalent behavioral strategy.*

Proof. Let $\mathcal{A}$ be an arena with observable actions. Thanks to Lemma 4.5, we can assume without loss of generality that Σ is a mixed strategy, that is, $\Sigma \in \Delta(C^* \to I)$. For a sequence $c_1 \cdots c_k$ of signals, possibly empty when $k = 0$, we define the set $E(c_1 \cdots c_k)$ of pure strategies that are consistent with the actions associated to signals $c_1 \cdots c_k$:

$$E(c_1 \cdots c_k) = \{\sigma : C^* \to I \mid \sigma(\varepsilon) = \mathrm{Obs}(c_1) \land \forall 1 \leq k \leq n-1, \sigma(c_1 \cdots c_k) = \mathrm{Act}_1(c_{k+1})\},$$

and for $i \in I$, $E(c_1 \cdots c_k, i) = \{\sigma \in E(c_1 \cdots c_k) \mid \sigma(c_1 \cdots c_k) = i\}$. Then let σ_b be the behavioral strategy in $\mathcal{A}$ defined for $c_1 \cdots c_n \in C^*$ and $i \in I$ by $\sigma_b(c_1 \cdots c_n)(i) = \Sigma(E(c_1 \cdots c_n, i) \mid E(c_1 \cdots c_n))$. By definition of $\mathbb{E}_\Sigma$, this guarantees $\mathbb{E}_\Sigma = \mathbb{E}_{\sigma_b}$. According to Lemma 4.4, the strategies σ_b and Σ are equivalent. □

Note that Lemma 4.6 does not hold if actions are not observable; a counterexample inspired from [14] is given in Figure 4 in the examples section (Section 3). The observability of actions is crucial in the proof of Lemma 4.6. For example, assume that Σ is a general strategy that selects with equal probability $\frac{1}{2}$ the two pure strategies that play always i_0 or always i_1. Then the behavioral strategy σ_b constructed by the proof selects randomly the first action and then repeats it forever, which is equivalent to Σ. Playing σ_b is possible only if actions are observable. In case actions are *not* observable, it is natural to consider the behavioral strategy σ'_b:

$$\sigma'_b(c_1 \cdots c_n)(i) = \int_{\sigma: C^* \to \Delta(I)} \sigma(c_1 \cdots c_n)(i) d\Sigma(\sigma).$$

However, there is no guarantee that σ'_b and Σ are equivalent. Using the same example, σ'_b is the strategy that always plays the lottery $\frac{1}{2}i_0 + \frac{1}{2}i_1$. Clearly, σ'_b and Σ are *not* equivalent: σ'_b plays almost-surely infinitely many times both actions i_0 and i_1, while this never happens when playing Σ.

Lemma 4.7. *For every general strategy Σ in $\mathrm{Obs}(\mathcal{A})$ there exists a general strategy Σ' in $\mathcal{A}$ such that Σ is equivalent to $\mathrm{Obs}(\Sigma')$.*

Proof. Thanks to Lemma 4.5, we can assume without loss of generality that Σ is a mixed strategy.

We start with the even simpler case where Σ is the Dirac distribution on a single pure strategy $\sigma : (C \times I)^* \to I$ in $\mathrm{Obs}(\mathcal{A})$. This case is easy since a player playing the pure strategy σ can use the definition of σ to compute his or her past actions, and thus the player can forget the actions included in the signals. Formally, we define a pure strategy σ_f in $\mathcal{A}$ by $\sigma_f(\varepsilon) = \sigma(\varepsilon)$ and the inductive formula $\sigma_f(c_1 \cdots c_n) = \sigma((c_1, \sigma_f(\varepsilon)) \cdot (c_2, \sigma_f(c_1)) \cdots (c_n, \sigma_f(c_1 \cdots c_{n-1}))$. Then clearly $\mathrm{Obs}(\sigma_f) = \sigma$. Then a sequence of signals $c_1 \cdots c_n$ is consistent with σ (in sense that $\forall k, \sigma(c_1 \cdots c_{k-1}) = \mathrm{Act}_1(c_k))$ if and only if $c_1 \cdots c_n$ is consistent with $\mathrm{Obs}(\sigma_f)$. As a consequence, $\mathbb{E}_\sigma = \mathbb{E}_{\mathrm{Obs}(\sigma_f)}$, and thus σ and $\mathrm{Obs}(\sigma_f)$ are equivalent according to Lemma 4.4.

Assume now that Σ is a mixed strategy in $\mathrm{Obs}(\mathcal{A})$ and let Σ' be the mixed strategy in $\mathcal{A}$ defined for $E \subseteq C^* \to I$ by $\Sigma'(E) = \Sigma(\{\sigma \mid \sigma_f \in E\})$. According to Lemma 4.4, it is enough to prove $\mathbb{E}_\Sigma = \mathbb{E}_{\mathrm{Obs}(\Sigma')}$. This holds because for every sequence $u = (i_0, (c_1, i_0), \ldots, (c_n, i_{n-1}), i_n) \in I((C \times I) \times I)^{n-1}$, $\mathbb{E}_\Sigma(u) = \int_{\sigma:(C\times I)^* \to I} \mathbb{E}_\sigma(u) d\Sigma(\sigma) = \int_{\sigma:(C\times I)^* \to I} \mathbb{E}_{\mathrm{Obs}(\sigma_f)}(u) d\Sigma(\sigma) = \int_{\sigma_f: C^* \to I} \mathbb{E}_{\mathrm{Obs}(\sigma_f)}(u) d$

$\Sigma'(\sigma_f) = \mathbb{E}_{\text{Obs}(\Sigma')}(u)$, where the first equality holds by definition of $\mathbb{E}_\Sigma$ in case Σ is a mixed strategy, the second because we proved already $\mathbb{E}_\sigma = \mathbb{E}_{\text{Obs}(\sigma_f)}$, and the third and fourth by definition of Σ' and $\mathbb{E}_{\Sigma'}$. □

Lemma 4.8. *For every finite-memory strategy σ with memory M in $\text{Obs}(\mathcal{A})$, there exists a finite-memory strategy σ' with memory $M \times I$ in $\mathcal{A}$, such that σ and $\text{Obs}(\sigma')$ are equivalent. Moreover, the action choice of σ' is simply the projection of $M \times I$ to I.*

Proof. From $\sigma = (\text{init}, \text{upd}, \sigma_M)$ on M, we define $\sigma' = (\text{init}', \text{upd}', \sigma'_M)$ on $M \times I$, where we encode action choices in the set of memory states: in σ', each transition not only performs the corresponding transition of σ to the next memory state m' but also simultaneously selects the next action to be played, according to the distribution $\sigma(m')$. Formally, the action choice of σ' is the projection $\sigma'_M(m,i) = i$, the memory update is $\text{upd}'((m,i), c, (m', i') = \text{upd}(m, c, m') \cdot \sigma_M(m')(i')$, and the initial memory choice is $\text{init}'(m,i) = \text{init}(m) \cdot \sigma_M(m,i)$. This terminates the proof of Lemma 4.8. □

In general, the transformation of Lemma 4.8 does not preserve deterministic updates: starting from a deterministic update function $\text{upd} : C \times M \to M$, the randomized action choice $\sigma_M : M \to \Delta(I)$ is integrated into the new (randomized) update function $\text{upd}' : C \times (M \times I) \to \Delta(M \times I)$. To guarantee that the resulting strategy has deterministic updates, we need both the update and the action choice to be deterministic.

4.3 Equivalence of Games with General Strategies and Games with Observable Actions and Behavioral Strategies

We combine the results obtained so far to prove that a player has a winning general strategy in a game if and only if he or she has a winning behavioral strategy in the variant of the same game where actions are included in signals and thus become observable.

Theorem 4.9. *Let $\mathcal{A}$ be an arena and $\text{Obs}(\mathcal{A})$ the action-observable arena associated with $\mathcal{A}$. Let Win be a winning condition on $\mathcal{A}$ and consider the two games $G = (\mathcal{A}, \text{Win})$ and $\text{Obs}(G) = (\text{Obs}(\mathcal{A}), \text{Obs}(\text{Win}))$. Then the three following statements are equivalent:*

i) *Player 1 wins G almost surely.*
ii) *Player 1 wins $\text{Obs}(G)$ almost surely.*
iii) *Player 1 has a behavioral strategy in $\text{Obs}(G)$ that is almost-surely winning.*

The same equivalence holds if ones replaces "almost surely" with "positively" and/or "player 1" with "player 2."

Moreover, every almost-surely (positively, respectively) winning finite-memory strategy in $\text{Obs}(G)$ can be turned in linear time into an almost-surely (positively, respectively) winning finite-memory strategy in G.

Proof. First, (iii) and (ii) are equivalent because (iii) implies (ii) trivially and Lemma 4.6 shows that (ii) implies (iii).

Now we prove that (i) and (ii) are equivalent. Assume player 1 has an almost-surely winning strategy Σ for G, and let us prove that $\text{Obs}(\Sigma)$ is almost-surely winning in $\text{Obs}(G)$. Let T be a strategy in $\text{Obs}(G)$ and τ' be the strategy in G given by Lemma 4.7, such that $\text{Obs}(\tau')$ is equivalent to T. Then

$$\mathbb{P}_\delta^{\text{Obs}(\Sigma),T}(\text{Obs}(\text{Win})) = \mathbb{P}_\delta^{\text{Obs}(\Sigma),\text{Obs}(\tau')}(\text{Obs}(\text{Win})) = \mathbb{P}_\delta^{\Sigma,\tau'}(\text{Win}) = 1,$$

where the first equality is by choice of τ', the second equality is Lemma 4.2, and the third equality is because Σ is almost-surely winning in $\mathcal{A}$. It proves that $\text{Obs}(\Sigma)$ is almost-surely winning in $\text{Obs}(G)$.

Last, assume (ii) holds and let us prove (i). Let Σ be a strategy of player 1 winning almost surely in $\text{Obs}(G)$. According to Lemma 4.7, there exists a strategy Σ' in G such that $Obs(\Sigma') \equiv \Sigma$. Let us prove that Σ' is almost-surely winning in G. For any strategy T of player 2 in G,

$$\mathbb{P}_{\delta}^{\Sigma',T}(\text{Win}) = \mathbb{P}_{\delta}^{\text{Obs}(\Sigma'),\text{Obs}(T)}(\text{Obs}(\text{Win})) = \mathbb{P}_{\delta}^{\Sigma,\text{Obs}(T)}(\text{Win}) = 1.$$

Indeed, the first equality is Lemma 4.2, the second is by choice of Σ', and the last is because Σ is almost-surely winning in $\text{Obs}(G)$. Thus (i), (ii), and (iii) are equivalent.

The last statement about finite-memory strategies is a consequence of Lemma 4.8. □

Theorem 4.9 has an algorithmic corollary: every algorithm that decides the existence of an almost-surely winning strategy in games with behavioral strategies and observable actions can be used to decide the same problem in games with general strategies and nonobservable actions. For that it suffices to compute the action-observable version of the game, as defined later, and Theorem 4.9 ensures that this transformation has no incidence on the winner and that winning finite-memory strategies in the observable game can be lifted to winning finite-memory strategies in the original game.

This reduction leaves open an algorithmic question, which is not addressed in the present article: in case players cannot observe their actions, is it possible to decide the existence of an almost-surely or a positively winning behavioral strategy?

5 BELIEF STRATEGIES

Beliefs and beliefs of beliefs formalize part of the knowledge of players during the game. They are used to define *belief strategies*, which are finite-memory strategies of particular interest. For these notions to be properly defined, the arena should have observable actions (in the sense of Definition 4.1).

5.1 Beliefs and 2-Beliefs

The *belief* of a player is the set of possible states of the game, according to the signals received by the player.

Definition 5.1 (Belief). Let $\mathcal{A}$ be an arena with observable actions. From an initial set of states $L \subseteq K$, the belief of player 1 after having received signal c is

$$\mathcal{B}_1(L,c) = \{k \in K \mid \exists l \in L, d \in D \text{ such that } p(k,c,d \mid l, \text{Act}_1(c), \text{Act}_2(d)) > 0\}.$$

Note that in this definition we use the fact that actions of player 1 are observable; thus, when he receives a signal $c \in C$, player 1 can deduce he played action $\text{Act}_1(c) \in I$.

The belief of player 1 after having received a sequence of signals $c_1, \ldots, c_n$ is defined inductively by

$$\mathcal{B}_1(L, c_1, c_2, \ldots, c_n) = \mathcal{B}_1(\mathcal{B}_1(L, c_1, \ldots, c_{n-1}), c_n).$$

Beliefs of player 2 are defined similarly. Given an initial distribution δ, we denote by $\mathcal{B}_1^n$ the random variable defined by

$$\mathcal{B}_1^0 = \text{supp}(\delta)$$
$$\mathcal{B}_1^{n+1} = \mathcal{B}_1(\text{supp}(\delta), C_1, \ldots, C_{n+1}) = \mathcal{B}_1(\mathcal{B}_1^n, C_{n+1}).$$

We will also rely on the notion of *belief of belief*, called here *2-belief*, which, roughly speaking, represents for one player the set of possible beliefs for his (or her) adversary, as well as the possible current state.

Definition 5.2 (2-Belief). Let $\mathcal{A}$ be an arena with observable actions. From an initial set $\mathcal{L} \subseteq K \times \mathcal{P}(K)$ of pairs composed of a state and a belief for player 2, the 2-belief of player 1 after having received signal c is the subset of $K \times \mathcal{P}(K)$ defined by

$$\mathcal{B}_1^{(2)}(\mathcal{L}, c) = \{(k, \mathcal{B}_2(L, d)) \mid \exists (l, L) \in \mathcal{L}, d \in D, p(k, c, d \mid l, \mathrm{Act}_1(i), \mathrm{Act}_2(j)) > 0\}.$$

From an initial set $\mathcal{L} \subseteq K \times \mathcal{P}(K)$ of pairs composed of a state and a belief for player 2, the 2-belief of player 1 after having received a sequence of signals $c_1, \ldots, c_n$ is defined inductively by

$$\mathcal{B}_1^{(2)}(\mathcal{L}, c_1, c_2, \ldots, c_n) = \mathcal{B}_1^{(2)}\left(\mathcal{B}_1^{(2)}(\mathcal{L}, c_1, \ldots, c_{n-1}), c_n\right).$$

There are natural definitions of 3-beliefs (beliefs on beliefs on beliefs) and even k-beliefs; however, in the present article we show that 2-beliefs are enough, in the following sense: in Büchi games, the positively winning sets of player 2 can be characterized by fix-point equations on sets of 2-beliefs, and some positively winning strategies of player 2 with finite memory can be implemented using 2-beliefs.

5.2 Belief Strategies

Based on the notions of beliefs and 2-beliefs, we introduce the following families of strategies with finite memory that will be sufficient to win stochastic games with signals either positively or almost surely.

Definition 5.3 (Belief Strategies and 2-Belief Strategies). Let $\mathcal{A}$ be an arena with observable actions. A *belief strategy* of player 1 is a strategy whose memory is $\mathcal{P}(K)$ and the update function coincides with $\mathcal{B}_1$ on $\mathcal{P}(K) \setminus \{\emptyset\}$. A *2-belief strategy* of player 1 is a strategy whose memory is a subset of $\mathcal{P}(K \times \mathcal{P}(K))$ and the update coincides with $\mathcal{B}_1^{(2)}$ on $\mathcal{P}(K \times \mathcal{P}(K)) \setminus \{\emptyset\}$.

Note that in a belief strategy, by definition, the memory update is deterministic from every memory state different from $\emptyset$. However, it may be randomized from $\emptyset$. Actually, in the positively winning 2-belief strategies of player 2 for Büchi games built in this article (cf Theorem 6.6), $\emptyset$ is the initial memory state and, whatever signal is received, the update function sets positive chance to stay in $\emptyset$ as well as perform a transition to other memory states.

5.3 Particular Signaling Structures

To give a complete picture of stochastic games with signals, and to compare with existing work on games with imperfect information, we will at some places consider restricted classes of games, based on their signaling structures, as defined next.

Definition 5.4. Player 1 is *perfectly informed about the state* if her signals reveal the state, that is, if for every signal $c \in C$ of player 1 there is a state $k_c \in K$ such that $p(k', c, d \mid k, i, j) > 0 \Rightarrow k' = k_c$.

Player 1 is *better informed* than player 2 if her signals reveal the signals received by player 2, that is, if for every signal $c \in C$ of player 1 there is a signal $d_c \in D$ of player 2 such that $p(k', c, d \mid k, i, j) > 0 \Rightarrow d = d_c$.

Player 1 is *perfectly informed* if she is both perfectly informed about the state and better informed than player 2.

In the games of incomplete information used in [11], being *perfectly informed* is equivalent to being *perfectly informed about the state*, as the signal received by a player is entirely determined by the state of the game. However, in stochastic games with signals, a player may be perfectly informed about the state and yet not know the signal received by his or her opponent.

6 MAIN RESULTS

In this section, we state our main contributions, and the proofs can be found in the next sections.

6.1 Qualitative Determinacy

The following theorem constitutes the core of the article.

THEOREM 6.1. *Stochastic games with signals and reachability, safety, and Büchi winning conditions are qualitatively determined.*

The proof can be found in Section 7.

Since reachability and safety games are dual, a consequence of Theorem 6.1 is that in a reachability game, every initial distribution is either almost-surely winning for player 1, almost-surely winning for player 2, or positively winning for both players. When a safety condition is satisfied almost surely for a fixed profile of strategies, it trivially implies that the safety condition is satisfied by all consistent plays; thus, for safety games, winning *surely* is the same as winning almost surely.

By contrast, Büchi games are *not* qualitatively determined; a counterexample is given in Section 7.2. For Theorem 6.1 to hold, players should be allowed to use general strategies (or equivalently mixed strategies) and finite-memory strategies with randomized updates. Otherwise, if players are restricted to behavioral strategies or finite memory with deterministic updates, then qualitative determinacy does not hold anymore, as demonstrated by Example 2.8.

6.2 Algorithmic Complexity of Deciding the Winner

We now turn to the result concerning the (time) complexity to decide stochastic games with signals, starting with the easy case of safety games.

PROPOSITION 6.2. *In a safety game with signals, deciding whether the initial distribution is almost-surely winning for player 1 is EXPTIME-complete. If player 1 is perfectly informed about the state, the decision problem is in PTIME.*

Almost-surely winning a safety game coincides with winning surely this safety game, which in turn coincides with winning surely against a perfectly informed opponent; thus, Proposition 6.2 can be obtained by applying [11, 30], which tackle sure winning in partially observable games.

Beside the determinacy result stated in Theorem 6.1, the main contribution of this article concerns the complexity of deciding reachability and Büchi games, for which we will establish the following theorem:

THEOREM 6.3. *In reachability and Büchi games with signals, deciding whether the initial distribution is almost-surely winning for player 1 is 2EXPTIME-complete.*

Concerning winning positively a *safety or co-Büchi game*, one can use Theorem 6.1 and the determinacy property: player 2 has a positively winning strategy in the aforementioned game if and only if player 1 has no almost-surely winning strategy. Therefore, deciding when player 2 has a positively winning strategy can also be done, with the same complexity. The proof of the upper bound of Theorem 6.3 can be found in Section 8. The lower bound can be found in Theorem 10.1.

For particular signaling structures, the complexity is better than 2EXPTIME, for example, EXPTIME when player 2 is perfectly informed [11]. This reduced complexity holds for other cases as well:

	Almost-surely	**Positively**
Reachability	exponential	memoryless
Safety	exponential	doubly-exponential
Büchi	exponential	infinite
Co-Büchi	infinite	doubly-exponential

Fig. 8. Tight memory requirements for finite-memory strategies with randomized updates.

Theorem 6.4. *For reachability and Büchi games where either player 1 is perfectly informed about the state or player 2 is better informed than player 1, deciding whether the initial distribution is almost-surely winning for player 1 is EXPTIME-complete.*

The upper bound in Theorem 6.4 is shown in Proposition 10.2. The winning states can be computed by the same fix-point algorithm used for Theorem 6.3 without any change. The lower bound derives from [11].

6.3 Complexity of Strategies

The doubly exponential time complexity of Theorem 6.3 is surprising. The main explanation to the time complexity is that a player may need doubly exponential memory to win positively. More generally, the algorithmic complexity of these games is highly related to the memory needed by winning strategies, and finite-memory is sufficient to win every decidable game we consider in this article. We give the precise tight memory requirements in Figure 8.

First, as already mentioned for Proposition 6.2, almost-surely winning safety games is equivalent to surely winning safety games. Hence, results of [11, 30] can be applied, giving the exponential upper bound for the memory size needed for (almost-)surely winning safety games. More precisely, belief strategies are sufficient to win (almost-)surely safety games.

The upper bound on memory for almost-surely winning reachability and Büchi games can be derived from the proof of the determinacy of reachability and Büchi games (see Corollary 8.1). Here again, belief-based strategies are sufficient to win almost-surely reachability and Büchi games. This is not very surprising since similar strategies were used in [11], where this result was used for games where player 2 has perfect information.

Proposition 6.5 (Belief Strategies are Sufficient to Win Almost Surely). *In safety, reachability, and Büchi games with observable actions, if a player wins almost surely, then the player has an almost-surely winning* belief strategy. *There are games for which strategies with an exponential number of memory states are necessary for a player to win almost surely.*

A very similar result holds in case the actions are *not* assumed to be observable. The only difference is that the finite-memory strategy is not exactly a belief strategy. Actually, there is a transformation of a belief strategy in $\mathrm{Obs}(\mathcal{A})$ to the corresponding equivalent finite-memory strategy in $\mathcal{A}$, as described by Lemma 4.8. Inspecting the proof shows that the resulting strategy has memory $\mathcal{P}(K) \times I$ and its update operator coincides with $\mathcal{B}_1$ on the first component.

We now turn to the memory needed to win positively. First, memoryless strategies playing uniformly at random are sufficient to win positively reachability games.

A surprising fact is the amount of memory needed for winning positively co-Büchi and safety games. In these situations, it is still enough for a player to use a strategy with finite memory, but an exponential memory size is not enough to win positively. Actually, 2-belief strategies are sufficient for positively winning safety and co-Büchi games, and there is a doubly exponential lower bound on memory for winning positively a safety or co-Büchi game (see Proposition 9.1). This result cannot be derived from the memory requirements for player 1 to win almost surely, nor from the work in [23].

These bounds on the memory hold for finite-memory strategies with randomized updates. When only deterministic updates are considered and actions are not observable, memory requirements can become nonelementary (see [8] and the discussion in Section 2.2).

Theorem 6.6 (2-belief Strategies are Sufficient to Win Positively). *In reachability and Büchi games with observable actions, if player 2 wins positively, then he has a positively winning* 2-belief strategy. *There are reachability games with signals where player 1 is better informed than player 2 and where strategies with a doubly exponential number of memory states are necessary for player 2 to win positively.*

Like for Proposition 6.5, a very similar result holds in case the actions are *not* assumed to be observable, except the finite-memory strategy is not exactly a 2-belief strategy but rather the result of the linear transformation of a 2-belief strategy described in Lemma 4.8.

Last, the infinite lower bound for positively winning Büchi games is a consequence of [3] and [9] and the infinite lower bound for almost-surely winning co-Büchi games follows, since the class of languages recognized by probabilistic Büchi automata [3] is closed by complementation.

7 QUALITATIVE DETERMINACY OF STOCHASTIC GAMES WITH SIGNALS

7.1 Qualitative Determinacy of Reachability, Büchi, and Safety Games

The goal of this subsection is to prove Theorem 6.1, which states the qualitative determinacy of reachability, Büchi, and safety games. Note 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. Qualitative determinacy of safety games is rather easy to establish, so we omit the proof here. Proving qualitative determinacy of Büchi games is harder and we provide full details.

7.1.1 Properties of Beliefs. The following properties of beliefs are useful.

Lemma 7.1. *Let $\mathcal{A}$ be an arena with observable actions and τ_{rand} be the strategy of player* 2, *which always plays the uniform distribution over J. For all* behavioral *strategies σ and τ, initial distribution δ, and $n \in \mathbb{N}$, the following statements hold $\mathbb{P}_\delta^{\sigma,\tau}$ almost surely:*

$$\mathcal{B}_1^n = \left\{k \in K \mid \mathbb{P}_\delta^{\sigma,\tau_{rand}}(K_n = k \mid C_1, \ldots, C_n) > 0\right\} \tag{6}$$

$$K_n \in \mathcal{B}_1^n. \tag{7}$$

Note that Equation (6) is an equality between random variables: $\mathcal{B}_1^n = \mathcal{B}_1(\mathrm{supp}(\delta), C_1 \cdots C_n)$ is $(C_1, \ldots, C_n)$-measurable.

The proof of Lemma 7.1 relies on the following lemma, called the shifting lemma, which describes the effect of shifting time on the probability measure induced by two behavioral strategies.

Lemma 7.2 (Shifting Lemma). *For every $n \in \mathbb{N}$, we denote by $P_{\geq n}$ the infinite suffix of the play truncated up to step n:*

$$P_{\geq n} = K_n, I_n, J_n, C_{n+1}, D_{n+1}, K_{n+1}, \ldots.$$

Let δ be an initial distribution and σ and τ be two behavioral *strategies. Let $c \in C$ and $d \in D$ and $\delta_{(c,d)} \in \Delta(K)$ and σ_c and τ_d be the strategies defined by*

$$\delta_{(c,d)}(k) = \mathbb{P}_\delta^{\sigma,\tau}(K_1 = k \mid C_1 = c, D_1 = d)$$
$$\sigma_c : c_2 c_3 \cdots c_n \mapsto \sigma(c c_2 c_3 \cdots c_n)$$
$$\tau_d : d_2 d_3 \cdots d_n \mapsto \tau(d d_2 d_3 \cdots d_n).$$

Then for every measurable event $E \subseteq K(IJCD)^\omega$,

$$\mathbb{P}_\delta^{\sigma,\tau}\left(P_{\geq 1} \in E \mid C_1 = c, D_1 = d\right) = \mathbb{P}_{\delta_{(c,d)}}^{\sigma_c,\tau_d}(E). \tag{8}$$

More generally, for every $n \in \mathbb{N}$,

$$\mathbb{P}_\delta^{\sigma,\tau}\left(P_{\geq n} \in E \mid C_1 \cdots C_n = c_1 \cdots c_n \wedge D_1 \cdots D_n = d_1 \cdots d_n\right) = \mathbb{P}_{\delta'}^{\sigma_{c_1\cdots c_n},\tau_{d_1\cdots d_n}}(E), \tag{9}$$

where $\sigma_{c_1\cdots c_n}(p) = \sigma(c_1 \cdots c_n p)$ *and* $\tau_{c_1\cdots c_n}$ *is defined similarly and* $\delta'(k) = \mathbb{P}_\delta^{\sigma,\tau}(K_n = k \mid C_1 \cdots C_n = c_1 \cdots c_n \wedge D_1 \cdots D_n = d_1 \cdots d_n)$.

Proof. Using the definition of the probability measure $\mathbb{P}_\delta^{\sigma,\tau}$, Equation (8) holds when E is a finite union of cylinders. Moreover, the class of events E that satisfy Equation (8) is clearly closed under countable monotone unions and intersections and thus it is a monotone class. Thus, according to the monotone class theorem [18, Theorem 6.1.3, page 235], all measurable events have the property in Equation (8). The proof of Equation (9) follows by induction. □

Proof of Lemma 7.1. First, Equation (6) holds for $n = 1$. Let $c \in C$ such that $\mathbb{P}_\delta^{\sigma,\tau}(C_1 = c) > 0$. Then $\sigma(\varepsilon)(\mathrm{Act}_1(c)) > 0$ and

$$\begin{aligned}
&\left\{k \in K \mid \mathbb{P}_\delta^{\sigma,\tau_{\mathrm{rand}}}(K_1 = k \mid C_1 = c) > 0\right\}\\
&\quad = \{k \in K \mid \exists k' \in K, \exists d \in D, \mathbb{P}_\delta^{\sigma,\tau_{\mathrm{rand}}}(K_1 = k', K_0 = k, D_1 = d \mid C_1 = c) > 0\}\\
&\quad = \{k \in K \mid \exists k' \in \mathrm{supp}(\delta), d \in D, p(k', C_1, d \mid k, \mathrm{Act}_1(c), \mathrm{Act}_2(d)) > 0\}\\
&\quad = \mathcal{B}_1(\mathrm{supp}(\delta), c),
\end{aligned}$$

where the first equality is by additivity, the second because all possible actions are played by τ_{rand} and $\sigma(\varepsilon)(\mathrm{Act}_1(c)) > 0$, and the last by definition of the operator $\mathcal{B}_1$. Since $\mathrm{supp}(\delta) = \mathcal{B}_1^0$, Equation (6) holds $\mathbb{P}_\delta^{\sigma,\tau}$ almost surely for $n = 1$. The case for arbitrary $n \in \mathbb{N}$ follows from an induction based on Equation (8) of the shifting lemma. We prove Equation (7). According to Equation (7), Equation (7) holds $\mathbb{P}_\delta^{\sigma,\tau_{\mathrm{rand}}}$ almost surely. Since τ_{rand} plays every possible action with positive probability, $(\mathbb{P}_\delta^{\sigma,\tau}(K_n = k) > 0) \Rightarrow (\mathbb{P}_\delta^{\sigma,\tau_{\mathrm{rand}}}(K_n = k) > 0)$, and thus Equation (7) holds $\mathbb{P}_\delta^{\sigma,\tau}$ almost surely. □

We use the following technical lemma about belief-based strategies several times.

Lemma 7.3. *Fix a Büchi game with observable actions. Let* $\mathcal{L} \subseteq \mathcal{P}(K)$ *and* σ *be a strategy for player 1. Assume that* σ *is a belief strategy and* $\mathcal{L}$ *is downward closed (i.e.,* $L \in \mathcal{L} \wedge L' \subseteq L \Rightarrow L' \in \mathcal{L}$*), and for every* $L \in \mathcal{L} \setminus \{\emptyset\}$ *and every strategy* τ,

$$\mathbb{P}_{\delta_L}^{\sigma,\tau}(\exists n \in \mathbb{N}, K_n \in T) > 0, \tag{10}$$

$$\mathbb{P}_{\delta_L}^{\sigma,\tau}(\forall n \in \mathbb{N}, \mathcal{B}_1^n \in \mathcal{L}) = 1. \tag{11}$$

Then σ *is almost-surely winning for the Büchi game from any support* $L \in \mathcal{L} \setminus \{\emptyset\}$.

Proof. Since $\mathcal{L}$ is downward closed, $\forall L \in \mathcal{L}, \forall l \in L, \{l\} \in \mathcal{L}$ thus Equation (10) implies

$$\forall L \in \mathcal{L}, \forall l \in L, \mathbb{P}_{\delta_L}^{\sigma,\tau}(\exists n \in \mathbb{N}, K_n \in T \mid K_0 = l) > 0. \tag{12}$$

Once σ is fixed, the game is a one-player game with state space $K \times 2^K$ and imperfect information, and Equation (12) implies that in this one-player game,

$$\forall L \in \mathcal{L}, \forall l \in L, \forall \tau, \mathbb{P}_{\delta_L}^{\tau}(\exists n \leq N, K_n \in T \mid K_0 = l) > \varepsilon, \tag{13}$$

where $N = |K| \cdot 2^{|K|}$ and $\varepsilon = p_{\min}^{|K|\cdot 2^{|K|}}$ and $p_{\min}$ is the minimal nonzero transition probability. Moreover, Equation (11) implies that in this one-player game the second component of the state space is always in $\mathcal{L}$, whatever strategy τ is played by player 2. As a consequence, in this one-player game, for every $m \in \mathbb{N}$ and every behavioral strategy τ and every $l \in K$,

$$\mathbb{P}^{\tau}_{\delta_L}\left(\exists m \leq n \leq m + N, K_n \in T \mid K_m = l\right) \geq \varepsilon, \tag{14}$$

whenever $\mathbb{P}^{\tau}_{\delta_L}\left(K_m = l\right) > 0$. We use the Borel-Cantelli Lemma to conclude the proof. According to Equation (14), for every $\tau, L \in \overline{\mathcal{L}}, m \in \mathbb{N}$,

$$\mathbb{P}^{\tau}_{\delta_L}\left(\exists n, mN \leq n < (m+1)N, K_n \in T \mid K_{mN}\right) \geq \varepsilon, \tag{15}$$

which implies that for every behavioral strategy τ and $k, m \in \mathbb{N}$,

$$\mathbb{P}^{\tau}_{\delta_L}\left(\forall n, ((m \cdot N) \leq n < ((m+k) \cdot N) \Rightarrow K_n \notin T)\right) \leq (1-\varepsilon)^k.$$

Since $\sum_k (1-\varepsilon)^k$ is finite, we can apply the Borel-Cantelli Lemma for the events $(\{\forall n, m \cdot N \leq n < (m+k) \cdot N \Rightarrow K_n \notin T\})_k$ and we get $\mathbb{P}^{\tau}_{\delta_L}\left(\forall n, m \cdot N \leq n \Rightarrow K_n \notin T\right) = 0$; thus,

$$\mathbb{P}^{\tau}_{\delta_L}\left(\text{Büchi}\right) = 1.$$

As a consequence, σ is almost-surely winning for the Büchi game. □

7.1.2 The Maximal Strategy. In every Büchi game with observable actions, we define a belief-based strategy $\sigma_{\max}$ called the *maximal strategy of player* 1 and we prove that this strategy is almost-surely winning from any initial distribution that is not positively winning for player 2. The maximal strategy is quite simple to define, as follows.

Definition 7.4 (Maximal Strategy). Fix a Büchi game with observable actions. Let $\mathcal{L} \subseteq \mathcal{P}(K)\backslash\{\emptyset\}$ be the set of supports that are positively winning for player 2. For every $L \subseteq K$ we define the set of L-safe actions:

$$\mathrm{ISafe}_{\mathcal{L}}(L) = \{i \in I \mid \forall c \in C, (\mathrm{Act}_1(c) = i) \Rightarrow (\mathcal{B}_1(L, c) \notin \mathcal{L})\}.$$

The maximal strategy is the belief strategy of player 1 that plays the uniform distribution on $\mathrm{ISafe}_{\mathcal{L}}(\mathcal{B}_1)$ when it is not empty and plays the uniform distribution on I otherwise.

An important feature of the maximal strategy is the following.

Lemma 7.5. *In a Büchi game with observable actions, let $\delta \in \Delta(K)$ be an initial distribution that is not positively winning for player 2, that is, $\mathrm{supp}(\delta) \notin \mathcal{L}$. Then, for every strategy τ of player 2 and every $n \in \mathbb{N}$,*

$$\mathbb{P}^{\sigma_{max}, \tau}_{\delta}\left(\mathcal{B}^n_1 \notin \mathcal{L}\right) = 1. \tag{16}$$

Proof. The proof is by induction on n. The case $n = 0$ is by hypothesis since $\mathbb{P}^{\sigma_{\max}, \tau}_{\delta}(\mathcal{B}^0_1 = \mathrm{supp}(\delta)) = 1$. Let $\overline{\mathcal{L}} = \mathcal{P}(K) \setminus (\mathcal{L} \cup \{\emptyset\})$. To perform the inductive step, it is actually enough to prove

$$\forall L \in \overline{\mathcal{L}}, \mathrm{ISafe}_{\mathcal{L}}(L) \neq \emptyset. \tag{17}$$

Assume that Equation (17) holds and that $\mathbb{P}^{\sigma_{\max}, \tau}_{\delta}(\mathcal{B}^n_1 \in \overline{\mathcal{L}}) = 1$. By definition of observability of actions, $\mathbb{P}^{\sigma_{\max}, \tau}_{\delta}\left(I_n = \mathrm{Act}_1(C_{n+1})\right) = 1$. Thus, by definition of $\mathrm{ISafe}_{\mathcal{L}}(L)$, $\mathbb{P}^{\sigma_{\max}, \tau}_{\delta}(\mathcal{B}_1(\mathcal{B}^n_1, C_{n+1}) \in \overline{\mathcal{L}} \mid I_n \in \mathrm{ISafe}_{\mathcal{L}}(L)) = 1$. This concludes the inductive step since $\mathcal{B}_1(\mathcal{B}^n_1, C_{n+1}) = \mathcal{B}^{n+1}_1$ and $\mathbb{P}^{\sigma_{\max}, \tau}_{\delta}\left(I_n \in \mathrm{ISafe}_{\mathcal{L}}(L)\right) = 1$.

The proof of Equation (17) is by contradiction. Assume that $\mathrm{ISafe}_{\mathcal{L}}(L) = \emptyset$ for some $L \in \overline{\mathcal{L}}$. Then, for every action $i \in I$ there exists a signal $c_i \in C$ such that $\mathcal{B}_1(L, c_i) \neq \emptyset$ and $\mathcal{B}_1(L, c_i) \in \mathcal{L}$. Since $\mathcal{B}_1(L, c_i) \neq \emptyset$, the definition of the belief operator implies

$$\exists l_i \in L, k_i \in K, j_i \in J, d_i \in D, \text{ such that } p(k_i, c_i, d_i \mid l_i, i, j_i) > 0.$$

We prove

$$\forall \sigma, \mathbb{P}_{\delta_L}^{\sigma, \tau_{\text{rand}}}(\mathcal{B}_1^1 \in \mathcal{L}) > 0. \tag{18}$$

Since the game has observable actions, then according to Lemma 4.6, it is enough to prove Equation (18) when σ is a behavioral strategy $\sigma : C^* \to \Delta(I)$. Let $I' = \text{supp}(\sigma(\varepsilon))$ and $i \in I'$. Since $\tau_{\text{rand}}(j_i) > 0$, there is nonzero probability that player 1 receives c_i and then by choice of c_i, $\mathcal{B}_1(L, c_i) \in \mathcal{L}$. This proves Equation (18).

To get the contradiction, we define a strategy τ' for player 2 that is positively winning from δ_L. By definition of $\mathcal{L}$, for every support $B \in \mathcal{L}$ there exists a positively winning strategy τ_B from the uniform initial distribution δ_B. Since actions are observable, then according to Theorem 4.9, we can assume w.l.o.g. that τ_B is behavioral. Let τ' be the general strategy that plays the uniform distribution over J for the first round, and then at the beginning of the second round it selects at random some support $B \in \mathcal{L}$ and then plays τ_B from the second round until the end. According to Equation (18), there exists $c \in C$ such that $\mathcal{B}_1(L, c) \in \mathcal{L}$ and

$$\mathbb{P}_{\delta_L}^{\sigma, \tau'}(C_1 = c) > 0. \tag{19}$$

We fix such a c and we set $B_c = \mathcal{B}_1(L, c)$.

Although τ' is not defined as a behavioral strategy, since actions are observable, τ' is equivalent to a behavioral strategy (Lemma 4.6). Thus, we can apply the shifting lemma (Lemma 7.2) to δ_L, σ, τ' and CoBüchi and with the same notations we get

$$\forall d \in D, \mathbb{P}_{\delta_L}^{\sigma, \tau'}(\text{CoBüchi} \mid C_1 = c, D_1 = d) = \mathbb{P}_{\delta'_{cd}}^{\sigma_c, \tau'_d}(\text{CoBüchi}), \tag{20}$$

with $\delta'_{cd}(k) = \mathbb{P}_{\delta_L}^{\sigma, \tau'}(K_1 = k \mid C_1 = c, D_1 = d)$. Since τ' plays the same way independently of the first signal D_1, τ'_d is independent of d, and actually τ'_d is the strategy that selects randomly any $B \in \mathcal{L}$ and plays τ_B forever. Denote by τ'' this strategy. Summing Equation (20) over all $d \in D$, weighted by $\mathbb{P}_{\delta_L}^{\sigma, \tau'}(D_1 = d)$, we get

$$\mathbb{P}_{\delta_L}^{\sigma, \tau'}(\text{CoBüchi} \mid C_1 = c) = \mathbb{P}_{\delta'_c}^{\sigma_c, \tau''}(\text{CoBüchi}), \tag{21}$$

with $\delta'_c(k) = \mathbb{P}_{\delta_L}^{\sigma, \tau'}(K_1 = k \mid C_1 = c)$. Let $B' = \text{supp}(\delta'_c)$. According to the properties of beliefs (Lemma 7.1), since τ' plays randomly for the first round, $B' = \mathcal{B}_1(L, c) = B_c$. By definition of τ'', there is a positive chance that τ'' plays like τ_{B_c} forever, and τ_{B_c} is positively winning from B_c; thus,

$$\mathbb{P}_{\delta'_c}^{\sigma_c, \tau''}(\text{CoBüchi}) > 0.$$

According to Equation (21), it implies $\mathbb{P}_{\delta_L}^{\sigma, \tau'}(\text{CoBüchi} \mid C_1 = c) > 0$, which together with Equation (19) implies

$$\mathbb{P}_{\delta_L}^{\sigma, \tau'}(\text{CoBüchi}) > 0.$$

Since this holds for every behavioral strategy σ, the strategy τ' is positively winning from support δ_L and thus $L \in \mathcal{L}$, a contradiction with $L \in \overline{\mathcal{L}}$. This completes the proof of Equation (17). □

With the notion of maximal strategy being defined, we can complete the proof of Theorem 6.1.

7.1.3 Proof of Theorem 6.1. Reachability and safety conditions can be easily encoded as Büchi conditions, and thus it is enough to prove Theorem 6.1 for Büchi games. We prove Theorem 6.1 in the case where actions are observable, which implies that Theorem 6.1 holds in every arena, according to Theorem 4.9.

In an arena with observable actions, the maximal strategy $\sigma_{\max}$ is well defined. Since $\mathcal{L}$ is the collection of positively winning supports for player 2, it is enough to show that $\sigma_{\max}$ is almost-surely winning from every support not in $\mathcal{L}$.

Let $\overline{\mathcal{L}} = \mathcal{P}(K) \setminus (\mathcal{L} \cup \{\emptyset\})$. The first step is to prove that for every $L \in \overline{\mathcal{L}}$,

$$\forall \tau, \mathbb{P}_{\delta_L}^{\sigma_{\max}, \tau}\,(\text{Safe}) < 1. \tag{22}$$

We prove Equation (22) by contradiction. Assume Equation (22) does not hold for some $L \in \overline{\mathcal{L}}$ and strategy τ:

$$\mathbb{P}_{\delta_L}^{\sigma_{\max}, \tau}\,(\text{Safe}) = 1. \tag{23}$$

Under this assumption, we use τ to build a strategy positively winning from L, which will contradict the hypothesis $L \in \overline{\mathcal{L}}$. Although τ is surely winning from L against the particular strategy $\sigma_{\max}$, there is no reason for τ to be positively winning from L against all other strategies of player 1. Instead, we define a general strategy $\tau' \in \Delta(C^* \to \Delta(I))$ as follows. The strategy τ' is any general strategy that gives positive probability to play τ as well as any strategy in the family of strategies $(\tau_{n,B})_{n\in\mathbb{N}, B\in\mathcal{L}}$ defined as follows. For every $B \in \mathcal{L}$ we choose a strategy τ_B positively winning from B. Then $\tau_{n,B}$ is the strategy that plays the uniform distribution on J for the first n steps and then forgets past signals and switches definitively to τ_B.

A possible way to implement the general strategy τ' is as follows. At the beginning of the play, player 2 tosses a fair coin. If the result is heads, then he plays τ. Otherwise, he keeps tossing coins and as long as the coin toss is heads, player 2 plays randomly an action in J . The day the coin toss is tails, he picks up randomly some $B \in \mathcal{L}$ and starts playing τ_B.

Now that τ' is defined, we prove it is positively winning from L. Let E be the event "player 1 plays only actions that are safe with respect to her belief", that is,

$$E = \{\forall n \in \mathbb{N}, I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n)\}.$$

Then for every behavioral strategy σ:

- Either $\mathbb{P}_{\delta_L}^{\sigma, \tau'}\,(E) = 1$. In this case

  $$\mathbb{P}_{\delta_L}^{\sigma, \tau'}\,(\text{Safe}) > 0,$$

  because for every finite play $\pi = k_0 i_0 j_0 c_1 d_1 k_1 \cdots k_n$,

  $$\left(\mathbb{P}_{\delta_L}^{\sigma, \tau}\,(\pi) > 0\right) \Rightarrow \left(\mathbb{P}_{\delta_L}^{\sigma_{\max}, \tau}\,(\pi) > 0\right) \Rightarrow (\forall 0 \le m \le n, k_m \notin T),$$

  where the first implication holds because, by definition of $\sigma_{\max}$ and E, for every $c_1 \cdots c_n \in C^*$, $\text{supp}(\sigma(c_1 \cdots c_n)) \subseteq \text{supp}(\sigma_{\max}(c_1 \cdots c_n))$, while the second implication is from Equation (23). Thus, $\mathbb{P}_{\delta_L}^{\sigma, \tau}\,(\text{Safe}) = 1$ and we get $\mathbb{P}_{\delta_L}^{\sigma, \tau'}\,(\text{Safe}) \ge \tau'(\tau) > 0$ by definition of τ'.
- Or $\mathbb{P}_{\delta_L}^{\sigma, \tau'}\,(E) < 1$. Then by definition of E, there exists $n \in \mathbb{N}$ such that $\mathbb{P}_{\delta_L}^{\sigma, \tau'}(I_n \notin \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n)) > 0$. By definition of $\text{ISafe}_{\mathcal{L}}$, it implies $\mathbb{P}_{\delta_L}^{\sigma, \tau'}(\mathcal{B}_1^{n+1} \in \mathcal{L}) > 0$, and thus there exists $B \in \mathcal{L}$ such that $\mathbb{P}_{\delta_L}^{\sigma, \tau'}(\mathcal{B}_1^{n+1} = B) > 0$. By definition of τ', we get $\mathbb{P}_{\delta_L}^{\sigma, \tau_{n+1,B}}(\mathcal{B}_1^{n+1} = B) > 0$, because whatever finite play $k_0, \ldots, k_{n+1}$ leads with positive probability to the event $\{\mathcal{B}_1^{n+1} = B\}$, the same finite play can occur with $\tau_{n+1,B}$ since $\tau_{n+1,B}$ plays every possible action for the $n+1$ first steps. Since $\tau_{n+1,B}$ coincides with τ_{rand} for the first $n+1$ steps, then according to Equation (6), $\mathbb{P}_{\delta_L}^{\sigma, \tau_{n+1,B}}(\mathcal{B}_1^{n+1} = B) > 0$ and $B \subseteq \{k \in K \mid \mathbb{P}_{\delta_L}^{\sigma, \tau_{n+1,B}}(K_{n+1} = k \mid \mathcal{B}_1^{n+1} = B) > 0\}$. Using the shifting lemma (both σ and $\tau_{n+1,B}$ are behavioral) and the definition of τ_B, we get $\mathbb{P}_{\delta_L}^{\sigma, \tau_{n+1,B}}(\text{CoBüchi}) > 0$. As a consequence, by definition of τ', we get that $\mathbb{P}_{\delta_L}^{\sigma, \tau'}(\text{CoBüchi}) > 0$.

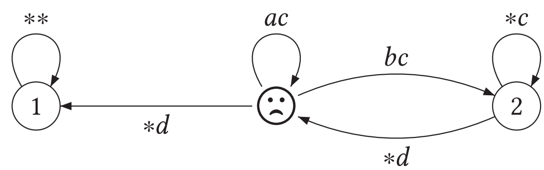

Fig. 9. Co-Büchi games are not qualitatively determined.

In both cases, for every σ, $\mathbb{P}^{\sigma,\tau'}_{\delta_L}$ (CoBüchi) > 0; thus, τ' is positively winning from L. This contradicts the hypothesis of $L \in \overline{\mathcal{L}}$. As a consequence, we get Equation (22) by contradiction.

Using Equation (22), we apply Lemma 7.3 to the collection $\overline{\mathcal{L}}$ and the strategy $\sigma_{\max}$. The collection $\overline{\mathcal{L}}$ is downward closed because $\mathcal{L}$ is upward closed: if a support is positively winning for player 2, then any greater support is positively winning as well, using the same positively winning strategy.

Thus, $\sigma_{\max}$ is almost-surely winning for the Büchi game from every support in $\overline{\mathcal{L}}$ (i.e., every support that is not positively winning for player 2), and hence the game is qualitatively determined.

7.2 Nondeterminacy of Co-Büchi Games

In contrast with Büchi games, not all co-Büchi games are qualitatively determined: a counterexample is represented in Figure 9. Similar examples can be used to prove that stochastic Büchi games with signals do not have a value [21]. In this game, player 1 observes everything, player 2 is blind (he only observes his own actions), and player 1's objective is to visit only finitely many times the ☹-state. The initial state is ☹.

On one hand, no strategy Σ is almost-surely winning for player 1 for the co-Büchi objective. According to Theorem 4.9, since both players can observe their actions, it is enough to prove that no behavioral strategy $\sigma \in C^* \to \Delta(I)$ of player 1 is almost-surely winning. Fix strategy σ and assume toward contradiction that σ is almost-surely winning. We define a strategy τ such that $\mathbb{P}^{\sigma,\tau}_{☹}$ (Büchi) > 0. Strategy τ starts by playing only c. The probability to be in state ☹ at step n is $x^0_n = \mathbb{P}^{\sigma,c^\omega}_{☹}(K_n = ☹)$, and since σ is almost-surely winning, then $x^0_n \to_n 0$; thus, there exists n_0 such that $x^0_{n_0} \leq \frac{1}{2}$. Then τ plays d at step n_0. Assuming the state was 2 when d was played, the probability to be in state ☹ at step $n \geq n_0$ is $x^1_n = \mathbb{P}^{\sigma,c^{n_0}dc^\omega}_{☹}(K_n = ☹ \mid K_{n_0} = ☹)$, and since σ is almost-surely winning, there exists n_1 such that $x^1_{n_1} \leq \frac{1}{4}$. Then τ plays d at step n_1. By induction, we keep defining τ this way so that $\tau = c^{n_0-1}dc^{n_1-n_0-1}dc^{n_2-n_1-1}d\ldots$. and for every $k \in \mathbb{N}$, $\mathbb{P}^{\sigma,\tau}_{☹}(K_{n_{k+1}} = ☹ \text{ and } K_{n_{k+1}-1} = 2 \mid K_{n_k} = ☹) \geq 1 - \frac{1}{2^{k+1}}$. Thus, finally, $\mathbb{P}^{\sigma,\tau}_{☹}$ (Büchi) $\geq \Pi_k(1 - \frac{1}{2^{k+1}}) > 0$, which contradicts the hypothesis.

On the other hand, player 2 does not have a positively winning strategy either. Intuitively, player 2 cannot win positively because as time passes, either the play reaches state 1 or the chances that player 2 plays action d drop to 0. When these chances are small, player 1 can play action c and she bets no more d will be played and the play will stay safe in state 2. If player 1 loses her bet, then again she waits until the chances to see another d are small and then plays action c. Player 1 may lose a couple of bets but almost surely she eventually is right and the CoBüchi condition is fulfilled. Formally, according to Theorem 4.9, since both players can observe their actions, it is enough to prove that no behavioral strategy $\tau \in D^* \to \Delta(J)$ of player 2 is positively winning. The strategy τ being fixed, we define a strategy σ for player 1 such that $\mathbb{P}^{\sigma,\tau}_{☹}$ (Büchi) $= 1$. The only state where player 1's action matters is ☹. After a play $p = k_0 i_0 j_0 \cdots k_n$ ending up in state ☹ (player 1 can observe the state), the strategy σ plays action a except if the trigger condition

$$\mathbb{P}_{☹}^{i_0 \cdots i_n a^\omega, \tau} (\forall m \geq n, J_m \neq d \mid P_n = p) \geq \frac{1}{2}$$

is satisfied—in this case action b is played. Let E_0 be the event in which finitely many d are played, that is, $E_0 = \{\exists n, \forall m \geq n, J_m \neq d\}$. According to Lévy law, $\mathbb{P}_{☹}^{\sigma,\tau} (E_0 \mid P_n)$ converges $\mathbb{P}_{☹}^{\sigma,\tau}$ almost surely to the indicator function 1_{E_0} of the event E_0. If E_0 holds, then finitely many d are played, and the play cannot stay forever in state ☹ after the last d because $\mathbb{P}_{☹}^{\sigma,\tau} (E_0 \mid P_n)$ converges to 1 and thus the trigger condition is eventually satisfied. Thus, when E_0 holds, the play eventually stays in state 1 or 2 and the CoBüchi condition is satisfied. If E_0 does not hold, then $\mathbb{P}_{☹}^{\sigma,\tau} (E_0 \mid P_n)$ converges to 0 and thus eventually the trigger condition is not satisfied anymore and hence player 1 eventually plays no more bs, only as. But E_0 does not hold and thus infinitely many d are played, and thus the play reaches state 1. In both cases, CoBüchi holds $\mathbb{P}_{☹}^{\sigma,\tau}$ almost surely, and thus τ is not positively winning.

Finally, neither player 1 wins almost surely nor player 2 wins positively.

8 ALGORITHMS

8.1 A Naïve Algorithm

As a corollary of the proof of qualitative determinacy (Theorem 6.1), we get a maximal strategy $\sigma_{\max}$ for player 1 (see Definition 7.4) to win almost-surely Büchi games.

Corollary 8.1. *If player 1 has an almost-surely winning strategy in a Büchi game with observable actions, then the maximal strategy σ_{max} is almost-surely winning.*

A simple algorithm to decide for which player a game is winning can be derived from Corollary 8.1: this simple algorithm enumerates all possible belief strategies and tests each one of them to see if it is almost-surely winning. The test reduces to checking positive winning in one-player co-Büchi games and can be done in exponential time. As there is a doubly exponential number of belief strategies, this can be done in time doubly exponential. This algorithm also appears in [23]. This settles the upper bound for Theorem 6.3. The lower bounds are established in Theorem 10.1, proving that this enumeration algorithm is optimal for worst-case complexity. While optimal in the worst case, this algorithm is likely to be inefficient in practice. For instance, if player 1 has no almost-surely winning strategy, then this algorithm will enumerate every single of the doubly exponential many possible belief strategies. Instead, we provide fix-point algorithms that do not enumerate every possible strategy in Theorem 8.2 for reachability games and Theorem 8.3 for Büchi games. Although they should perform better on games with particular structures, these fix-point algorithms still have a worst-case 2-EXPTIME complexity.

8.2 A Fix-Point Algorithm for Reachability Games

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

Theorem 8.2 (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* $\mathrm{supp}(\delta) \subseteq K$. *The corresponding partition of $\mathcal{P}(K)$ is computable in time $O(|G| \cdot 2^{|K|})$, where $|G|$ denotes the size of the description of the game, as the largest fix point of a monotonic operator $\Phi : \mathcal{P}(\mathcal{P}(K)) \to \mathcal{P}(\mathcal{P}(K))$ computable in time linear in $|G|$.*

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

$$\Phi(\mathcal{L}) = \{L \in \mathcal{L} \mid \exists j_L \in J, \forall d \in D, (\mathrm{Act}_2(d) = j_L) \Rightarrow (\mathcal{B}_2(L,d) \in \mathcal{L} \cup \{\emptyset\})\}; \tag{24}$$

in other words, $\Phi(\mathcal{L})$ is the set of supports such that player 2 has an action that ensures his next belief will be in $\mathcal{L}$, whatever signal d he might receive. Let σ_{rand} 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, and
(B) σ_{rand} is positively winning from any support $L \subseteq K$ that is not in $\mathcal{L}_\infty$.

We start with proving (A). To win surely from any support $L \in \mathcal{L}_\infty$, player 2 uses the following belief strategy τ_B: when the current belief of player 2 is $L \in \mathcal{L}_\infty$, player 2 plays an action j_L defined as in Equation (24). By definition of Φ and since $\mathcal{L}_\infty$ is a fix point of Φ, there always exists such an action. When playing with the belief strategy τ_B, starting from a support in $\mathcal{L}_\infty$, the beliefs of player 2 stay in $\mathcal{L}_\infty$ and never intersect T because $\mathcal{L}_\infty \subseteq \mathcal{P}(K \backslash T)$. According to Equation (7) of beliefs (Lemma 7.1), this guarantees the play never visits T, whatever strategy is used by player 1.

We now prove (B). 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 fix point of Φ. We prove that for any support $L \in \mathcal{P}(K)$, if $L \notin \mathcal{L}_\infty$, then

$$\sigma_{\mathrm{rand}} \text{ is positively winning for player 1 from } L. \tag{25}$$

If $L \cap T \neq \emptyset$, Equation (25) is obvious. To deal with the case where $L \cap T = \emptyset$, 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 behavioral strategy τ,

$$\mathbb{P}_{\delta_L}^{\sigma_{\mathrm{rand}},\tau}(\exists m, 2 \leq m \leq n+1, K_m \in T) > 0. \tag{26}$$

For $n = 0$, Equation (26) is obvious because $\mathcal{K}_0 = \emptyset$. Suppose that for some $n \in \mathbb{N}$, Equation (26) holds for every $L' \in \mathcal{K}_n$, and let $L \in \mathcal{K}_{n+1}\backslash\mathcal{K}_n$. Then, by definition of $\mathcal{K}_{n+1}$,

$$L \in \mathcal{L}_n \backslash \Phi(\mathcal{L}_n). \tag{27}$$

Let δ_L be an initial distribution with support L and τ any behavioral strategy for player 2. Let $J_0 \subseteq J$ be the support of $\tau(\delta_L)$ and $j_L \in J_0$. According to Equation (27), by definition of Φ, there exists a signal $d \in D$ such that $\mathrm{Act}_2(d) = j_L$ and $\mathcal{B}_2(L,d) \notin \mathcal{L}_n$ and $\mathcal{B}_2(L,d) \neq \emptyset$. According to the property in Equation (6) of beliefs (Lemma 7.1), $\forall k \in \mathcal{B}_2(L,d), \mathbb{P}_{\delta_L}^{\sigma_{\mathrm{rand}},\tau}(K_2 = k \wedge D_1 = d) > 0$. If $\mathcal{B}_2(L,d) \cap T \neq \emptyset$, then, according to the definition of beliefs, $\mathbb{P}_{\delta_L}^{\sigma_{\mathrm{rand}},\tau}(K_2 \in T) > 0$. Otherwise, $\mathcal{B}_2(L,d) \in \mathcal{P}(K\backslash T)\backslash\mathcal{L}_n = \mathcal{K}_n$ and hence distribution $\delta_d : k \to \mathbb{P}_{\delta_L}^{\sigma_{\mathrm{rand}},\tau}(K_2 = k \mid D_1 = d)$ has its support in $\mathcal{K}_n$. By inductive hypothesis, for every behavioral strategy τ',

$$\mathbb{P}_{\delta_d}^{\sigma_{\mathrm{rand}},\tau'}(\exists m \in \mathbb{N}, 2 \leq m \leq n+1, K_m \in T) > 0;$$

hence, using the shifting lemma and the definition of δ_d,

$$\mathbb{P}_{\delta}^{\sigma_{\mathrm{rand}},\tau}(\exists m \in \mathbb{N}, 3 \leq m \leq n+2, K_m \in T) > 0,$$

which completes the proof of the inductive step. Hence, Equation (26) holds for every behavioral strategy τ. Thus, according to Lemma 4.6, Equation (26) holds as well for every general strategy τ.

To compute 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 fix point 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 $|G| \cdot 2^{|K|}$. To compute the strategy τ_B, it is enough to compute for each $L \in \mathcal{L}_\infty$ one action j_L such that $(\mathrm{Act}_2(d) = j_L) \Rightarrow (\mathcal{B}_2(L,d) \in \mathcal{L}_\infty)$. □

As a byproduct of the proof, one obtains the following bounds on time and probabilities before reaching a target state, when player 1 uses the uniform memoryless strategy σ_{rand}. From an initial distribution positively winning for the reachability objective, for every strategy τ,

$$\mathbb{P}_{\delta}^{\sigma_{\text{rand}},\tau}\left(\exists n \leq 2^{|K|}, K_n \in T\right) \geq \left(\frac{1}{p_{\min} \mid I \mid}\right)^{2^{|K|}}, \tag{28}$$

where $p_{\min}$ is the smallest nonzero transition probability.

8.3 A Fix-Point Algorithm for Büchi Games

To decide whether player 1 wins almost surely a Büchi game, we provide an algorithm that runs in doubly exponential time. It uses the algorithm for reachability games as a subprocedure.

Theorem 8.3 (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* $\text{supp}(\delta) \subseteq K$. *The corresponding partition of $\mathcal{P}(K)$ is computable in time $O(2^{2^{|G|}})$, where $|G|$ denotes the size of the description of the game, as a projection of the greatest fix-point $\mathcal{L}_\infty$ of a monotonic operator*

$$\Psi : \mathcal{P}(\mathcal{P}(K) \times K) \to \mathcal{P}(\mathcal{P}(K) \times K).$$

The operator Ψ is computable using as a nested fix point the operator Φ of Theorem 8.2. The almost-surely winning belief strategy of player 1 *and the positively winning 2-belief strategy of player* 2 *can be extracted from $\mathcal{L}_\infty$.*

The proof of Theorem 8.3 is detailed in Section 8.4. We sketch here the main ideas.

First, suppose that from *every* initial support, player 1 can win positively the reachability game. Then she can do so using a belief strategy, and according to Lemma 7.3, this strategy guarantees almost surely the Büchi condition.

In general, though, player 1 is not in such an easy situation and there exists a support L that is *not* positively winning for her for the reachability objective. Then, by qualitative determinacy, player 2 has a strategy to achieve surely her safety objective from L, which is a fortiori surely winning for her co-Büchi objective as well.

We prove that in case player 2 can *force with positive probability the belief of player* 1 to be L eventually from another support L', then player 2 has a general strategy to win positively from L'. This is not completely obvious because in general, player 2 cannot know exactly *when* the belief of player 1 is L (he can only compute the 2-belief, letting him know all the possible beliefs player 1 can have). However, player 2 can make blind guesses and be right with greater than zero probability. For winning positively from L', player 2 plays totally randomly until he guesses randomly that the belief of player 1 is L, and at that moment he 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 he makes a wrong guess about the belief of player 1. However, there is a nonzero probability for his guess to be right.

Hence, player 1 should surely avoid her belief to be L or L' if she wants to win almost surely. However, by 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. This is the basis of our fix-point algorithm.

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 which player 1 has a strategy that wins positively the reachability game

and also ensures 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 guarantees for sure her beliefs to stay outside $\mathcal{L}'_i$, and
(b) from supports in $\mathcal{L}'_{i+1}$, player 2 can ensure with positive probability the belief of player 1 to be in $\mathcal{L}''_{i+1}$ eventually, under the same hypothesis.

The overall strategy of player 2 positively winning for the co-Büchi objective consists of playing randomly for some time until he decides to pick up randomly a belief L of player 1 in some $\mathcal{L}''_i$ and bets that the current belief of player 1 is L and that player 1 guarantees for sure her future beliefs will stay outside $\mathcal{L}'_i$. He forgets the signals he has received up to that moment and switches definitively to a strategy that guarantees (a). With positive probability, player 2 guesses 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 and player 2 wins.

In order to ensure (a), player 2 makes use of the hypothesis about player 1 beliefs staying outside $\mathcal{L}'_i$. For that player 2 needs to keep track of all the possible beliefs of player 1, hence the doubly exponential memory. The reason is that player 2 can infer from this data structure some information about the possible actions played by player 1: in case for every possible belief of player 1 an action $i \in I$ creates a risk to reach $\mathcal{L}'_i$, then player 2 knows for sure this action is not played by player 1. This in turn helps player 2 to know which are the possible states of the game. Finally, when player 2 estimates the state of the game using his 2-beliefs, this gives a potentially more accurate estimation of the possible states than simply computing his 1-beliefs.

The positively winning 2-belief strategy of player 2 has a particular structure. All memory updates are deterministic except for one: from the initial memory state $\emptyset$, whatever signal is received, there is a nonzero chance that the memory state stays $\emptyset$, but it may as well be updated to several other memory states.

8.4 Proof of Theorem 8.3

According to Theorem 4.9, it is enough to prove Theorem 8.3 in the case where actions are observable, which we assume in all of this proof.

We start with formalizing what it means for player 1 to enforce her beliefs to stay outside a certain set.

Definition 8.4. Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of nonempty 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 $\mathcal{L}$,

$$\mathbb{P}_\delta^{\sigma,\tau}\left(\forall n \in \mathbb{N}, \mathcal{B}_1^n \notin \mathcal{L}\right) = 1. \tag{29}$$

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

$$\mathrm{ISafe}_{\mathcal{L}}(L) = \{i \in I \mid \forall c \in C, (\mathrm{Act}_1(c) = i) \Rightarrow (\mathcal{B}_1(L, c) \notin \mathcal{L})\}$$

of actions that guarantees the next belief of player 1 to stay outside $\mathcal{L}$ is not empty.

Note that the same operator $\mathrm{ISafe}_{\mathcal{L}}$ is also used in the proof of qualitative determinacy.

Proof. The equivalence is straightforward. In one direction, let σ be a strategy with the property in Equation (29), $L \notin \mathcal{L}$, δ_L a distribution with support L. Then, according to Equation (29), $\mathrm{supp}(\sigma(\delta_L)(i)) \subseteq \mathrm{ISafe}_{\mathcal{L}}(L)$ and hence $\mathrm{ISafe}_{\mathcal{L}}(L)$ is not empty. In the other direction, if $\mathrm{ISafe}_{\mathcal{L}}(L)$ is not empty for every $L \notin \mathcal{L}$, then consider the finite-memory strategy σ for player 1, which plays an action in $\mathrm{ISafe}_{\mathcal{L}}(L)$ when the belief of player 1 is L. Then, by definition of $\mathrm{ISafe}_{\mathcal{L}}(L)$, and according to Lemma 7.1, the property in Equation (29) holds. □

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

Definition 8.5 ($\mathcal{L}$-games). Let $\mathcal{L}$ be an upward-closed set of supports such that player 1 can enforce her beliefs to stay outside $\mathcal{L}$. The $\mathcal{L}$-game has the same actions, transitions, and signals as the original partial observation game, only the winning condition changes: player 1 wins if the play reaches a target state, and moreover, player 1 is restricted to use actions in $\mathrm{ISafe}_{\mathcal{L}}(L)$ whenever her belief is L. The winning condition is

$$\mathrm{Win}_{\mathcal{L}} = \left\{ \exists n, K_n \in T \text{ and } \forall n, I_n \in \mathrm{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n) \right\}. \tag{30}$$

Note that, strictly speaking, $\mathcal{L}$-games are not reachability games, but the proof shows that they can be encoded into reachability games. The following properties of $\mathcal{L}$-games are crucial.

Proposition 8.6 ($\mathcal{L}$-games). *Let G be a Büchi game with observable actions. Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be a set of nonempty supports such that $\mathcal{L}$ is upward closed and such that player 1 can enforce her 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 by $\mathcal{L}''$ the set of supports that are not in $\mathcal{L}$ and are surely winning for player 2 in the $\mathcal{L}$-game.*
(ii) *Assume $\mathcal{L}''$ is empty. Then every support not in $\mathcal{L}$ is almost-surely winning for player 1, both in the $\mathcal{L}$-game and for the Büchi objective in game G.*
(iii) *Assume $\mathcal{L}''$ is not empty. Then player 2 has a 2-belief strategy τ with memory $\mathcal{P}(\mathcal{L}'' \times K) \setminus \{\emptyset\}$ to win surely the $\mathcal{L}$-game from any support in $\mathcal{L}''$.*
(iv) *There is an algorithm running in time doubly exponential in the size of G to compute $\mathcal{L}''$ and, in case (iii) holds, strategy τ. This algorithm performs the fix-point computation of Theorem 8.2 on a game with state space $\mathcal{P}((\mathcal{P}(K)\backslash\mathcal{L}) \times K)$.*

Proof. We define a reachability game $G_{\mathcal{L}}$ that is similar to the $\mathcal{L}$-game. The game $G_{\mathcal{L}}$ is a synchronized product of the original game G with beliefs of player 1, with a few modifications. The state space is $K_{\mathcal{L}} = K \times (\mathcal{P}(K)\backslash\mathcal{L} \cup \{\emptyset\})$. The first component is the state K_n of the original game $G_{\mathcal{L}}$ and performs transitions according to the transition rules of the original game G. The second component keeps track of the belief of player 1, and in case player 1 plays a forbidden action $i \notin \mathrm{ISafe}_{\mathcal{L}}(B)$, this component is emptied definitively. Target states $T_{\mathcal{L}}$ of $G_{\mathcal{L}}$ are $T_{\mathcal{L}} = \{(s, B) \mid s \in T \wedge B \neq \emptyset\}$, so to win the game, a target state of the game G should be entered while the belief has never been emptied.

Formally, the nonzero values of the transition function $p_{\mathcal{L}}$ of $G_{\mathcal{L}}$ are defined for every $i \in I, j \in J$ and $k, k' \in K$ and $B, B' \subseteq K$ by $p_{\mathcal{L}}((k', B'), c, d \mid (k, B)i, j) = p(k', c, d \mid k, i, j)$, where, if $i \in \mathrm{ISafe}_{\mathcal{L}}(B)$,

$$B' = \begin{cases} \mathcal{B}_1(B, c) & \text{if } (B \neq \emptyset) \wedge (i \in \mathrm{ISafe}_{\mathcal{L}}(B)) \\ \emptyset & \text{otherwise.} \end{cases}$$

To get (i), (iii), and (iv), we apply Theorem 8.2 to the reachability game $G_{\mathcal{L}}$: for every L, let $\delta_{\mathcal{L}}(L)$ be the uniform distribution on $L \times \{L\}$, and then $\delta_{\mathcal{L}}(L)$ is either positively winning for 1 or surely winning for 2 in $G_{\mathcal{L}}$. We show that the same holds for L in the $\mathcal{L}$-game.

Assume $\delta_{\mathcal{L}}(L)$ is positively winning for 1 in $G_{\mathcal{L}}$; then according to Theorem 8.2, the strategy σ_{rand} that plays randomly all actions is positively winning in $G_{\mathcal{L}}$. By construction of $G_{\mathcal{L}}$, after signals $c_1 \cdots c_n$, playing an action $i \notin \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1(c_1 \cdots c_n))$ is useless for player 1 since it empties the second component and thus the probability to reach $T_{\mathcal{L}}$ is 0 onward. Thus, the strategy $\sigma_{\mathcal{L}}$ that plays randomly any action in $\text{ISafe}_{\mathcal{L}}(\mathcal{B}_1(c_1 \cdots c_n))$ after signals $c_1 \cdots c_n$ is positively winning as well in $G_{\mathcal{L}}$. Moreover, $\sigma_{\mathcal{L}}$ guarantees in G that $\forall \tau, \mathbb{P}^{\sigma,\tau}_{\delta}(\forall n, \sigma(C_1, \ldots, C_n) \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}^n_1)) = 1$ and thus it is positively winning in the $\mathcal{L}$-game.

Assume now that $\delta_{\mathcal{L}}(L)$ is surely winning for player 2 in $G_{\mathcal{L}}$; then, according to Theorem 8.2, player 2 can win surely with a belief strategy τ. A belief strategy in $G_{\mathcal{L}}$ is a 2-belief strategy in G. According to the definition of $T_{\mathcal{L}}$, τ guarantees for sure in $G_{\mathcal{L}}$ that $\forall n \in \mathbb{N}, (K_n \in T \times \mathcal{P}(K)) \Rightarrow (K_n \in T \times \{\emptyset\})$. Since τ is a strategy in the $\mathcal{L}$-game as well, and by definition of transitions in $G_{\mathcal{L}}$, τ guarantees in the $\mathcal{L}$-game that $\forall n \in \mathbb{N}, (K_n \in T) \Rightarrow (\exists m \leq n, I_m \notin \text{ISafe}_{\mathcal{L}}(\mathcal{B}^n_1))$, τ is surely winning for player 2 in the $\mathcal{L}$-game.

This terminates the proof of (i), (iii), and (iv).

Now we suppose $\mathcal{L}''$ is empty and prove (ii). We use again the positively winning belief strategy $\sigma_{\mathcal{L}}$ defined earlier. We apply Lemma 7.3 to $\sigma_{\mathcal{L}}$ and $\overline{\mathcal{L}} = \mathcal{P}(K) \setminus \mathcal{L}$, which is downward closed because $\mathcal{L}$ is upward closed. For that we shall prove that the two hypotheses in Equations (10) and (11) are satisfied. The hypothesis in Equation (11) holds because $\sigma_{\mathcal{L}}$ only plays action in $I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}^n_1)$; thus, if the initial support is in $\overline{\mathcal{L}}$, then $\sigma_{\mathcal{L}}$ guarantees for sure $\forall n, \mathcal{B}^n_1 \in \overline{\mathcal{L}}$. To prove Equation (10), we need to show that for every $L \in \overline{\mathcal{L}}$ and $l \in L$, $\mathbb{P}^{\sigma_{\mathcal{L}},\tau}_{\delta_L}(\exists n \in \mathbb{N}, K_n \in T \mid K_0 = l) > 0$. Since $\mathcal{L}$ is upward closed, $\overline{\mathcal{L}}$ is downward closed, and since $\mathcal{L}'' = \emptyset$, $\sigma_{\mathcal{L}}$ is positively winning in $G_{\mathcal{L}}$ from every nonempty $L' \in \overline{\mathcal{L}}$. This proves Equation (10), and thus all hypotheses of Lemma 7.3 are satisfied. According to Lemma 7.3, $\sigma_{\mathcal{L}}$ is almost-surely winning the Büchi game from every initial support in $\overline{\mathcal{L}}$. This terminates the proof of (ii). □

The properties of $\mathcal{L}$-games lead to a fix-point characterization of almost-surely winning supports for player 1.

Proposition 8.7 (Fix-Point Characterization of Almost-Surely Winning). *Let G be a Büchi game with observable actions. Let $\mathcal{L} \subseteq \mathcal{P}(K)$ be an upward-closed set of supports such that player 1 can enforce her beliefs to stay outside $\mathcal{L}$. Let $\mathcal{L}''$ be the set of supports surely winning for player 2 in the $\mathcal{L}$-game and*

$$\mathcal{L}' = \left\{L \notin \mathcal{L} \mid \forall \sigma, \mathbb{P}^{\sigma,\tau_{rand}}_{\delta_L}(\exists n, \mathcal{B}^n_1 \in \mathcal{L} \cup \mathcal{L}'') > 0\right\}, \tag{31}$$

where τ_{rand} is the strategy for player 2 playing randomly any action. Then,

(i) *either $\mathcal{L}' = \emptyset$, and in this case every support $L \notin \mathcal{L}$ is almost-surely winning for player 1 and her Büchi objective;*
(ii) *or $\mathcal{L}' \neq \emptyset$, and in this case:*
  (a) *$\mathcal{L} \cap \mathcal{L}' = \emptyset$,*
  (b) *player 1 can enforce her beliefs to stay outside $\mathcal{L} \cup \mathcal{L}'$,*
  (c) *there is a 2-belief strategy τ^* for player 2 with memory $\mathcal{P}(\mathcal{L}' \times K)$ such that:*

$$\forall \sigma, \forall L \in \mathcal{L}', \mathbb{P}^{\sigma,\tau^*}_{\delta_L}(\text{CoBüchi} \mid \forall n, I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}^n_1)) > 0. \tag{32}$$

There exists an algorithm running in time doubly exponential in the size of G for deciding whether (i) or (ii) holds. In case (ii) holds, the algorithm computes as well $\mathcal{L}'$ and τ^*.

Proof. We start with proving that if $\mathcal{L}''$ is empty and then (i) holds. In this case, 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 8.6, every support not in $\mathcal{L}$ is almost-surely winning for player 1 for the Büchi condition, and hence (i) holds.

Suppose now that $\mathcal{L}''$ is *not* empty, Then we prove (ii)(a), (ii)(b), and (ii)(c). Property (ii)(a) is obvious because $\mathcal{L}'$ contains $\mathcal{L}''$. Property (ii)(b) follows from the characterization in Definition 8.4: if for some $L \in \mathcal{P}(K) \setminus \emptyset$ the set $\text{ISafe}_{\mathcal{L}\cup\mathcal{L}'}(L)$ is empty, then $\forall \sigma, \mathbb{P}_{\delta_L}^{\sigma,\tau_{\text{rand}}}(\mathcal{B}_1^1 \in \mathcal{L}' \cup \mathcal{L}) > 0$ and thus $L \in \mathcal{L} \cup \mathcal{L}$.

Now we prove (ii)(c). According to (iii) of Proposition 8.6, there exists a 2-belief strategy τ' for player 2 that is surely winning in the $\mathcal{L}$-game from any support in $\mathcal{L}''$. We define a 2-belief strategy τ^* for player 2 such that Equation (32) holds. The initial state is $\emptyset$, and in this state player 2 throws a coin. As long as the result is "tails," player 2 plays randomly any action and the memory state is $\emptyset$. If the result is "heads," then player 2 picks randomly a memory state $L \in \mathcal{L}''$ and switches to the 2-belief strategy τ'. Intuitively, player 2 guesses the belief of player 1 and bets that player 1 will only play safe actions from that moment on. Thus, when playing against τ^*, the opponent player 1 does not know whether she faces strategy τ' or strategy τ_{rand}, because everything is possible with strategy τ_{rand}.

Let us prove that τ^* guarantees the property in Equation (32). By definition of the probability distribution induced by a general strategy, w.l.o.g. it is enough to prove Equation (32) in the case where σ is a behavioral strategy. Let $L \in \mathcal{L}'$. We assume w.l.o.g. that

$$\mathbb{P}_{\delta_L}^{\sigma,\tau'}\left(\forall n, I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n)\right) > 0; \tag{33}$$

otherwise, Equation (32) is undefined.

We first prove Equation (32) in case $L \in \mathcal{L}''$. By definition of $\mathcal{L}''$, L is surely winning for player 2 in the $\mathcal{L}$-game, and τ' guarantees $\mathbb{P}_{\delta_L}^{\sigma,\tau'}(\text{Win}_{\mathcal{L}}) = 0$. Since $\text{Win}_{\mathcal{L}} = \{\exists n, K_n \in T \text{ and } \forall n, I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n)\}$, then $\mathbb{P}_{\delta_L}^{\sigma,\tau'}(\exists n, K_n \in T \mid \forall n, I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n)) = 0$. There is positive probability that τ^* plays like τ', and thus

$$\mathbb{P}_{\delta_L}^{\sigma,\tau^*}\left(\exists n, K_n \in T \mid \forall n, I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n)\right) < 1, \tag{34}$$

which implies Equation (32).

Now we prove Equation (32) in case $L \in \mathcal{L}'$. For every $n \in \mathbb{N}$ there is positive probability that τ^* plays like τ_{rand} up to step n. Thus, according to the definition of $\mathcal{L}'$,

$$\mathbb{P}_{\delta_L}^{\sigma,\tau^*}\left(\exists n, \mathcal{B}_1^n \in \mathcal{L}'' \cup \mathcal{L}\right) > 0. \tag{35}$$

By definition of $\text{ISafe}_{\mathcal{L}}$, if $\mathcal{B}_1^n \notin \mathcal{L}$ and $I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n)$, this guarantees for sure that $\mathcal{B}_1^{n+1} \notin \mathcal{L}$. Since $L \notin \mathcal{L}$, Equation (35) implies

$$\mathbb{P}_{\delta_L}^{\sigma,\tau^*}\left(\exists n, \mathcal{B}_1^n \in \mathcal{L}'' \mid \forall n, I_n \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^n)\right) > 0. \tag{36}$$

As a consequence, according to the assumption in Equation (33), there exists a finite play $\pi = k_0 i_0 j_0 c_1 d_1 k_1 \cdots k_n$ such that $\mathbb{P}_{\delta_L}^{\sigma,\tau^*}(P_n = \pi \wedge \forall m, I_m \in \text{ISafe}_{\mathcal{L}}(\mathcal{B}_1^m)) > 0$ and $\mathcal{B}_1(L, c_1 \ldots c_n) \in \mathcal{L}''$. Denote $B = \mathcal{B}_1(L, c_1 \ldots c_n)$.

Since σ and τ^* are behavioral, we can apply the shifting lemma (Lemma 7.2) to δ_L, σ, τ^* and $E = \{\forall m, K_m \notin T\}$; hence,

$$\mathbb{P}_{\delta_L}^{\sigma,\tau^*}\left(P_{\geq n} \in E \mid R\right) = \mathbb{P}_{\delta'}^{\sigma_{c_1\cdots c_n},\tau^*_{d_1\cdots d_n}}(E), \tag{37}$$

with $R = \{C_1 \cdots C_n = c_1 \cdots c_n \wedge D_1 \cdots D_n = d_1 \cdots d_n\}$ and $\delta'(k) = \mathbb{P}_{\delta_L}^{\sigma,\tau^*}(K_n = k \mid R)$.

We show that

$$\operatorname{supp}(\delta') = B. \tag{38}$$

Since there is positive probability that τ^* plays like τ_{rand} for any number of steps, the property in Equation (6) of Lemma 7.1 implies $B = \{k \in K \mid \mathbb{P}^{\sigma,\tau^*}_{\delta_L}(K_n = k \mid C_1 \cdots C_n = c_1 \cdots c_n) > 0\}$. Moreover, again because τ^* may play any action at any time whatever signal is received by player 2, $\mathbb{P}^{\sigma,\tau^*}_{\delta_L}(K_n = k \mid C_1 \cdots C_n = c_1 \cdots c_n) > 0 \iff \mathbb{P}^{\sigma,\tau^*}_{\delta_L}(K_n = k \mid R) > 0$. This shows Equation (38).

We have already proved that Equation (32) holds for $L \in \mathcal{L}''$, and according to Equation (38), $\operatorname{supp}(\delta') = B \in \mathcal{L}''$ and thus we get $\mathbb{P}^{\sigma_{c_1\cdots c_n},\tau^*}_{\delta'}(E) > 0$. Since there is positive probability that $\tau^*_{d_1\cdots d_n}$ coincides with τ^*, $\mathbb{P}^{\sigma_{c_1\cdots c_n},\tau^*_{d_1\cdots d_n}}_{\delta'}(E) > 0$. Then Equation (37) implies

$$\mathbb{P}^{\sigma,\tau^*}_{\delta_L}(P_{\geq n} \in E \mid R) > 0. \tag{39}$$

By choice of π, $\mathbb{P}^{\sigma,\tau^*}_{\delta_L}(P_n = \pi \wedge \forall m, I_m \in \mathrm{ISafe}_{\mathcal{L}}(\mathcal{B}_1^m)) > 0$ and $\{P_n = \pi\} \subseteq R$; thus, $\mathbb{P}^{\sigma,\tau^*}_{\delta_L}(R \mid \forall m, I_m \in \mathrm{ISafe}_{\mathcal{L}}(\mathcal{B}_1^m)) > 0$. Together with Equation (39) we get $\mathbb{P}^{\sigma,\tau^*}_{\delta_L}(P_{\geq n} \in E \mid \forall m, I_m \in \mathrm{ISafe}_{\mathcal{L}}(\mathcal{B}_1^m)) > 0$. By definition of E, it implies

$$\mathbb{P}^{\sigma,\tau^*}_{\delta_L}(\mathrm{CoBüchi} \mid \forall m, I_m \in \mathrm{ISafe}_{\mathcal{L}}(\mathcal{B}_1^m)) > 0;$$

thus, Equation (32) is proved.

Description of the algorithm. To terminate the proof of Proposition 8.7, we have to describe the doubly exponential time algorithm.

First, we compute $\mathcal{L}''$ using the algorithm of Proposition 8.6 on the game $G_{\mathcal{L}}$. In case $\mathcal{L}''$ is not empty, the algorithm computes $\mathcal{L}'$ defined by Equation (31). This can be performed by solving a one-player game with a sure-winning safety condition. The game is a synchronized product of the one-player version of G, where player 2 plays totally randomly with the beliefs of player 1; this is similar to and easier than computing $\mathcal{L}''$ and we do not give more details.

Once $\mathcal{L}'$ has been computed, the algorithm outputs the 2-belief strategy τ^* with memory $\mathcal{P}(\mathcal{L}' \times K)$, whose construction is described in (ii)(b). For that it uses the algorithm of Proposition 8.6 to output a 2-belief strategy with memory $\mathcal{P}(\mathcal{L}' \times K) \setminus \emptyset$ and adds an initial memory state $\emptyset$ with nonzero transition probabilities to all other memory states including $\emptyset$ itself. □

Now we are done with preliminary results and we turn to the proof of Theorem 8.3.

Proof of Theorem 8.3. We start with $\mathcal{L}_0 = \emptyset$ and apply iteratively Proposition 8.7 in order to obtain a sequence $\mathcal{L}'_0, \mathcal{L}'_1, \ldots, \mathcal{L}'_M$ of disjoint nonempty sets of supports such that

- if $1 \leq m \leq M - 1$, then $\mathcal{L}_m = \mathcal{L}'_0 \cup \cdots \cup \mathcal{L}'_{m-1}$ matches case (ii) of Proposition 8.7, which defines a set $\mathcal{L}'$ and a strategy τ^* that we rename $\mathcal{L}'_{m+1}$ and τ^*_{m+1}, and
- $\mathcal{L}_M$ matches case (i) of Proposition 8.7.

Then, according to Proposition 8.7, 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.

The sequence $\mathcal{L}'_0, \mathcal{L}'_1, \ldots, \mathcal{L}'_M$ is computable in doubly exponential time, because each application of Proposition 8.7 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 2-belief strategy τ^+ for player 2. Strategy τ^+ consists of playing randomly any action as long as a coin gives the result "heads." When the coin gives the result "tails," 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. Intuitively, the strategy bets that the belief of player 1 is exactly L and that, from that moment on, for every step n player 1 will play actions in $\text{ISafe}_{\mathcal{L}_m}(\mathcal{B}_1^n)$. Since each strategy τ_m^* has memory $\mathcal{P}(\mathcal{L}'_m \times K)\backslash\{\emptyset\}$ and the $\mathcal{L}'_m$ are distinct, 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 behavioral strategy for player 1 and $L \in \mathcal{L}_M$. Let

$$m_0 = \min\{0 \leq m < M \mid \mathbb{P}_{\delta_L}^{\sigma,\tau^+}(\exists n \in \mathbb{N}, \mathcal{B}_1^n \in \mathcal{L}'_m) > 0\}.$$

By minimality of m_0,

$$\mathbb{P}_{\delta_L}^{\sigma,\tau^+}\left(\forall n, I_n \in \text{ISafe}_{\mathcal{L}_{m_0}}(\mathcal{B}_1^n)\right) = 1; \tag{40}$$

otherwise, since τ^+ may play any action at any moment, there would be n such that $\mathbb{P}_{\delta_L}^{\sigma,\tau^+}(\mathcal{B}_1^{n+1} \in \mathcal{L}_{m_0}) > 0$. Since $\mathcal{L}_{m_0} = \mathcal{L}'_0 \cup \ldots \cup \mathcal{L}'_{m_0-1}$, this would contradict the minimality of m_0.

By definition of m_0, there exists a finite play $p = k_0 i_0 j_0 c_1 d_1 \ldots k_n$ such that $\mathcal{B}_1(L, c_1 \cdots c_n) \in \mathcal{L}'_{m_0}$ and $\mathbb{P}_{\delta_L}^{\sigma,\tau^+}(P_n = p) > 0$. Denote $B = \mathcal{B}_1(L, c_1, \ldots, c_n)$. Since the game has observable actions, we can replace w.l.o.g. τ^+ by an equivalent behavioral strategy (Lemma 4.6). Thus, we can apply the shifting lemma (Lemma 7.2) to δ_L, σ, τ^+ and CoBüchi; hence,

$$\mathbb{P}_{\delta_L}^{\sigma,\tau^+}(\text{CoBüchi} \mid R) = \mathbb{P}_{\delta'}^{\sigma_{c_1\cdots c_n},\tau^+_{d_1\cdots d_n}}(\text{CoBüchi}), \tag{41}$$

with $R = \{C_1 \cdots C_n = c_1 \cdots c_n \wedge D_1 \cdots D_n = d_1 \cdots d_n\}$ and $\delta'(k) = \mathbb{P}_{\delta_L}^{\sigma,\tau^+}(K_n = k \mid R)$. We show

$$\text{supp}(\delta') = B. \tag{42}$$

Since whatever signals are received by player 2 there is positive probability that τ^+ plays any action at any step, then the property in Equation (6) of Lemma 7.1 implies $B = \{k \in K \mid \mathbb{P}_{\delta_L}^{\sigma,\tau^+}(K_n = k \mid C_1 \cdots C_n = c_1 \cdots c_n) > 0\}$. Moreover, again because τ^+ may play any action at any time whatever his signals, $\mathbb{P}_{\delta_L}^{\sigma,\tau^+}(K_n = k \mid C_1 \cdots C_n = c_1 \cdots c_n) > 0 \iff \mathbb{P}_{\delta_L}^{\sigma,\tau^+}(K_n = k \mid R) > 0$. This shows Equation (42). Since $\text{supp}(\delta') = B \in \mathcal{L}'_{m_0}$, and by choice of τ_{m_0}, which satisfies Equation (32) of Proposition 8.7, we get $\mathbb{P}_{\delta'}^{\sigma_{c_1\cdots c_n},\tau_{m_0}}\left(\text{CoBüchi} \mid \forall n, I_n \in \text{ISafe}_{\mathcal{L}_{m_0}}(\mathcal{B}_1^n)\right) > 0$. Thus, according to Equation (40), $\mathbb{P}_{\delta'}^{\sigma_{c_1\cdots c_n},\tau_{m_0}}(\text{CoBüchi}) > 0$. According to the definition of τ^+, there is positive probability that τ^+ is equivalent to τ_{m_0} after signals $d_1 \cdots d_n$ are received; thus,

$$\mathbb{P}_{\delta'}^{\sigma_{c_1\cdots c_n},\tau^+_{d_1\cdots d_n}}(\text{CoBüchi}) > 0.$$

Together with Equation (41), this last inequality implies $\mathbb{P}_{\delta_L}^{\sigma,\tau^+}(\text{CoBüchi} \mid R) > 0$. Moreover, $\{P_n = \pi\} \subseteq R$ and by choice of π, $\mathbb{P}_{\delta_L}^{\sigma,\tau^+}(P_n = \pi) > 0$; thus, $\mathbb{P}_{\delta_L}^{\sigma,\tau^+}(\text{CoBüchi}) > 0$. Since this holds for any behavioral strategy σ, the strategy τ^+ is positively winning from any $L \in \mathcal{L}_M$. □

9 LOWER BOUND ON MEMORY NEEDED BY STRATEGIES

In this section, we give the proof of the lower bound in Theorem 6.6, stating that doubly exponential memory is necessary to win positively. This lower bound holds for *both* finite-memory strategies with *randomized* updates and for finite-memory strategies with *deterministic* updates. This should be compared to the doubly exponential upper bound of Theorem 6.6, obtained for strategies for player 2 with randomized updates, built from the fix-point algorithm of the previous section.

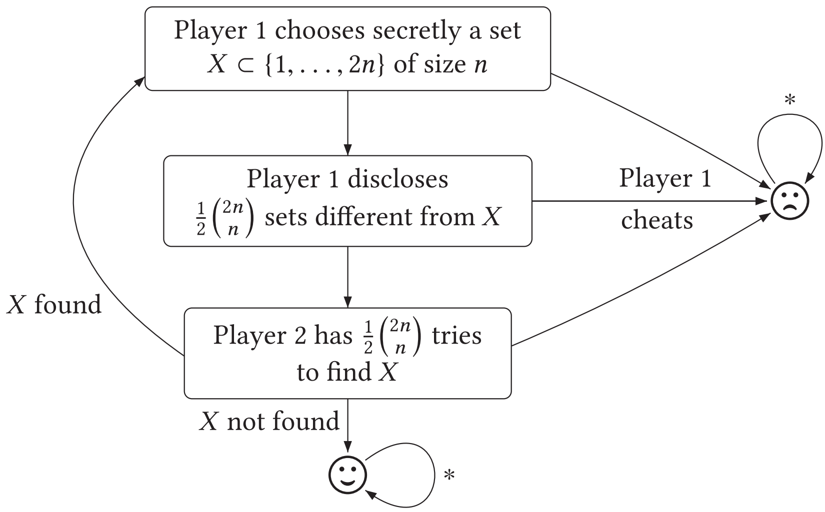

Fig. 10. Player 2 needs doubly exponential memory to avoid the ☺-state with positive probability.

9.1 Overview of the Proof

We show in this section that a doubly exponential memory is necessary to win positively safety (and hence co-Büchi) games.

To this aim, we construct, for each integer n, a reachability game of size polynomial in n. A high-level description of this game, called guess_my_set$_n$, is given in Figure 10. The objective of player 1 is to reach ☺, while player 2 has the dual objective of avoiding ☺. We will establish that player 2 wins positively, and that the memory of any positively winning strategy for player 2 is at least doubly exponential in n. These properties are postponed to Proposition 9.1.

Let us start by describing the high-level structure of guess_my_set$_n$ for a fixed $n \in \mathbb{N}$.

Idea of the game. The game guess_my_set$_n$ is divided into three phases, represented by blocks in Figure 10. In the first phase, player 1 chooses a set $X \subsetneq \{1, \ldots, 2n\}$ of size n. There are $\binom{2n}{n}$ possibilities of such sets X. Player 2 is blind in this phase and has no action to play.

In the second phase, player 1 discloses through her actions $\frac{1}{2}\binom{2n}{n}$ pairwise distinct sets of size n that are all different from X. Player 2 has no action to play in that phase, yet he observes the actions of player 1 and thus the sets disclosed by player 1.

In the third phase, player 2 aims at guessing X by trying up to $\frac{1}{2}\binom{2n}{n}$ sets of size n. Similarly to player 1 in phase 2, here player 2 discloses these sets by his actions. In this phase, player 1 has no action to play, yet she observes actions of her opponent. If player 2 succeeds in guessing X, the game restarts from the beginning. Otherwise, state ☺ is reached and player 1 wins.

In order for guess_my_set$_n$ to be of polynomial size in n, the various sets X and the ones disclosed by player 1 or tried by player 2 cannot be stored in the arena. A consequence of this is to allow player 1 to cheat: either in the first phase by picking a set of sizes not equal to n, or in the second phase by disclosing set X, or in the third phase by pretending player 2 did not guess X. To prevent player 1 from cheating, we rely on probabilities and store in the state of the game short random information, for example, one element of X as opposed to the whole set. Therefore, if player 1 cheats, she will be caught with positive probability, yielding to a sink losing state ☹. To win almost surely, player 1 will thus have to play according to the rules. Our concise encoding, however, will not allow player 2 to cheat. Notice that player 1 is better informed than player 2 in this game.

Concise encoding. Let us explain in more detail the encoding of the game guess_my_set$_n$ to justify that its number of states is polynomial in n. There are three issues to be addressed. First, storing set X in the state of the game would require exponentially many states. Instead, we use a fairly standard technique: store a single element $x \in X$ at random. In order to check that a set Y of size n, disclosed by player 1, is different from X, we challenge player 1 to pinpoint an element $y \in Y \setminus X$. We ensure by construction that $y \in Y$: player 1 has to pinpoint y when disclosing Y. If player 1 cheats in phase 2, she pinpoints $y \in X$, and with positive probability $y = x$, in which case the game moves to ☹ and player 1 loses. The second issue is to make sure that player 1 discloses an exponential number of pairwise different sets $X_1, X_2, \ldots, X_{\frac{1}{2}\binom{2n}{n}}$, while the game cannot store even one of these sets. Instead, player 1 will disclose the sets in some total order, denoted $<$. Thus, it will suffice to check only one inequality each time a set X_{i+1} is given, namely, $X_i < X_{i+1}$. The precise encoding is more involved than with the previous issue but relies on similar ideas (see Section 9.3).

The last issue is to count up to $\frac{1}{2} \cdot \binom{2n}{n}$, with a logarithmic number of bits, to check that that number of sets have been disclosed by player 1 or tried by player 2. Here again, we ask player 1 to increment a counter, while storing only one of the bits. If she is caught cheating when incrementing the counter, the game moves to ☹ (see Section 9.2).

Now that we gave a high-level description of the game, we can state its properties:

Proposition 9.1. *Player 2 has a positively winning finite-memory strategy with deterministic updates with* $3 \times 2^{\frac{1}{2}\cdot\binom{2n}{n}}$ *different memory states in the game* guess_my_set$_n$.

No finite-memory strategy with randomised updates of player 2 with less than $2^{\frac{1}{2}\cdot\binom{2n}{n}}$ *memory states wins positively* guess_my_set$_n$.

Proof. Let us describe a positively winning strategy for player 2 with at most $3 \times 2^{\frac{1}{2}\cdot\binom{2n}{n}}$ memory states. First of all, player 2 remembers in which the phase the game is (three different possibilities). In phase 2, player 2 remembers all the sets disclosed by player 1 ($2^{\frac{1}{2}\cdot\binom{2n}{n}}$ possibilities). Between phase 2 and phase 3, he reverses his memory to remember the sets player 1 did not disclose (still $2^{\frac{1}{2}\cdot\binom{2n}{n}}$ possibilities). Then he tries each of these sets, one by one, in phase 3, deleting the set from his memory after he tried it.

Let us assume first that player 1 does not cheat. Then each set of size n is either disclosed by player 1 or tried by player 2, since there are $\binom{2n}{n}$ such sets. As a consequence, X has been found, and the game starts another round, and avoids ☺. Otherwise, if player 1 cheats at some point, there is a positive probability to reach the losing state ☹, and player 2 also wins *positively* his safety objective.

To show the second claim, assume by contradiction that there exists a positively winning finite-memory strategy τ for player 2 that has less than $2^{\frac{1}{2}\cdot\binom{2n}{n}}$ memory states. We build a counter strategy σ to τ for player 1. Note that σ shall not cheat, or else the game would enter the sink losing state with positive probability. Strategy σ actually takes all its decisions at random: it chooses the secret set X at random in phase 1; then in phase 2, it chooses pairwise distinct sets $Y \neq X$ uniformly at random and discloses them following the total order. At the end of phase 2 for each round of the game, player 1 has disclosed a family $\mathcal{A}$ of $\frac{1}{2} \cdot \binom{2n}{n}$ sets of size n. The distribution over memory states of player 2 at that moment only depends on $\mathcal{A}$ and on the distribution of his memory state at the beginning of the round. Let us fix a round (thus the initial distribution over memory states of τ is fixed) and denote by $m_{\mathcal{A}}$ the distribution of memory states of player 2 after $\mathcal{A}$ has been disclosed. As τ has less than $2^{\frac{1}{2}\cdot\binom{2n}{n}}$ memory states, there exists at least one memory state $\mathfrak{m}$ and two families $\mathcal{B} \neq \mathcal{C}$ such that $m_{\mathcal{B}}(\mathfrak{m}) \neq 0$ and $m_{\mathcal{C}}(\mathfrak{m}) \neq 0$. Let $\overline{\mathcal{B}}$ ($\overline{\mathcal{C}}$, respectively) be the complement of $\mathcal{B}$ ($\mathcal{C}$, respectively) among the set of sets of n elements. Since $\mathcal{B} \neq \mathcal{C}$, $\overline{\mathcal{B}} \cup \overline{\mathcal{C}}$ has strictly more than $\frac{1}{2} \cdot \binom{2n}{n}$ sets of n elements. Hence, there exists a set $Y \in \overline{\mathcal{B}} \cup \overline{\mathcal{C}}$ that is tried by player 2 with probability less than 1 after memory state $\mathfrak{m}$. Without loss of generality, we can assume that $Y \notin \mathcal{B}$ (the case in which $Y \notin \mathcal{C}$ is symmetrical). The probability is nonzero that player 1 chose set Y in the first phase of that round and discloses $\mathcal{B}$. Hence, there is a nonzero probability that player 2 does not try set Y in phase 3, in which case ☺ is reached in that round. More precisely, there is a uniform lower bound $p > 0$ on the probability to reach ☺ at each round. As it is true for every round, almost surely ☺ is reached, a contradiction with the fact that τ is positively winning. Thus, no finite-memory strategy for player 2 with less than $2^{\frac{1}{2} \cdot \binom{2n}{n}}$ memory states can be positively winning. □

9.2 Concise Encoding of Exponentially Many Steps

As a first step to the formal definition of guess_my_set$_n$, we explain how to concisely count up to a number exponential in n with only a number of states polynomial in n.

Let $y_1 \cdots y_n$ be the binary encoding of a number y exponential in n, where y_n is the parity of y. We describe a single-player reachability game, in which the player surely wins if the game lasts for $n \cdot y$ steps. Intuitively, the player increments a counter from 0 to $y_1 \cdots y_n$. For a counter value x, let $x'_1 \cdots x'_n$ be the binary encoding of $x' = x + 1$. In order to check that the player does not cheat in the incrementation step, some bit x'_i for a random i is stored in the game state and hidden to the player. The value of x'_i can easily be computed on the fly while reading $x_i \ldots x_n$: indeed, $x'_i = x_i$ if and only if there exists some $k > i$ with $x_k = 0$.

In this game, the set of signals is the same as the set of actions, namely, $\{0, 1, 2\}$. Actions $a \in \{0, 1\}$ stand for the value of bits, while $a = 2$ represents that the player claims to have reached y. The state space is the following: $\{(i, b, j, b', j', c) \mid (i, j, j') \in \{1, \ldots, n\}^3, (b, b', c) \in \{0, 1\}^3\}$. The intuition of such a state is that the player will play action a_i corresponding to bit x_i, while b, j is the check to make to the current number (checking that $x_j = b$), b', j' is 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 action 2, claiming that y has been reached, then if $y_j \neq b$, the player is caught cheating (since the current counter value is certainly not y), and the game moves to a losing sink state ☹. Otherwise, when $y_j = b$, the game moves to the goal state ☺. Thus, there is a transition in the arena with $p((i, b, j, b', j', c), a, ☹) = 1$ if $i = j$ and $a \neq b$, corresponding to the player being caught cheating. Otherwise, if $i \neq n$, the stochastic transitions are

- the current bit a at position i may be checked for the successor x' of x: $p((i, b, j, b', j', c), a, (i + 1, b, j, a, i, 1)) = 1/2$ (carry initialised at 1), and
- the current bit will not be checked: $p((i, b, j, b', j', c), a, (i + 1, b, j, b', j', c \wedge a)) = \frac{1}{2}$ (the carry is 1 if both c and a are 1).

Last, for $i = n$, there is a transition $p((i, b, j, b', j', c), a, (1, b' \wedge c, j', a, 1, 1)) = 1$ (the bit of the next number becomes the bit for the current configuration, taking care of the carry c). Clearly, if the game does not last $n \cdot y$ steps, then the player did not faithfully encode the counter increment at some step, and she has a chance to get caught and lose, so that the probability to reach ☺ is less than 1.

9.3 Implementing guess_my_set$_n$ with a Polynomial Size Game

We finally turn to the formal definition of the game guess_my_set$_n$, with a number of states polynomial in n. Recall that player 1 has a reachability objective, namely, the target state ☺.

In the first phase of each round, player 1 chooses a set X of n elements in $\{1, \cdots 2n\}$. Formally, each number from 1 to $2n$ is called in increasing order, and player 1 has two actions, "yes" or "no," to define the set X. She has to play "yes" for exactly n numbers. The states of that phase of the game are of the form (x, i, r), where x is the number currently called, i counts the number of "yes" actions so far, and r is some element for which player 1 played "yes," which the system stores, and which is hidden to both players. Signals of player 1 coincide with her actions. Player 2 does not participate in this phase: his actions have no effect on the state and he receives always the same dummy signal whatever happens.

Formally, whenever player 1 plays "yes" for a number x, there are two stochastic transitions $p((x, i, r), yes, (x + 1, i + 1, x)) = 1/2$ and $p((x, i, r), yes, (x + 1, i + 1, r) = 1/2$. In both cases, x is selected as the $i + 1$-th number in set X, the current size of X is increased by 1, and the next number called is $x + 1$. In the former case, the randomly stored number r is updated to x, while in the latter case r is not updated. The stored number r at the end of phase 1 will be used in the other phases of this round. If player 1 plays action "no" for a number x, this triggers the transition $p((x, i, r), no, (x + 1, i, r)) = 1$.

State $(2n + 1, i, rx)$ with $i \neq n$ encodes that player 1 did not select with "yes" actions exactly n numbers, and the game moves directly to the sink losing state ☹.

In the second phase, player 1 discloses $\frac{1}{2} \cdot \binom{2n}{n}$ distinct sets Y of size n, all different of X. In order to be sure that every set Y she proposes is different from X, player 1 is asked to pinpoint an element in $Y \setminus X$. This number is not visible to player 2. In case player 1 pinpoints y equal to the stored number r, which belongs to X by construction, then she is caught cheating, and the game moves to the sink losing state ☹. Since player 1 does not know the number r, pinpointing any number in X is risky as it makes her lose with fixed positive probability.

To force player 1 to disclose distinct sets Y, she enumerates them in lexicographic order $<$. Formally, $Y < Y'$ if there exists a position i such that the $i - 1$-th smallest numbers of Y and of Y' agree, and the ith smallest number y of Y is less than the ith smallest number of Y'. The number y is called the distinguishing number between Y and Y'. To check that player 1 generates sets in lexicographic order, when disclosing Y, player 1 must announce which is the distinguishing number y between Y and Y', for Y' the next disclosed set. The actions of player 1 relevant to ensure that sets are enumerated in lexicographic order and distinct are thus $\{yesdif, yes, no\}$. Similarly to the first phase, one such action is played for each number in $\{1, \ldots 2n\}$ when they are called. Intuitively, $yesdif$ represents that y, the number being called, belongs to Y and is the distinguishing number between Y and the next set Y'. Otherwise, yes represents that $y \in Y$ and no stands for $y \notin Y$.

Formally, the part of the states relevant to ensure that the disclosed sets are distinct is of the form $(y, i, pry, py, pi, nry, ny, ni)$, with:

- y the number being called,
- i the number of yeses played by player 1 so far for the current set Y',
- py, pi the distinguishing number and its position as announced for the previous set Y,
- pry a random number in Y and less than py (hidden to both players),
- ny, ni the distinguishing number and its position for the current set Y' (initially 0,0), and
- nry a random number in Y less than ny (hidden to both players).

The stored numbers (py, pi, pry) yield a chance to catch player 1 if she does not generate Y in lexicographic order, in which case the game is sent to the sink state ☹. First, the game checks that the pi-th number in Y is larger than py. Also, the game checks that pry belongs to the current set. Hence, player 1 needs to ensure that the first $pry - 1$ elements of the previous set and of the current set agree, and that the the pi-th number in Y is larger than py; that is, Y is larger lexicographically than the previous set. Otherwise, she has a fixed positive probability to be caught, and she will lose the game in ☹.

In this phase, player 2 has no actions to play. The signals of player 1 and 2 are the same as the actions of player 1; that is, player 2 is informed of the sets disclosed by player 1.

Formally, the transitions depend on whether player 1 is caught cheating using (y, i, pry, py, pi) or not. She is caught cheating using (y, i, pry, py, pi) with the following transitions, with $yes* = yes$ or $yes* = yesdif$:

- $p((y, i, pry, py, pi), yes*, ☹) = 1$ for $y = py$,
- $p((y, i, pry, py, pi), yes*, ☹) = 1$ for $i = pi - 1$ and $y \geq py$.

Otherwise, we have the following transitions:

- $p((y, i, nry, ny, ni), a, ☹) = 1$ for $ny \neq 0$ and $a = yesdif$, corresponding to player 1 announcing two distinguishing numbers, or for $y = 2n$, $ny = 0$, and $a \neq yesdif$, corresponding to no distinguishing numbers. Otherwise,
- for $a = no$, $p((y, i, nry, ny, ni), no, (y + 1, i, nry, ny, ni)) = 1$, because when y does not belong to the current set, nothing has to be updated and the next number $y + 1$ is called.
- for $a = yesdif$, $p((y, i, nry, ny, ni), yesdif, (y + 1, i + 1, nry, ny', ni')) = 1$ with $ny' = y$ and $ni' = i$, because player 1 just pinpointed the number called with $yesdif$,
- for $a = yes$ and $ny > 0$, meaning that the distinguishing number was already pinpointed, $p((y, i, nry, ny, ni), yes, (y + 1, i + 1, nry, ny, ni)) = 1$,
- else, $a = yes$ and $ny = 0$, meaning that no distinguishing number was pinpointed yet, and then we have $p((y, i, nry, ny, ni), a, (y + 1, i + 1, nry, ny, ni)) = 1/2$ and $p((y, i, nry, ny, ni), a, (y + 1, i + 1, nry', ny, ni)) = 1/2$ with $nry' = y$. This corresponds to the fact that $y \in Y$ and y is less than the distinguishing number, so that it can be randomly chosen to be stored in nry.

Last, to ensure that player 1 discloses $\frac{1}{2} \cdot \binom{2n}{n}$ sets, she has to encode the increment of a counter, as described in Section 9.2, where the counter is incremented exactly when a set Y is disclosed. When she has given $\frac{1}{2} \cdot \binom{2n}{n}$ sets, the game proceeds to the third phase.

In the third phase, player 2 has at most $\frac{1}{2} \cdot \binom{2n}{n}$ tries to guess the set X chosen by player 1 in the first phase. To do so, player 2 tries $\frac{1}{2} \cdot \binom{2n}{n}$ sets of size n, and player 1 observes these sets. For each set Y tried by player 2, player 1 has to announce a witness in $Y \setminus X$, which is not observed by player 2. Similarly to phases 1 and 2, numbers in $\{1, \dots 2n\}$ are called in increasing order, and player 2 plays yes or no to tell whether they belong to Y. Just after player 2 announces that y is in the guessed set Y, player 1 can announce secretly that "$y \in Y \setminus X$." Player 1 is caught cheating if y coincides with r, the stored number from the first phase. If $Y = X$, player 1 cannot announce $y \in Y \setminus X$ without having a chance to be caught. Instead, she can play a reset action to restart the game in its first phase. Note that player 1 only has an incentive to play that reset action in case $Y \neq X$, since otherwise, she would rather select $y \in Y \setminus X$. After each set tried by player 2, the counter, as described in Section 9.2, is incremented. When $\frac{1}{2} \cdot \binom{2n}{n}$ sets $Y \neq X$ have been tried by player 2 without guessing X, the game moves to winning state ☺, and player 1 wins.

10 COMPLEXITY LOWER BOUND AND SPECIAL CASES

In this section, we show that our 2EXPTIME algorithms are optimal regarding complexity. Furthermore, we show that these algorithms enjoy better complexity in restricted cases. In particular, we generaliZe a result of [11, 30], extending EXPTIME complexity to a larger subclass of systems with particular signaling structures, as described in Section 10.2.

10.1 Complexity Lower Bound for Reachability and Büchi Games

We prove here that the problem of knowing whether the initial support of a reachability game or a Büchi 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 10.1. *In a reachability or Büchi game, deciding whether player* 1 *has an almost-surely winning strategy is 2EXPTIME-hard, even if player 1 is more informed than player* 2.

We provide a proof for reachability games. The lower bound of course extends to Büchi games since any reachability game can be turned into an equivalent Büchi one by making target states absorbing.

Proof. We do a reduction from the membership problem for EXPSPACE alternating Turing machines. Let $\mathcal{M}$ be an EXPSPACE alternating Turing machine and w be an input word of length n. From $\mathcal{M}$ and w 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, he enumerates the tape contents of successive configurations. Moreover, he 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. Both players will be able to deviate from these rules, but then they will have a nonzero probability to be caught cheating, immediately ending the game in a state where the other player wins. In this game, if player 2 implements some execution of $\mathcal{M}$ on w without cheating, 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.

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. He 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 him 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, a position k of the tape is chosen at random and is revealed to player 1 as a sequence of n bits. Player 2 is not told anything about k. The game stores 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 should announce to the game that position k has been reached again (player 1 can cheat by announcing a different $k' \neq k$, but we will first assume it is not the case). The game checks that the letter player 2 gives is indeed a. This way, each time player 2 cheats, the game has a fixed positive probability to detect it. If so, the game goes to a sink state that is winning for player 1.

On top of that, player 1 has the possibility to reset the whole execution whenever she wants and restart a fresh computation.

Assuming that w is accepted by $\mathcal{M}$, we show that player 1 wins almost surely. Consider a strategy where player 1 does not cheat, plays a strategy ensuring that the computation on w is accepting, and resets as soon as player 2 cheats and the system does not detect it. There are two kinds of plays: those where player 2 plays fair during at least one whole computation, and those where player 2 cheats at least once after each reset. In the first case, the computation on w terminates in an accepting state. In the second case, player 2 gets caught almost surely: each time player 2 cheats, there is probability at least $\frac{1}{2^n}$ that player 2 gets caught (the probability that k chosen by the system is the cheating position). In both cases, player 1 wins.

We now have to take into account that player 1 could cheat: she could call to a position different from k in the next step. To avoid this kind of behavior, or at least contain it, a piece of information about the position pointed to by player 1 is kept secret (to both players) in the state of the game. More precisely, a bit b of the binary encoding of k is randomly chosen among the at most n possible bits, and the bit and its position are remembered. If player 1 is caught cheating (i.e., if the bits at the position remembered differ between both steps), 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.

Assume that w is not accepted by $\mathcal{M}$; we show that player 2 wins positively. For that player 2 plays a strategy not cheating and ensuring w is not accepted by $\mathcal{M}$. Either player 1 cheats and has probably <1 to win or she does not cheat and has probability 0 to win.

Finally, w is accepted by $\mathcal{M}$ if and only if player 1 has an almost-surely winning strategy to reach the goal state.

Notice that the game is stochastic (a bit and a position are remembered randomly in states of the game); player 1 is not perfectly informed about the state (she does not know which bit is remembered in the state), but she is better informed than player 2 (the latter does not know what letter player 1 decided to memorize). □

Finally, let us comment on the almost-surely winning strategy of player 1 built in the proof. This strategy requires a doubly exponential number of memory states for detecting whenever player 2 is cheating. This contrasts with our exponential upper bound on the memory needed by player 1 to win almost surely. Actually, player 1 has a simpler strategy to win almost surely: it is enough for her to play almost totally randomly. Resets are triggered randomly and the existential choices of the computation of the machine are also performed randomly. Only one thing should be done with care by player 1: she should remember exactly the value of the position k in order to announce it accurately when the next configuration occurs. This requires exponentially many memory states.

10.2 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. In [11], it is shown that this problem is EXPTIME-complete when player 2 is perfectly informed. The following proposition shows that actually, player 2 being better informed than player 1 is a sufficient condition for the complexity to drop from 2EXPTIME to EXPTIME. Orthogonally, player 1 being perfectly informed about the state is also sufficient to obtain EXPTIME complexity.

Proposition 10.2. *In a Büchi game where either player* 2 *is better informed than player* 1 *or player* 1 *is perfectly informed about the state, deciding whether player* 1 *has an almost-surely winning strategy can be done in exponential time.*

Proof. The reason for the single EXPTIME complexity in these special cases is that in both cases, there are at most an exponential number of 2-beliefs for player 2. If player 1 is perfectly informed about the state, then the belief of player 1 is a singleton$\{k\}$. Thus, the 2-belief of player 2 is a collection of pairs $(k, \{k\})$ with $k \in K$. There is an exponential number of such 2-beliefs.

If player 2 is better informed than player 1, then at every moment player 2 can compute exactly the belief $\mathcal{B}_1$ of player 1. Thus, the 2-belief of player 2 is a collection $\{(k, \mathcal{B}_1), k \in K'\}$ with $K' \subseteq K$

(actually $K' \subseteq \mathcal{B}_1$). For a given value of $\mathcal{B}_1$ there are at most $2^{|K|}$ such collections and thus in total there are less than $2^{2|K|}$ possible 2-beliefs. □

Note that the latter proposition does not hold when player 1 is better informed than player 2. Indeed, in the game presented for the lower bound, in the proof of Theorem 10.1, player 1 is better informed than player 2 (yet player 1 is not perfectly informed about the state).

11 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. Further, we showed that both the memory and the algorithmic complexities are tight.

Changing the notion of reaching a Büchi objective with positive probability for the notion where the frequency at which a target state is visited does not converge toward 0 leads to decidability of the emptiness problem of probabilistic finite automaton [36]. It would be interesting to extend this result to stochastic games with signals.

ACKNOWLEDGMENTS

We would like to thank the anonymous referees for their much valuable comments which allowed us to significantly improve this paper. In particular, one of the referees found a flaw in an earlier version, which led us to establish the equivalence between games with non-observable actions and general strategies and games with observable actions and behavioral strategies.

REFERENCES

[1] R. J. Aumann. 1964. Mixed and behaviour strategies in infinite extensive games. 1964. In *Advances in Game Theory, Annals of Mathematics Studies*, Dresher, Shapley & Tucker (Eds.). vol. 52, Princeton University Press, 627–650.

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

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

[4] N. Bertrand, B. Genest, and H. Gimbert. 2009. Qualitative determinacy and decidability of stochastic games with signals. In *LICS*. IEEE Computer Society, 319–328.

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

[6] T. Brázdil, V. Brozek, A. Kucera, and J. Obdrzálek. 2011. Qualitative reachability in stochastic BPA games. *Information and Computation* 209, 8 (2011), 1160–1183. DOI: http://dx.doi.org/10.1016/j.ic.2011.02.002

[7] K. Chatterjee, L. de Alfaro, and T. A. Henzinger. 2005. The complexity of stochastic rabin and streett games. In *Proc. of ICALP'05 (LNCS)*, Vol. 3580. Springer, 878–890.

[8] K. Chatterjee and L. Doyen. 2012. Partial-observation stochastic games: How to win when belief fails. In *Proc. of LICS'12*. IEEE, 175–184. Long version available at http://arxiv.org/abs/1107.2141.

[9] K. Chatterjee, L. Doyen, H. Gimbert, and T. A. Henzinger. 2010. Randomness for free. In *Proc. of MFCS'10 (LNCS)*, Vol. 6281. Springer, 246–257.

[10] K. Chatterjee, L. Doyen, and T. A. Henzinger. 2013. A survey of partial-observation stochastic parity games. *Formal Methods in System Design* 43, 2 (2013), 268–284.

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

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

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

[14] J. Cristau, C. David, and F. Horn. 2010. How do we remember the past in randomised strategies? In *Proceedings First Symposium on Games, Automata, Logic, and Formal Verification (GANDALF'10)*. 30–39. DOI: http://dx.doi.org/10.4204/EPTCS.25.7

[15] L. de Alfaro and T. A. Henzinger. 2000. Concurrent omega-regular games. In *Proc. of LICS'00*. IEEE, 141–154.
[16] L. de Alfaro, T. A. Henzinger, and O. Kupferman. 2007. Concurrent reachability games. *Theoretical Computer Science* 386, 3 (2007), 188–217.
[17] L. de Alfaro and R. Majumdar. 2001. Quantitative solution of omega-regular games. In *Proc. of STOC'01*. ACM, 675–683.
[18] Rick Durrett. 2010. *Probability: Theory and Examples*, (4th ed.). Cambridge University Press.
[19] H. Gimbert and F. Horn. 2008. Simple stochastic games with few random vertices are easy to solve. In *Proc. of FOSSACS'08 (LNCS)*, Vol. 4972. Springer, 5–19.
[20] H. Gimbert and Y. Oualhadj. 2010. Probabilistic automata on finite words: Decidable and undecidable problems. In *Proc. of the 37th International Colloquium on Automata, Languages and Programming (ICALP'10) (Lecture Notes in Computer Science)*, Vol. 6199. Springer, 527–538.
[21] H. Gimbert, J. Renault, S. Sorin, X. Venel, and W. Zielonka. 2016. On the values of repeated games with signals. *Annals of Applied Probability* 26, 1 (February2016), 402–424.
[22] E. Grädel, W. Thomas, and T. Wilke. 2002. *Automata, Logics and Infinite Games. LNCS*, Vol. 2500. Springer.
[23] V. Gripon and O. Serre. 2009. Qualitative concurrent stochastic games with imperfect information. In *Proceedings of the 36th International Colloquium on Automata, Languages and Programming (ICALP'09) (Lecture Notes in Computer Science)*, Vol. 5556. Springer, 200–211.
[24] V. Gripon and O. Serre. 2011. Qualitative concurrent stochastic games with imperfect information. *CoRR* abs/0902.2108 (2011).
[25] F. Horn. 2008. *Random Games*. PhD Dissertation. Université Denis-Diderot.
[26] D. A. Martin. 1998. The determinacy of Blackwell games. *Journal of Symbolic Logic* 63, 4 (1998), 1565–1581. DOI: http://dx.doi.org/10.2307/2586667
[27] J.-F. Mertens and A. Neyman. 1982. Stochastic games have a value. *Proceedings of the National Academy of Sciences USA*, 79 (1982), 2145–2146.
[28] A. Paz. 1971. *Introduction to Probabilistic Automata*. Academic Press.
[29] M. O. Rabin. 1963. Probabilistic automata. *Information and Control* 6, 3 (1963), 230–245.
[30] J. H. Reif. 1979. Universal games of incomplete information. In *Proc. of STOC'79*. ACM, 288–308.
[31] J. Renault. 2007. *The Value of Repeated Games with an Informed Controller*. Technical Report. CEREMADE, Paris.
[32] J. Renault and S. Sorin. 2008. *Personal Communications*. (2008).
[33] D. Rosenberg, E. Solan, and N. Vieille. 2003. *Stochastic Games with Imperfect Monitoring*. Technical Report 1376. Northwestern University.
[34] L. S. Shapley. 1953. Stochastic games. *Proceedings of the National Academy of Sciences USA*, 39 (1953), 1095–1100.
[35] S. Sorin. 2002. *A First Course on Zero-Sum Repeated Games*. Springer.
[36] M. Tracol. 2011. Recurrence and transience for finite probabilistic tables. *Theoretical Computer Science* 412, 12–14 (2011), 1154–1168.
[37] J. von Neumann and O. Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.

Received December 2013; revised December 2016; accepted June 2017