Games with Imperfect Information: Theory and Algorithms ⋆

Laurent Doyen[1] and Jean-François Raskin[2]

[1] LSV, ENS Cachan & CNRS, France
[2] Université Libre de Bruxelles (ULB), Belgium

Abstract. We study observation-based strategies for two-player turn-based games played on graphs with parity objectives. An observation-based strategy relies on imperfect information about the history of a play, namely, on the past sequence of observations. Such games occur in the synthesis of a controller that does not see the private state of the plant. Our main results are twofold. First, we give a fixed-point algorithm for computing the set of states from which a player can win with a deterministic observation-based strategy for a parity objective. Second, we give an algorithm for computing the set of states from which a player can win with probability 1 with a randomized observation-based strategy for a reachability objective. This set is of interest because in the absence of perfect information, randomized strategies are more powerful than deterministic ones.

1 Introduction

Games are natural models for reactive systems. We consider zero-sum two-player turn-based games of infinite duration played on finite graphs. One player represents a control program, and the second player represents its environment. The graph describes the possible interactions of the system, and the game is of infinite duration because reactive systems are usually not expected to terminate. In the simplest setting, the game is turn-based and with perfect information, meaning that the players have full knowledge of both the game structure and the sequence of moves played by the adversary. The winning condition in a zero-sum graph game is defined by a set of plays that the first player aims to enforce, and that the second player aims to avoid. We focus on ω-regular sets of plays expressed by the parity condition (see Section 2) and we briefly present properties and algorithmic solutions for such games. The theory and algorithms for games with perfect information has been extensively studied [Mar75,EJ91,Tho95,Tho02,Hen07].

Turn-based games of perfect information make the strong assumption that the players can observe the state of the game and the previous moves before playing. This is however unrealistic in the design of reactive systems because the

⋆ This research was supported in part by the European Quasimodo and Gasics projects, by the PAI program Moves funded by the Belgian Federal Government, and by the CFV (Federated Center in Verification) funded by the F.R.S.-FNRS.

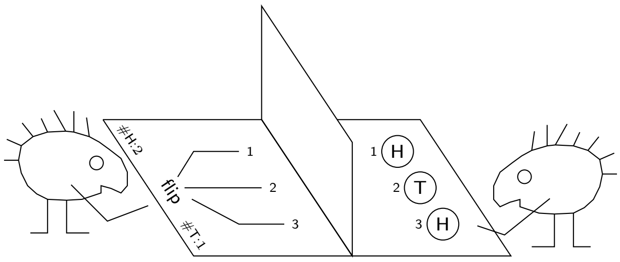

Fig. 1. The 3-COIN game.

components of a system have an internal state that is not visible to the other components, and because their execution is concurrent, each component choosing moves independently of the others. Such situations require to introduce games with *imperfect information* where the players have partial information about the play. We illustrate the games with imperfect information with the 3-COIN game, shown in Fig. 1.

> Three coins c_1, c_2, c_3 are arranged on a table, either *head* or *tail* up. Player 1 does not see the coins, but he is informed of the number of heads (H) and tails (T). The coins are manipulated by Player 2. The objective of Player 1 is to have all coins head up (HHH) while avoiding at all cost a configuration where all coins show tail (TTT). The game is played as follows. Initially, Player 2 chooses a configuration of the coins with two heads and one tails. Then, the following rounds are played: Player 1 can choose one coin in the set $\{c_1, c_2, c_3\}$ and ask Player 2 to toggle that coin. Player 2 must execute the choice of Player 1 and he may further decide to exchange the positions of the two other coins. The game stops whenever the three coins are all head up (Player 1 wins) or all tail up (Player 2 wins). Otherwise Player 2 announces the number of heads and tails, and the next round starts.

This is a game with imperfect information for Player 1 as she does not know the exact position of the coins, but only the number of heads and tails. In this game, does Player 1 have a strategy such that for all strategies of Player 2, the game reaches HHH and avoids TTT? We are interested in *observation-based strategies* which rely on the information available to Player 1. In fact, Player 1 has no deterministic observation-based strategy to win the 3-COIN game, because Player 2 can always find a spoiling counter-strategy using his ability to exchange coins after Player 1's move. If we do not allow Player 2 to exchange the coins, then Player 1 has a deterministic observation-based winning strategy consisting in successively trying to toggle every coin. This strategy requires *memory* and it is easy to see that memory is necessary to win this game. On the other hand, if we allow Player 1 to take his decision using a source of randomization, then she would be able to win the original 3-COIN game with probability 1. This shows that *randomized* strategies are in general more powerful than deterministic strategies.

We study in this course mathematical models and algorithms for games with imperfect information. The model that we consider is asymmetric in the sense that Player 1 has imperfect information about the state while Player 2 has perfect knowledge [Rei84,CDHR07,DDR06]. This model is useful for the design of control programs embedded in an environment that provides observations about its state via shared variables or sensors. We discuss the asymmetry of the definition in Section 3.1 and we argue that the existence of deterministic winning strategies for the Player 1 does not depend on the ability or not for Player 2 to see the exact position of the game. In the rest of Section 3, we present the theory and algorithms to decide the existence of observation-based winning strategies. We use a reduction of games with imperfect information to games with perfect information, and we exploit the structure of this reduction to obtain a tailored data-structure and symbolic algorithms. We focus on *reachability* and *safety* objectives which ask Player 1 to respectively reach and avoid a designated set of target configurations. For parity objectives, we choose to provide a reduction to safety games. We also briefly present algorithms to construct winning strategies.

In Section 4, we introduce randomized observation-based strategies and we present an algorithmic solution for reachability and Büchi objectives. The algorithm computes the set of winning positions of the game and constructs a randomized observation-based winning strategy.

2 Games with perfect information

Game graphs A *game graph* is a tuple $G = \langle L, l_I, \Sigma, \Delta \rangle$, where L is a finite set of states, $l_I \in L$ is the initial state, Σ is a finite alphabet of actions, and $\Delta \subseteq L \times \Sigma \times L$ is a set of labeled transitions. We require the game graph G to be *total* i.e., for all $\ell \in L$ and all $\sigma \in \Sigma$, there exists $\ell' \in L$ such that $(\ell, \sigma, \ell') \in \Delta$.

The turn-based game on G is played by two players for infinitely many rounds. The first round starts in the initial location l_I of the game graph. In each round, if the current location is ℓ, Player 1 chooses an action $\sigma \in \Sigma$, and then Player 2 chooses a location ℓ' such that $(\ell, \sigma, \ell') \in \Delta$. The next round starts in ℓ'.

Plays and strategies A *play* in G is an infinite sequence $\pi = \ell_0 \ell_1 \dots$ such that $\ell_0 = l_I$, and for all $i \geq 0$, there exists $\sigma_i \in \Sigma$ such that $(\ell_i, \sigma_i, \ell_{i+1}) \in \Delta$. We denote by $\mathsf{Inf}(\pi)$ the set of locations that occur infinitely often in π. A *history* is a finite prefix $\pi(i) = \ell_0 \dots \ell_i$ of a play, and its *length* is $|\pi(i)| = i$. We denote by $\mathsf{Last}(\pi(i)) = \ell_i$ the last location in $\pi(i)$.

A *deterministic strategy* in G for Player 1 is a function $\alpha : L^+ \to \Sigma$ that maps histories to actions, and for Player 2 it is a function $\beta : L^+ \times \Sigma \to L$ such that for all $\pi \in L^+$ and all $\sigma \in \Sigma$, we have $(\mathsf{Last}(\pi), \sigma, \beta(\pi, \sigma)) \in \Delta$. We denote by $\mathcal{A}_G$ and $\mathcal{B}_G$ the set of all Player 1 and Player 2 strategies in G, respectively.

A strategy $\alpha \in \mathcal{A}_G$ is *memoryless* if $\mathsf{Last}(\pi) = \mathsf{Last}(\pi')$ implies $\alpha(\pi) = \alpha(\pi')$ for all $\pi, \pi' \in L^+$, that is the strategy only depends on the last location of the history. We define memoryless strategies for Player 2 analogously.

The *outcome* of deterministic strategies α (for Player 1) and β (for Player 2) in G is the play $\pi = \ell_0 \ell_1 \dots$ such that $\sigma_i = \alpha(\pi(i))$ and $\ell_{i+1} = \beta(\pi(i), \sigma_i)$ for all $i \geq 0$. This play is denoted $\mathsf{outcome}(G, \alpha, \beta)$. A play π is *consistent* with a deterministic strategy α for Player 1 if $\pi = \mathsf{outcome}(G, \alpha, \beta)$ for some deterministic strategy β for Player 2. We denote by $\mathsf{Outcome}_1(G, \alpha)$ the set of plays that are consistent with α. Plays that are consistent with a deterministic strategy for Player 2 and the set $\mathsf{Outcome}_2(G, \beta)$ are defined analogously.

Objectives An *objective* for a game graph $G = \langle L, l_I, \Sigma, \Delta \rangle$ is a set $\varphi \subseteq L^\omega$. We denote by $\overline{\varphi} = L^\omega \setminus \varphi$ the complement of φ. A deterministic strategy α for Player 1 (resp. β for Player 2) is *surely-winning* for an objective φ in G if $\mathsf{Outcome}_1(G, \alpha) \subseteq \varphi$ (resp. if $\mathsf{Outcome}_2(G, \beta) \subseteq \varphi$). We consider the following objectives:

- *Reachability and safety objectives.* Given a set $\mathcal{T} \subseteq L$ of target locations, the *reachability* objective $\mathsf{Reach}(\mathcal{T}) = \{\ell_0 \ell_1 \dots \mid \exists k \geq 0 : \ell_k \in \mathcal{T}\}$ requires that an observation in $\mathcal{T}$ is visited at least once. Dually, the *safety* objective $\mathsf{Safe}(\mathcal{T}) = \{\ell_0 \ell_1 \dots \mid \forall k \geq 0 : \ell_k \in \mathcal{T}\}$ requires that only locations in $\mathcal{T}$ are visited.
- *Büchi and coBüchi objectives.* Given a set $\mathcal{T} \subseteq L$ of target locations, the *Büchi* objective $\mathsf{Buchi}(\mathcal{T}) = \{\pi \mid \mathsf{Inf}(\pi) \cap \mathcal{T} \neq \varnothing\}$ requires that at least one location in $\mathcal{T}$ is visited infinitely often. Dually, the *coBüchi* objective $\mathsf{coBuchi}(\mathcal{T}) = \{\pi \mid \mathsf{Inf}(\pi) \subseteq \mathcal{T}\}$ requires that only locations in $\mathcal{T}$ are visited infinitely often.
- *Parity objectives.* For $d \in \mathbb{N}$, let $pr : L \to \{0, 1, \dots, d\}$ be a *priority function* that maps each location to a nonnegative integer priority. The *parity* objective $\mathsf{Parity}(pr) = \{\pi \mid \min\{pr(\ell) \mid \ell \in \mathsf{Inf}(\pi)\} \text{ is even}\}$ requires that the minimal priority occurring infinitely often is even.

Given a location $\hat{\ell}$, we also say that Player i ($i = 1, 2$) is surely-winning from $\hat{\ell}$ (or that $\hat{\ell}$ is surely-winning) for an objective φ in G if Player i has a surely-winning strategy in for φ in the game $\hat{G} = \langle L, \hat{\ell}, \Sigma, \Delta \rangle$ where $\hat{\ell}$ is the initial location. A game is *determined* if when player i does not have a surely-winning strategy from a location ℓ for an objective φ, then Player $3 - i$ has a surely-winning strategy from ℓ for the complement objective $\overline{\varphi}$.

Exercise 1 Prove the following:

(a) Büchi and coBüchi objectives are special cases of parity objectives.
(b) The complement of a parity objective is again a parity objective.

□

The following result shows that (i) parity games are determined and (ii) memoryless strategies are sufficient to win parity games.

Theorem 1 (Memoryless determinacy [EJ91]). *In all game graphs G with parity objective φ, the following hold:*

- *either Player 1 has a surely-winning strategy in $\langle G, \varphi \rangle$, or Player 2 has a surely-winning strategy in $\langle G, \bar{\varphi} \rangle$;*
- *Player 1 has a surely-winning strategy in $\langle G, \varphi \rangle$ if and only if she has a memoryless surely-winning strategy in $\langle G, \varphi \rangle$;*
- *Player 2 has a surely-winning strategy in $\langle G, \varphi \rangle$ if and only if he has a memoryless surely-winning strategy in $\langle G, \varphi \rangle$.*

Exercise 2 Consider a game graph $G = \langle L, l_I, \Sigma, \Delta \rangle$ which is not total, and assume that we modify the rules of the game as follows: if in a round where the current location is ℓ, Player 1 chooses an action $\sigma \in \Sigma$ such that there exists no transition $(\ell, \sigma, \ell') \in \Delta$, then Player 1 is declared losing the game. Given a non-total game graph G and parity objective φ in G, define a generic construction of a total game graph G' along with a parity objective φ' such that Player 1 has a surely-winning strategy in $\langle G, \varphi \rangle$ if and only if he has a surely-winning strategy in $\langle G', \varphi' \rangle$. $\square$

Exercise 3 Traditionally, a two-player game is a directed graph $\langle V, v_I, E \rangle$ where V is partitioned into V_1, V_2 the sets of vertices of Player 1 and Player 2 respectively, $v_I \in V$ is the initial vertex, and $E \subseteq V \times V$ is a set of edges. We call this model *edge-game*. A parity objective is defined by a priority function $pr : V \to \{0, 1, \ldots, d\}$ as above. A (memoryless) strategy for player i ($i = 1, 2$) is a function $\gamma_i : V_i \to E$ such that $(v, \gamma_i(v)) \in E$ for all $v \in V_i$. The definition of plays and outcomes is adapted accordingly. Show that the edge-games are equivalent to our game graphs by defining a generic transformation (a) from parity edge-games to parity game graphs, and (b) from parity game graphs to parity edge-games, such that player 1 has a surely-winning strategy in one game if and only if he has a surely-winning strategy in the other game.

Hint: for (a), first define an equivalent bipartite graph $\langle V', v'_I, E' \rangle$ such that for all edges $(v, v') \in E'$, $v \in V'_1$ if and only if $v' \in V'_2$. $\square$

Algorithms We present an algorithmic solution to the problem of deciding, given a game graph G and an objective φ, if Player 1 has a surely-winning strategy for φ in G. The set of locations in which Player 1 has a surely-winning strategy can be computed symbolically as the solution of certain nested fixpoint formulas, based on the *controllable predecessor operator* $\mathsf{Cpre} : 2^L \to 2^L$ which, given a set of locations $s \subseteq L$, computes the set of locations $\ell \in L$ from which Player 1 can force the game to be in a location of s in the next round, i.e. she has an action $\sigma \in \Sigma$ such that all transitions from ℓ labeled by σ lead to s. Formally,

$$\mathsf{Cpre}(s) = \{\ell \in L \mid \exists \sigma \in \Sigma \cdot \forall \ell' \in L : \text{ if } (\ell, \sigma, \ell') \in \Delta \text{ then } \ell' \in s\}.$$

Exercise 4 (a) Show that Cpre is a monotone operator for the subset ordering i.e., $s \subseteq s'$ implies $\mathsf{Cpre}(s) \subseteq \mathsf{Cpre}(s')$ for all $s, s' \subseteq L$.
(b) Define the controllable predecessor operator for the two-player edge-games of Exercise 3. $\square$

Consider a game with safety objective $\mathsf{Safe}(\mathcal{T})$. To win such a game, Player 1 has to be able to maintain the game in the set $\mathcal{T}$ for infinitely many rounds. For all $i \geq 0$, let $W^i \subseteq L$ be the set of locations from which Player 1 can maintain the game in the set $\mathcal{T}$ for at least the next i rounds. Clearly $W^{i+1} \subseteq W^i \subseteq \mathcal{T}$ for all $i \geq 0$, and therefore the sequence of sets $(W^i)_{i\geq 0}$ is decreasing and eventually stabilizes. The limit of this sequence is defined as

$$W = \bigcap_{i\geq 0} W^i$$

and this is the set of surely-winning locations for Player 1. This result follows from the facts that for all $i \geq 0$ and from all locations $\ell \in W^{i+1}$, Player 1 can force the game to be in a location of W^i in the next round, and that $W = W^{j+1} = W^j$ for some $j \geq 0$. We can compute the sets W^i as follows:

$$\begin{aligned} W^0 &= \mathcal{T} \\ W^{i+1} &= \mathcal{T} \cap \mathsf{Cpre}(W^i) \text{ for all } i \geq 0 \end{aligned}$$

Note that the limit W is obtained after at most n iterations where $n = |\mathcal{T}|$ is the number of target locations. The set W can also be viewed as the *greatest solution* of the equation $W = \mathcal{T} \cap \mathsf{Cpre}(W)$, noted $\nu W \cdot \mathcal{T} \cap \mathsf{Cpre}(W)$. The argument showing that a unique greatest fixpoint exists is not developed in this tutorial. We simply mention that it relies on the theory of complete lattices and Kleene's fixpoint theorem.

Theorem 2 (Safety games). *The set of surely-winning positions for Player 1 in safety games with perfect information can be computed in linear time.*

For reachability objectives, the algorithmic solution based on Cpre computes a sequence of sets W^i $(i \geq 0)$ such that from every $\ell \in W^i$, Player 1 can force the game to reach some location $\ell \in \mathcal{T}$ within the next i rounds. It can be computed as follows:

$$\begin{aligned} W^0 &= \mathcal{T} \\ W^{i+1} &= \mathcal{T} \cup \mathsf{Cpre}(W^i) \text{ for all } i \geq 0 \end{aligned}$$

The necessary number of iterations is at most $|L \setminus \mathcal{T}|$. In terms of fixpoint, the set W is the *least solution* of the equation $W = \mathcal{T} \cup \mathsf{Cpre}(W)$, noted $\mu W \cdot \mathcal{T} \cup \mathsf{Cpre}(W)$.

Theorem 3 (Reachability games). *The set of surely-winning positions for Player 1 in reachability games with perfect information can be computed in linear time.*

For parity objectives, several algorithms have been proposed in the literature (see e.g. [Zie98,Jur00,Sch08], and [FL09] for a survey). Using the result of memoryless determinacy (Theorem 1), it is easy to show that parity games can be solved in NP $\cap$ coNP. A major open problem is to know whether parity games with perfect information can be solved in polynomial time.

We present an algorithmic solution for parity games using a reduction to safety games. A variant of this reduction has been presented in [BJW02]. In the worst case, it yields safety games of size exponentially larger than the parity game. Such a blow-up is not surprising since safety games can be solved in polynomial time. The reduction gives some insight on the structure of parity games.

Consider a game graph $G = \langle L, l_I, \Sigma, \Delta \rangle$ and a priority function $pr : L \to \{0, 1, \ldots, d\}$ defining the parity objective $\mathsf{Parity}(pr)$ that requires the minimal priority occurring infinitely often to be even. We extend the locations of G with tuples $\langle c_1, c_3, \ldots, c_d \rangle$ of *counters* associated with the odd priorities (we assume that d is odd). The counters are initialized to 0, and each visit to a state with *odd* priority p increments the counter c_p. Intuitively, accumulating visits to an odd priority is potentially bad, except if a smaller even priority is also eventually visited. Therefore, each visit to a state with *even* priority p resets all counters $c_{p'}$ with $p' > p$.

Under these rules, if player 1 has a surely-winning strategy in G for the objective $\mathsf{Parity}(pr)$, then player 1 also has a memoryless surely-winning strategy, and thus can enforce that each counter c_p remains bounded by n_p the number of locations with priority p. On the other hand, if Player 1 has no strategy that maintains all counter c_p below n_p, then it means that no matter the strategy of Player 1, there exists a strategy of Player 2 such that the outcome of the game visits some location with odd priority p at least twice, without visiting a location of smaller priority. Since we can assume that Player 1 uses a memoryless strategy, this shows that Player 2 can force infinitely many visits to an odd priority without visiting a smaller priority, thus Player 1 cannot win the parity game.

Formally, we define $G' = \langle L', l'_I, \Sigma, \Delta' \rangle$ where

- $L' = L \times [n_1] \times [n_3] \times \ldots \times [n_d]$ where $[n_i]$ denotes the set $\{0, 1, \ldots, n_i\} \cup \{\infty\}$, and n_i is the number of locations with priority i in G;
- $l'_I = (l_I, 0, 0, \ldots, 0)$;
- $\Delta' = \{((\ell_1, c), \sigma, (\ell_2, \mathsf{update}(c, p))) \mid (\ell_1, \sigma, \ell_2) \in \Delta \text{ and } p = pr(q)\}$ where

$$\mathsf{update}(\langle c_1, c_3, \ldots, c_d \rangle, p) = \begin{cases} \langle c_1, \ldots, c_{p-1}, 0, \ldots, 0 \rangle & \text{if } p \text{ is even} \\ \langle c_1, \ldots, c_{p-1}, c_p + 1, c_{p+1}, \ldots, c_d \rangle & \text{if } p \text{ is odd} \end{cases}$$

  where we let $c_p + 1 = \infty$ for $c_p \in \{n_p, \infty\}$.

The safety objective for G' is $\mathsf{Safe}(\mathcal{T}^G_{pr})$ where $\mathcal{T}^G_{pr} = L' \cap (L \times \mathbb{N}^{\lceil \frac{d}{2} \rceil})$ is the set of locations in which no overflow occurred. The following lemma states the correctness of the construction.

Lemma 1. *For all game graphs G and priority functions pr, Player 1 is surely-winning in G for the objective $\mathsf{Parity}(pr)$ if and only if Player 1 is surely-winning in G' for the objective $\mathsf{Safe}(\mathcal{T}^G_{pr})$.*

Proof. First, let α be a winning strategy for Player 1 in G for the parity objective $\mathsf{Parity}(pr)$. We construct a strategy α' for Player 1 in the game G' and we show that this strategy is surely winning for the objective $\mathsf{Safe}(\mathcal{T}_{pr}^{G})$. First, without loss of generality we can assume that α is memoryless. We define α' as follows, for all histories π in G', let $(\ell, c) = \mathsf{Last}(\pi)$, we take $\alpha'(\pi) = \alpha(\ell)$. We show that α' is winning for the objective $\mathsf{Safe}(\mathcal{T}_{pr}^{G})$. Towards contradiction, assume that it is not the case. Then there exists a strategy β' of Player 2 such that $\mathsf{outcome}(G', \alpha', \beta') = (\ell_0, c_0)(\ell_1, c_1)\dots(\ell_n, c_n)\dots$ leaves $\mathcal{T}_{pr}^{G}$. Let $0 \leq k_1 < k_2$ be such that (ℓ_{k_2}, c_{k_2}) is the first location where a counter (say c_p) reaches the value ∞ (p is the odd priority associated with this counter), and k_1 is the last index where this counter has been reset (k_1 is equal to 0 if the counter has never been reset). As c_p overflows, we know that the subsequence $(\ell_{k_1}, c_{k_1})(\ell_{k_1+1}, c_{k_1+1})\dots(\ell_{k_2}, c_{k_2})$ visits $n_p{+}1$ locations with priority p. As there are n_p locations with priority p in G, we know that there is at least one location with priority p which is repeating in the subsequence. Let i_1 and i_2 be the two indexes associated with such a repeating location. Between i_1 and i_2, there is no visit to an even priority smaller than p. Because Player 1 is playing a memoryless strategy in G, Player 2 can spoil the strategy of Player 1 by repeating his sequence of choices between i_1 and i_2. This contradicts our hypothesis that α is a winning strategy in G for the parity objective $\mathsf{Parity}(pr)$.

Second, let us consider the case where Player 1 is not surely winning in G for the objective $\mathsf{Parity}(pr)$. By determinacy, we know that Player 2 has a surely winning strategy β for the parity objective $\overline{\mathsf{Parity}(pr)}$. Using a similar argument as above we can construct a strategy β' for Player 2 for surely winning the reachability objective $\mathsf{Reach}(\overline{\mathcal{T}_{pr}^{G}})$. By determinacy, this shows that Player 1 is not surely-winning in G' for the objective $\mathsf{Safe}(\mathcal{T}_{pr}^{G})$. ■

Note that since Büchi and coBüchi objectives are parity objectives (see Exercise 1), the above reduction to safety games applies and yields a game G' of quadratic size, thus a quadratic-time algorithm for solving Büchi and coBüchi games.

3 Games with imperfect information: surely-winning

In a game with imperfect information, the set of locations is partitioned into information sets called *observations*. Player 1 is not allowed to see what is the current location of the game, but only what is the current observation. Observations provide imperfect information about the current location. For example, if a location encodes the state of a distributed system, the observation may disclose the value of the shared variables, and hide the value of the private variables; or in a physical system, an observation may give a range of possible values for parameters such as temperature, modeling sensor imprecision. Note that the structure of the game itself is known to both players, imperfect information arising only about the current location while playing the game.

3.1 Game structure with imperfect information

A *game structure with imperfect information* is a tuple $G = \langle L, l_I, \Sigma, \Delta, \mathcal{O}\rangle$, where $\langle L, l_I, \Sigma, \Delta\rangle$ is a game graph (see Section 2) and $\mathcal{O}$ is a set of *observations* that partitions the set L of locations. For each location $\ell \in L$, we denote by $\mathsf{obs}(\ell)$ the unique observation $o \in \mathcal{O}$ such that $\ell \in o$. For each play $\pi = \ell_0\ell_1 \ldots$, we denote by $\mathsf{obs}(\pi)$ the sequence $\mathsf{obs}(\ell_0)\mathsf{obs}(\ell_1)\ldots$ and we analogously extend $\mathsf{obs}(\cdot)$ to histories, sets of plays, etc.

The game on G is played in the same way as in the perfect information case, but now only the observation of the current location is revealed to Player 1. The effect of the uncertainty about the history of the play is formally captured by the notion of observation-based strategy.

An *observation-based strategy* for Player 1 is a function $\alpha : L^+ \to \Sigma$ such that $\alpha(\pi) = \alpha(\pi')$ for all histories $\pi, \pi' \in L^+$ with $\mathsf{obs}(\pi) = \mathsf{obs}(\pi')$. We often use the notation α^o to emphasize that α is observation-based. Outcome and consistent plays are defined as in games with perfect information.

An objective φ in a game with imperfect information is a set of plays as before, but we require that φ is *observable* by Player 1 i.e., for all $\pi \in \varphi$, for all π' such that $\mathsf{obs}(\pi') = \mathsf{obs}(\pi)$, we have $\pi' \in \varphi$. In the sequel, we often view objectives as sets of infinite sequences of observations, i.e. $\varphi \in \mathcal{O}^\omega$, and we also call them observable objectives. For example, we assume that reachability and safety objectives are specified by a union of target observations, and parity objectives are specified by priority functions of the form $p : \mathcal{O} \to \{0, \ldots, d\}$. The definition of surely-winning strategies is adapted accordingly, namely, a deterministic observation-based strategy α for player 1 is *surely-winning* for an objective $\varphi \in \mathcal{O}^\omega$ in G if $\mathsf{obs}(\mathsf{Outcome}_1(G, \alpha)) \subseteq \varphi$. Note that games with perfect information can be obtained as the special case where $\mathcal{O} = \{\{\ell\} \mid \ell \in L\}$.

Example Consider the game structure with imperfect information in Fig. 2. The observations are $o_1 = \{\ell_1\}$, $o_2 = \{\ell_2, \ell_2'\}$, $o_3 = \{\ell_3, \ell_3'\}$, and $o_4 = \{\ell_4\}$. The transitions are shown as labeled edges, and the initial state is ℓ_1. The objective of Player 1 is $\varphi = \mathsf{Reach}(o_4)$ i.e., to reach location ℓ_4. We argue that the game is not surely-winning for Player 1. Let α be an arbitrary deterministic strategy for Player 1. Consider the strategy β for Player 2 as follows: for all $\pi \in L^+$ such that $\mathsf{Last}(\pi) \in o_2$, if $\alpha(\pi) = a$, then in the previous round β chooses the state ℓ_2, and if $\alpha(\pi) = b$, then in the previous round β chooses the state ℓ_2'. Given α and β, the play $\mathsf{outcome}(G, \alpha, \beta)$ never reaches ℓ_4. Similarly, Player 2 has no strategy β to ensure that $\mathsf{obs}(\mathsf{outcome}_2(G, \beta)) \subseteq \bar{\varphi}$ where $\bar{\varphi} = \mathsf{Safe}(\{o_1, o_2, o_3\})$, is the complement of φ. Hence the game is not determined.

We briefly discuss the definition of games with imperfect information. In traditional games with perfect information played on graphs (see exercice 3), locations are partitioned into locations of Player 1 and locations of Player 2, and the players choose edges from the locations they own. It can be shown that for perfect information games, this model is equivalent to our definition (see

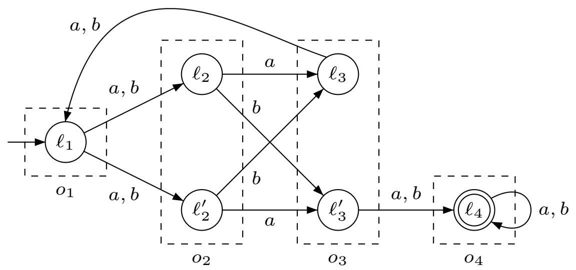

Fig. 2. A game structure with imperfect information G.

Exercise 3). When extending the classical game model to imperfect information, we need to remember that Player 1 does not see what is the current location, and therefore he could not in general choose an edge from the current location. Instead, one may ask Player 1 to choose in each round one edge per location, thus to be prepared to all situations. This would require an alphabet of actions of the form $L \to \Delta$ which is of exponential size. We prefer a simpler definition where an alphabet Σ of actions is fixed, and each action selects some outgoing edges. In this definition, all locations belong to Player 1 and the choices of Player 2 are modeled by nondeterminism.

Another point of interest is the fact that games with imperfect information sound asymmetric, as only Player 1 has partial view of the play. It should be noted however that for surely-winning, it would be of no help to Player 1 that Player 2 also has imperfect information. Indeed, a surely-winning strategy of Player 1 has to ensure that *all* outcomes are in the objective, and this requirement is somehow independent of the ability or not of Player 2 to see the current location. In terms of strategies, one can show that to spoil a not surely-winning strategy of Player 1, Player 2 does not need to remember the history of the play, but only needs to count the number of rounds that have been played. We say that a deterministic strategy $\beta : L^+ \times \Sigma \to L$ for Player 2 is *counting* if for all $\pi, \pi' \in L^+$ such that $|\pi| = |\pi'|$ and $\mathsf{Last}(\pi) = \mathsf{Last}(\pi')$, and for all $\sigma \in \Sigma$, we have $\beta(\pi, \sigma) = \beta(\pi', \sigma)$.

Theorem 4 ([CDHR07]). *Let G be a game structure with imperfect information and φ be an observable objective. There exists an observation-based strategy $\alpha^o \in \mathcal{A}_G$ such that for all $\beta \in \mathcal{B}_G$ we have* $\mathsf{outcome}(G, \alpha^o, \beta) \in \varphi$ *if and only if there exists an observation-based strategy $\alpha^o \in \mathcal{A}_G^O$ such that for all counting strategies $\beta^c \in \mathcal{B}_G$ we have* $\mathsf{outcome}(G, \alpha^o, \beta^c) \in \varphi$.

Exercise 5 Prove Theorem 4. □

The requirement that observations partition the set of locations of the games may seem to be restrictive. For example in a system using sensors, it would be more natural to allow overlapping observations. For instance, if a control program measures the temperature using sensors, the values that are obtained

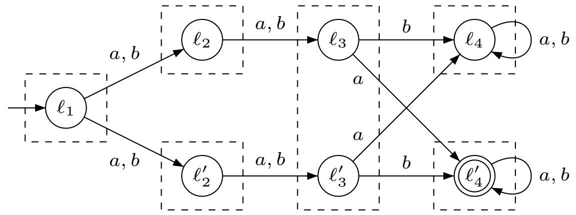

Fig. 3. Memory is necessary for Player 1 to surely-win the objective $\mathsf{Reach}(\ell'_4)$.

have finite precision ε. When the sensor returns value t, the actual temperature lies within the interval $[t-\varepsilon, t+\varepsilon]$. Clearly, for a measure t' such that $|t'-t| < \varepsilon$, we have that $[t-\varepsilon, t+\varepsilon] \cap [t'-\varepsilon, t'+\varepsilon] \neq \varnothing$. As a consequence, the temperature observations overlap and do not form a partition of the space of values.

Exercise 6 Show that a game structure with imperfect information in which the observations do not partition the state space can be transformed into an equivalent game structure with imperfect information with partitioning observations in polynomial time. □

Consider the game structure with imperfect information in Fig. 3. The alphabet of actions is $\Sigma = \{a, b\}$ and the objective for Player 1 is to reach location ℓ'_4. The partition induced by the observations is represented by the dashed sets. We claim that Player 1 has no memoryless observation-based surely-winning strategy in this game. This is because from locations ℓ_3 and ℓ'_3, different actions need to be played to reach ℓ'_4, but since ℓ_3 and ℓ'_3 have the same observation, Player 1 has to play the same action in a memoryless observation-based strategy. However, if Player 1 remembers the previous observation, then he has a surely-winning strategy, namely if $\{\ell_2\}$ was observed in the previous round, then play a, and if $\{\ell'_2\}$ was observed in the previous round, then play b. This shows that memory may be necessary for surely-winning in a game with imperfect information even for a reachability objective. Intuitively, a sequence of past observations provides more precise knowledge about the current location of the game than the current observation only. Therefore, Player 1 should store and update this knowledge along the play to maintain the most precise information possible. Initially, his knowledge is the singleton $\{l_I\}$ (we assume that the structure of the game is known to both players), and if the current knowledge is a set $s \subseteq L$, Player 1 chooses action $\sigma \in \Sigma$, and observation $o \in \mathcal{O}$ is disclosed, then the updated knowledge is $\mathsf{post}^G_\sigma(s) \cap o$ where $\mathsf{post}^G_\sigma(s) = \{\ell' \in L \mid \exists \ell \in s : (\ell, \sigma, \ell') \in \Delta\}$ i.e., the set of all locations reachable from locations in s by playing σ.

3.2 Reduction to games with perfect information

Given a game structure with imperfect information $G = \langle L, l_I, \Sigma, \Delta, \mathcal{O} \rangle$ with observable parity objective φ, we construct an equivalent game structure (with perfect information) $G^{\mathsf{K}} = \langle S, s_I, \Sigma, \Delta^{\mathsf{K}} \rangle$ with a parity objective φ^{K} which, intuitively, monitors the knowledge that Player 1 has about the current location of the play. The game G^{K} is called *knowledge-based subset construction.* The structure $G^{\mathsf{K}} = \langle S, s_I, \Sigma, \Delta^{\mathsf{K}} \rangle$ is defined as follows.

- The set of locations is $S = \{s \in 2^L \backslash \{\varnothing\} \mid \exists o \in \mathcal{O} \cdot s \subseteq o\}$. In the sequel, we call a set $s \in S$ a *cell.*
- The initial location is $s_I = \{l_I\}$.
- The set of labeled transitions $\Delta^{\mathsf{K}} \subseteq S \times \Sigma \times S$ contains all (s, σ, s') for which there exists $o \in \mathcal{O}$ such that $s' = \mathsf{post}^G_\sigma(s) \cap o$.

Note that since the game graph G is total and the observations form a partition of the locations, the game graph G^{K} is also total.

To complete the reduction, we show how to translate the objectives. Given a priority function $pr : \mathcal{O} \to \{0, \dots, d\}$ defining the parity objective φ in G, we define the parity objective φ^{K} in G^{K} using the priority function pr^{K} such that $pr^{\mathsf{K}}(s) = pr(o)$ for all $s \in S$ and $o \in \mathcal{O}$ such that $s \subseteq o$.

Theorem 5 ([CDHR07]). *Player* 1 *has an observation-based surely-winning strategy in a game structure G with imperfect information for an observable parity objective φ if and only if Player* 1 *has a surely-winning strategy in the game structure G^{K} with perfect information for the parity objective φ^{K}.*

Exercise 7 Write a proof of Theorem 5. □

Observable safety and reachability objectives are defined by sets $\mathcal{T} \subseteq L$ of target locations that are a union of observations. Hence for all cells $s \in S$, either $s \cap \mathcal{T} = \varnothing$ or $s \subseteq \mathcal{T}$. In the above reduction, such an objective is transformed into an objective of the same type with set of target cells $\mathcal{T}^{\mathsf{K}} = \{s \in S \mid s \subseteq \mathcal{T}\}$.

Exercise 8 Consider a game structure with imperfect information $G = \langle L, l_I, \Sigma, \Delta, \mathcal{O} \rangle$ and a *non-observable* reachability objective defined by $\mathcal{T} \subseteq L$. Construct an equivalent game structure with imperfect information G' with an observable reachability objective $\mathsf{Reach}(\mathcal{T}')$, i.e. such that Player 1 has an observation-based surely-winning strategy in G for $\mathsf{Reach}(\mathcal{T})$ if and only if Player 1 has an observation-based surely-winning strategy in G' for $\mathsf{Reach}(\mathcal{T}')$. Hint: take $G' = \langle L, l_I, \Sigma, \Delta, \mathcal{O}' \rangle$ where $\mathcal{O}' = \{o \cap \mathcal{T} \mid o \in \mathcal{O}\} \cup \{o \cap (L \backslash \mathcal{T}) \mid o \in \mathcal{O}\}$.

Note that non-observable Büchi objectives are more difficult to handle. For such objectives and more generally for non-observable parity objectives, our knowledge-subset construction is not valid and techniques related to Safra's determinization need to be used [Saf88]. □

3.3 Symbolic algorithms and antichains

Theorem 5 gives a natural algorithm for solving games with imperfect information with observable objective: apply the algorithms for solving games with perfect information to the knowledge-based subset construction presented above[3].

[3] Note that the symbolic algorithm can be applied without explicitly constructing the knowledge-based construction.

The symbolic algorithms presented in Section 2 are based on the controllable predecessor operator $\mathsf{Cpre} : 2^S \rightarrow 2^S$ whose definition can be rewritten for all $q \subseteq S$ as:

$$\begin{aligned}\mathsf{Cpre}(q) &= \{s \in S \mid \exists \sigma \in \Sigma \cdot \forall s' \in S : \text{ if } (s, \sigma, s') \in \Delta^{\mathsf{K}} \text{ then } s' \in q\} \\ &= \{s \in S \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathcal{O} : \text{ if } s' = \mathsf{post}^G_\sigma(s) \cap o \neq \varnothing \text{ then } s' \in q\}.\end{aligned}$$

A crucial property of this operator is that it preserves downward-closedness of sets of cells. Intuitively, Player 1 is in a better situation when her knowledge is more precise, i.e. when her knowledge is a smaller cell according to set inclusion. A set q of cells is *downward-closed* if $s \in q$ implies $s' \in q$ for all $s' \subseteq s$. If Player 1 can force the game G^{K} to be in a cell of a downward-closed set of cells q in the next round from a cell s, then she is also able to do so from all cells $s' \subseteq s$. Formally, if q is downward-closed, then so is $\mathsf{Cpre}(q)$. It is easy to show that $\cap$ and $\cup$ also preserve downward-closedness, and therefore all sets of cells that are computed for solving games of imperfect information are downward-closed.

As the symbolic algorithms are manipulating downward closed sets, it is valuable to design a data-structure to represent them compactly. We define such a data-structure here. The idea is to represent a set of cells by a set of sets of locations and interpret this set as defining all the cells that are included in one of its element. Clearly, in such a representation having a set of sets with two $\subseteq$-comparable element is not useful, so we can restrict our symbolic representations to be *antichains*, i.e. set of sets of locations that are $\subseteq$-incomparable.

Antichains for representing downward-closed sets Let us note $\mathcal{A}$ the set of $\subseteq$-antichains of sets of locations, that is

$$\mathcal{A} = \{\{s_1, s_2, \ldots, s_n\} \subseteq 2^L \mid \forall 1 \leq i, j \leq n : s_i \subseteq s_j \rightarrow i = j\}.$$

Note that an antichain is a set of subsets of locations that are not necessary cells. We denote by $\mathcal{A}$ the set of antichains. The set $\mathcal{A}$ is partially ordered as follows. For $q, q' \in \mathcal{A}$, let $q \sqsubseteq q'$ iff $\forall s \in q \cdot \exists s' \in q' : s \subseteq s'$. The least upper bound of $q, q' \in \mathcal{A}$ is $q \sqcup q' = \lceil\{s \mid s \in q \text{ or } s \in q'\}\rceil$, and their greatest lower bound is $q \sqcap q' = \lceil\{s \cap s' \mid s \in q \text{ and } s' \in q'\}\rceil$. We view antichains as a symbolic representation of $\subseteq$-downward-closed sets of cells. Given an antichain $q \in \mathcal{A}$, let $q{\downarrow} = \{s \in S \mid \exists s' \in q : s \subseteq s'\}$ be the *downward closure* of q i.e., the set of cells that it represents.

Exercise 9 Show that $\sqcup$ and $\sqcap$ are indeed the operators of least upper bound and greatest lower bound respectively. By establishing this, you have shown that the set of antichains forms a complete lattice. What are the least and greatest elements of this complete lattice ? $\square$

To define a controllable predecessor operator $\mathsf{Cpre}^{\mathcal{A}}$ over antichains, we observe that for all $q \in A$,

$$\begin{aligned}\mathsf{Cpre}(q{\downarrow}) &= \{s \in S \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathcal{O} \cdot \exists s' \in q : \mathsf{post}^G_\sigma(s) \cap o \subseteq s'\} \\ &= \{s \in S \mid \exists \sigma \in \Sigma \cdot \forall o \in \mathcal{O} \cdot \exists s' \in q : s \subseteq \widetilde{\mathsf{pre}}^G_\sigma(s' \cup \overline{o})\}\end{aligned}$$

where $\overline{o} = L \setminus o$ and $\widetilde{\mathsf{pre}}^G_\sigma(s) = \{\ell \in L \mid \mathsf{post}^G_\sigma(\{\ell\}) \subseteq s\}$. Hence, we define

$$\mathsf{Cpre}^{\mathcal{A}}(q) = \bigsqcup_{\sigma \in \Sigma} \bigsqcap_{o \in \mathcal{O}} \bigsqcup_{s' \in q} \left\{\widetilde{\mathsf{pre}}_\sigma(s' \cup \overline{o})\right\}$$

and this operator computes a symbolic representation of the controllable predecessors of the downward-closed set of cells symbolically represented by q.

Lemma 2. *For all antichains $q \in \mathcal{A}$, we have $\mathsf{Cpre}^{\mathcal{A}}(q){\downarrow} = \mathsf{Cpre}(q{\downarrow})$.*

Exercise 10 Prove Lemma 2. $\square$

In the definition of $\mathsf{Cpre}^{\mathcal{A}}$, the operations $\widetilde{\mathsf{pre}}$, $\bigsqcup_{\sigma \in \Sigma}$ and $\bigsqcup_{s' \in q}$ can be computed in polynomial time, while $\bigsqcap_{o \in \mathcal{O}}$ can be computed in exponential time by simple application of the definitions. Unfortunately, it turns out that a polynomial-time algorithm is unlikely to exist for computing $\bigsqcap_{o \in \mathcal{O}}$ as the NP-complete problem IndependentSet can be reduced to it.

Consider a graph $\mathcal{G} = (V, E)$ where V is a set of vertices and $E \subseteq V \times V$ is a set of edges. An *independent set* of $\mathcal{G}$ is a set $W \subseteq V$ of vertices such that for all $(v, v') \in E$, either $v \notin W$ or $v' \notin W$ i.e., there is no edge of $\mathcal{G}$ connecting vertices of W. The IndependentSet problem asks given a graph $\mathcal{G}$ and size k to decide if there exists an independent set of $\mathcal{G}$ of size larger than k. This problem is known to be NP-complete. We show that IndependentSet reduces to computing $\sqcap$.

Let $\mathcal{G} = (V, E)$ be a graph, and for each $e = (v, v') \in E$ let $q_e = \big\{V \setminus \{v\}, V \setminus \{v'\}\big\}$. The set $q_e{\downarrow}$ contains all sets of vertices that are independent of the edge e. Therefore, the antichain $q = \lceil \bigcap_{e \in E} q_e{\downarrow} \rceil$ contains the maximal independent sets of $\mathcal{G}$, and an algorithm to compute q would immediately solve IndependentSet, showing that such an algorithm running in polynomial time cannot exist unless $P = NP$. The idea of this reduction can be extended to show that $\mathsf{Cpre}^{\mathcal{A}}$ requires exponential time [BCD+08,FJR09].

Exercise 11 Compute the winning cells in the two versions of the 3-coin game with the symbolic algorithm using the antichain representation. The 3-coin game graph is given in Fig. 4. We give here the solutions to this exercise.

- We first consider the version in which Player 2 is allowed to exchange the positions of the coins that are not toggled. To compute the winning cells of the game with imperfect information, we compute the set of all cells that are able to force the cell $\{7\}$. We give here the sequence of antichains computed by our algorithm: $X_0 = \{\{7\}\}$, $X_1 = X_0 \sqcup \mathsf{Cpre}(X_0) = \{\{1\}, \{2\}, \{3\}, \{7\}\}$, $X_2 = X_1 \sqcup \mathsf{Cpre}(X_1) = \{\{1\}, \{2\}, \{3\}, \{4\}, \{5\}, \{6\}, \{7\}\} = X_1$ and the fixed point is reached. As $\{0\} \notin X_1{\downarrow}$, this shows that Player 1 does not have a deterministic winning strategy in this game.
- We now consider the version where Player 2 is not allowed to exchange the position of the coins that are not toggled. To compute the winning cells of the game with imperfect information, we compute the set of all cells that are able

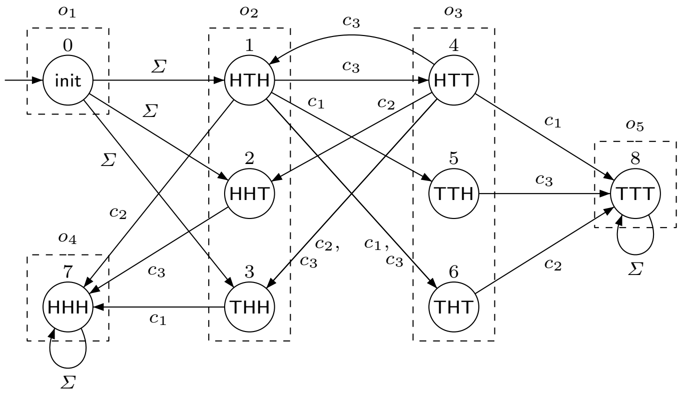

Fig. 4. The 3-COIN game graph with alphabet $\Sigma = \{c_1, c_2, c_3\}$. The transitions between states 2, 3, 5, and 6 are omitted for the sake of clarity.

to force the cell $\{7\}$. We give here the sequence of antichains computed by our algorithm. $X_0 = \{\{7\}\}$, $X_1 = X_0 \sqcup \mathsf{Cpre}(X_0) = \{\{1\},\{2\},\{3\},\{7\}\}$, $X_2 = X_1 \sqcup \mathsf{Cpre}(X_1) = \{\{1\},\{2\},\{3\},\{4\},\{5\},\{6\},\{7\}\}$, $X_3 = X_2 \sqcup \mathsf{Cpre}(X_2) = \{\{1,2\},\{2,3\},\{1,3\},\{4\},\{5\},\{6\},\{7\}\}$, $X_4 = X_3 \sqcup \mathsf{Cpre}(X_3) = \{\{1,2\},\{2,3\},\{1,3\},\{4,6\},\{5,6\},\{4,5\},\{7\}\}$, $X_5 = X_4 \sqcup \mathsf{Cpre}(X_4) = \{\{1,2,3\},\{4,6\},\{5,6\},\{4,5\},\{7\}\}$, $X_6 = X_5 \sqcup \mathsf{Cpre}(X_5) = \{\{0,1,2,3\},\{0,4,6\},\{0,5,6\},\{0,4,5\},\{0,7\}\}$, $X_7 = X_6$. As $\{0\} \in X_7{\downarrow}$, this shows that Player 1 has a deterministic winning strategy in this version of the 3-coin game.

□

3.4 Strategy construction

The algorithms presented in Section 2 for safety and reachability games compute the set of winning positions for Player 1. We can use these algorithms to compute the set of winning cells in a game with imperfect information, using the controllable predecessor operator $\mathsf{Cpre}^{\mathcal{A}}$. This gives a compact representation (an antichain) of the downward-closed set of winning cells. However, it does not construct surely-winning strategies. We show that in general, there is a direct way to construct a surely-winning strategy for safety games, but not for reachability and parity games.

For safety games with perfect information, the fixed point computation shows that the set of winning positions satisfies the equation $W = \mathcal{T} \cap \mathsf{Cpre}(W)$. Therefore, $W \subseteq \mathsf{Cpre}(W)$, and thus for each $\ell \in W$, there exists an action $\sigma_\ell \in \Sigma$ such that $\mathsf{post}^G_{\sigma_\ell}(\{\ell\}) \subseteq W$. Since, $W \subseteq \mathcal{T}$, it is easy to see that the memoryless strategy playing σ_ℓ in each location $\ell \in W$ is surely-winning.

For safety games with imperfect information, the fixed point W is represented by an antichain q_{win} such that $W = q_{\mathsf{win}}{\downarrow}$. The strategy construction for safety games with perfect information can be extended as follows. By definition of

$\mathsf{Cpre}^{\mathcal{A}}$, for each $s \in q_{\mathsf{win}}$ there exists $\sigma_s \in \Sigma$ such that for all $o \in \mathcal{O}$, we have $\mathsf{post}^G_{\sigma_s}(s) \cap o \subseteq s'$ for some $s' \in q_{\mathsf{win}}$. It is easy to see that the strategy playing σ_s in every cell $s'' \subseteq s$ is surely-winning.

Thus, we can define a surely-winning strategy by the Moore machine $\langle M, m_I, \mathsf{update}, \mu\rangle$ where $M = q_{\mathsf{win}}$, $m_I = s$ such that $s_I \subseteq s$ for some $s \in q_{\mathsf{win}}$, $\mu : M \to \Sigma$ is an output function such that $\mu(s) = \sigma_s$ as defined above for all $s \in M$, and $\mathsf{update} : M \times \mathcal{O} \to M$ is such that if $\mathsf{update}(s, o) = s'$, then $\mathsf{post}^G_{\sigma}(s) \cap o \subseteq s'$ for $\sigma = \mu(s)$ (note that such an s' exists by the above remark). The automaton A defines the observation-based strategy α such that $\alpha(\pi) = \sigma$ where $\sigma = \mu(s)$ and $s = \mathsf{update}(m_I, \mathsf{obs}(\pi))$ for all $\pi \in L^+$ (where the update function is extended to sequences of observations in the usual way, i.e. $\mathsf{update}(m, o_1 \dots o_n) = \mathsf{update}(\mathsf{update}(m, o_1), o_2 \dots o_n))$.

For reachability games, the information contained in the fixpoint of winning positions is not sufficient to extract a surely-winning strategy. Intuitively, a surely-winning strategy needs to stay in the winning set (as in safety games), and moreover should ensure some kind of progress with respect to the target set $\mathcal{T}$ to guarantee that $\mathcal{T}$ is eventually reached. The notion of progress can be formalized by a number $\mathsf{rank}(s)$ associated to each cell s such that Player 1 can enforce to reach the target from cell s within at most $\mathsf{rank}(s)$ rounds.

In a reachability game with perfect information, the rank of a location ℓ in the set of winning positions W is the least i such that $\ell \in W^i$. From a location $\ell \in W$ with rank $r > 0$, a surely-winning strategy can play an action $\sigma_\ell \in \Sigma$ such that $\mathsf{post}^G_{\sigma_\ell}(\{\ell\}) \subseteq W^{r-1}$.

In a game with imperfect information, knowing the rank of the cells in the antichain q_{win} may still not be sufficient to obtain a surely-winning strategy. Consider the game G in Fig. 5, with reachability objective $\mathsf{Reach}(\{\ell_2\})$ and observations $\{\ell_0, \ell_1\}$ and $\{\ell_2\}$. Since $\mathsf{Cpre}(\{\{\ell_2\}\}) = \{\{\ell_1\}\}$ (by playing action b) and $\mathsf{Cpre}(\{\{\ell_1\}, \{\ell_2\}\}) = \{\{\ell_0, \ell_1\}\}$ (by playing action a), the fixed point computed by the antichain algorithm is $\{\{\ell_2\}, \{\ell_0, \ell_1\}\}$. However, from $\{\ell_0, \ell_1\}$, after playing a, Player 1 reaches the cell $\{\ell_1\}$ which is not in the fixed-point (however, it is subsumed by the cell $\{\ell_0, \ell_1\}$). Intuitively, the antichain algorithm has forgotten which action is to be played next from $\{\ell_1\}$. Note that playing a again (and thus forever) is not winning.

This example illustrates the fact that the rank of a cell s is not necessarily the same as the rank of a cell $s' \subseteq s$. Therefore, for the purpose of strategy construction, the fixpoint computation needs to store the rank associated with a cell, and refine the rule of eliminating the cells that are subsumed by larger cells to take ranks into account [BCD+09]. In fact, it can be shown that for some family of reachability games with imperfect information, the fixpoint computed in the antichain representation (without rank) is of polynomial size while any finite-memory surely-winning strategy is of exponential size [BCD+08].

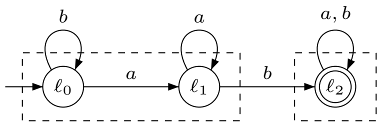

Fig. 5. A reachability game G.

4 Games with imperfect information: almost-surely winning

We revisit the 3-COIN game. In Exercise 11, we have seen that Player 1 does not have an observation-based deterministic winning strategy in this game when Player 2 is allowed to exchange the position of the coins that are not toggled. This is because Player 2 can guess the choice that Player 1 will make in the next round. When a deterministic strategy for Player 1 is fixed, this information is formally available to Player 2 but this is not realistic in practice. Player 1 should use a source of randomization in order to avoid that Player 2 can guess the choice she will make in the next round. Whenever the game is in a configuration with two heads, Player 1 chooses uniformly at random one of the three coins. Clearly the probability to choose the coin showing tail is $\frac{1}{3}$ no matter if Player 2 has decided to exchange the coins or not at the previous step. Otherwise, she should play the same coin a second time to make sure to come back to a configuration with two heads. She then repeats the same randomized strategy. Every two rounds, Player 1 has a $\frac{1}{3}$ probability to reach the winning configuration. Note also that she is sure to avoid the loosing configuration (all coins on tails). This simple strategy is thus winning the reachability objective with probability one. This illustrates the power of randomized strategies in games with imperfect information.

4.1 Playing with randomized strategies

Before going into the formalization, let us take a look at the example of Fig. 3. From the initial location ℓ_1, we have seen that Player 1 has no surely-winning strategy for reaching ℓ_4. This is because for all strategies α of Player 1, there exists a play $\pi \in \mathsf{Outcome}_1(G, \alpha)$ that visits ℓ_3 infinitely often, and therefore never visits ℓ_4. However, the strategy β of Player 2 such that $\pi = \mathsf{outcome}(G, \alpha, \beta)$ chooses the successor $\hat{\ell}$ of ℓ_1 in a way that depends on the next move of Player 1, namely $\hat{\ell} = \ell_2$ if α plays action a next, and $\hat{\ell} = \ell_2'$ if α plays action b next. In a concrete implementation of the system, this means that Player 2 needs to predict the behavior of Player 1 infinitely often in order to win. In practice, since one wrong guess make Player 1 win, this suggests that the probability that Player 2 wins (making infinitely many right guesses) is 0, and thus Player 1 can win with probability 1.

We now formally define a notion of probabilistic winning. First, a *randomized strategy* for Player 1 is a function $\alpha : (L \times \Sigma)^* L \to \mathcal{D}(\Sigma)$ where $\mathcal{D}(\Sigma)$ denotes the set of probability distributions over Σ i.e., the set of all functions $f : \Sigma \to [0, 1]$

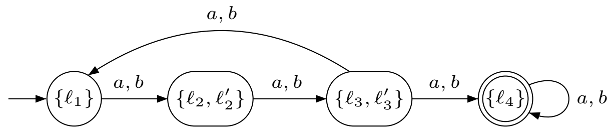

Fig. 6. The knowledge-based subset construction for the game of Fig. 3.

such that $\sum_{\sigma \in \Sigma} f(\sigma) = 1$. Intuitively, if Player 1 uses distribution f, then he plays each action σ with probability $f(\sigma)$. We assume that Player 1 is informed about which actual actions he played. Hence, strategies are mappings from interleaved sequences of states and actions of the form $\rho = \ell_0\sigma_0\ell_1\sigma_1 \dots \sigma_{n-1}\ell_n$ that we call *labeled histories*. We denote by $L(\rho) = \ell_0\ell_1 \dots \ell_n$ the *projection* of ρ onto L^*. A strategy α is *observation-based* if for all pairs of labeled histories ρ, ρ' of the form $\rho = \ell_0\sigma_0\ell_1\sigma_1 \dots \sigma_{n-1}\ell_n$ and $\rho' = \ell_0'\sigma_0\ell_1'\sigma_1 \dots \sigma_{n-1}\ell_n'$ such that for all i, $1 \leq i \leq n$, $\mathsf{obs}(\ell_i) = \mathsf{obs}(\ell_i')$, we have that $\alpha(\rho) = \alpha(\rho')$.

A randomized strategy for Player 2 is a function $\beta : (L \times \Sigma)^+ \to \mathcal{D}(L)$ such that for all labeled histories $\rho = \ell_0\sigma_0\ell_1\sigma_1 \dots \sigma_{n-1}\ell_n$ and $\sigma \in \Sigma$, for all ℓ such that $\beta(\rho, \sigma)(\ell) > 0$, we have $(\ell_n, \sigma, \ell) \in \Delta$. We extend in the expected way (using projection of labeled histories onto L^* when necessary) the notions of observation-based randomized strategies for Player 2, memoryless strategies, consistent plays, outcome, etc.

Given strategies α and β for Player 1 and Player 2 respectively, and an initial location ℓ_0, the probability of a labeled history $\rho = \ell_0\sigma_0\ell_1\sigma_1 \dots \sigma_{n-1}\ell_n$ is $\mathbb{P}(\rho) = \prod_{i=0}^{n-1} \alpha(\ell_0\sigma_0 \dots \sigma_{i-1}\ell_i)(\sigma_i) \cdot \beta(\ell_0\sigma_0 \dots \sigma_{i-1}\ell_i\sigma_i)(\ell_i)$. The probability of a history $\pi = \ell_0\ell_1 \dots \ell_n$ is $\mathbb{P}(\pi) = \sum_{\rho \in L^{-1}(\pi)} \mathbb{P}(\rho)$, which uniquely defines the probabilities of measurable sets of (infinite) plays [Var85]. The safety, reachability, and parity objectives being Borel objectives, they are measurable [Kec95]. We denote by $\mathrm{Pr}_{\ell}^{\alpha,\beta}(\varphi)$ the probability that an objective φ is satisfied by a play starting in ℓ in the game G played with strategies α and β. A randomized strategy α for Player 1 in G is *almost-surely winning* for the objective φ if for all randomized strategies β of Player 2, we have $\mathrm{Pr}_{l_I}^{\alpha,\beta}(\varphi) = 1$. A location $\hat{\ell} \in L$ is *almost-surely winning* for φ if Player 1 has an almost-surely winning randomized strategy α in the game $\hat{G} = \langle L, \hat{\ell}, \Sigma, \Delta \rangle$ where $\hat{\ell}$ is the initial location.

Note that our definition is again asymmetric in the sense that Player 1 has imperfect information about the location of the game while Player 2 has perfect information. While having perfect information does not help Player 2 in the case of surely-winning, it makes Player 2 stronger in the probabilistic case. Recent works [BGG09,GS09] study a symmetric setting where the two players have imperfect information. The decision problems are computationally harder to solve (deciding if a location is almost-surely winning is EXPTIME-complete in our setting, and it becomes 2EXPTIME-complete in the symmetric setting). We choose to present the asymmetric setting for the sake of consistency with the first part of this tutorial, because it is a simpler setting, and because the techniques that we present can be adapted to solve the more general case.

4.2 An algorithm for reachability objectives

We present an algorithm for computing the locations of a reachability game with imperfect information G from which Player 1 has an almost-surely winning strategy. The algorithm can be extended to solve Büchi objectives [CDHR07]. The case of coBüchi and parity objectives remains open.

Extended subset construction First, note that the reduction to games with perfect information G^{K} of Section 3 does not preserve the notion of almost-surely winning. The knowledge-based subset construction for the the game of Fig. 3 is given in Fig. 6. It is easy to see that for all strategies of Player 1, Player 2 can avoid $\{\ell_4\}$ by always choosing from $\{\ell_3, \ell'_3\}$ the transition back to $\{\ell_1\}$. In the original game, this amounts to allow Player 2 to freely "switch" between location ℓ_3 and ℓ'_3. However, against Player 1 strategy playing a and b uniformly at random, Player 2 cannot really decide which location of ℓ_3 or ℓ'_3 is reached, since both have probability $\frac{1}{2}$ to be reached regardless of Player 2 strategy. So, we have to enrich the knowledge-based subset construction to take this phenomenon into account. In the new construction, locations are pairs (s, ℓ) consisting of a cell s and a location $\ell \in s$. To reduce ambiguity, we call such pairs *states*. The cell s encodes the knowledge of Player 1, and the location ℓ keeps track of the choice of Player 2, forcing Player 2 to stick to his choice. Of course, we need to take care that the decisions of Player 1 do not depend on the location ℓ, but only on the cell s.

Given a game structure with imperfect information $G = \langle L, l_I, \Sigma, \Delta, \mathcal{O}\rangle$, we construct the game structure (with perfect information) $H = \mathsf{Knw}(G) = \langle Q, q_I, \Sigma, \Delta_H\rangle$ as follows:

- $Q = \{(s, \ell) \mid \exists o \in \mathcal{O} : s \subseteq o \text{ and } \ell \in s\}$;
- the initial state is $q_I = (\{\, l_I \,\}, l_I)$;
- the transition relation $\Delta_H \subseteq Q \times \Sigma \times Q$ is defined by $((s, \ell), \sigma, (s', \ell')) \in \Delta_H$ iff there is an observation $o \in \mathcal{O}$ such that $s' = \mathsf{post}^G_\sigma(s) \cap o$ and $(\ell, \sigma, \ell') \in \Delta$.

The structure H is called the *extended knowledge-based subset construction* of G. Intuitively, when H is in state (s, ℓ), it corresponds to G being in location ℓ and the knowledge of Player 1 being s. The game $H = \mathsf{Knw}(G)$ is given in Fig. 7 for the game G of Fig. 2. Reachability and safety objectives defined by a target set $\mathcal{T} \subseteq L$ are transformed into an objective of the same type where the target set of states is $\mathcal{T}' = \{(s, \ell) \in Q \mid \ell \in \mathcal{T}\}$. A parity objective φ in G defined by a priority function $pr : L \to \mathbb{N}$ is transformed into a parity objective φ^{Knw} in H using the priority function pr^{Knw} such that $pr^{\mathsf{Knw}}(s, \ell) = pr(o)$ for all $(s, \ell) \in Q$ and $o \in \mathcal{O}$ such that $s \subseteq o$.

Equivalence preserving strategies Since we are interested in observation-based strategies for Player 1 in G, we require that the strategies of Player 1 in H only depend on the sequence of knowledges $s_0 \ldots s_i$ in the sequence of previously visited states $(s_0, \ell_0) \ldots (s_i, \ell_i)$. Two states $q = (s, \ell)$ and $q' = (s', \ell')$ of H are *equivalent*, written $q \approx q'$, if $s = s'$, i.e. when the knowledge of Player 1 is the same in the two states. For a state $q \in Q$, we denote by $[q]_\approx = \{q' \in Q \mid q \approx q'\}$ the $\approx$-equivalence class of q. Equivalence and equivalence classes for plays and labeled histories are defined in the expected way. A strategy α for Player 1 in H is *equivalence-preserving* if $\alpha(\rho) = \alpha(\rho')$ for all labeled histories ρ, ρ' of H such that $\rho \approx \rho'$.

Theorem 6 ([CDHR07]). *For all game structures G with imperfect information, Player 1 has an observation-based almost-surely winning strategy in G for a parity objective φ if and only if Player 1 has an equivalence-preserving almost-surely winning strategy in $H = \mathsf{Knw}(G)$ for the parity objective φ^{Knw}.*

Solving reachability objectives It can be shown that for reachability and Büchi objectives, memoryless strategies are sufficient for Player 1 to almost-surely win the game with perfect information $H = \mathsf{Knw}(G)$. Let $H = \mathsf{Knw}(G) = \langle Q, q_I, \Sigma, \Delta_H \rangle$, let $\mathsf{Reach}(\mathcal{T})$ with $\mathcal{T} \subseteq Q$ be an observable reachability objective in H, and $\approx$ the equivalence relation between states of H as defined above. Player 1 almost-surely wins from the set of states $W \subseteq Q$ if there exist functions $\mathsf{Allow} : Q \to 2^\Sigma$ and $\mathsf{Good} : Q \to \Sigma$ such that for all $q \in W$:

1. for all $q' \approx q$ and for all $\sigma \in \mathsf{Allow}(q)$, $\mathsf{post}^H_\sigma(q') \subseteq W$,
2. in the graph (W, E) with $E = \{(q, q') \in W \times W \mid (q, \mathsf{Good}(q), q') \in \Delta_H\}$, all infinite paths visit a state in $\mathcal{T}$,
3. $\mathsf{Good}(q) \in \mathsf{Allow}(q)$.

Condition 1 ensures that the set W of winning states is never left. This is necessary because if there was a positive probability to leave W, then Player 1 would not win the game with probability 1. Condition 2 ensures that from every state $q \in W$, the target $\mathcal{T}$ is entered with some positive probability (remember that the action $\mathsf{Good}(q)$ is played with some positive probability). Note that if all infinite paths in (W, E) eventually visit $\mathcal{T}$, then all finite paths of length $n = |W|$ do so. Therefore, the probability to reach $\mathcal{T}$ within n rounds can be bounded by a constant $\kappa > 0$, and thus after every n rounds the target set $\mathcal{T}$ is reached with probability at least κ. Since Condition 1 ensures the set W is never left, the probability that the target set has not been visited after $m \cdot n$ rounds is at most $(1 - \kappa)^m$. Since the game is played for infinitely many rounds, the probability to reach $\mathcal{T}$ is $\lim_{m \to \infty} 1 - (1 - \kappa)^m = 1$. By Condition 3, the actions that ensure progress towards the target set can be safely played.

The algorithm to compute the set of states $W \subseteq Q$ from which Player 1 has an equivalence-preserving almost-surely winning strategy for $\mathsf{Reach}(\mathcal{T})$ is the limit of the following computations:

$$\begin{aligned} W^0 &= Q \\ W^{i+1} &= \mathsf{PosReach}(W^i) \text{ for all } i \geq 0 \end{aligned}$$

where the $\mathsf{PosReach}(W^i)$ operator is the limit of the sequence X^j defined by

$$\begin{aligned} X^0 &= \mathcal{T} \\ X^{j+1} &= X^j \cup \mathsf{Apre}(W^i, X^j) \text{ for all } j \geq 0 \end{aligned}$$

where

$$\mathsf{Apre}(W, X) = \{q \in W \mid \exists \sigma \in \Sigma : \mathsf{post}^H_\sigma(q) \subseteq X \text{ and } \forall q' \approx q : \mathsf{post}^H_\sigma(q') \subseteq W\}.$$

The operator $\mathsf{Apre}(W, X)$ computes the set of states q from which Player 1 can ensure that some state of X is visited in the next round with positive probability, while ensuring that W is not left, even if the current state is $q' \approx q$ (because if the game is actually in q, then it means that Player 1 cannot be sure that the game is not in q' with some positive probability).

Note that for $W = Q$, the condition $\mathsf{post}^H_\sigma(q') \subseteq W$ is trivial. Hence, for $W^0 = Q$ the set $W^1 = \mathsf{PosReach}(W^0)$ contains all states from which Player 1 can enforce to reach $\mathcal{T}$ with positive probability. Clearly, this set is an over-approximation of the almost-surely winning states, since from $Q \setminus W^1$ and no matter the strategy of Player 1, the probability that $\mathcal{T}$ is reached is 0. Therefore, we compute in $W^2 = \mathsf{PosReach}(W^1)$ the set of states from which Player 1 can enforce to reach $\mathcal{T}$ with positive probability without leaving W^1, giving a better over-approximation of the set of almost-surely winning states. The iteration continues until a fixpoint is obtained. Note that $W^0 \supseteq W^1 \supseteq W^2 \supseteq \cdots$ is a decreasing sequence, and $X^0 \subseteq X^1 \subseteq X^2 \subseteq \cdots$ is an increasing sequence for each computation of $\mathsf{PosReach}(W^i)$. This algorithm is thus quadratic in the size of H, and exponential in the size of G.

Theorem 7. *The problem of deciding whether Player* 1 *is almost-surely winning in a reachability game with imperfect information is EXPTIME-complete.*

It can be shown that the problem is EXPTIME-hard, see [CDHR07], and thus the algorithm developed above is worst-case optimal. For Büchi objectives, an EXPTIME algorithm can be obtained by substituting the first line of the $\mathsf{PosReach}(W^i)$ operator by $X^0 = \mathcal{T} \cap \mathsf{Spre}(W^i)$ where

$$\mathsf{Spre}(W^i) = \mathsf{Apre}(W^i, W^i) = \{q \in W^i \mid \exists \sigma \in \Sigma \cdot \forall q' \approx q : \mathsf{post}^H_\sigma(q') \subseteq W^i\}.$$

Intuitively, we start the iteration in $\mathsf{PosReach}(W^i)$ with those target states from which Player 1 can force to stay in W^i in the next round. This ensures that whenever a target state is reached (which will happen with probability one), Player 1 can continue to play and will again force a visit to the target set with probability one, thus realizing the objective $\mathsf{Buchi}(\mathcal{T})$ with probability 1.

Antichains for randomized strategies When computing the set of surely-winning locations of a game with imperfect information, we have shown that antichains of sets of locations are a well-suited data-structure. This idea can be extended for computing the sets of almost-surely winning locations of a game with imperfect information.

Let $G = \langle L, l_I, \Sigma, \Delta, \mathcal{O} \rangle$ be a game structure with imperfect information, and let H be its extended knowledge based construction, i.e. $H = \mathsf{Knw}(G) = \langle Q, q_I, \Sigma, \Delta_H \rangle$. We define $\preceq \subseteq Q \times Q$ as $(s, \ell) \preceq (s', \ell')$ iff $s \subseteq s'$ and $\ell = \ell'$. This order has the following properties:

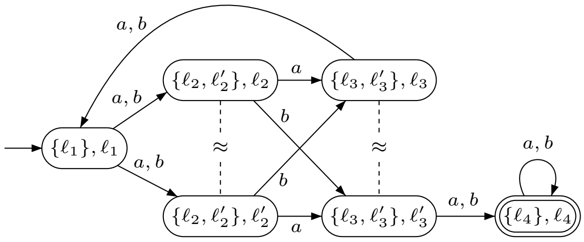

Fig. 7. Game structure $H = \mathsf{Knw}(G)$ (for G of Fig. 2).

- if a state q in H is almost-surely winning for the observable reachability objective $\mathsf{Reach}(\mathcal{T})$, then for all $q' \preceq q$ in H, q' is almost-surely winning for the objective $\mathsf{Reach}(\mathcal{T})$;
- given an observable reachability objective $\mathcal{T}$, all the sets $W^0, W^1, \ldots,$ and $X^0, X^1, \ldots$ are $\preceq$-downward closed.

Exercise 12 Define the operations $\sqcap$, $\sqcup$ for the order $\preceq$. Define the operations $\mathsf{PosReach}$ and Apre so that they operate directly on $\preceq$-antichains. $\square$

Exercise 13 Apply the fixed point algorithm above to compute the almost-surely winning positions in the 3-coin example when Player 2 is allowed to switch coins. Make sure to use antichains during your computations. Extract from the fixed point an almost-surely winning observation-based randomized strategy.

We give the solution to the exercise below. To determine the set of cells in our 3-coin game from which Player 1 has a randomized strategy that allows her to win the game with probability one, we execute our fixed point algorithm. In the computations, we may denote sets of locations by the sequence of their elements, e.g., $\langle 01235 \rangle$ denotes the set $\{0, 1, 2, 3, 5\}$.

$W^0 = \{\langle 012345678 \rangle\} \times \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$. $W^1 = \mathsf{PosReach}(W^0)$ is obtained by the following fixed point computation. $X^0 = (\langle 7 \rangle, 7)$, $X^1 = X^0 \sqcup \mathsf{Apre}(W^0, X^0) = \{\langle 01234578 \rangle\} \times \{0, 1, 2, 3, 4, 5, 7\} \sqcup \{\langle 01235678 \rangle\} \times \{0, 1, 2, 3, 5, 6, 7\} \sqcup \{\langle 01234678 \rangle\} \times \{0, 1, 2, 3, 4, 6, 7\} = X^2$. $W^2 = W^1$. This fixed point tells us that Player 1 has a randomized strategy to win the 3-coin game with probability one. The randomized strategy can be extracted from the antichain W^1 and is as follows. In the first round, all choices of Player 1 are equivalent, so she can play c_1. Then she receives the observation o_2 and update her knowledge to $\{1, 2, 3\}$ which is subsumed by all the elements of the antichain. Then, she plays any action which is associated to those elements with positive probability. The action c_1 is associated with $\{\langle 01235678 \rangle\} \times \{0, 1, 2, 3, 5, 6, 7\}$, action c_2 is associated with $\{\langle 01234578 \rangle\} \times \{0, 1, 2, 3, 4, 5, 7\}$, and action c_3 is associated with $\{\langle 01234678 \rangle\} \times \{0, 1, 2, 3, 4, 6, 7\}$. Let us consider the different cases:

- If the action c_1 is played then the knowledge of Player 1 becomes $\{5, 6\}$. This knowledge is subsumed by all the elements in $\{\langle 01235678 \rangle\} \times \{0, 1, 2, 3, 5, 6, 7\}$ and the action associated with those element is 1. After playing 1 the knowledge of Player 1 is now $\{1, 2\}$. Again this knowledge is subsumed by all the elements of the fixed point so Player 1 can play each action in $\{c_1, c_2, c_3\}$ with positive probability. Note that with this knowledge, it is sufficient to choose play with positive probability in the set of actions $\{c_2, c_3\}$, but this optimization is not necessary if we want to win with probability one, it only reduces the expected time for winning.

– if the action c_2 is played then the knowledge of Player 1 becomes $\{4, 5\}$. This knowledge is subsumed by all the elements in $\{\langle 01234578\rangle\} \times \{0, 1, 2, 3, 4, 5, 7\}$ and the action associated with those element is c_2. After playing c_2 the knowledge of Player 1 is now $\{2, 3\}$. And we can start again playing all actions in $\{c_1, c_2, c_3\}$ with positive probability.
– the reasoning is similar for action c_3.

So we see that our algorithm proposes at each even round to play a action at random then to replay the same action. With this strategy, if Player 1 plays each action with probability $\frac{1}{3}$ when her knowledge is subsumed by $\{1, 2, 3\}$, she has a probability $\frac{1}{3}$ to reach 7 every two rounds and so she wins with probability 1. $\square$

References

[BCD⁺08] D. Berwanger, K. Chatterjee, L. Doyen, T. A. Henzinger, and S. Raje. Strategy construction for parity games with imperfect information. In *Proc. of CONCUR: Concurrency Theory*, Lecture Notes in Computer Science 5201, pages 325–339. Springer-Verlag, 2008.

[BCD⁺09] D. Berwanger, K. Chatterjee, M. De Wulf, L. Doyen, and T. A Henzinger. Alpaga: A tool for solving parity games with imperfect information. In *Proc. of TACAS: Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 5505, pages 58–61. Springer-Verlag, 2009.

[BGG09] N. Bertrand, B. Genest, and H. Gimbert. Qualitative determinacy and decidability of stochastic games with signals. In *Proc. of LICS: Logic in Computer Science*. IEEE Computer Society Press, 2009. To appear.

[BJW02] J. Bernet, D. Janin, and I. Walukiewicz. Permissive strategies: from parity games to safety games. *Inf. Théorique et Applications*, 36(3):261–275, 2002.

[CDHR07] K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin. Algorithms for omega-regular games of incomplete information. *Logical Methods in Computer Science*, 3(3:4), 2007.

[DDR06] M. De Wulf, L. Doyen, and J.-F. Raskin. A lattice theory for solving games of imperfect information. In *Proc. of HSCC 2006: Hybrid Systems—Computation and Control*, Lecture Notes in Computer Science 3927, pages 153–168. Springer-Verlag, 2006.

[EJ91] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *Proc. of FoCS: Foundations of Computer Science*, pages 368–377. IEEE, 1991.

[FJR09] E. Filiot, N. Jin, and J.-F. Raskin. An antichain algorithm for LTL realizability. In *Proc. of CAV: Computer-aided verification*, Lecture Notes in Computer Science 5643, pages 263–277. Springer, 2009.

[FL09] O. Friedmann and M. Lange. The PGSolver collection of parity game solvers. Technical report, Ludwig-Maximilians-Universität München, 2009.

[GS09] V. Gripon and O. Serre. Qualitative concurrent games with imperfect information. In *Proc. of ICALP: Automata, Languages and Programming*, 2009. To appear.

[Hen07] T. A. Henzinger. Games, time, and probability: Graph models for system design and analysis. In *Proc. of SOFSEM (1): Theory and Practice of Computer Science*, Lecture Notes in Computer Science 4362, pages 103–110. Springer, 2007.

[Jur00] M. Jurdziński. Small progress measures for solving parity games. In *Proc. of STACS: Theor. Aspects of Comp. Sc.*, LNCS 1770, pages 290–301. Springer, 2000.

[Kec95] A. Kechris. *Classical Descriptive Set Theory*. Springer, 1995.

[Mar75] D.A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

[Rei84] J. H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and System Sciences*, 29(2):274–301, 1984.

[Saf88] S. Safra. On the complexity of omega-automata. In *Proc. of FOCS: Foundations of Computer Science*, pages 319–327. IEEE, 1988.

[Sch08] S. Schewe. An optimal strategy improvement algorithm for solving parity and payoff games. In *Proc. of CSL: Computer Science Logic*, Lecture Notes in Computer Science 5213, pages 369–384. Springer, 2008.

[Tho95] W. Thomas. On the synthesis of strategies in infinite games. In *Proc. of STACS: Symposium on Theoretical Aspects of Computer Science*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995.

[Tho02] W. Thomas. Infinite games and verification. In *Proc. of CAV: Computer Aided Verification*, Lecture Notes in Computer Science 2404, pages 58–64. Springer, 2002.

[Var85] M.Y. Vardi. Automatic verification of probabilistic concurrent finite-state systems. In *Proc. of FOCS: Foundations of Computer Science*, pages 327–338. IEEE Computer Society Press, 1985.

[Zie98] W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theoretical Computer Science*, 200:135–183, 1998.