In *Proc. of ICALP 2003:* 30th International Colloquium on Automata, Languages, and Programming, Lectures Notes in Computer Science, Springer-Verlag, 2003.

Information Flow in Concurrent Games*

Luca de Alfaro[1] and Marco Faella[1,2]

[1] Department of Computer Engineering, UC Santa Cruz, USA
[2] Dipartimento di Informatica ed Applicazioni, Università degli Studi di Salerno, Italy

Abstract. We consider games where the players have perfect information about the game's state and history, and we focus on the information exchange that takes place at each round as the players choose their moves. The ability of a player to gather information on the opponent's choice of move in a round determines her ability to counteract the move, and win the game. When the game is played between teams, rather than single players, the amount of intra-team communication determines the ability of the team members to coordinate their moves and win the game. We consider games with quantitative bounds on inter-team and intra-team information flow, and we provide algorithms and complexity bounds for their solution.

1 Introduction

We consider repeated games played for an infinite number of rounds on a finite state space [Sha53]. At each round of the game, each player selects a move; the selected moves jointly determine the next state of the game [Sha53,FV97,AHK97]. This process, repeated, gives rise to a play of the game, consisting in the infinite sequence of visited states. We consider safety games, where the goal consists in staying forever in a safe set of states, and reachability games, where the goal consists in reaching a desired subset of states [Tho95,Zie98]. The ability of a player to win such games depends on the information available to the player. In *partial information* games, players have incomplete information about the current state of the game and the past history; computing the sets of winning states for safety and reachability goals is EXPTIME complete [Rei84,KV97]. In this paper, we consider instead games where the players have perfect information about the game's current state and history, and we focus instead on the information exchange that takes place at each round, between players and within players, as the players choose their moves.

We first consider the distinction between *turn-based* and *concurrent* games. Usually, this distinction is defined structurally: a game is *concurrent* if at each state both players may have a choice of moves [FV97,AHK97]. and is turn-based if at each state at most one player has the choice among multiple moves [BL69,GH82,EJ91,Tho95]. We argue that the difference is best captured in terms of information: a game is concurrent if the two players must choose their moves independently, on the basis of the same information, and it is turn-based if one of the two players has full information about the opponent's choice of move when choosing her own move. Indeed, in a game where the players play simultaneously, if one player has full information about the other player's choice of move, the game is in effect turn-based, the player with full information playing second. While this may seem an odd way to play a game, it occurs in hardware design whenever a Moore machine, whose next outputs (the next move) can depend on the current inputs only, is composed with a Mealy machine, whose next outputs (move) can depend both on the current inputs, and on the *next* inputs from other machines. Effectively, the Moore machine chooses first, while the Mealy machine can look at the move chosen by the Moore machine before choosing its own move.

* This research was supported in part by the NSF CAREER award CCR-0132780, the NSF grant CCR-0234690, the ONR grant N00014-02-1-0671, and the MIUR grant "Metodi Formali per la Sicurezza e il Tempo" (MEFISTO).

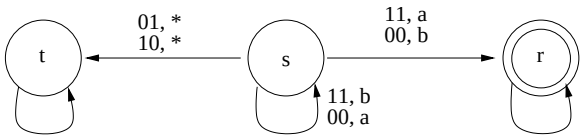

Fig. 1. A concurrent game. An edge label such as $11, a$ indicates that the edge is followed when player 1 chooses '11' and player 2 chooses 'a'. The game starts in s, and the goal is to reach r. States t and r are sink states, without outgoing transitions.

Conversely, whenever in a turn-based game one of the players is prevented from observing the preceding opponent move (along with its effects), the game is effectively concurrent, even though the choice of moves is not simultaneous. Indeed, there would hardly be any concurrent game, if the distinction between concurrent and turn-based were based on truly simultaneous choice, rather than on independent choice under the same information.

Once the distinction between concurrent and turn-based games is phrased in terms of information, concurrent and turn-based games constitute the two extremes in a spectrum of games, which we call *semi-concurrent,* where one player is able to gather a bounded amount of information about the opponent's move before choosing her own. Games where players exchange information in a round have been considered in [dAHM00,dAHM01] to model the interaction of synchronous hardware; in those works the communication scheme is fixed, and is specified together with the game. We consider here games where the amount of information exchanged between players is bounded, but the information content, and the way it is gathered, is left to the discretion of the players. Semi-concurrent games have several applications. In the design of controllers for digital circuits, semi-concurrent games model the case where together with the controller, we can design combinatorial signals that provide information about the next state of the controlled system. Moreover, semi-concurrent games can be used to model games played with untrustworthy adversaries, who can exploit leaked information about our choice of move.

We provide algorithms for solving semi-concurrent games with respect to safety and reachability conditions. We consider both the case when the goal must be attained for all plays (*sure* winning), and the case when the goal must be attained with probability 1 (*almost* winning) [dAHK98], and we consider both the case when the player striving to achieve the goal can spy on the opponent, or is spied upon. We give tight bounds for the complexity of these algorithms, proving that for several combinations of goals, spying, and winning mode (sure or almost), deciding whether a player can win from a state is *NP*-complete. We also show that the larger the amount of information a player can gather about the opponent's choice of move, the more games the player can win; our results enable the determination of the minimum amount of information about the opponent's move that is required in order to win. Finally, investigate the need for randomization in winning reachability games. From [dAHK98] it is known that randomization is needed to win reachability games with probability 1 if the game is concurrent, but not if it is turn-based. We show that randomization in general is always needed to win semi-concurrent reachability games with probability 1, regardless of whether a player is spying or is being spied upon, as long as one of the players does not have perfect information about the other player's choice of move.

Concurrent games can also be seen as the extreme point of another spectrum, concerning the amount of communication *within* a player. While some players are single entities, others are internally composed of separate entities: such composite players are called *teams,* and the entities they comprise are called *team members*. We consider games where the move chosen by a team is a tuple, each team member choosing a component of the tuple. This problem was first studied in [PR79], where it was shown that team games where the players have incomplete information about

the state of the game are in general undecidable. Later, [PR90] and [KV01] considered team games with linear or cyclic communication structure between team members, and showed that solving such games with respect to linear or branching time temporal logic conditions is decidable. These previous works considered games where each team member has a different, partial view of the state of the game. Here, we consider instead the situation in which the team members share complete information about the state of the game, but must coordinate their moves at each round.

A team can readily play any deterministic choice of move that can be played by a single-entity player: each team member simply chooses deterministically the desired move component. However, if the team members cannot communicate while choosing the move components, the team can only play *randomized* distributions of moves that result from the independent randomization of each member's choice. This limits the team's ability to win reachability games, as illustrated by the game of Figure 1. Player 1 can reach r from state s with probability 1 by choosing moves 00 and 11 with probability $1/2$ each, and by choosing moves 01 and 10 with probability 0. However, assume player 1 consists in a two-member team, where each member chooses one of the bits. If the two team members cannot communicate while choosing the bits, the team can only play probability distributions $p(i,j)$ for $i,j \in \{0,1\}$ of the form $p(i,j) = q_1(i)q_2(j)$, where q_1 and q_2 are the distributions chosen by the team members. It is easy to see that team 1 cannot reach r with probability 1 from s using these distributions. If an arbitrary amount of communication can take place in a round between team members, the team can replicate the behavior of a single-entity player: thus, concurrent games constitute the limit case of team games for arbitrary communication.

Here, we study team games where a bounded (possibly 0) amount of communication can take place among team members in a round. Team games model controller-design problems where the controller consists of distributed sub-controllers that can observe the current state of the controlled system, but that have limited communication ability to coordinate their next move. For instance, in synchronous digital circuits, it may not be feasible for the sub-controllers to communicate their next state if they communicate through links that are slow compared to the system clock. Moreover, team games model the real-world situation when members of the same team must coordinate their next move covertly using limited bandwidth.

Team safety games can be solved in the same manner as concurrent safety games, since no randomization is required in the winning strategies. We present algorithms for solving team reachability games with communicating and non-communicating team members, and we provide tight bounds for their complexity, showing in particular that solving non-communicating reachability games is an *NP*-complete problem. While in the case of semi-concurrent games, the more the information is communicated, the more the games that can be won, we show that for team games, a single bit of information communicated between team members is as efficient as complete coordination. On the other hand, we show that probability 1 reachability team games with are in general not determined: if one team cannot win the game with probability 1, this does not imply the existence of a single adversary strategy that prevents the team from winning.

2 Games

For a finite set A, a *probability distribution* on A is a function $p\colon A \mapsto [0,1]$ such that $\sum_{a\in A} p(a) = 1$; we denote the set of probability distributions on A by $\mathcal{D}(A)$. A *game structure* is a tuple $G = (S, M, \Gamma_1, \Gamma_2, \tau)$, where:

- S is a finite set of *states;*
- M is a finite sets of *moves;*
- $\Gamma_1, \Gamma_2 : S \mapsto 2^M \setminus \{\emptyset\}$ are the *move assignments* of the players;
- $\tau\colon S \times M \times M \mapsto \mathcal{D}(S)$ is the *probabilistic transition function.*

At every state $s \in S$, player 1 chooses a move $a \in \Gamma_1(s)$, and player 2 chooses a move $b \in \Gamma_2(s)$; the game then proceeds to state $t \in S$ with probability $\tau(s,a,b)(t)$. For $s \in S$ and $a, b \in M$, we denote by $\delta(s,a,b) = \{t \in S \mid \tau(s,a,b)(t) > 0\}$ the set of possible successors of s when moves a and b are played. A *play* of G is an infinite sequence $s_0, \langle a_0, b_0\rangle, s_1, \langle a_1, b_1\rangle, \ldots$ such that for all $n \geq 0$, we have $a_n \in \Gamma_1(s_n)$, $b_n \in \Gamma_2(s_n)$, and $s_{n+1} \in \delta(s_n, a_n, b_n)$. We denote by $Plays_{s_0}$ the set of plays starting from $s_0 \in S$, and by $Plays = \bigcup_{s_0 \in S} Plays_{s_0}$ the set of all plays of G. A *history* σ is a finite play prefix $\sigma = s_0, \langle a_0, b_0\rangle, s_1, \langle a_1, b_1\rangle, \ldots, s_n$ that terminates in a state; we denote by $last(\sigma)$ the last state s_n of σ. We denote by $Hist_{s_0}$ the set of histories of G starting from s_0, and by $Hist = \bigcup_{s_0 \in S} Hist_{s_0}$ the set of all histories of G. We define the *size* of G to be equal to the number of entries of the transition function δ; specifically, $|G| = \sum_{s \in S} \sum_{a \in \Gamma_1(s)} \sum_{b \in \Gamma_2(s)} |\delta(s,a,b)|$. Given $s \in S$, $Y \subseteq S$ and $B \subseteq M$, it is useful to define $Safe_1(s, Y, B)$ as the set of moves of player 1 that ensure staying in Y when the player 2 chooses moves from B. Formally, $Safe_1(s, Y, B) = \{a \in \Gamma_1(s) \mid \forall b \in B.\delta(s,a,b) \subseteq Y\}$. We define symmetrically $Safe_2(s, Y, A)$.

2.1 Strategies, Winning Conditions, and Games

Let Ω_s be the set of measurable subsets of $Plays_s$, defined as usual (see e.g. [Wil91]). A *family of strategies* $\langle \Upsilon_1, \Upsilon_2, \Pr\rangle$ consists of two sets Υ_1 and Υ_2 of strategies for players 1 and 2, together with a mapping Pr that associates a probability measure $\Pr_s^{\pi_1,\pi_2} : \Omega_s \mapsto [0,1]$ with each initial state s and pair of strategies $\pi_1 \in \Upsilon_1$ and $\pi_2 \in \Upsilon_2$. Thus, $\Pr_s^{\pi_1,\pi_2}(\mathcal{E})$ is the probability that a game starting from $s \in S$ follows a play in $\mathcal{E} \in \Omega_s$ when players 1 and 2 play according to strategies π_1 and π_2, respectively. A probability measure $\Pr_s^{\pi_1,\pi_2}$ over Ω_s gives rise to a set $Outcomes(s, \Pr, \pi_1, \pi_2) \subseteq Plays_s$, of *outcome plays,* consisting of the plays whose finite prefixes can be followed with non-zero probability. Formally, for $\rho \in Plays_s$ and $n > 0$, let $\mathcal{E}(\rho, n) \in \Omega_s$ be the set of plays that agree with ρ up to round n: then, $\rho \in Outcomes(s, \Pr, \pi_1, \pi_2)$ if $\Pr_s^{\pi_1,\pi_2}(\mathcal{E}(\rho, n)) > 0$ for all $n > 0$.

We consider *safety games,* in which the winning condition $\Box R$ consists in remaining forever in a subset $R \subseteq S$ of states, and *reachability games,* in which the winning condition $\Diamond R$ consists in reaching a subset $R \subseteq S$ of states; we define $[\![\Box R]\!] = \{s_0, \langle a_0, b_0\rangle, s_1, \langle a_1, b_1\rangle, \ldots \in Plays \mid \forall n \in \mathbb{N} \,.\, s_n \in R\}$ and $[\![\Diamond R]\!] = \{s_0, \langle a_0, b_0\rangle, s_1, \langle a_1, b_1\rangle, \ldots \in Plays \mid \exists n \in \mathbb{N} \,.\, s_n \in R\}$. A *game* is thus a tuple $(G, \phi, i, \mathcal{M} \mid \Upsilon_1, \Upsilon_2, \Pr)$, composed of a game structure G, a winning condition $\phi \in \{\Box R, \Diamond R\}$, an integer $i \in \{1, 2\}$, a modality $\mathcal{M} \in \{sure, almost\}$, and a family of strategies. Given a family of strategies $\langle \Upsilon_1, \Upsilon_2, \Pr\rangle$ and $\phi \in \{\Box R, \Diamond R\}$, we define the sets $win_1(G, \phi, sure \mid \Upsilon_1, \Upsilon_2, \Pr)$ of player-1 *sure-winning* states and the set $win_1(G, \phi, almost \mid \Upsilon_1, \Upsilon_2, \Pr)$ of player-1 *almost-winning* states as follows [dAHK98]:

Sure-winning. For all $s \in S$, we have $s \in win_1(G, \phi, sure \mid \Upsilon_1, \Upsilon_2, \Pr)$ if there is $\pi_1 \in \Upsilon_1$ such that for all $\pi_2 \in \Upsilon_2$ we have $Outcomes(s, \Pr, \pi_1, \pi_2) \subseteq [\![\phi]\!]$.

Almost-winning. For all $s \in S$, we have $s \in win_1(G, \phi, almost \mid \Upsilon_1, \Upsilon_2, \Pr)$ if there is $\pi_1 \in \Upsilon_1$ such that for all $\pi_2 \in \Upsilon_2$ we have $\Pr_s^{\pi_1,\pi_2}([\![\phi]\!]) = 1$.

The sets of player-2 sure and almost-sure winning states are defined symmetrically. A *winning strategy* is a strategy that ensures victory to a player with the prescribed mode (sure, or almost), for all winning states. Precisely, for $\mathcal{M} \in \{sure, almost\}$, a winning strategy for $(G, \phi, 1, \mathcal{M} \mid \Upsilon_1, \Upsilon_2, \Pr)$ is a strategy $\pi_1 \in \Upsilon_1$ such that, for all $s \in win_1(G, \phi, almost \mid \Upsilon_1, \Upsilon_2, \Pr)$ and all $\pi_2 \in \Upsilon_2$, we have that $Outcomes(s, \Pr, \pi_1, \pi_2) \subseteq \phi$ if $\mathcal{M} = sure$, and $\Pr_s^{\pi_1,\pi_2}([\![\phi]\!]) = 1$ if $\mathcal{M} = almost$. A *spoiling strategy* is an adversary strategy that prevents a player from winning whenever victory cannot be assured. Precisely, for $\mathcal{M} \in \{sure, almost\}$, a spoiling strategy for $(G, \phi, 1, \mathcal{M} \mid \Upsilon_1, \Upsilon_2, \Pr)$ is a strategy $\pi_2 \in \Upsilon_2$ such that, for all $s \notin win_1(G, \phi, almost \mid \Upsilon_1, \Upsilon_2, \Pr)$ and all $\pi_1 \in \Upsilon_1$, we have that $Outcomes(s, \Pr, \pi_1, \pi_2) \not\subseteq \phi$ if $\mathcal{M} = sure$, and $\Pr_s^{\pi_1,\pi_2}([\![\phi]\!]) < 1$ if $\mathcal{M} = almost$. Analogous definitions hold for the winning problems that refer to player 2.

A *game type* is a tuple $(\Delta, i, \mathcal{M} \mid \Upsilon_1, \Upsilon_2, \mathrm{Pr})$ where $i \in 1,2$, $\mathcal{M} \in \{sure, almost\}$, and $\Delta \in \{\Box, \Diamond\}$. We say that a game type is *determined* iff, for all game structures G, player $i \in \{1,2\}$, and $R \subseteq S$, both winning and spoiling strategies exist for $(G, \Delta R, i, \mathcal{M} \mid \Upsilon_1, \Upsilon_2, \mathrm{Pr})$.

2.2 Concurrent Games

In concurrent games, the players choose their moves simultaneously and independently. A *concurrent strategy* for player $i \in \{1,2\}$ is a mapping $\pi_i : Hist \mapsto \mathcal{D}(M)$ that associates with every history σ of the game a probability distribution $\pi_i(\sigma)$ used to select the next move; for all $a \in M$, we require that $\pi_i(\sigma)(a) > 0$, implies $a \in \Gamma_i(last(\sigma))$, ensuring that the strategy selects only moves that are available to the players. We denote by Π_i^c the set of all concurrent strategies for player $i \in \{1,2\}$. Given a strategy $\pi \in \Pi_1^c \cup \Pi_2^c$, we say that π is *memoryless* if for all $\sigma \in Hist$ we have $\pi(\sigma) = \pi(last(\sigma))$, and we say that π is *deterministic* if for all $\sigma \in Hist$ and all $a \in M$ we have $\pi(\sigma)(a) \in \{0,1\}$. An initial state s_0 and a pair of strategies $\pi_1 \in \Pi_1^c$ and $\pi_2 \in \Pi_2^c$ give rise to a probability $\mathrm{Prb}_{s_0}^{\pi_1,\pi_2}$ on histories defined inductively by $\mathrm{PrbC}_{s_0}^{\pi_1,\pi_2}(s_0) = 1$ and, for $n \geq 0$, by $\mathrm{PrbC}_{s_0}^{\pi_1,\pi_2}(s_0,\ldots,s_n,\langle a_n,b_n\rangle,s_{n+1}) = \mathrm{PrbC}_{s_0}^{\pi_1,\pi_2}(s_0,\ldots,s_n)\cdot\pi_1(s_0,\ldots,s_n)(a_n)\cdot\pi_2(s_0,\ldots,s_n)(b_n)\cdot\tau(s_n,a_n,b_n)(s_{n+1})$. These probabilities on histories give rise to a probability measure $\mathrm{PrbC}_{s_0}^{\pi_1,\pi_2}$ on Ω_{s_0} [Wil91]. A *concurrent game* is a game in which the players use concurrent strategies. The winning states of concurrent safety and reachability games can be computed using μ-calculus; we briefly review the approach, as it will be the starting point of the algorithms we will present for other families of strategies.

Safety. The solution of concurrent safety games is entirely classical. The set of winning states can be computed using the *controllable predecessor* operator $CPre_1 : 2^S \mapsto 2^S$, defined for all $X \subseteq S$ by

$$CPre_1(X) = \{s \in S \mid \exists a \in \Gamma_1(s).\forall b \in \Gamma_2(s).\delta(s,a,b) \subseteq X\}.$$

Intuitively, $CPre_1(X)$ consists of all the states from which player 1 can force the game to X in one round; the operator $CPre_2$ for player 2 can be defined symmetrically. For $i \in \{1,2\}$ and $R \subseteq S$ we have then:

$$win_i(G, \Box R, sure \mid \Pi_1^c, \Pi_2^c, \mathrm{PrbC}) = win_i(G, \Box R, almost \mid \Pi_1^c, \Pi_2^c, \mathrm{PrbC}) = \nu X.(R \cap CPre_i(X)), \tag{1}$$

where ν denotes the greatest fixpoint operator. The fixpoint can be computed by Picard iteration by letting $X_0 = S$ and, for $n \geq 0$, $X_{n+1} = R \cap CPre_i(X_n)$; the solution is then given by the limit $\lim_{n\to\infty} X_n$, which can be computed in at most $|S|$ iterations.

Reachability. For mode *sure*, the solution of concurrent reachability games is also classical: for player $i \in \{1,2\}$ and target set $R \subseteq S$, we have

$$win_i(G, \Diamond R, sure \mid \Pi_1^c, \Pi_2^c, \mathrm{PrbC}) = \mu X.(R \cup CPre_i(X)) \tag{2}$$

where μ denotes the least fixpoint operator. The solution can again be computed iteratively, as the limit $\lim_{n\to\infty} X_n$ of the sequence $X_0, X_1, X_2, \ldots$ defined by $X_0 = \emptyset$, and for $n \geq 0$, by $X_{n+1} = R \cup CPre_i(X_n)$. The solution for mode *almost* and player $i \in \{1,2\}$ relies on the two-argument predecessor operator $APre_i : 2^S \times 2^S \mapsto 2^S$ [dAHK98,dAH00]. For $X, Y \subseteq S$, we have $s \in APre_1(Y,X)$ iff player i can force the game to stay in Y, while at the same time forcing a transition to X with positive probability:

$$APre_1(Y,X) = \{s \in S \mid \forall b \in \Gamma_2(s).\exists a \in Safe_1(s,Y,\Gamma_2(s)).\delta(s,a,b) \cap X \neq \emptyset\}.$$

The operator $APre_2$ for player 2 can be defined symmetrically. The set of states from which player $i \in \{1,2\}$ wins with probability 1 with respect to the winning condition $\Diamond R$ can then be computed as a nested fixpoint [dAHK98,dAH00]:

$$win_i(G, \Diamond R, sure \mid \Pi_1^c, \Pi_2^c, \mathrm{PrbC}) = \nu Y.\mu X.(R \cup APre_i(Y, X)). \tag{3}$$

To understand this algorithm, let Y^* set of winning states computed by (3). Since $Y^* = \mu X.(R \cup APre_i(Y^*, X))$, we can write $Y^* = \lim_{k\to\infty} X_k$, where $X_0 = R$, and for $k \geq 0$, where $X_{k+1} = (R \cup APre_i(Y^*, X_k))$. For $k \geq 0$, from $X_{k+1} \setminus X_k$ player i can ensure some probability of going to X_k, while never leaving Y^*. Hence, from any state in Y^*, player i can play a sequence of $|Y^*|$ rounds that ensure that (i) R is reached with positive probability, and (ii) Y^* is left only after R is reached. By repeating this $|Y^*|$-round sequence indefinitely, player i is able to reach R with probability 1. The following theorem summarizes the results on concurrent games.

Theorem 1 [dAHK98] *For all game structures G, players $i \in \{1,2\}$, and sets $R \subseteq S$, the following assertions hold.*

Safety. *For $\mathcal{M} \in \{sure, almost\}$, the set $win_i(G, \Box R, \mathcal{M} \mid \Pi_1^c, \Pi_2^c, PrbC)$ can be computed in time $\mathcal{O}(|G|)$ by (1). The game type $(\Box, i, \mathcal{M} \mid \Pi_1^c, \Pi_2^c, PrbC)$ is determined, and there always exist winning strategies that are both deterministic and memoryless.*

Sure reachability. *The set $win_i(G, \Diamond R, sure \mid \Pi_1^c, \Pi_2^c, PrbC)$ can be computed in time $\mathcal{O}(|G|)$ by (2). The game type $(\Diamond, i, sure \mid \Pi_1^c, \Pi_2^c, PrbC)$ is determined, and there always exist winning strategies that are both deterministic and memoryless.*

Almost sure reachability. *The set $win_i(G, \Diamond R, almost \mid \Pi_1^c, \Pi_2^c, PrbC)$ can be computed in time $\mathcal{O}(|G|^2)$ by (3). The game type $(\Diamond, i, almost \mid \Pi_1^c, \Pi_2^c, PrbC)$ is determined, there always exist winning strategies that are memoryless, but the existence of deterministic winning strategies is not guaranteed.*

3 Semi-concurrent Games

In *semi-concurrent* games, one of the players, when choosing her move, has access to a bounded amount of information about the opponent's choice of move. To model inter-player communication within a round, we introduce semi-concurrent strategies. Let $\Sigma_k = \{1, 2, \ldots, k\}$. A *semi-concurrent strategy* of order $k > 0$ for player $i \in \{1,2\}$ is a pair $\pi_i = \langle \pi_i^s, \pi_i^d \rangle$ consisting of a *spy strategy* $\pi_i^s : Hist \times M \mapsto \mathcal{D}(\Sigma_k)$ and of a *decision strategy* $\pi_i^d : Hist \times \Sigma_k \mapsto \mathcal{D}(M)$, such that for all $\sigma \in Hist$, all $1 \leq j \leq k$ and all $a \in M$, we have that $\pi_i^d(\sigma, j)(a) > 0$ implies $a \in \Gamma_i(last(\sigma))$. The spy strategy represents the method used by player i to gather information about the opponent's move: after the history σ, if the opponent chooses move b, one of the integers in Σ_k is received by player i, according to the distribution $\pi_i^s(\sigma, b)$. Once player i receives an integer n, it chooses the move $a \in M$ with probability $\pi_i^d(\sigma, n)(a)$. Note that semi-concurrent strategies of order 1 are essentially concurrent strategies, as the only symbol carries no information, and semi-concurrent strategies of order $|M|$ give rise to turn-based games, since one of the players can obtain full information about the move of the other. For $k > 0$, we let $\vec{\Pi}_i^k$ be the set of all semi-concurrent strategies of order k for player $i \in \{1,2\}$.

A *semi-concurrent game* is a game where one player uses semi-concurrent strategies, and the other uses concurrent strategies. We arbitrarily fix player 1 to be the player using semi-concurrent strategies. Hence, we consider the families of strategies $\langle \vec{\Pi}_1^k, \Pi_2^c, \mathrm{PrbS} \rangle$ for $k > 0$, where for $\pi_1 = \langle \pi_1^s, \pi_1^d \rangle \in \vec{\Pi}_1^k$ and $\pi_2 \in \Pi_2^c$, $\mathrm{PrbS}_{s_0}^{\pi_1,\pi_2}$ is defined inductively on histories by $\mathrm{PrbS}_{s_0}^{\pi_1,\pi_2}(s_0) = 1$ and,

for $n \geq 0$ and $\sigma \in Hist_{s_0}$, by

$$\mathrm{PrbS}_{s_0}^{\pi_1,\pi_2}(\sigma, \langle a_n, b_n \rangle, s_{n+1}) = \mathrm{PrbS}_{s_0}^{\pi_1,\pi_2}(\sigma) \cdot \pi_2(\sigma)(b_n) \cdot \tau(last(\sigma), a_n, b_n)(s_{n+1}) \cdot \sum_{j \in \Sigma_k} \pi_1^d(\sigma, j)(a_n) \cdot \pi_1^s(\sigma, b_n)(j).$$

Again, these probabilities on histories give rise to a probability measure $\mathrm{PrbS}_{s_0}^{\pi_1,\pi_2}$ on Ω_{s_0}.

In general, both the spy strategy π_i^s and the decision strategy π_i^d can be history-dependent and randomized. For $i \in \{1,2\}$, we say that a decision strategy π_i^d of order k is *memoryless* if $\pi_i^d(\sigma, j) = \pi_i^d(last(\sigma), j)$ for all $\sigma \in Hist$ and all $j \in \{1, 2, \ldots, k\}$, and we say that π_i^d is *deterministic* if $\pi_i^d(\sigma, j)(a) \in \{0, 1\}$ for all $\sigma \in Hist$, all $j \in \{1, 2, \ldots, k\}$, and all $a \in M$. Analogously, for $i \in \{1,2\}$ and $k > 0$, we say that a spy strategy π_i^s is *memoryless* if $\pi_i^s(\sigma, b) = \pi_i^s(last(\sigma), b)$ for all $\sigma \in Hist$ and all $b \in M$, and we say that π_i^s is *deterministic* if $\pi_i^s(\sigma, b)(j) \in \{0, 1\}$, for all $\sigma \in Hist$, all $j \in \{1, 2, \ldots, k\}$, and all $b \in M$. We say that a semi-concurrent strategy $\pi_1 = \langle \pi_1^s, \pi_1^d \rangle$ is *memoryless* (respectively *deterministic*) if both π_1^s and π_1^d are memoryless (resp. *deterministic*).

3.1 Semi-concurrent Safety Games

Since the information in each round flows from player 2 to player 1, the solution of semi-concurrent safety games is not symmetrical with respect to players 1 and 2. In order to win a safety game, player 1 must be able at each round to issue a move that keeps the game into the safe region, regardless of the opponent's move. If player 1 can use an order-k semi-concurrent strategy, the best approach consists, at each round, in partitioning the moves of player 2 in k groups, and in using the spy strategy to communicate the group of the move chosen by player 2. If player 1 has a move for each of the k groups that ensures the game stays in the safe region, player 1 can win the game. Hence, we define the *order-k semi-concurrent* predecessor operator $SPre_1^k : 2^S \mapsto 2^S$ as follows. A *k-partition* of a set A consists in k subsets $A_1, \ldots, A_k \subseteq A$ such that $A = \bigcup_{j=1}^k A_j$.

> *For all $X \subseteq S$ and $s \in S$, we have $s \in SPre_1^k(X)$ iff there is a k-partition $B_1, \ldots, B_k$ of $\Gamma_2(s)$ and $a_1, \ldots, a_k \in \Gamma_1(s)$ (possibly not all distinct) such that, for all $b \in \Gamma_2(s)$, if $b \in B_j$ then $\delta(s, a_j, b) \subseteq X$.*

Thus, when player 2 chooses move $b \in B_j$, player 1 can force the game to X by playing move a_j.

When player 2 tries to win a safety game using a concurrent strategy, the fact that player 1 uses a concurrent strategy, or a semi-concurrent strategy, is irrelevant: in fact, if player 2 had a move that guaranteed safety when not spied upon, the same move will guarantee safety also when spied upon by player 1. Thus, the game can be solved with the usual controllable predecessor operator $CPre_2$. The following theorem summarizes the results about semi-concurrent safety games.

Theorem 2 *For all game structures G, sets $R \subseteq S$, $k > 1$, and $\mathcal{M} \in \{sure, almost\}$, the following assertions hold:*

1. *We have $win_1(G, \Box R, \mathcal{M} \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS) = \nu X.(R \cap SPre_1^k(X))$; the set can be computed in time $\mathcal{O}(|G|^k)$. There are always winning strategies that are both deterministic and memoryless.*
2. *We have $win_2(G, \Box R, \mathcal{M} \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS) = win_2(G, \Box R, \mathcal{M} \mid \Pi_1^c, \Pi_2^c, Prb)$, and as in the case of concurrent games, the above sets can be computed in time $\mathcal{O}(|G|)$ by (1). There are always winning strategies that are both deterministic and memoryless.*

As for determinacy, the following theorem holds.

Theorem 3 *For $i \in \{1,2\}$, the game type $(\Box, i, \mathcal{M} \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS)$ is determined.*

If we consider the order $k > 0$ to be part of the input, we obtain the following NP-completeness result. The result is proved by reducing Vertex Cover [GJ79] to the problem of computing $SPre_1^k(\cdot)$.

Theorem 4 *Given input (k, G, R), the membership problem in $win_1(G, \Box R, \mathcal{M} \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS)$ is NP-complete.*

3.2 Player One Reachability Games

In order to win a reachability game with mode *sure*, player 1 must guarantee that, at each round, deterministic progress is made toward the goal. Hence, the solution of player 1 reachability games for mode *sure* uses again the controllable predecessor operator $SPre_1^k$.

When the desired winning mode is *almost*, rather than *sure*, the game is solved using a semi-concurrent version $SAPre_1^k$ of the operator $APre_1$ for concurrent games, for $k > 0$. Again, the best approach for player 1 consists in partitioning the adversary's moves into k subsets $B_1, \ldots, B_k$, using the spy strategy to learn the subset in which the move played by player 2 lies. Thus, if the conditions of operator $APre_1$ hold for each subset $B_1, \ldots, B_k$ of moves, then player 1 is able to ensure probabilistic progress toward the goal. The definition is as follows.

> *Given two sets $X, Y \subseteq S$ and a state $s \in S$, we say that $s \in SAPre_1^k(Y, X)$ if and only if there exist a k-partition $B_1, \ldots, B_k$ of $\Gamma_2(s)$ such that, for all $b \in \Gamma_2(s)$, if $b \in B_j$ then there is $a \in Safe_1(s, Y, B_j)$ such that $\delta(s, a, b) \cap X \neq \emptyset$.*

Theorem 5 *For all game structures G, $R \subseteq S$, and $k > 1$, the following assertions hold:*

1. *We have $win_1(G, \Diamond R, sure \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS) = \mu X.(R \cup SPre_1^k(X))$; the fixpoint can be computed in time $\mathcal{O}(|G|^k)$. There always exist deterministic and memoryless winning strategies.*
2. *We have $win_1(G, \Diamond R, almost \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS) = \nu Y.\mu X.(R \cup SAPre_1^k(Y, X))$. Deciding whether a state belongs to the above fixpoint is NP-complete in $|G|$. There always exist memoryless winning strategies, but there may not be deterministic winning strategies.*

The theorem states, in particular, that computing the set of winning states of a player 1 semi-concurrent reachability game is an NP-complete problem even when $k = 2$, i.e., when the spy strategies can communicate at most 1 bit of information about player 2's choice of move. The NP-completeness result is proved by reducing 3-SAT to the problem of deciding $s \in SAPre_1^2(\cdot, \cdot)$. Then, it is shown that the result for $k = 2$ implies the result for all $k > 1$.

Theorem 6 *The following assertions hold:*

1. *The game type $(\Diamond, 1, sure \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS)$ is determined.*
2. *The game type $(\Diamond, 1, almost \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS)$ is* not *determined. On the other hand, if only memoryless spy strategies are considered, the latter game type is determined.*

The following theorem states that the more information is available to player 1, the more games player 1 can win.

Theorem 7 *For $\triangle \in \{\Diamond, \Box\}$ and $\mathcal{M} \in \{sure, almost\}$, if $k_1 > k_2 > 0$ then there is a game structure G and a subset of states $R \subseteq S$ such that*

$$win_1(G, \triangle R, \mathcal{M} \mid \vec{\Pi}_1^{k_2}, \Pi_2^c, PrbS) \subsetneq win_1(G, \triangle R, \mathcal{M} \mid \vec{\Pi}_1^{k_1}, \Pi_2^c, PrbS).$$

The theorem is proved by constructing a game where player 1 has moves $a_1, \ldots, a_m$, and player 2 has moves $b_1, \ldots, b_m$. In order to win, player 1 must match each move b_j, for $1 \leq j \leq m$, with move a_j. Obviously, player 1 can do this only in a semi-concurrent game of order $k \geq m$. Since semi-concurrent games of order 1 are concurrent games, the following corollary follows.

Corollary 1 *For $\triangle \in \{\Diamond, \Box\}$, $\mathcal{M} \in \{sure, almost\}$, and $k > 1$, there is a game structure G and a subset of states $R \subseteq S$ such that*

$$win_1(G, \triangle R, \mathcal{M} \mid \Pi_1^c, \Pi_2^c, PrbS) \subsetneq win_1(G, \triangle R, \mathcal{M} \mid \vec{\Pi}_1^{k_1}, \Pi_2^c, PrbS).$$

3.3 Player Two Reachability Games

We now consider the case when player 2 has to reach a region R, while player 1 is able to gather information about her choice of moves using a semi-concurrent strategy. Again, for winning mode *sure*, the solution of the game coincides with that of concurrent games. Informally, if player 2 must ensure that *all* outcome plays reach R (as opposed to a set of outcome plays with measure 1), player 1 does not need to get information about player 2's choice of moves: he can just guess it. For mode *almost*, semi-concurrent reachability games of order k is solved using a predecessor operator $VAPre_2^k : 2^S \times 2^S \mapsto 2^S$, that plays the same role as the operator $APre_2$ for concurrent games: for $X \subseteq Y \subseteq S$, the set $VAPre_2^k(Y, X) \subseteq S$ consists of the states from which player 2 can ensure a positive probability of going to X in one round, while staying in Y.

> *For $s \in S$ and $X, Y \subseteq S$, we have $s \in VAPre_2^k(Y, X)$ iff for all k-partitions $B_1, \ldots, B_k$ of $Safe_2(s, Y, \Gamma_2(s))$, there is $j \in \{1, \ldots, k\}$ such that:*
>
> $$\forall a \in \Gamma_1(s). \exists b \in B_j. \delta(s, a, b) \cap X \neq \emptyset.$$

The idea is as follows: the best strategy for player 2 at a state $s \in S$ consists in playing uniformly at random all moves in $Safe_2(s, Y, \Gamma_2(s))$; the above definition ensures that, if $s \in VAPre_2^k(Y, X)$, then a transition to X happens with positive probability, regardless of the partition of $\Gamma_2(s)$ chosen by player 1's spy strategy. We are now ready to state the following:

Theorem 8 *For all game structures G, $R \subseteq S$, and $k > 1$, the following holds:*

1. *We have $win_2(G, \Diamond R, sure \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS) = win_1(G, \Diamond R, sure \mid \Pi_1^c, \Pi_2^c, Prb)$; as in the case of concurrent games, the above sets can be computed in time $\mathcal{O}(|G|)$ by (2). There are are always memoryless and deterministic winning strategies.*
2. *We have $win_2(G, \Diamond R, almost \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS) = \nu Y. \mu X. (R \cup VAPre_2^k(Y, X))$; the fixpoint can be computed in time $\mathcal{O}(|G|^k)$. There are always winning strategies that are memoryless, but there may not be deterministic winning strategies.*

Theorem 9 *For $\mathcal{M} \in \{sure, almost\}$, the game type $(\Diamond, 1, \mathcal{M} \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS)$ is determined.*

If we consider the order $k > 0$ to be part of the input, we obtain the following result (compare with Theorem 4). The result is proved by reducing Vertex Cover to non-membership in $VAPre_2^k(\cdot, \cdot)$.

Theorem 10 *Given input (k, G, R), the membership problem in $win_2(G, \Box R, almost \mid \vec{\Pi}_1^k, \Pi_2^c, PrbS)$ is co-NP-complete.*

4 Team Games

In *team games,* one of the players does not consist of a single player, but rather of a team, composed by *members.* At each state, each team member can choose a move; the resulting team move is a tuple consisting of the choices of the members. We assume that each team member has complete information about the past history of the game, and we explicitly model the coordination used by the team members in choosing their moves. In particular, we consider both *non-communicating* and

communicating strategies for team members. When using non-communicating strategies, the team members must select their moves simultaneously and independently not only from the opposing player, but also from each other. When using communicating strategies, the team members are allowed to communicate some information before choosing the moves.

Formally, a (m_1, m_2)-*team game structure* is a concurrent game structure $G = (S, M, \Gamma_1, \Gamma_2, \tau)$, where $M = \prod_{j=1}^{m_1} M_1^j \cup \prod_{j=1}^{m_2} M_2^j$, and for all $s \in S$, $\Gamma_1(s) = \prod_{j=1}^{m_1} \Gamma_1^j(s)$ and $\Gamma_2(s) = \prod_{j=1}^{m_2} \Gamma_2^j(s)$. Intuitively, the game is played by teams 1 and 2, and the team $i \in \{1,2\}$ is composed by members $1, \ldots, m_i$. At a state $s \in S$, the set $\Gamma_i^j(s)$ contains the moves that can be chosen by member $j \in \{1, \ldots, m_i\}$. A move of team i is thus a tuple $\langle a_1, \ldots, a_{m_i} \rangle$, consisting of the choices of the members. We let Π_1^c and Π_2^c be the sets of concurrent strategies for G, as defined in Section 2.

4.1 Team Games with Non-communicating Strategies

A *non-communicating team strategy* for team $i \in \{1,2\}$ is a function $\bar{\pi}_i : \mathit{Hist} \mapsto \mathcal{D}(M)$ that prescribes for each game history a move distribution to be played by the team. Since we forbid communication between the members of the team, the distributions chosen by the team members must be mutually independent. Hence, we require that there are m_i functions $\pi_i^j : \mathit{Hist} \mapsto \mathcal{D}(M_i^j)$, for $1 \leq j \leq m_i$, such that $\bar{\pi}_i(\sigma)(\langle a_1, \ldots, a_{m_i} \rangle) = \prod_{j=1}^{m_i} \pi_i^j(\sigma)(a_j)$ for all $\sigma \in \mathit{Hist}$ and all $\langle a_1, \ldots, a_{m_i} \rangle \in \Gamma_i(s)$. We denote by $\bar{\Pi}_i$ the set of all team strategies for team $i \in \{1,2\}$. In a (m_1, m_2)-team game we have $\bar{\Pi}_i \subseteq \Pi_i^c$, and the inclusion is strict whenever $m_i > 1$ and there is a state $s \in S$ with $|\Gamma_i(s)| > 1$. The probability measure $\mathrm{PrbT}_s^{\pi_1, \pi_2}$ for $s \in S$ and $\pi_1 \in \bar{\Pi}_1$, $\pi_2 \in \bar{\Pi}_2$ can then be defined in a straightforward way, yielding the family of (m_1, m_2)-team strategies $\langle \bar{\Pi}_1, \bar{\Pi}_2, \mathrm{PrbT} \rangle$.

A team game with non-communicating strategies (also called *non-communicating team game*) differs from a concurrent game because, at each state, each team must choose a probability distribution over moves that can be written as the product of the distributions chosen by the team members. Since deterministic distributions can always be written in product form (if team i wants to play tuple $\langle a_1, \ldots, a_{m_i} \rangle$, each member $1 \leq j \leq m_i$ simply plays a_j), for the games where the existence of deterministic winning strategies is assured, the winning states of non-communicating team games coincide with the winning states of concurrent games.

Theorem 11 *For all $m_1, m_2 > 0$, all (m_1, m_2)-team game structures G, sets $R \subseteq S$, and teams $i \in \{1,2\}$, the following assertions hold:*

1. *For $\mathcal{M} \in \{sure, almost\}$, we have $win_i(G, \Box R, \mathcal{M} \mid \bar{\Pi}_1, \bar{\Pi}_2, PrbT) = win_i(G, \Box R, \mathcal{M} \mid \Pi_1^c, \Pi_2^c, Prb)$. There are always winning strategies that are memoryless and deterministic.*
2. *We have $win_i(G, \Diamond R, sure \mid \bar{\Pi}_1, \bar{\Pi}_2, PrbT) = win_i(G, \Diamond R, sure \mid \Pi_1^c, \Pi_2^c, Prb)$. There are always winning strategies that are memoryless and deterministic.*

Corollary 2 *For $\mathcal{M} \in \{sure, almost\}$ and $i \in \{1,2\}$, the game type $(\Box, i, \mathcal{M} \mid \bar{\Pi}_1, \bar{\Pi}_2, PrbT)$ is determined. The game type $(\Diamond, i, sure \mid \bar{\Pi}_1, \bar{\Pi}_2, PrbT)$ is also determined.*

Hence, the interesting problem in team games consists in solving reachability games with probability 1, where the winning strategies need randomization in the general case [dAHK98]. Such games can be solved using the predecessor operator $TAPre_1$, defined as follows. In the following, we call *cube* any set $C \in \prod_{j=1}^{m_1} (2^{M_1^j} \setminus \{\emptyset\})$.

> *Given two sets $X, Y \subseteq S$ and a state $s \in S$, we say that $s \in TAPre_1(Y, X)$ if and only if there exists a cube $C \in \prod_{j=1}^{m_1} (2^{M_1^j} \setminus \{\emptyset\})$. such that:*
>
> $$\forall b \in \Gamma_2(s). (\forall a \in C. \delta(s, a, b) \subseteq Y \text{ and } \exists a \in C. \delta(s, a, b) \subseteq X).$$

Theorem 12 *For all $(2,1)$-team game structures G and sets $R \subseteq S$, we have $win_1(G, \Diamond R, almost \mid \bar{\Pi}_1, \bar{\Pi}_2, PrbT) = \nu Y.\mu X.(R \cup TAPre_1(Y, X))$, and membership in this fixpoint is an NP-complete problem. Moreover, there are always winning strategies that are memoryless, but there may not be deterministic winning strategies.*

In order to prove the *NP*-hardness, a non-trivial reduction is developed, transforming the classical 3-CNF-SAT problem to membership in $TAPre_1(\cdot, \cdot)$.

By means of a counterexample, the following can be shown.

Theorem 13 *For $i \in \{1, 2\}$, the game type $(\Diamond, i, almost \mid \bar{\Pi}_1, \bar{\Pi}_2, PrbT)$ is not determined.*

4.2 Team Games with Communication

In this section we consider the case in which members of the same team are allowed to communicate some information in each turn of the game. To simplify the notation, rather than considering arbitrary flows of information between team members, we consider a team composed of only two members. This special case suffices to capture the interesting features of the general case.

Formally, given a $(2,1)$-team game structure, a *communicating team strategy of order $k > 0$* for team 1 is a function $\hat{\pi}_1^k : Hist \mapsto \mathcal{D}(M)$ subject to the following requirements. There are three functions $\pi_1^t : Hist(G) \times \Sigma_k^* \mapsto \mathcal{D}(\Sigma_k)$, $\pi_1^1 : Hist(G) \times \Sigma_k^* \mapsto \mathcal{D}(M_1^1)$, and $\pi_1^2 : Hist(G) \times \Sigma_k^* \mapsto \mathcal{D}(M_1^2)$. The function π_1^t represents the generation of random symbols; these symbols are then communicated to the team members, which then use the functions π_1^1 and π_1^2 to choose their moves, on the basis of the game history and of teh received symbols. Note that if we consider both the generation of symbols π_1^t and the choice π_1^i to be done by team member i, for $i \in \{1, 2\}$, then communication effectively takes place from team member i to team member $3 - i$. For $n \geq 0$, $\langle r_0, \ldots, r_n \rangle \in \Sigma_k^*$ and $\langle s_0, \ldots, s_{n+1} \rangle \in Hist$, we set

$$Pr_{s_0}^{\pi_1^t}(\langle r_0, \ldots, r_n \rangle \mid \langle s_0, \ldots, s_n \rangle) = \prod_{j=0}^{n} \pi_1^t(\langle s_0, \ldots, s_j \rangle, \langle r_0, \ldots, r_{j-1} \rangle)(r_j),$$

with the convention that $r_0, \ldots, r_{-1} = \varepsilon$ (where ε denotes the empty string). Then, for all $n \geq 0$, all $\sigma = \langle s_0, \ldots, s_n \rangle \in Hist_{s_0}$, all $a_1 \in M_1^1$, and all $a_2 \in M_1^2$ we define the overall team strategy $\hat{\pi}_1^k$ by

$$\hat{\pi}_1^k(\sigma)(\langle a_1, a_2 \rangle) = \sum_{\rho \in \Sigma_k^n} Pr_{s_0}^{\pi_1^t}(\rho \mid \sigma) \cdot \pi_1^1(\sigma, \rho)(a_1) \cdot \pi_1^2(\sigma, \rho)(a_2).$$

We denote by $\widehat{\Pi}_i^k$ the set of communicating team strategies of order k for team i, and by PrbTC the probability measure on Ω induced by $\widehat{\Pi}_1^k$ and Π_2^c, defined as for concurrent games. We prove that, for all $k > 1$, team 1 has a winning strategy if and only if player 1 has a winning strategy in the corresponding concurrent game.

Theorem 14 *For all $(2,1)$-team game structures G, sets $R \subseteq S$, and $k > 1$, we have $win_1(G, \Diamond R, almost \mid \widehat{\Pi}_1^k, \Pi_2^c, PrbTC) = win_1(G, \Diamond R, almost \mid \Pi_1^c, \Pi_2^c, Prb)$.*

This theorem implies that, from the point of view of winning reachability games with probability 1, being able to communicate 1 bit per round, or even just sharing 1 bit of random information, is as good as the ability to communicate an arbitrary amount of information.

We outline the idea of the proof for $k = 2$, the case for a one-bit channel. Consider the solution $Y^* = win_1(G, \Diamond R, almost \mid \Pi_1^c, \Pi_2^c, \mathrm{Prb}) = \mu X.(R \cup APre_i(Y^*, X))$ of the concurrent reachability game. As remarked in Section 2.2, if the team members could coordinate perfectly, they could play a sequence of $|Y^*|$ moves that leads to R with positive probability, and that does not leave Y^* otherwise. The problem, here, is that the two team members cannot communicate perfectly. To

communicate a sequence of $|Y^*|$ moves, they need $|Y^*| \cdot \lceil \log |M| \rceil$ bits. However, the two team members can play a *deterministic* strategy to stay in Y^*.

The winning strategy of the team thus consists in the alternation of two phases, a *planning* phase, and an *execution* phase. In the planning phase, which lasts $|Y^*| \cdot \lceil \log |M| \rceil$ rounds, the two team members play a deterministic strategy to stay in Y^*. In the meantime, the symbol generator generates $|Y^*| \cdot \lceil \log |M| \rceil$ random bits (notice that the bits are not visible to the adversary). In the subsequent execution phase, which lasts $|Y^*|$ rounds, the team members use the sequence of bits to coordinate their actions, and play at each $s \in Y^*$ all the moves in $Safe_1(s, Y^*, \Gamma_2(s))$ with strictly positive probability. It is easy to see that each cycle consisting in a planning and in an execution phase results in (i) either reaching R, or (ii) in remaining in Y^*, and outcome (i) occurs with positive probability. This leads to the result. The result can be easily extended to the case of more than two team members.

References

[AHK97] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proc. 38th IEEE Symp. Found. of Comp. Sci.*, pages 100–109. IEEE Computer Society Press, 1997.

[BL69] J.R. Büchi and L.H. Landweber. Solving sequential conditions by finite-state strategies. *Trans. Amer. Math. Soc.*, 138:295–311, 1969.

[dAH00] L. de Alfaro and T.A. Henzinger. Concurrent omega-regular games. In *Proc. 15th IEEE Symp. Logic in Comp. Sci.*, pages 141–154, 2000.

[dAHK98] L. de Alfaro, T.A. Henzinger, and O. Kupferman. Concurrent reachability games. In *Proc. 39th IEEE Symp. Found. of Comp. Sci.*, pages 564–575. IEEE Computer Society Press, 1998.

[dAHM00] L. de Alfaro, T.A. Henzinger, and F.Y.C. Mang. The control of synchronous systems. In *CONCUR 00: Concurrency Theory. 11th Int. Conf.*, volume 1877 of *Lect. Notes in Comp. Sci.*, pages 458–473. Springer-Verlag, 2000.

[dAHM01] L. de Alfaro, T.A. Henzinger, and F.Y.C. Mang. The control of synchronous systems part II. In *CONCUR 01: Concurrency Theory. 12th Int. Conf.*, volume 2154 of *Lect. Notes in Comp. Sci.*, pages 566–581. Springer-Verlag, 2001.

[EJ91] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *Proc. 32nd IEEE Symp. Found. of Comp. Sci.*, pages 368–377. IEEE Computer Society Press, 1991.

[FV97] J. Filar and K. Vrieze. *Competitive Markov Decision Processes.* Springer-Verlag, 1997.

[GH82] Y. Gurevich and L. Harrington. Trees, automata, and games. In *Proc. 14th ACM Symp. Theory of Comp.*, pages 60–65. ACM Press, 1982.

[GJ79] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* Freeman and Co., 1979.

[KV97] O. Kupferman and M.Y. Vardi. Synthesis with incomplete informatio. In *2nd International Conference on Temporal Logic*, pages 91–106, Manchester, July 1997.

[KV01] O. Kupferman and M.Y. Vardi. Synthesizing distributed systems. In *Proc. 16th IEEE Symp. on Logic in Computer Science*, July 2001.

[PR79] G.L. Peterson and J.H. Reif. Multiple-person alternation. In *Proc. 20th IEEE Symp. Found. of Comp. Sci.*, pages 348–363, 1979.

[PR90] A. Pnueli and R. Rosner. Distributed-reactive systems are hard to synthesize. In *Proc. 31th IEEE Symp. Found. of Comp. Sci.*, pages 746–757, 1990.

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

[Sha53] L.S. Shapley. Stochastic games. *Proc. Nat. Acad. Sci. USA*, 39:1095–1100, 1953.

[Tho95] W. Thomas. On the synthesis of strategies in infinite games. In *Proc. of 12th Annual Symp. on Theor. Asp. of Comp. Sci.*, volume 900 of *Lect. Notes in Comp. Sci.*, pages 1–13. Springer-Verlag, 1995.

[Wil91] D. Williams. *Probability With Martingales.* Cambridge University Press, 1991.

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