Information Tracking in Games on Graphs

Dietmar Berwanger[1] and Łukasz Kaiser[2]

[1] LSV, CNRS & ENS Cachan, France
[2] RWTH Aachen, Germany

Abstract When seeking to coordinate in a game with imperfect information, it is often relevant for a player to know what other players know. Keeping track of the information acquired in a play of infinite duration may, however, lead to infinite hierarchies of higher-order knowledge. We present a construction that makes explicit which higher-order knowledge is relevant in a game and allows us to describe a class of games that admit coordinated winning strategies with finite memory.

1 Introduction

The paradigm of games is intrinsically linked to uncertainty. Most fundamental concepts in non-cooperative game theory are aimed at reducing the uncertainty that a player may have about the intended choices of other players. Ideally, the postulate that all participants in a game act rationally allows to deduce meaningful predictions about their choices. However, the outcome of a decision situation is often interrelated not only with decisions of fellow players, but also with moves of nature which does not subsume to rationality assumptions.

In this paper, we discuss a model of games with imperfect information where several players coordinate against nature. The games are played by moving for an unbounded number of rounds along the edges on a graph. To make their choice, the players receive observations that may allow them to reduce the uncertainty about the current position of the play without identifying it, in general; observations received by different players are unrelated. The game graph has finitely many positions among which some are marked as forbidden. The goal of the grand coalition of players is to never reach a forbidden state, regardless of the moves of nature. Our model is a variant of the one introduced by Peterson and Reif [10] which incorporates imperfect information into the well-studied framework of infinite Gale-Stewart games on finite graphs [5, 9] and has important applications to various fields of computing science, notably distributed control theory of discrete-event systems and the verification of reactive systems; for a comprehensive account on algorithmic questions regarding this basic model, see [4].

Games on finite graphs are succinct descriptions of infinite extensive games. In particular, the knowledge that a player may acquire in the course of a play under imperfect information is not represented explicitly. Such an explicit representation cannot be straightforward as the amount of information generated during an infinite play is unbounded; already to maintain a counter for the number of played rounds may require unbounded resources. On the other hand, as the game is supported by a finite graph, i.e., only finitely many factual situations can occur, one may hope to identify a reduced, possibly finite, amount of relevant knowledge that is sufficient to play optimally.

Our investigation addresses two questions, one of epistemic and one of more technical nature.

(i) Which information do players need to acquire during a play to ensure a win against nature in coordination with the other players?

(ii) In which games is the grand coalition able to coordinate on finite-state winning strategies, and how to construct them?

We should emphasise that our problem setting assumes explicit pre-play coordination among the players of the grand coalition. This is in contrast to the question of how a player can do his part to achieve an outcome for which he shares a common interest with the other players, considered, for instance in [2].

The pre-play coordination setting has pragmatic justifications and a relatively long history in computing science. The initial model of games with imperfect information on graphs introduced in [10] was conceived as an extension of the alternating Turing-machine model of Chandra, Kozen, and Stockmeyer [3], firstly from the existential-universal duality to more than two players, and then beyond the assumption of perfect information. The manifest motivation for these extensions was to capture concurrent computation among processes equipped with shared and private variables in terms of games. The technical concern of the paper was to synthesise a winning strategy, i.e. a program for a process – or a set of programs for a system composed of several processes under the control of the system designer– in such a way that it guarantees a successful global run regardless of the variable assignments that cannot be observed. The authors proved that the problem can be solved algorithmically for one player against nature, and presented an instance with two plyers against nature where the problem already becomes undecidable.

After a period of discouragement in face of the general undecidability result, the research on games with partial information experienced increasing attention since the late eighties with the foundation of supervisory control theory ([12, 14, 15, 17]) at the one hand, and the development of automata-theoretic methods for the verification of reactive systems ([11, 16, 8, 7]). The modelling paradigm supporting the game-theoretic approach to these frameworks consists in associating the internal states of a computational component with decision points and the possible transitions in the system with moves in a sequential game. Then, the specification of correct behaviour of a component is interpreted as a utility function of the associated player and the question of whether the component can be implemented to comply with the specification translates into the question of whether the player can ensure to win in the resulting game. As the model targets non-terminating systems, the plays are typically unbounded, or infinite; to cope with such games, utility functions are in general described by automata and often feature only two values: win or lose. The basic solution concept is that of a finite-state winning strategy. For systems involving several component, the focus lies on *distributed* winning strategies; the main problem consists in constructing a profile of strategies such that their interplay results in a win. Thus, at the moment of choosing a strategy the controllable components are seen as one player in strict competition with the uncontrollable environment. However, when the strategy is executed, the components act as if they were individual players, none has access to the observation of the others. Accordingly, (distributed) finite-state winning strategies for the controllable part of a system translate into implementations of components that ensure correct behaviour of the system regardless of the behaviour of the environment.

Imperfect information arises naturally in computational models as an effect of locality of components, internal variables, privacy constraints, or incomplete specifications. One highly non-trivial issue regarding uncertainty in computational systems is whether the different players have the means to synchronise their moves. The general approach of solving games with imperfect information in the synchronous setting is the power-set construction proposed in [13] for solving games for one player against nature. Several system architectures have been identified, where the problem for several players can be reduced to the one player setting, with the general pattern that the information available to players can be ordered. See [18, 1, 8], for more background. Asynchronous composition raises additional difficulties already in the case of one player against nature performing unobservable moves, and thus the classes of games for which distributed winning strategies can be effectively constructed are even more restrictive. A comprehensive classification of the current state of research is presented in [6].

Our objective is to explore why games that involve two or more players with imperfect information may be undecidable in the synchronous setting. The determining factor in the undecidability arguments presented in [13] and [1] seems to be that along the course of a play, the agents need to keep track of increasingly many details about the current state of the game. What makes the process intractable is not the uncertainty that a player has about the effect of previous moves on his own state of knowledge, but the fact that he needs to take into account what other players know, and what they know about the knowledge of others and so on. Indeed, the information acquired by players can be represented effectively for any finite stage of the game, but it may require infinite resources to maintain the derived knowledge along a non-terminating play.

The description of a game in the distributed-systems perspective carries the characteristics of a pure coordination game in the sense of non-cooperative game theory: all participating players have the same utility, and the resolution of uncertainty about past moves raises similar problems as when a player chooses his actions on the basis of rationality assumptions. The twist is in the way of asking the question: while the game-theoretic issue consists in describing individual strategic choices that compose to efficient outcomes, the fundamental problem in distributed systems is that of decomposing a global winning strategy in such a way that the individual agents can execute it on basis of the information that they can access.

Nevertheless, we argue that coordinated strategies are significant in the game-theoretic perspective. Firstly, the existence of such winning strategies is a condition for the existence of Pareto-efficient solutions to the classical coordination problem. Secondly, the questions regarding the memory required for storing the knowledge acquired in a coordination game may be of technical interest beyond the applications to computational systems. Finally, the investigations on coordinated strategies may lead to new insights about the focality of a solution, for instance, if there exists a unique coordinated winning strategy.

The technical contribution of this paper is a generalisation of the power-set construction proposed by Reif [13] which replicates the determinisation of non-deterministic automata over finite words and underlies virtually all approaches to solving games with imperfect information today. The key concept is that of information tracking, a process that unravels a given game graph to a certain degree keeping the balance between two antagonistic constraints: to maintain possibly few information types – these are induced by the past play and may grow beyond any limit over time –, and to distinguish possibly many bisimulation types – these are relevant for the future of the play. Essentially, the tracking of a game corresponds to the minimal bisimulation quotient of its representation as an extensive game. We show that in every game with finite tracking the knowledge accumulated during the game can be represented in a finite way and recovered by strategy automata, which implies that these games admit solutions in distributed strategies. On the other hand, we point out that there is no simple criterion for determining the size of the tracking for a given game graph. Nevertheless, our analysis shows that the knowledge structure for a game can be constructed on the fly, yielding a semi-decision algorithm for the synthesis of strategy automata in graph games with imperfect information and several players.

To keep the analysis simple, we work with a particularly simple winning condition: a safety condition that requires that players avoid a designated bad state forever (one may think of minesweeper). We introduce the basic model of games with imperfect information on graphs in Section 2. In Section 3, we discuss several notions of game equivalence to establish the link between graph games and games in extensive form. Finally, in Section 4 we present our tracking construction and conclude presenting two examples with implications for the computational complexity of the strategy construction problem.

Acknowledgement. The authors gratefully acknowledge partial support from the GAMES program and the LINT EUROCORES project of the European Science Foundation.

2 Games with imperfect information on graphs

Notation. When we speak about situations that involve n players, we refer to a list of elements $x = (x^i)_{i<n}$, one for each player, as a *profile*. For any such profile we write x^{-i} to denote the list $(x^j)_{j<n, j \neq i}$ of elements in x for each player

except i. Given an element x^i and a list x^{-i}, we denote by (x^i, x^{-i}) the profile $(x^i)_{i<n}$. Similarly, for a coalition of players $I \subseteq \{0, \dots, n-1\}$, we write x^I to denote $(x^i)_{i \in I}$. The *grand coalition* is $\{0, \dots, n-1\}$. We do not regard nature as a player. For clarity, we will always use superscripts to specify to which player, or coalition, an element belongs. If not quantified otherwise, we usually refer to player i meaning *any* player.

A graph with edge labels from a fixed alphabet A is a structure $G = (V, \Delta)$ over a set V of nodes with an edge relation $\Delta \subseteq V \times A \times V$. Between two nodes v and w, there may be multiple edges (v, a, w) with different labels a. We write $\Delta(v, a)$ to designate the set $\{w : (v, a, w) \in \Delta\}$ of a-successors to a node $v \in V$. A path in G is a sequence $v_0, a_0, v_1, a_1, \dots$ alternating between nodes and edges such that $v_{\ell+1} \in \Delta(v_\ell, a_\ell)$; for a finite path of length ℓ, the sequence ends with the node v_ℓ. A node u is a root for the graph, if every node $v \in V$ is reachable via a path starting from u. A tree is a graph with a unique root $v_0 \in V$ from which every node is reachable via precisely one path. The depth of a node is the length of this path. Notice that, in contrast to arbitrary graphs, we do not allow multiple edges in trees: for every $v \in V, a \in A$, there is at most one predecessor w such that $(w, a, v) \in \Delta$.

Game graphs. We consider games played by n players, 0, 1, …, and $n-1$, against nature. Beforehand, we fix a set A^i of *actions* available to Player i and a set B^i of *observations* of Player i. We denote by A the set of action profiles and by B the set of observation profiles.

A *game graph* is a structure $G = (V, \Delta, (\beta^i)_{i<n})$ with a set V of *positions*, a *move* relation $\Delta \subseteq V \times A \times V$ and an *observation* function $\beta^i : V \to B^i$ for each player $i < n$. We refer to G as a *game tree* if (V, Δ) is a tree. For convenience, we will assume that each position in a game graph has at least one outgoing move for every action profile, i.e., $\Delta(v, a) \neq \emptyset$, for all $v \in V$ and all $a \in A$. This implies, in particular, that all game trees are infinite.

To play on a game graph, the n players interact with nature in forming a path by moving a token along the edges. At the beginning, the token is at a designated initial position $v_0 \in V$ known to all participants. The game is played in rounds; in every round, each player i chooses an action $a^i \in A^i$. Then, according to the current position v of the token and the joint profile a of actions, nature chooses a successor $v' \in \Delta(v, a)$ to which the token is moved. Here, each player i receives the observation $\beta^i(v')$ and the play continues with v' as the current position. Notice that the players are neither informed about the current position of the token, nor about the action chosen by other players.

Formally, a *play* starting from a designated position $v_0 \in V$ in G is an infinite sequence $\pi = v_0, a_0, v_1, a_1, \dots$ alternating between positions and action profiles with $(v_\ell, a, v_{\ell+1}) \in \Delta$, for all $\ell \geq 0$. An *initial play* is a prefix $v_0, a_0, \dots, a_{\ell-1}, v_\ell$ of a play. We extend the observation function from positions to (initial) plays $\pi = v_0, a_0, v_1, a_1, \dots$ by setting $\beta^i(\pi) = \beta^i(v_0), \beta^i(v_1), \dots$

A *strategy* for Player i is a partial function $s^i : (VA)^*V \to A^i$ that assigns an action to every initial play such that $s^i(\pi) = s^i(\pi')$ for any plays π and π' that induce the same sequence of observations $\beta^i(\pi) = \beta^i(\pi')$. We denote the set of all strategies of Player i with S^i and the set of all strategy profiles by S. A play (or an initial play) $\pi = v_0, a_0, v_1, a_1, \dots$ *follows* the strategy $s^i \in S^i$, if $a^i_\ell = s^i(v_0, a_0, v_1, \dots, a_{\ell-1}, v_\ell)$ for every index $\ell > 0$. For a coalition $I \subseteq C$, the play π follows a profile s^I of strategies, if it follows all strategies s^i of players $i \in I$. The set of possible *outcomes* of a strategy profile s^I is the set of plays that follow s^I.

We are particularly interested in strategies implemented by finite-state machines. A *strategy automaton* for Player i over a game graph G is a structure (M, m_0, μ, σ) consisting of a finite set M of *memory states* with an initial state m_0, a memory *update* function $\mu : M \times (A^i \times B^i) \to M$ and a *choice* function $\sigma : M \times B^i \to A^i$. We extend μ to sequences of actions-observations of Player i along initial plays starting with $\mu(\beta^i(v_0)) := m_0$ and setting

$$\mu(a_0, b_1, \dots, b_{\ell-1}, a_{\ell-1}, b_\ell) := \mu(\mu(a_0, b_1, \dots, b_{\ell-1}), a_{\ell-1}, b_\ell).$$

Depending on the current memory state and on the observation at the present position, the choice function prescribes an action at every position, thus implementing the strategy

$$s^i(v_0, a_0, \dots, v_{\ell-1}, a_{\ell-1}, v_\ell) := \sigma(\, \mu(\beta^i(v_0), a_0, \dots, a_{\ell-1}, \beta^i(v_\ell)), \beta^i(v_\ell)\,).$$

A strategy with finite memory is one that can be implemented by a strategy automaton.

Winning. A *winning condition* over a game graph G is a set $W \subseteq (VA)^\omega$ of plays. A *game* $\mathcal{G} = (G, W)$ consists of a game graph and a winning condition. We say that a play π on G is winning in $\mathcal{G}$ if $\pi \in W$; a strategy profile s^I is winning in $\mathcal{G}$, if all its possible outcomes are so.

To describe winning conditions in a more succinct way, and to relate games played on different graphs under similar conditions, we refer to a finite alphabet C of colours and a colouring function $\gamma : V \to C$ and describe winning conditions over G as sets $W \subseteq C^\omega$ comprising all plays $v_0, a_0, v_1, a_1, \dots$ with $\gamma(v_0), \gamma(v_1), \dots \in W$. Strictly speaking, colouring functions do not belong to the game form, but to the winning condition. However, it is technically convenient to include the colouring in the description of the game graph. Therefore, whenever we refer to a game graph throughout the paper without further specification, we mean a game graph expanded with a colouring function $(G, \Delta, (\beta^i)_{i<n}, \gamma)$.

As part of the description of a game, the colouring function is available to the players. However, if a player is uncertain about the current position, he may not know its colour either. In contrast to this, we say that a colouring is *observable* by player i, if $\beta^i(v) \neq \beta^i(v')$ whenever $\gamma(v) \neq \gamma(v')$.

A *(simple) safety game* is a game with two colours assigned by $\gamma : V \to \{\text{safe}, \#\}$ that are observable to all players, describing the winning condition $W = \{\text{safe}^\omega\}$. Hence, a play is winning if, and only if, it avoids the positions coloured by #. For simplicity, we assume that the symbol # belongs to the observation alphabet of every player and that $\beta^i(v) = \#$ for a player i precisely when $\gamma(v) = \#$.

Questions. Given a safety game, we ask the following questions:

- *Winner determination:* does there exists a coordinated winning strategy for the grand coalition ?
- *Distributed winning:* Does there exist a distributed finite-state winning strategy, that is, a winning strategy profile s where s^i can be implemented by a finite-state automaton, for every player i ?
- *Strategy construction:* If distributed winning strategy profiles exist, construct one.

We restrict our attention to safety conditions as they are the simplest winning conditions that are genuinely infinite, and because they feature a notion of forbidden move which is helpful for forming intuitions.

Before we proceed let us point out that the simultaneous moves are just a syntactic convenience. Equivalently, we could consider games where the players take turns. The questions in which we are interested have no concurrent content.

3 Equivalences between games

We aim at enriching the structure of a game to obtain an explicit representation of the knowledge that a player can derive from the history. To relate games played on different structures, we introduce different notions of game equivalence.

Definition 1. Let $\mathcal{G}$ be a game. Two strategies $s^i, r^i \in S^i$ are *payoff equivalent* if, for all $t^{-i} \in S^{-i}$, the outcome of (s^i, t^{-i}) is winning if, and only if, the outcome of (r^i, t^{-i}) is winning. The game $\mathcal{G}$ is *embeddable* into a game $\mathcal{G}'$, if there exists a function f that maps the strategies for each player i in $\mathcal{G}$ to strategies for the same player in $\mathcal{G}'$, such that

(i) a profile $(s^0, \dots, s^{n-1}) \in S$ is winning if, and only if, $(f(s^0), \dots, f(s^{n-1}))$ is winning, and

(ii) two strategies $s^i, r^i \in S^i$ are payoff equivalent, if, and only if, their images $f(s^i)$ and $f(r^i)$ are so.

We say that two games $\mathcal{G}, \mathcal{G}'$ are *(normal-form) equivalent* if $\mathcal{G}$ is embeddable into $\mathcal{G}'$ and, vice-versa. In this case, we write $\mathcal{G} \equiv \mathcal{G}'$.

We remark that $\equiv$ captures equivalence up to renaming of strategies and removal of redundant ones: consider two games $\mathcal{G} \equiv \mathcal{G}'$ and transform each of them by keeping only one strategy from each payoff-equivalence class. Between the residual games $\hat{\mathcal{G}}$ and $\hat{\mathcal{G}}'$ obtained in this way, there exist injective embeddings. By the Cantor-Bernstein-Schröder argument, it follows that there exists an isomorphism between the normal forms of $\hat{\mathcal{G}}$ and $\hat{\mathcal{G}}'$, that is, a bijective embedding.

Normal-form equivalence is in general too coarse to speak about the sequential structure of a game, but sufficient to preserving the questions about existence of a coordinated winning strategy. Modal bisimulation induces a much finer equivalence which will allow us to operate on the internal structure of a game graph.

Definition 2. A *bisimulation* between two game graphs G and G' is a binary relation $Z \subseteq V \times V'$ that preserves the colours and the observations, i.e.,

$$\text{if } (v, v') \in Z \text{ then } \gamma(v) = \gamma'(v) \text{ and } \beta^i(v) = \beta^i(v'), \text{ for all } i < n,$$

and that satisfies the following back-and-forth condition:

- for all $(v, v') \in Z$, $a \in A$, and every $w \in \Delta(v, a)$, there exists a position $w' \in \Delta(v', a)$ such that $(w, w') \in Z$, and
- for all $(v, v') \in Z$, $a \in A$, and every $w' \in \Delta(v', a)$, there exists a position $w \in \Delta(v, a)$ such that $(w, w') \in Z$.

For designated initial positions, we say that G, v_0 is bisimilar to G', v'_0 and write $G, v_0 \simeq G', v'_0$, if there exists a bisimulation between G and G' that contains (v_0, v'_0).

Lemma 3. *If two game trees T and T' are bisimilar, then the games (T, W) and (T', W) are equivalent, for every winning condition W.*

Proof. Let $T \simeq T'$ be game trees with the usual notation, and let $W \subseteq C^\omega$ be an appropriate winning condition.

To witness that the games (T, W) and (T', W) are equivalent, we set out with a bisimulation relation $Z \in T \times T$ and extract two functions $z : T \to T'$ and $z' : T' \to T$ such that $\{(v, z(v)) : v \in V\}$ and $\{(z(v'), v') : v' \in V\}$ are bisimulations. We proceed by induction over the depth of positions. For the roots u, u' of T and T', set $z(u) = u'$ and $z'(u') = u$. Assuming that the values of z and z' have been defined for all positions of depth up to ℓ, consider an arbitrary position v of depth ℓ and let $v' = z(v)$. Then, for every a-successor $w \in \Delta(v, a)$, choose an a-successor $w' \in \Delta'(v', a)$ such that $(w, w') \in Z$ and set $z(w) := w'$.

Clearly, $\{(v, z(v)) : v \in V\}$ is a bisimulation relation with the additional property that, for any initial play $\pi = v_0, a_0, v_1, a_1, \dots$ over T, the image $z(\pi) = z(v_0), a_0, z(v_1), a_1, \dots$ is a play over T' which induces the same sequence of colours and observations, that is, $\gamma(\pi) = \gamma'(z(\pi))$ and $\beta^i(\pi) = \beta'^i(z(\pi))$ for all players i. Analogously, we construct a bisimulation function $z' : T' \to T$ for the converse direction.

Now, we can translate each strategy s^i on T into a strategy s'^i on T' by setting $s'^i(\pi') = s(\pi)$, for every initial play π' that is the image of some initial play π over T'; for every other initial play π' over T', we choose a companion $\tau' := z(z'(\pi'))$ and set $s'^i(\pi') = s'^i(\tau')$. The mapping $s^i \to s'^i$ constructed in this way has the property that every profile of strategies s^I for a coalition $I \in \{0, \dots, n-1\}$ has the same set of possible outcomes, in terms of colours, as its image s'^I. This implies that the translation $s \to s'$ is an embedding of T into T' that preserves payoff equivalence. An embedding for the converse direction can be constructed analogously. This concludes the proof that the games (T, W) and (T', W') are equivalent for any winning condition defined in terms of colours. □

Definition 4. The *unravelling* of a game graph G from a position v_0 is a game tree $\mathcal{T}(G, v_0)$ with the following ingredients:

- the set of positions consists of all initial plays in G, v_0,
- the move relation consists of edges (π, a, π'), for all plays $\pi = v_0, a_0, \ldots, v_\ell$ and $\pi' = v_0, a_0, \ldots, v_\ell, a_\ell, v_{\ell+1}$ with $a_\ell = a$.
- the observation function of Player i maps every play $v_0, a_0, \ldots, v_\ell$ to $\beta^i(v_\ell)$, and
- the colouring function maps every play $v_0, a_0, \ldots, v_\ell$ to $\gamma^i(v_\ell)$.

Obviously, the projection $\mathcal{T}(G, v_0) \mapsto G, v_0$ that sends every initial play to its last position defines a bisimulation between $\mathcal{T}(G, v_0)$ and G, v_0. Extending this projection to entire plays allows us to view any winning condition W over a game graph as a winning condition over its unravelling, and thus speak of $(\mathcal{T}(G, v_0), W)$ as the unravelling of the game (G, v_0, W).

Lemma 5. *Every game is equivalent to its unravelling.*

Proof. The bijection f that maps every initial play $\pi = v_0, a_0, v_1, \ldots v_\ell$ on a graph game G to the sequence

$$f(\pi) = v_0, (v_0, a_0, v_1), (v_0, a_0, v_1, a_1, v_2), \ldots, (v_0, a_0, v_1, \ldots, v_\ell)$$

of its prefixes, which corresponds to an initial play in the unravelling of G, induces embeddings of strategies from G into its unravelling and vice versa by setting $s'^i(f(\pi)) := s(\pi)$ and $s^i(\pi) := s'^i(f^{-1}(\pi))$, respectively. These embeddings preserve payoff equivalence and thus witness that the games on G and its unravelling are equivalent. $\square$

Proposition 6. *If two game graphs G, v_0 and G', v_0' are bisimilar, then the games (G, v_0, W) and (G', v_0', W) are equivalent, for any winning condition W.*

Proof. By Lemma 5, each of (G, v_0, W) and (G', v_0', W) is equivalent to its unravelling. As bisimilar game graphs unravel to game trees that are again bisimilar, and thus equivalent under any winning condition defined in terms of colours, according to Lemma 3, transitivity of game equivalence implies that $(G, v_0, W) \equiv (G', v_0', W)$. $\square$

The unravelling of a game reflects the sequential order of decisions and actions in more detail than a finite graph. The extensive form of a game is a representation that additionally makes the information structure of a game explicit.

Definition 7. An *extensive* game form $\mathcal{T} = (T, \Delta, (\sim^i)_{i<n})$ consists of a tree over a set T of positions with a move relation $\Delta \subseteq T \times A \times T$, and a profile of equivalence relations $\sim^i \in T \times T$, called *indistinguishability* relations.

An extensive-form strategy for Player i is a function $f^i : T \to T$ that maps every position of T to an action $a^i \in A^i$ such that $f^i(v) = f^i(v')$, whenever $v \sim^i v'$. Plays, consistency of a play with a strategy, and possible outcomes are defined as for game graphs, by identifying initial plays with positions in the tree.

An extensive game is an extensive game form equipped with a winning condition, which we describe in terms of a colouring function $\gamma : T \to C$ as a set $W \subseteq C^\omega$.

The equivalence classes of $\sim^i$ are called the *information sets* of Player i; the family of all information sets of a player forms his *information partition*.

Notice that the general notion of indistinguishability allows to describe games that do not correspond to any game on graphs. This is because our definition of graph games with synchronous moves makes an implicit assumption of perfect recall: each player is aware of the number of previous rounds, of his previous own actions and observations, and any information that has been once available is never forgotten.

Formally, an indistinguishability relation $\sim^i$ satisfies *perfect recall* if every pair of nodes that are distinct but indistinguishable for Player i satisfies the following conditions:

(i) v and v' are of the same depth in $\mathcal{T}$, and

(ii) for the unique w, a and w', a' with $\Delta(w, a) = v$ and $\Delta(w', a') = v'$, we have $w \sim^i w'$ and $a^i = a'^i$.

We will be interested in unwrapping the information sets represented in a graph game.

Definition 8. Let $\mathcal{T} = (T, \Delta, (\beta^i)_{i<n})$ be a game tree. The *indistinguishability* relation of Player i on $\mathcal{T}$ is defined by

$$\pi \sim^i \pi' \quad \text{if} \quad \alpha^i(\pi) = \alpha^i(\pi') \text{ and } \beta^i(\pi) = \beta^i(\pi).$$

The *indistinguishability* relation of Player i on a game graph $\mathcal{G}, v_0$ is his indistinguishability relation on the unravelling of G, v_0.

Lemma 9. *The indistinguishability relation on a game tree $\mathcal{T}$ is the coarsest equivalence relation that satisfies perfect recall and refines the observational equivalence relation $\{(\pi, \tau) \ : \ \beta(\pi) = \beta(\tau)\}$.*

This follows because $\sim^i$ is reflexive and closed under the following propagation rule:

> if $w \sim^i w'$ then, for all action profiles a, a' that agree on the action of Player i and for all successors $v = \Delta(w, a)$ and $v' = \Delta(w', a')$ with $\beta^i(v) = \beta^i(v')$, we have $v \sim^i v'$.

Definition 10. The extensive form $\mathrm{Ext}(G, v_0)$ of a game graph G, v_0 is the extensive game form $(T, \Delta, (\sim^i)_{i<n})$ obtained by unravelling the game graph from the initial position v_0 and expanding it with the indistinguishability relations of all players on the unravelling.

Notice that, in spite of their similar shape, graph games and extensive forms induce different notions of strategies. Intuitively, extensive-form strategies can be seen as mappings from information sets to actions whereas game-tree strategies are mappings from *sequences* of observations to actions. On the other hand, in the extensive form $\sim^i$ can distinguish between positions reached with the same sequence of observations, but with different actions, which is not possible in graph games. We show that, the games are nevertheless equivalent in the sense in which we are interested.

Lemma 11. *Every game is equivalent to its extensive form.*

Proof. By Lemma 5, we know that every game is bisimilar – and thus equivalent – to its unravelling. Accordingly, it is sufficient to prove the statement for games on (infinite) trees.

Let $\mathcal{G} = (V, \Delta, (\beta^i)_{i<n}, W)$ be a graph game on a tree and let $\mathrm{Ext}(\mathcal{G}) = (V, \Delta, (\sim^i)_{i<n}, W)$ be its extensive form. As the two games have the same set of positions, and because all strategies in the extensive form respect the observation equivalence of $\mathcal{G}$, by construction of $\sim^i$, the identity mapping from strategies of $\mathcal{G}$ to strategies of $\mathrm{Ext}(\mathcal{G})$ is readily an embedding that respects payoff equivalence.

To construct an appropriate strategy embedding from $\mathrm{Ext}(\mathcal{G})$ to $\mathcal{G}$ for the reverse direction, let r^i be a strategy in the extensive game. We shall construct a strategy s^i for $\mathcal{G}$ by induction over the length of initial plays, such that

– s^i agrees with r^i on all initial plays π of length at most ℓ that follow r^i, that is, $s^i(\pi) = r^i(\pi)$, and

– s^i does not distinguish between any initial plays π, π' of length at most ℓ that are observationally equivalent in $\mathcal{G}$, i.e., if $\beta^i(\pi) = \beta^i(\pi')$ then $s^i(\pi) = s^i(\pi')$.

To begin with, we set $s^i(v_0) = r^i(v_0)$, for the root v_0 of the game tree. As an induction hypothesis, let us assume that s^i has been defined for all initial plays of length up to ℓ. To extend the definition to plays of length $\ell + 1$, consider, for each sequence $\beta \in (B^i)^\ell$ of observations, the set Π_β of all initial plays π with $\beta^i(\pi) = \beta$ that follow r^i. By induction hypothesis, all plays $\pi' \in \Pi_\beta$ also follow s^i and hence $\alpha^i(\pi) = \alpha^i(\pi')$. Thus, Player i cannot distinguish the plays by his own actions or observations, i.e., $\pi \sim^i \pi'$. As r^i respects $\sim^i$, we can therefore pick an action $a^i_\beta = r^i(\pi)$, for an arbitrary representant $\pi \in \Pi_\beta$, provided the set is non-empty; otherwise we choose an arbitrary action $a^i_\beta \in A^i$. Now, for every play π of length $\ell + 1$ with $\beta^i(\pi) = \beta$, we assign $s^i(\pi) := a_\beta$.

Clearly, s^i respects observation equivalence induced by β^i and is thus a valid strategy for Player i in the graph game $\mathcal{G}$. Moreover, every initial play that follows s^i agrees with r^i on the continuing action. Accordingly, the sets of outcomes of (s^i, t^{-i}) and (r^i, t^{-i}) are the same in the two games, for all strategies $t^{-i} \in S^{-i}$ of the other players, which shows that the mapping $r^i \mapsto s^i$ is an embedding that preserves payoff equivalence. □

Lemma 12. *If two extensive game forms $\mathcal{T}$ and $\mathcal{T}'$ are bisimilar, then the games $(\mathcal{T}, W)$ and $(\mathcal{T}', W)$ are equivalent for every winning condition W.*

The proof follows the lines of Lemma 3.

4 Tracking structure

Definition 13. Let $\mathcal{T} = (T, \mathit{\Delta}, (\beta^i)_{i \le n}, \gamma)$ be a game tree and let $\simeq$ be a maximal bisimulation relation over its expansion $(\mathcal{T}, (\sim^i)_{i \le n})$ with the indistinguishability relations of all players. Then, the *tracking* $\mathrm{Tr}(\mathcal{T})$ of $\mathcal{T}$ is the game graph

$$G = (V, \hat{\mathit{\Delta}}, (\hat{\beta}^i)_{i<n}, \hat{\gamma})$$

obtained as a quotient of $\mathcal{T}$ by $\simeq$ with

- positions corresponding to the equivalence classes of $\simeq$;
- moves $\hat{\mathit{\Delta}} := \{([\pi], a, [\pi']) : (\tau, a, \tau') \in \mathit{\Delta}$ for any $\tau \simeq \pi$ and $\tau' \simeq \pi'\}$,
- observations $\hat{\beta}^i([\pi]) = \beta^i(\pi)$, for each player i,
- colouring function $\hat{\gamma}([\pi]) = \gamma(\pi)$, and
- initial position corresponding to the equivalence class of the root of $\mathcal{T}$.

The tracking $\mathrm{Tr}(\mathcal{G}, v_0)$ of a game graph, is the tracking of its unravelling.

Clearly, every game is bisimilar to its tracking; in fact, the two games have the same extensive form and the identity function induces payoff-preserving mutual embeddings.

Lemma 14. *Every game is equivalent to its tracking.*

Moreover, we argue that the tracking of a game is rich enough for allowing to store in its elements all the information needed to distinguish information sets, at least with respect to safety conditions.

Theorem 15. *For every safety game with finite tracking, if the grand coalition has a winning strategy, it also has one with finite memory.*

Our proof proceeds in two stages.

Definition 16. For an extensive-game tree $\mathcal{T}$, the knowledge-equivalence $x \approx^i y$ for a player i is the transitive closure of $\{(x, y) \in T \times T \ : \ x \sim^i y \text{ or } x \simeq y\}$.

For each player i, the knowledge equivalence thus induces the finest partition into classes closed under indistinguishability and bisimulation with respect to the extensive form.

Proposition 17. *For any extensive game with safety condition, if the grand coalition has a winning strategy, then it also has one that does not distinguish between knowledge-equivalent positions, i.e., there exists a winning strategy s with $s^i(x) = s^i(y)$ whenever $x \approx^i y$, for all $i < n$.*

Proof. Let $\mathcal{T}$ be an extensive game with the usual notation, and let t be a winning strategy for the grand coalition. To construct a $\approx^i$-invariant winning strategy, we define a profile s of strategies that map every (play of an) $\approx^i$-class $[\pi]$ of a position $\pi \in T$ to an action in A^i maintaining the following property:

[$\approx^i \implies \sim^i$] for any pair of plays $\pi \approx^i \pi'$ of the same length that follow s, we have $\pi \sim^i \pi'$.

This is done simultaneously for all players, by induction on length of initial plays. For the root of the game tree, we set $s^i([v_0]) = t^i(v_0)$, for all players. For every initial play π of length ℓ and each player i, if $s^i([\pi])$ is not yet assigned, all representatives of $[\pi]$ of length ℓ that are reachable with s^i are $\sim^i$-equivalent, by induction hypothesis, and hence t^i maps them to the same action a^i. We set $s^i([\pi]) := a^i$, and argue that the invariant propagates to plays of length $\ell+1$: if a pair of prolongations $\tau \approx^i \tau'$ of two plays $\pi \approx^i \pi'$ is reached via s in round $\ell+1$, we have either directly $\tau \sim^i \tau'$ or $\tau \simeq \tau'$, and in this case indistinguishability follows from $\beta^i(\tau) = \beta^i(\tau')$.

Finally, we argue that the profile s is winning. Towards a contradiction suppose that, after an initial play τ, an unsafe successor with observation # is reached upon taking the recommended action profile $a = s(\pi)$, and let π be the representative of $[\tau]$ responsible for the assignment of the action. Then, there exists a finite path alternating between $\sim^i$ and $\simeq$-edges leading from τ to π which propagates the property that a-actions avoid unsafe positions. A contradiction. □

To reason in terms of knowledge equivalence within a game graph G rather than in its extensive form, let $\mathcal{K}^i$ be the partition induced by $\approx^i$ in $\mathrm{Ext}(\mathcal{G}, v_0)$ and let the knowledge set $\mathcal{K}^i(\pi)$ of a play π be the unique set $K \in \mathcal{K}^i$ with $\pi \in K$. Intuitively, the predicate $\mathcal{K}^i(\pi)$ reflects the knowledge that a player may need to acquire in order to play a safety game on $\mathcal{G}, v_0$.

As the knowledge equivalence $\approx^i$ on a game tree is coarser than the bisimulation $\simeq$ that generates its tracking, we can conclude that any two initial plays which end at the same position of a tracked game $\mathrm{Tr}(G, v_0)$ are knowledge equivalent.

Lemma 18. *The knowledge-set predicate is positional in every tracked game* $\mathrm{Tr}(G, v_0)$ *: for all initial plays π, τ that end at the same node v in* $\mathrm{Tr}(G, v_0)$*, we have $\mathcal{K}^i(\pi) = \mathcal{K}^i(\tau)$.*

Accordingly, we can project $\mathcal{K}^i$ from the extensive form $\mathrm{Ext}(G, v_0)$ of a game (G, v_0) to its tracking $\mathrm{Tr}(G, v_0)$ by setting $\mathcal{K}^i(v)$ to be the (name of the) set $\mathcal{K}(\pi)$ for any play π that ends at v. Notice that this projection would not be well-defined on arbitrary games. It is here where we exploit that the tracking of a game has enough structure to receive the knowledge predicate.

In the next step, we argue that, if each player i could observe the predicate $\mathcal{K}^i(v)$, strategies that depend only on this observation at the current position would be sufficient. To make this more precise, consider an observation alphabet consisting of (names of) knowledge sets with the intended meaning $\beta^i(v) := \mathcal{K}^i(v)$.

Corollary 19. *If the grand coalition has a winning strategy in a safety game* (G, v_0)*, it also has a memoryless strategy on the tracking game* $\mathrm{Tr}(G, v_0)$ *with knowledge-set observations.*

Proof. If the grand coalition wins on $\mathrm{Ext}(G, v_0)$, it also has a winning profile s of strategies s^i that depend only on $\mathcal{K}^i(\pi)$, according to Proposition 17. Any such strategy yields a memoryless winning strategy on $\mathrm{Tr}(G, v_0)$ by choosing the action $s^i(\mathcal{K}^i(v))$, for any play that ends at a position v. □

As it appears, the knowledge predicate would be very helpful for finding winning strategies in a safety game. This raises the question of whether and how it can be derived from directly from a game graph rather than from its tracking. The following lemma points out that this can be accomplished with finite resources, for games with finite tracking.

Lemma 20. *If the tracking of a game $\mathcal{G}, v_0$ is finite, then for every player i, there exists a finite-state automaton over $A^i \times B^i$ that recognises the knowledge set $\mathcal{K}(\pi)$ of any initial play $\pi = v_0, a_0, v_1, a_1, \dots, a_\ell, v_\ell$ upon input of its action-observation sequence $a^i_0, \beta^i(v_1), a^i_1, \beta^i(v_2), \dots, a^i_\ell, \beta^i(v_\ell)$.*

Proof. Let s be a winning strategy profile for the grand coalition that is invariant under knowledge-equivalence as in Proposition 17 and Corollary 19. On basis of this profile, we construct for each player i, a deterministic finite-state automaton that runs along the plays over (G, v_0) accepting the actions and the observations of the player as follows.

The state set Q consists of the finitely many (names of) knowledge sets in $\mathcal{K}$, the initial state is $\mathcal{K}^i(v_0)$, and the transition $\delta(K, a^i, b^i)$ leads to the knowledge set K' of any initial play π' that follows s and prolongs a play π with $\mathcal{K}^i(\pi) = K$ by the action a^i and the observation b^i.

To see that the transition function is well-defined and does not depend on the choice of the representatives π, π', one shows by induction over the length of plays according to the profile s, that the following invariant holds:

> for any pair π, τ of initial plays that follow s and end at positions in the same knowledge set K, any pair of continuations π', τ' along the action profile $(s^i(K^i))_{i<n}$ and with the same observation profile $(b_i)_{i<n}$, agrees on the knowledge set, for each player.

Now, if the automaton constructed for Player i runs along an initial play $\pi = v_0, a_0, v_1, a_1, \dots, a_\ell, v_\ell$ that follows the strategy s receiving the action-observation sequence $a^i_0, \beta^i(v_1), a^i_1, \beta^i(v_2), \dots, a^i_{\ell-1}, \beta^i(v_\ell)$, it assumes the state $\mathcal{K}^i(\pi)$, which we regard as the output of the automaton. □

If we add to the automaton constructed for Player i, a choice function returning $s^i(K)$ at every internal state $K \in \mathcal{K}^i$, we readily obtain a strategy automaton that implements s^i, for each player i. The profile of automata resulting from the construction is thus a distributed finite-state winning strategy.

This concludes the proof of Theorem 15.

Notice that in the above proof, it is essential that all players follow the strategy s (or, equivalently, the strategy automaton that implements it); if any player j deviates from this strategy, the internal state K of the automata of other players i will in general not correspond to the current knowledge set.

We conjecture that if a game has finite tracking, then finite memory is sufficient for general regular strategies over the original game.

4.1 Finite and infinite tracking

In this section we discuss examples and algorithmic applications of the tracking construction.

Proposition 21. *Let $(G, W), v_0$ be a finite safety game with finite tracking.*

(i) *The* winner determination *problem of establishing whether the grand coalition has a winning strategy is decidable.*

(ii) *The* distributed winning *problem of determining whether the grand coalition has a finite-state distributed winning strategy and its construction variant can be solved in nondeterministic polynomial time with respect to the size of the tracking.*

Proof. The tracking $(G', v'_0) = \mathrm{Tr}(G, v_0)$ can be constructed incrementally by traversing the (minimal bisimulation quotient of the) game graph G, v_0 in a breadth-first manner while maintaining the knowledge sets and introducing a fresh copy of a position if, and only if, it is not knowledge-equivalent to a previously visited one. This process terminates in time linear in the size of the tracking.

To verify that a winning strategy exists, it is sufficient to guess a mapping from knowledge sets to actions, for each player, and test whether it corresponds to a winning strategy. The testing can be done in linear time with respect to the size of the tracking. using the standard propagation procedure. Overall, this yields a generic complexity of nondeterministic polynomial time. Every distributed winning strategy found in this way translates into a a profile of strategy automata, one for each player.

The above proof sketches a semi-decision algorithm for solving safety games in general, even if it is not known that the tracking is finite. Essentially, the tracking predicate can be constructed on the fly.

Proposition 22. *There are finite safety games with two players that require infinite memory.*

To see an example of such a game, consider the graph depicted in Figure 1. Here, Player 1 and Player 2 play against nature with actions in $A^1 = A^2 = \{\mathsf{o}, \mathsf{x}\}$, and receive observations $B^1 = B^2 = \{\bullet, \neg, \#\}$. All moves that are not represented in the picture are meant to lead to a sink position (not represented either) where both players receive the fatal observation # and lose; the observation ¬ is left void in the picture.

The plays have a similar scenario for both players: they need to perform action o until receiving the observation •, then, perform a number of x action, and finally switch back to o, forever. (The option of playing x will be ruled out soon.) Thus, for each player, the question is how many x actions to perform after observing •. Accordingly, the strategies available to player i correspond to functions $f^i : \omega \to \omega$ with $f(n) = m$ meaning, that if • occurs after n steps, play x for precisely m rounds.

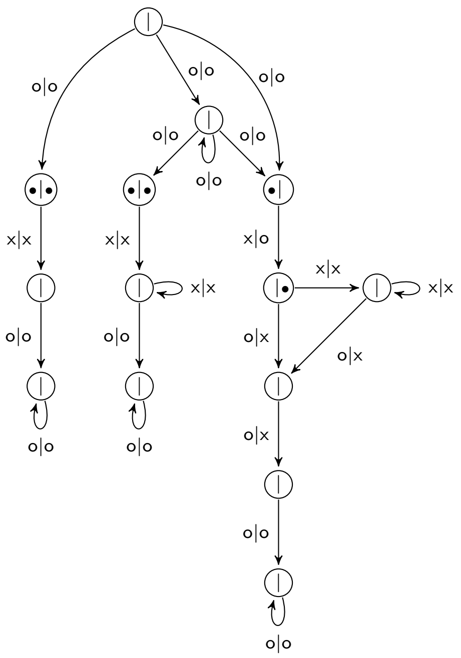

Figure 1. A game that requires infinite memory

The game graph gives us even more clues about which choices the players could make. One important detail is that Player 2 receives • either in the same round as Player 1, or in the subsequent one. In the former case, the two players need to perform the same number of x actions. Hence, any safe strategy must satisfy $f^1(n) = f^2(n)$, for all n. In the latter case, Player 2 must produce one more x-action than Player 1, implying $f^2(n+1) = f^1(n) + 1$, for all n. Finally, we can observe that, if Player 2 receives • in the first round, his only choice is to produce one x. Hence, $f^2(0) = 0$. But to satisfy all these constraints, there is only one pair of strategies that can be safely chosen $f^i(n) = n$. Thus, this unique winning strategy for the coalition needs unbounded memory.

Figure 2 represents a fragment of the extensive form of the game in our example; it can be seen that no two nodes where the players receive the •-observation in the same round are bisimilar with respect to the indistinguishability relation. This is because the alternating $\sim\ ^1$-$\sim^2$ paths leading to the the leftmost position in the tree have different length at each level. The leftmost position is identifiable as the only one which is reachable via a $\sim^1$-$\sim^2$ alternation and that has no $\sim^2$ indistinguishable companion in a different bisimulation class.

Proposition 23. *The question of whether a game graph has finite tracking is undecidable.*

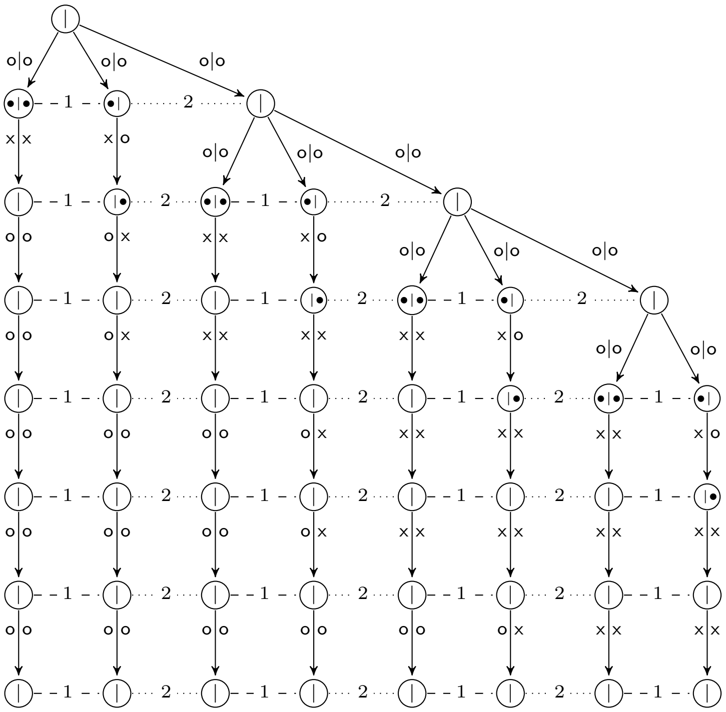
o|o o|o o|o
1 2
x|x x|o o|o o|o o|o
1 2 1 2
o|o o|x x|x x|o o|o o|o o|o
1 2 1 2 1 2
o|o o|x x|x x|x x|x x|o o|o o|o
1 2 1 2 1 2 1
o|o o|o o|o o|x x|x x|x x|x x|o
1 2 1 2 1 2 1
o|o o|o o|o o|x x|x x|x x|x x|x
1 2 1 2 1 2 1
o|o o|o o|o o|o o|o o|x x|x x|x
1 2 1 2 1 2 1

Figure 2. Extensive form of the game in Figure 1 (excerpt)

To see this, we modify our counting example from Figure 1 by replacing the test for the (unary) successor relation with a test for a correct transition between two configurations of a Turing machine.

Let $\mathcal{M} = (Q, \Sigma, q_0, \delta, F)$ be a Turing machine over state space Q with initial state $q_0 \in Q$, final states $F \subseteq Q$, tape alphabet Σ, and a deterministic transition function $\delta : Q \times \Sigma \to Q \times \Sigma \times \{-1, 0, 1\}$. The intention is to encode every configuration of the machine as a word $w(cq)w' \in \Sigma^* \times (\Sigma \times Q)\Sigma^*$ where ww' is the contents of the tape, q the current state, and the length of $|w|$ indicates the current head position with contents c. Accordingly, we choose $\Sigma \cup Q$ as an action alphabet for both players, and $\{\bullet, \neg, \sharp\}$ as the observation alphabet as before. The game graph features a test that the players produce the fixed initial configuration with empty tape when the play starts with a $\bullet$-observation for both players. All later simultaneous $\bullet$-observations lead to a testing that both players produce the same configurations and finally, if Player 2 receives the $\bullet$-observation after Player 1, we have a test that the configuration described of Player 2 succeeds the one described by Player 1, except when the current state is accepting in which case the test for equality applies.

In this scenario, the intended winning strategy for the coalition is the one that produces the n-th configuration of the machine $\mathcal{M}$ upon observing $\bullet$ in round n. This profile can be implemented with a finite-memory strategy, if and only if the machine $\mathcal{M}$ halts. In that case, a simple diagonalisation argument shows that the amount of different memory states that need to be distinguished, and thus the size of the tracking graph, is not bounded by any computable function. Finally, the game admits a winning strategy, if and only if the machine halts on the empty tape – this problem is undecidable.

Corollary 24. *There exists a family of safety games for which the grand coalition has finite-state distributed winning strategies, but the amount of required memory grows faster than any computable function.*

References

[1] A. Arnold and I. Walukiewicz, *Nondeterministic controllers of nondeterministic processes*, in Logic and Automata, vol. 2, Amsterdam University Press, 2007.

[2] D. Berwanger, *Infinite coordination games*, in Logic and the Foundations of Game and Decision Theory (LOFT9), Texts in Logic and Games, Amsterdam University Press, 2010. To appear.

[3] A. K. Chandra, D. Kozen, and L. J. Stockmeyer, *Alternation*, J. ACM, 28 (1981), pp. 114–133.

[4] K. Chatterjee, L. Doyen, T. A. Henzinger, and J.-F. Raskin, *Algorithms for omega-regular games with imperfect information.*, in Proc. of the 20th International Workshop on Computer Science Logic (CSL), 15th Annual Conference of the EACSL, vol. 4207 of LNCS, Springer, 2006, pp. 287–302.

[5] D. Gale and F. M. Stewart, *Infinite games with perfect information*, Annals of Mathematics Studies, 28 (1953), pp. 245–266.

[6] P. Gastin, N. Sznajder, and M. Zeitoun, *Distributed synthesis for well-connected architectures*, Form. Methods Syst. Des., 34 (2009), pp. 215–237.

[7] T. A. Henzinger, *Games in system design and verification.*, in Proc. Theoretical Aspects of Rationality and Knowledge (TARK-2005), 2005.

[8] O. Kupferman and M. Y. Vardi, *Synthesizing distributed systems*, in Proc. Logic in Computer Science (LICS 2004), 2001.

[9] R. McNaughton, *Infinite Games Played on Finite Graphs.*, Ann. Pure Appl. Logic, 65 (1993), pp. 149–184.

[10] G. L. Peterson and J. H. Reif, *Multiple-Person Alternation*, in Proc 20th Annual Symposium on Foundations of Computer Science, (FOCS 1979), IEEE, 1979, pp. 348–363.

[11] A. Pnueli and E. Rosner, *On the synthesis of a reactive module*, in Proceedings of the 16th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, ACM Press, 1989, pp. 179 – 190.

[12] P. J. Ramadge and W. M. Wonham, *Supervisory control of a class of discrete event processes*, SIAM J. Control Optim., 25 (1987), pp. 206–230.

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

[14] K. Rudie and W. M. Wonham, *Supervisory control of communicating processes.*, in Proc. PSTV 1990, Tenth International Symposium on Protocol Specification, Testing and Verification, L. Logrippo, R. L. Probert, and H. Ural, eds., North-Holland, 1990, pp. 243–257.

[15] K. Rudie and W. M. Wonham, *Think Globally, Act Locally: Decentralized Supervisory Control*, IEEE Trans. Autom. Control, 37 (1992), pp. 1692–1708.

[16] W. Thomas, *On the Synthesis of Strategies in Infinite Games.*, in STACS 95, Proc. Symposium on Theoretical Aspects of Computer Science, 1995, pp. 1–13.

[17] S. Tripakis, *Undecidable problems of decentralized observation and control on regular languages.*, Inf. Process. Lett., 90 (2004), pp. 21–28.

[18] I. Walukiewicz, *A Landscape with Games in the Background*, in Proc. Logic in Computer Science (LICS 2004), 2004, pp. 356–366.