The Complexity of Admissibility in Omega-Regular Games*

Romain Brenguier Jean-François Raskin Mathieu Sassolas

Département d'informatique, Université Libre de Bruxelles (U.L.B), Belgium

Abstract

Iterated admissibility is a well-known and important concept in classical game theory, e.g. to determine *rational* behaviors in multi-player matrix games. As recently shown by Berwanger, this concept can be soundly extended to infinite games played on graphs with ω-regular objectives. In this paper, we study the algorithmic properties of this concept for such games. We settle the exact complexity of natural decision problems on the set of strategies that survive iterated elimination of dominated strategies. As a byproduct of our construction, we obtain automata which recognize all the possible outcomes of such strategies.

1 Introduction

Two-player games played on graphs are central in many applications of computer science. For example, in the synthesis problem for reactive systems, implementations are obtained from winning strategies in games with a ω-regular objectives [1]. To analyze systems composed of several components, two-player games are extended to multi-player games with non zero-sum objectives, i.e. each player has his own objective expressed as a ω-regular specification which is not necessarily adversarial w.r.t. the objectives of the other players.

To analyze multi-player games in normal form (a.k.a. *matrix games*), concepts like the celebrated Nash equilibrium [2] have been proposed. Another central concept is the notion of *dominated strategy* [3]. A strategy of a player *dominates* another one if the outcome of the first strategy is better than the outcome of the second no matter how the other players play. In two-player matrix game of Figure 1, strategies of player 1 (of player 2 respectively) are given as rows of the matrix (as columns respectively), and the payoffs to be maximized, are given as pairs of integers (the first for player 1 and the second for player 2). Strategy B of player 1 dominates strategy A: no matter how player 2 plays, B provides an outcome which is larger than or equal to the one of A, and if player 2 plays C then the outcome provided by B is strictly larger than the outcome of A. On the other hand, player 2 at first sight has no preference between C and D. But if player 2 knows that player 1 prefers strategy B to strategy A, then he will in turn prefer D to C, and it is then reasonable to predict that (B, D) will be played. This process is called the *iterated elimination of dominated strategies*, and it is valid under the hypothesis that rationality is *common knowledge* among the players [4]. Strategies that survive the iterated elimination of strategies are called *iteratively admissible strategies.*

	C	D
A	$(0,2)$	$(1,1)$
B	$(1,1)$	$(1,2)$

Figure 1: A two-player matrix game.

In [5], Berwanger has shown that the notion of strategy dominance and iterated elimination of dominated strategies can be soundly extended to infinite games played on graphs with

*Work supported by ERC Starting Grant inVEST (279499).

ω-regular objectives. This extension is challenging as the set of strategies is infinite and may lead to infinite dominance chains. His main results are as follows: all iteration stages are dominated by admissible strategies, the iteration is non-stagnating, and, under regular objectives, admissible strategies form a regular set. In particular, for the last result, Berwanger suggests a procedure that uses tree automata to represent sets of strategies, which is also the case of [6]. The closure of tree automata to projection and Boolean operations naturally provide an algorithm to compute admissible strategies in parity games but this algorithm has non-elementary complexity.

In this paper, we study in details the algorithmic aspects of admissibility in infinite games played on graphs. We provide techniques that avoid the non-elementary complexity of the plain automata-based approach. We study games with weak Muller and (classical) Muller winning conditions given as circuits. Circuits offer a concise representation of Muller conditions and are closed (while remaining succinct) under Boolean operations. Weak Muller conditions define objectives based on the set of states that occur along a run, they generalize safety and reachability objectives. (Classical) Muller conditions define objectives based on the set of states that appear *infinitely often* along a run. They generalize Büchi and parity objectives and are canonical representations of ω-regular objectives. We study the *winning coalition problem*: given a game and two subsets W, L of players, to determine whether there exists an iteratively admissible profile of strategies that guarantees that (i) all players of W win the game, and (ii) all players of L lose the game (other players may either win or lose). For weak and classical Muller objectives, we provide a procedure in PSPACE, with matching lower-bounds for safety, reachability, and Muller objectives. For Büchi objectives, we provide an algorithm that calls a polynomial number of times an oracle solving parity games (hence this would lead to a polynomial time solution if a polynomial time algorithm is found for parity games – the current best known complexity is $\mathsf{UP} \cap \mathsf{coUP}$ [7], although a deterministic subexponential algorithm exists [8]).

As a byproduct of our constructions, we obtain automata which recognize all the possible outcomes of iteratively admissible strategies. In turn, this allows to solve the problem of *model-checking under admissibility*: given φ, an LTL formula [9, 10], does the outcome of every iteratively admissible profiles satisfy φ? The complexity of this problem remains in PSPACE, as is the “classical” model-checking problem for this logic.

Related work. Dominance can be expressed in strategy logics [6, 11] but not unbounded iterated dominance. Bounded iterated dominance is expressible but leads to classes of formulas with a non-elementary model-checking algorithm. Other paradigms of rationality have been studied for games on finite graphs, like Nash-equilibria [12, 13] or *regret minimization* [14]. In [13], the authors build an automaton that recognizes outcomes of Nash equilibria.

Organization of the paper. The rest of the paper is organized as follows. We first formalize the setting and the notations in Section 2. Then we solve the case of games with safety objectives, in Section 3, which gives rise to a simple notion of dominance.

We then give the more general algorithm and constructions for Muller objectives in Section 4. This include the special case of Büchi objectives (Section 4.5) as well as the algorithm solving the model-checking under admissibility problem (Section 4.6).

A comprehensive example of iterated elimination of dominated strategies and model-checking under admissibility is provided in Section 5.

In Section 6, we generalize the algorithm for safety with weak Muller objectives.

2 Definitions

2.1 Multiplayer Games

Definition 2.1 (Multiplayer games)**.** *A* turn-based multiplayer (non zero-sum) game *is a tuple* $\mathcal{G} = \langle P, (V_i)_{i\in P}, E, (\textsc{Win}_i)_{i\in P}\rangle$ *where:*

- P *is the non-empty and finite set of players;*
- $V = \biguplus_{i\in P} V_i$ *and for every* i *in* P, V_i *is the finite set of player* i*'s states;*
- $E \subseteq V \times V$ *is the set of edges*[1]*; we write* $s \to t$ *for* $(s,t) \in E$ *when* E *is clear from context.*
- *For every* i *in* P, $\textsc{Win}_i \subseteq V^\omega$ *is a winning condition.*

A *path* ρ is a sequence of states $(\rho_j)_{0\le j<n}$ with $n \in \mathbb{N}\cup\{\infty\}$ s.t. for all $j < n-1$, $\rho_j \to \rho_{j+1}$. The *length* $|\rho|$ of the path ρ is n. A *history* is a finite path and a *run* is an infinite path. Given a run $\rho = (\rho_j)_{j\in\mathbb{N}}$ and an integer k, we write $\rho_{\le k}$ the history $(\rho_j)_{0\le j<k+1}$, that is, the prefix of length $k+1$ of ρ. For a history ρ and a (finite or infinite) path ρ', ρ is a *prefix* of ρ' is written $\rho \otimes \rho'$. The *last state* of a history ρ is $\mathrm{last}(\rho) = \rho_{|\rho|-1}$. The set of states *occuring* in a path ρ is $\mathrm{Occ}(\rho) = \{s \in V \mid \exists i \in \mathbb{N}.\ i < |\rho|, \rho_i = s\}$. The set of states *occuring infinitely often* in a run ρ is $\mathrm{Inf}(\rho) = \{s \in V \mid \forall j \in \mathbb{N}.\ \exists i > j, \rho_i = s\}$.

Definition 2.2 (Strategies)**.** *A* strategy *of player* i *is a function* $\sigma_i : (V^* \cdot V_i) \to V$*, such that if* $\sigma_i(\rho) = s$ *then* $(\mathrm{last}(\rho), s) \in E$*. A* strategy profile *for the set of players* $P' \subseteq P$ *is a tuple of strategies, one for each player of* P'*.*

Let $\mathcal{S}_i(\mathcal{G})$ be the set of all strategies of player i in $\mathcal{G}$; we write $\mathcal{S}_i$ when $\mathcal{G}$ is clear from the context. We write $\mathcal{S} = \prod_{i\in P}\mathcal{S}_i$ for the set of all strategy profiles, and $\mathcal{S}_{-i}$ for the set of strategy profiles for all players but i. If $\sigma_{-i} = (\sigma_j)_{j\in P\setminus\{i\}} \in \mathcal{S}_{-i}$, we write (σ_i, σ_{-i}) for $(\sigma_j)_{j\in P}$. Similarly, if S is a set of profiles, S_i denotes the i-th projection of S, i.e. a set of strategies for player i. A *rectangular set* of strategy profiles is a set that can be decomposed as a Cartesian product of strategy sets, one for each player.

A strategy profile $\sigma_P \in \mathcal{S}$ defines a unique *outcome* from state s: $Out_s(\sigma_P)$ is the run $\rho = (\rho_j)_{j\in\mathbb{N}}$ s.t. $\rho_0 = s$ and for $j \ge 0$, if $\rho_j \in V_i$, then $\rho_{j+1} = \sigma_i(\rho_{\le j})$. If S_i is a set of strategies for player i, we write $Out_s(S_i)$ for $\{\rho \mid \exists \sigma_i \in S_i, \sigma_{-i} \in \mathcal{S}_{-i}.\ Out_s(\sigma_i, \sigma_{-i}) = \rho\}$. For a tuple of sets of strategies $S_{P'}$ with $P' \subseteq P$, we write $Out_s(S_{P'}) = \bigcap_{i\in P'} Out_s(S_i)$. A strategy σ_i of player i is said to be *winning* from state s against a rectangular set $S_{-i} \subseteq \mathcal{S}_{-i}$, if for all $\sigma_{-i} \in S_{-i}$, $Out_s(\sigma_i, \sigma_{-i}) \in \textsc{Win}_i$. The set S_{-i} against which a strategy is winning can be omitted if it is clear from context. If σ_i is winning from s for all states s, we simply say that it is *winning*. For each player i, we write $\textsc{Win}_i^s(\sigma_P)$ if $Out_s(\sigma_P) \in \textsc{Win}_i$.

Definition 2.3 (Subgame)**.** *A* subgame *of* $\mathcal{G}$ *is a game on a subset* $((V_i')_{i\in P}, E')$ *of* $((V_i)_{i\in P}, E)$ *such that each state of* $V' = \biguplus_{i\in P} V_i'$ *has at least one successor by* E' *in* V'*. If* T *denotes a set of transitions, the game* $\mathcal{G} \setminus T$ *is the game on the graph* $(V, E\setminus T)$*.*

If $\mathcal{G}' = (V', E')$ is a subgame of $\mathcal{G}$, a strategy σ in $\mathcal{G}$ is a strategy in $\mathcal{G}'$ if for any run $\rho \in V'^*$, $\sigma(\rho) \in V'$. The set of these strategies is written $\mathcal{S}(\mathcal{G}')$. A set of strategies S can be restricted w.r.t. a subgame $\mathcal{G}' \subseteq \mathcal{G}$: $S[\mathcal{G}'] = S \cap \mathcal{S}(\mathcal{G}')$.

[1]It is assumed that each state in V has at least one outgoing edge.

Transition-based strategy sets. A set S_i of strategies of a player i is *transition-based* if it is the set of all strategies of player i in some subgame $\mathcal{G}'$, i.e. if there exists $T \subseteq E$ such that $S_i = \mathcal{S}_i(\mathcal{G} \setminus T)$. A set of profiles which is the product of transition-based strategy sets is also called *transition-based.*

Winning conditions. Winning conditions for each player are given by Boolean circuits [15], either on the set of states occurring along the run, or the set of states occurring infinitely often. Particular cases are safety, reachability, and Büchi winning conditions.

- A *safety condition* is defined by a set $Bad_i \subseteq V$: $\textsc{Win}_i = (V \setminus Bad_i)^\omega$.
- A *reachability condition* is defined by a set $Good_i \subseteq V$: $\textsc{Win}_i = V^* \cdot Good_i \cdot V^\omega$.
- A *Büchi condition* is defined by a set $F_i \subseteq V$: $\textsc{Win}_i = (V^* \cdot F_i)^\omega$.
- A *circuit condition* is given by a Boolean circuit whose inputs are all the states of V that encodes a Boolean formula φ_i which has states as free variables. A run ρ defines a valuation through the set of states visited infinitely often: $v_\rho(s) = 1$ if $s \in \mathrm{Inf}(\rho)$ and $v_\rho(s) = 0$ otherwise. Then $\textsc{Win}_i = \{\rho \mid v_\rho \vDash \varphi_i\}$.
- A *weak circuit condition* is given in the same way except that the inputs concern states occurring during the run (not necessarily infinitely often): $v_\rho(s) = 1$ when $s \in \mathrm{Occ}(\rho)$ and then $\textsc{Win}_i = \{\rho \mid v_\rho \vDash \varphi_i\}$.

Circuit conditions generalize Büchi and other classical conditions such as parity or Muller, in that these can be encoded by a circuit of polynomial size [15]. Note that circuit conditions are *prefix-independent*: for any finite path ρ and infinite path ρ', $\rho \cdot \rho' \in \textsc{Win}_i \Leftrightarrow \rho' \in \textsc{Win}_i$. In two-player zero-sum games with a circuit condition, deciding the winner is PSPACE-complete [15]. Weak circuit conditions generalize safety and reachability.

2.2 Admissibility

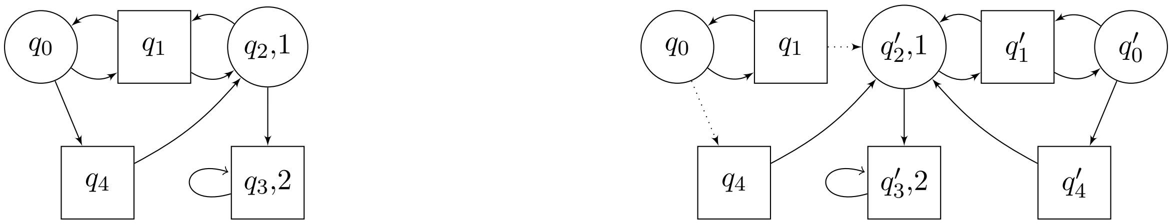
Definition 2.4 (Dominance for strategies). *Let $S = \prod_{i \in P} S_i \subseteq \mathcal{S}$ be a rectangular set of strategy profiles. Let $\sigma, \sigma' \in S_i$. Strategy σ* weakly dominates *strategy σ' with respect to S, written $\sigma \succcurlyeq_S \sigma'$, if from all states s:*

$$\forall \tau \in S_{-i}, \textsc{Win}_i^s(\sigma', \tau) \Rightarrow \textsc{Win}_i^s(\sigma, \tau).$$

Strategy σ strictly dominates *strategy σ' with respect to S, written $\sigma \succ_S \sigma'$, if $\sigma \succcurlyeq_S \sigma'$ and $\neg(\sigma' \succcurlyeq_S \sigma)$. A strategy $\sigma \in S_i$ is dominated in S if there exists $\sigma' \in S_i$ such that $\sigma' \succ_S \sigma$. A strategy that is not dominated is* admissible. *The subscripts on $\succcurlyeq$ and $\succ$ are omitted when the sets of strategies are clear from the context.*

The set of *iteratively admissible* strategies is the fix-point of the operator that associates to a rectangular set of strategies the rectangular set of its admissible strategies, starting from the set $\mathcal{S}$. We write $\mathcal{S}^n$ the rectangular set of strategies obtained after n steps of elimination (and thus $\mathcal{S}^0 = \mathcal{S}$) and $\mathcal{S}^*$ the fix-point. For ω-regular winning conditions, this fix-point always exists and is reached after finitely many iterations [5].

Example 2.5. Figure 2(a) presents a safety game that starts in q_0. Strategies of player 1 that from q_0 go to q_4 are losing. Whereas for those that go to q_1, there is a strategy of player 2 which helps player 1 to win. Hence the former are dominated by the later, and so there are eliminated at the first application of the operator that removes dominated strategies.



(a) A safety game.

(b) The transitions eliminated after two steps of elimination are dotted.

Figure 2: A safety game and its unfolding. States controlled by player 1 are represented with circles and states controlled by player 2 with squares. We will keep this convention through the paper. Label i next to the name of the state denotes that it belongs to Bad_i.

Winning coalition problem: Given a game $\mathcal{G}$ and two subsets W, L of players, does there exists an iteratively admissible profile s.t. all players of W win the game, and all players of L lose the game (other players may either win or lose)?

Model-checking under admissibility problem: Given a game and an LTL [9, 10] formula ψ, does the outcome of every iteratively admissible profiles satisfy ψ?

Values. Our algorithms are based on the notion of *value* of a history. It characterizes whether a player can win (alone) or cannot win (even with the help of other players), restricting the strategies to the ones that have not been eliminated so far. This notion is also a central tool in [5] to characterize admissible strategies. Later, we will show how to compute these values.

Definition 2.6 (Value)**.** *The* value *of history h for player i after the n-th step of elimination, written* $\mathrm{Val}_i^n(h)$*, is given by:*

- *if there is no strategy profile σ_P in $\mathcal{S}^n$ whose outcome ρ from* $\mathrm{last}(h)$ *is such that $h_{<|h|-1} \cdot \rho$ is winning for player i then* $\mathrm{Val}_i^n(h) = -1$*;*
- *if there is a strategy of $\sigma_i \in \mathcal{S}_i^n$ such that for all strategy profiles σ_{-i} in $\mathcal{S}_{-i}^n$, the outcome ρ of (σ_i, σ_{-i}) from s is such that $h_{<|h|-1} \cdot \rho$ is winning for player i then* $\mathrm{Val}_i^n(h) = 1$*;*
- *otherwise* $\mathrm{Val}_i^n(h) = 0$*;*

By convention, $\mathrm{Val}_i^{-1}(h) = 0$.

The following lemma illustrates a property of values and admissible strategies:

Lemma 2.7. *For any n, if* $\mathrm{last}(h) \in V_i$ *and $\sigma_i \in \mathcal{S}_i^{n+1}$ then* $\mathrm{Val}_i^n(h) = \mathrm{Val}_i^n(h \cdot \sigma_i(h))$.

Hence a player which plays according to an admissible strategy cannot go to a state that would produce a history with a different value. This condition is not always sufficient, but in the following sections we characterize runs of admissible strategies relying on this notion of value.

3 Safety objectives

The main result of this section is stated in the following theorem:

Theorem 3.1. *The winning coalition problem with safety winning conditions is* PSPACE-*complete.*

Proposition 3.2. *For safety winning conditions, the value of a history only depends on the players that have already lost and on the last state of the history.*

Proof. Since a safety objective can be transformed into a prefix-independent one by remembering which player have already lost, this is a consequence of the fact that for prefix-independent objectives the value depends only on the last state of the history 4.8. □

So, we write $\mathrm{Val}_i^n(\lambda, s)$ for $\mathrm{Val}_i^n(h)$ where $\lambda = \{i \in P \mid \exists k < |h|.\ h_k \in Bad_i\}$ and $\mathrm{last}(h) = s$. This yields a local notion of dominance for safety conditions. Moreover, in this case the necessary condition of Lemma 2.7 becomes sufficient, as shown below. We encode the set λ in the state of the game, through an unfolding – at the price of an exponential blowup (in the number of players). The *unfolded game* has states in $2^P \times V$ and set of transitions $(\lambda, s) \to (\lambda \cup \{i \mid s' \in Bad_i\}, s')$ for any $\lambda \subseteq P$, if $s \to s'$. In this unfolded game, the value depends only on the current state, hence is written $\mathrm{Val}_i^n(s)$. For example, the game of Figure 2(a) is unfolded as the game of Figure 2(b); states $q'_0, \ldots, q'_4$ are states where player 1 has already lost. Now, let us assume for the remainder of this section that game $\mathcal{G}$ is unfolded.

3.1 Admissible strategies in the unfolded game

For safety winning conditions, dominance can be expressed locally on transitions:

Definition 3.3 (Dominance for transitions)**.** *We write T_i^n for the set of transitions $s \to s' \in E$, such that s is controlled by player i and $\mathrm{Val}_i^n(s) > \mathrm{Val}_i^n(s')$. Such transitions are said to be* dominated *after the n-th step of elimination. We also write T^n for the union of all T_i^n.*

Now, we need to establish links between the two notions of dominance. Note that since all states s have at least one successor with a value greater or equal to that of s, removing transitions of T_i^n do yield a subgame.

Proposition 3.4. *All strategies of $\mathcal{S}(\mathcal{G}) \setminus \mathcal{S}(\mathcal{G} \setminus T_i^n)$ are dominated w.r.t. $\mathcal{S}^n$.*

Proof. Since the objective is now prefix independent, we can use the more general result of Lemma 4.11, proved in the next section.Namely, if $s \in V_i$, $\sigma_i \in \mathcal{S}_i^{n+1}$ then for any history h, $\mathrm{Val}_i^n(s) = \mathrm{Val}_i^n(s')$ where $s' = \sigma_i(h \cdot s)$. Hence no strategy which takes a transition in T_i^n can be in $\mathcal{S}_i^{n+1}$. □

Example 3.5. In Figure 2(a), initially, q_4 has value -1 for player 1, but q_0 has value 0 since it is possible to loop in q_1 and q_0 (if player 2 helps). So, the transition to state q_4 is dominated and removed at the first iteration. Then, player 2 has a winning strategy from q_1, by always going back to q_0, whereas the state q'_2 has value 0 for him. Hence $q_1 \to q'_2$ is removed after this iteration. The fix-point is obtained at that step, it is represented in Figure 2(b).

So removing dominated transitions only removes strictly dominated strategies. The converse is also true: all strategies that remain are not dominated:

Proposition 3.6. *If $\mathcal{S}^n$ is transition-based, then all strategies of $\mathcal{S}(\mathcal{G} \setminus T_i^n)$ are admissible with respect to $\mathcal{S}^n$.*

Proof. Let $\sigma_i, \sigma'_i \in \mathcal{S}_i(\mathcal{G} \setminus T_i^n)$ and assume $\sigma'_i \succ_{\mathcal{S}^n} \sigma_i$. Then there is a state s and strategy profile $\sigma_{-i} \in \mathcal{S}_{-i}^n$ such that $\textsc{Win}_i^s(\sigma'_i, \sigma_{-i}) \wedge \neg\textsc{Win}_i^s(\sigma_i, \sigma_{-i})$. Let $\rho = Out_s(\sigma_i, \sigma_{-i})$ and $\rho' = Out_s(\sigma'_i, \sigma_{-i})$. Consider the first position where these runs differ: write $\rho = w \cdot s' \cdot s_2 \cdot w'$ and $\rho' = w \cdot s' \cdot s_1 \cdot w''$. Note that s' belongs to player i.

First remark that since $\textsc{Win}_i(\sigma'_i, \sigma_{-i})$, it is clear that $\mathrm{Val}_i^n(s_1) \geq 0$. Moreover, since $s' \to s_1$ and $s \to s_2$ do not belong to T_i^n, states s', s_1 and s_2 must have the same value.

- Assume $\mathrm{Val}_i^n(s') = 0$. We show that there is a profile[2] $\sigma_{-i}^2 \in \mathcal{S}_{-i}^n$ such that $\textsc{Win}_i(\sigma_i, \sigma_{-i}^2)$ from s_2.

  Let T^{n-1} be the transitions defining the set of strategy $\mathcal{S}^n$: i.e. $\mathcal{S}^n = \mathcal{S}(\mathcal{G} \setminus T^n)$. Note that if all the successors of a state s'' by transitions that are not in T^{n-1} have value -1, then $\mathrm{Val}_i^n(s'') = -1$. Therefore it is possible to define a strategy profile $\sigma_{-i}^2 \in \mathcal{S}_{-i}^n$ that never decreases the value from 0 or 1 to -1. The strategy σ_i itself does not decrease the value of player i because it does not take transitions of T_i^n. So the outcome of $(\sigma_i, \sigma_{-i}^2)$ never reaches a state of value -1. Hence it never reaches a state in Bad_i and therefore it is winning for player i.

  Now, $\mathrm{Val}_i^n(s_1) = 0$ so there is no winning strategy for player i from s_1 against all strategies of $\mathcal{S}_{-i}^n$. Then there exists a strategy profile $\sigma_{-i}^1 \in \mathcal{S}_{-i}^n$ such that σ_i' loses from s_1. Now consider strategy profile σ_{-i}' that plays like σ_{-i} if the play does not start with w, then σ_{-i}^1 after s_1 and σ_{-i}^2 after s_2. Given a history h:

$$\sigma'_{-i}(h) = \begin{cases} \sigma_{-i}^1((ws_1)^{-1}h) & \text{if } ws_1 \oslash h \\ \sigma_{-i}^2((ws_2)^{-1}h) & \text{if } ws_2 \oslash h \\ \sigma_{-i}(h) & \text{otherwise} \end{cases}$$

  Clearly we have $\textsc{Win}_i^s(\sigma_i, \sigma'_{-i}) \wedge \neg\textsc{Win}_i^s(\sigma'_i, \sigma'_{-i})$, which contradicts $\sigma'_i \succcurlyeq_{\mathcal{S}^n} \sigma_i$.

- Now assume $\mathrm{Val}_i^n(s_2) = 1$. Since $\neg\textsc{Win}_i(\sigma_i, \sigma_{-i})$, the produced outcome ρ reaches a state of Bad_i, hence the value of states along ρ is -1 after some point. Consider the first state ρ_k which has value smaller or equal to 0: $k = \min_{k'}\{\rho_{k'} \mid \mathrm{Val}_i^n(\rho_{k'}) \leq 0\}$. The state ρ_{k-1} has value 1, it is necessarily controlled by a player j different from player i, since transitions of T_i^n cannot be taken by σ_i. Since there exists a winning strategy $\sigma_i \in \mathcal{S}_i^n$ from ρ_{k-1} against strategies of $\mathcal{S}_{-i}^n$, then this strategy is still winning at ρ_k. Therefore $\mathrm{Val}_i^n(\rho_k) = 1$, which is a contradiction. □

Theorem 3.7. *The admissible strategies w.r.t. $\mathcal{S}^n$ is the transition-based set $\mathcal{S}(\mathcal{G} \setminus T^n)$.*

This result yields a polynomial procedure in the unfolded game to compute the set of all iteratively admissible strategies:

- for each state, compute its value as follows: if there is a winning strategy for player i from s in $\mathcal{G} \setminus T^n$, then $\mathrm{Val}_i^n(s) = 1$; if there is no winning run for player i from s in $\mathcal{G} \setminus T^n$, then $\mathrm{Val}_i^n(s) = -1$; otherwise $\mathrm{Val}_i^n(s) = 0$;
- remove dominated transitions;
- start again until no transition is dominated (this will happen after at most $|E|$ steps, where $|E|$ is the number of transitions in the unfolded game).

However, this procedure assumes that the information of which players have already violated their safety condition is encoded in the state. So in the general case, the procedure has a complexity which is exponential in the number of players (but still polynomial if the number of players is fixed).

[2]Although the definition of the value yields the existence of a profile winning for i, it remains to be shown that there is such profile where i plays strategy σ_i.

3.2 The winning coalition problem for safety objectives

To decide the winning coalition problem, only the existence of a particular profile is required and the explicitly construction the unfolded graph is not necessary. By guessing a lasso path produced by such a profile, and checking recursively that it does not contain dominated transition, we get PSPACE membership for Theorem 3.1.

This theorem is proved in the following two lemmata.

Proposition 3.8. *The winning coalition problem is in* PSPACE*.*

Proof. First remark that although there is an exponential number of copies of the game over the graph (V, E) that need to be considered with respect to which players have already lost, states can be ordered the following way: we say that $(\lambda, s) \leq (\lambda', s')$ if $\lambda \subseteq \lambda'$. Along any path the states are increasing for this order, it can increase strictly at most $|P|$ times, and there are at most $|V|$ equivalent states. In addition, the value, hence the elimination of transitions, only depends on the values of greater states, so the iterations stops after at most $|P| \cdot |E|$ phases.

Therefore a procedure to find an iteratively admissible strategy winning at least for players of W and losing at least for players of L consists in guessing a lasso path ρ that ends in a copy where W has not lost and L has. This path has length bounded by $|P| \cdot |\mathcal{G}|$.

However the algorithm needs to check that each transition taken by ρ has indeed survived the elimination of transitions: this transition should not be dominated by any other. This is done by recursively checking that a transition has survived the j-th elimination phase. Note that the (recall that there can be at most $|P| \cdot |E|$ such phases). For a transition $\rho_k \to \rho_{k+1}$ to survive the j-th phase, the value of ρ_{k+1} needs to be the same than that of ρ_k for the player controlling s.

To check $\mathrm{Val}_i^n(\rho_k)$, we use the following procedure:

- if we fail to guess a lasso which does not intersect with Bad_i in $\mathcal{G} \setminus T^n$ from state ρ_k, then $\mathrm{Val}_i^n(\rho_k) = -1$. Note that looking for a path in $\mathcal{G} \setminus T^n$ implies recursively computing some values of iteration $j - 1$;

- if there is a winning strategy for player i in the safety game $\mathcal{G} \setminus T^n$ with target Bad_i, then $\mathrm{Val}_i^n(\rho_k) = 1$; note that this can be done by finding a strategy that either never visits a new set Bad_j (hence not increasing for $\leq$) or visiting a new set Bad_j through a state of value 1 for i (this value being computed recursively, for details, see the more general proof of Theorem 6.1).

- in the other cases $\mathrm{Val}_i^n(\rho_k) = 0$.

In all cases, the recursive calls can stack up to ..., since they always traverse upwards (for $\leq$) the set of states. ☐

Proposition 3.9. *The winning coalition problem is* PSPACE*-hard, even for sets of players $W, L \subseteq P$ such that $|W| = 1$ and $L = \emptyset$.*

Example 3.10. The hardness proof of Theorem 3.1 is done by encoding instances of QSAT. The construction is illustrated in Figure 3 for the following formula $\mu = \exists x_1 \forall x_2 \exists x_3 (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$. There are two players x and $\neg x$ for each variable x, plus two players `Eve` and `Adam`. The moves of `Eve` and `Adam` in the left part of the game determine a valuation: x_i is said to be true if Bad_i was reached. If a player x_i has not yet lost, in the right part of the game, it is better for this player to visit the losing state of `Eve` than its own. Hence, at the first step of elimination, the edges removed in the unfolded game correspond to the ones going to a state Bad_{x_i} if x_i is false (and $\neg x_i$ if x_i is true). At the second step of elimination,

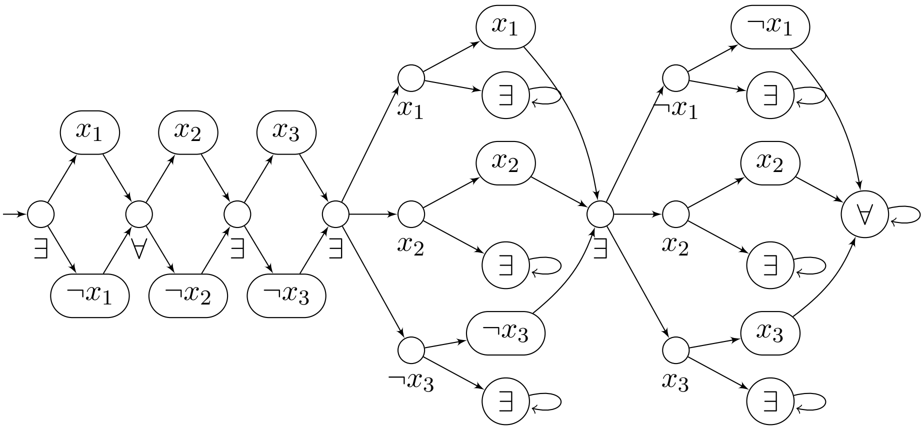


Figure 3: Game $\mathcal{G}_\mu$ with $\mu = \exists x_1 \forall x_2 \exists x_3 (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$. A label y *inside* a state s denotes that $s \in Bad_y$; a label y *below* a state s denotes that $s \in V_y$. Note that `Eve` is abbreviated to $\exists$ and `Adam` is abbreviated to $\forall$.

`Eve` should avoid whenever possible, states corresponding to a literal whose valuation is not true, since those states will necessarily lead to $Bad_\exists$. If the valuation satisfies each clause, then she has the possibility to do so, and one admissible profile is winning for her: so μ is true if, and only if, there is a admissible profile where `Eve` is winning.

Proof. We encode a instance of QSAT into a game in which there is an admissible strategy profile which is winning for `Eve` if, and only if, the formula is valid.

Given a formula $\phi = \exists x_1 \forall x_2 \dots \psi$ we associate a game $\mathcal{G}_\phi$ in which there are one player for each literal x_i or $\neg x_i$ and two players `Eve` and `Adam`. The construction is recursive separately over the quantifiers and over the propositional part. If ψ is a propositional formula:

- If $\psi = x_i$ then we define the module M_ψ in which player x_i has a choice between making `Eve` lose or lose himself and let the game continue, this is represented in Figure 4(a).
- If $\psi = \neg x_i$ then the construction is similar, with player $\neg x_i$ replacing x_i, see Figure 4(b).
- If $\psi = \psi_1 \wedge \psi_2$ then we put the modules M_{ψ_1} and M_{ψ_2} in sequence, see Figure 4(c).
- If $\psi = \bigvee_i \psi_i$ then `Eve` has the choice between all modules M_{ψ_i}, see Figure 4(d).

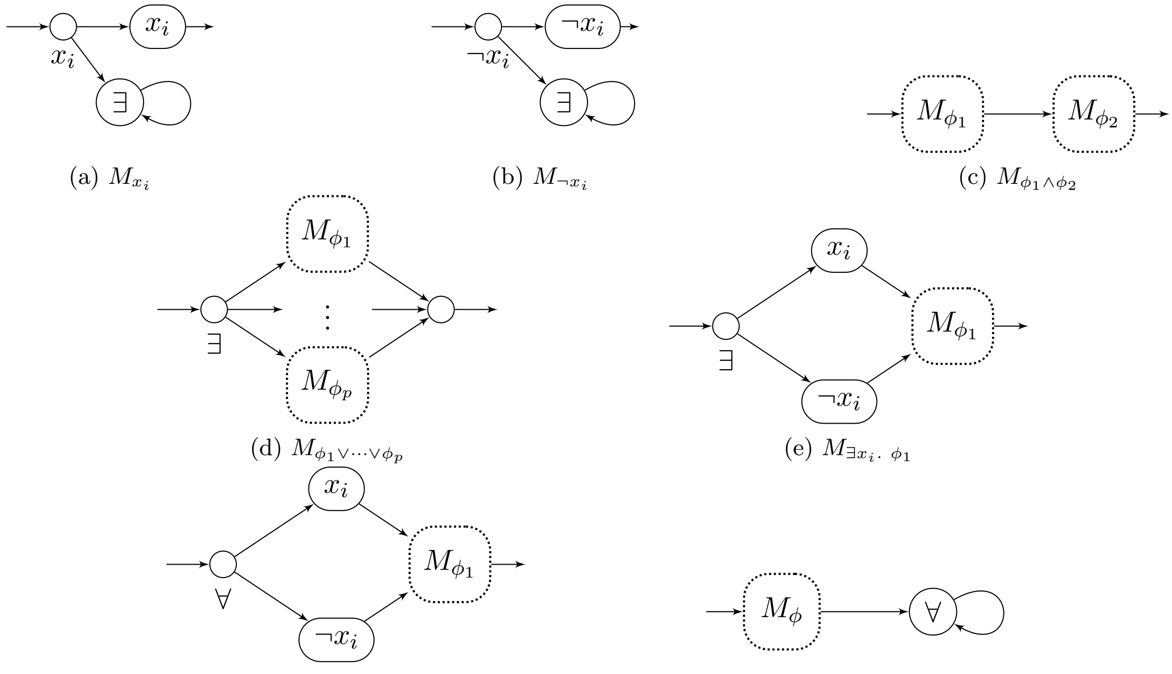
If ϕ is a quantified formula:

- If $\phi = \exists x_i.\ \phi_1$ then `Eve` has the choice between making x_i or $\neg x_i$ lose before continuing to M_{ϕ_1}, see Figure 4(e).
- If $\phi = \forall x_i.\ \phi_1$ is similar but `Adam` controls the choice, see Figure 4(f).

Finally $\mathcal{G}_\phi$ is obtained by directing the remaining outgoing transitions of M_ϕ to a state losing for `Adam`, see Figure 4(g). A full example of $\mathcal{G}_\mu$ with $\mu = \exists x_1 \forall x_2 \exists x_3 (x_1 \vee x_2 \vee \neg x_3) \wedge (\neg x_1 \vee x_2 \vee x_3)$ is given in Figure 3. Note that any run in $\mathcal{G}_\phi$ winning for `Eve` is losing for `Adam`, and *vice versa*.

Given a history of the game, we write $\lambda(h) = \{p \mid \exists i.\ h_i \in Bad_p\}$ the set of player who already lost on that path. We associate to such a set of players λ, a partial valuation v_λ such that:

$$v_\lambda(x_i) = \begin{cases} 1 & \text{if } x_i \in \lambda \\ 0 & \text{if } \neg x_i \in \lambda \wedge x_i \notin \lambda \\ \text{undefined} & \text{otherwise} \end{cases}$$



(a) M_{x_i} (b) $M_{\neg x_i}$ (c) $M_{\phi_1 \wedge \phi_2}$

(d) $M_{\phi_1 \vee \cdots \vee \phi_p}$ (e) $M_{\exists x_i.\ \phi_1}$

(f) $M_{\forall x_i.\ \phi_1}$ (g) $\mathcal{G}_\phi$

Figure 4: Modules for the definition of the game $\mathcal{G}_\phi$.

Note that upon entering M_ψ, the module for the propositional part of ϕ, v_λ is a valuation over variables $\{x_i\}_i$: each variable had exactly one literal visited. We write $v_{\exists\forall}$ this valuation.

Let ℓ be a literal. Let S^1_ℓ be the set of strategies of player ℓ admissible with respect to the all other strategies, *i.e.* S^1_ℓ is the set of admissible strategies after one elimination phase. We claim that S^1_ℓ is exactly the set of strategies that, given a history h such that $\ell \notin \lambda(h)$ and $\text{last}(h) \in V_\ell$, do not take the transition to state labeled ℓ.

Indeed, if such a transition is taken by strategy σ_i, player x_i loses while the strategy σ'_i that mimics σ_i up to this M_{x_i} module then chooses the other transition is winning in this case and preforming in the same way otherwise, thus $\sigma'_i \succ \sigma_i$.

Now if σ_i is a strategy that never chooses the transition to states labeled ℓ unless $\ell \in \lambda(h)$. Assume $\sigma'_i \succ \sigma_i$. Let σ_{-i} be a profile for the other players such that $\textsc{Win}_\ell(\sigma'_i, \sigma_{-i})$ and $\neg\textsc{Win}_\ell(\sigma_i, \sigma_{-i})$. Consider the first time σ_i and σ'_i diverged in these plays, after a history h. If $\ell \in \lambda(h)$, both plays are losing, hence it is the case that $\ell \notin \lambda(h)$. According to its constraints, σ_i chooses to go state losing for `Eve`, and thus wins since no other state, hence no state labeled ℓ, is ever visited, which is a contradiction.

Remark that now all the choices that remain for player ℓ is when he has already lost. Thus the set of admissible strategies for player ℓ is stabilized for these players after one iteration.

The set of strategies $S^1_{\texttt{Eve}}$ and $S^1_{\texttt{Adam}}$ of admissible strategies for the first iterations for players `Eve` and `Adam`, respectively, are identical to the initial sets of strategies for these players. Indeed, `Eve` can be made to lose or win by the coalition of other players regardless of her choices. Since the objectives of `Adam` and `Eve` are opposite, this is also the case for `Adam`. For any history that is not already in a sink state where `Eve` has lost (then nobody has choices), other players can play the two following profiles:

1. To make `Eve` win:

   - when in a module M_ℓ, player ℓ always chooses to go to the state labeled ℓ (even though he loses);

- what Adam plays is irrelevant.

This ensures progress toward the end state where Eve wins (and Adam loses).

2. To make Eve lose:

   - when in a module M_ℓ, player ℓ always chooses to go to the state labeled $\exists$;
   - what Adam plays is irrelevant.

   Since these modules are encountered for every path Eve may choose, she loses[3] (and Adam wins).

First, assume that ϕ is satisfiable. We then show that the admissible strategies of Eve in the second phase of strategy elimination, $S^2_{\texttt{Eve}}$, are the ones corresponding to a satisfaction of ϕ. Namely, the choices of Eve in choosing the value of existentially quantified variables should yield a true propositional formula ψ, which truth is proved by Eve choosing the proper clause in the case of disjunctions. Indeed, if Eve follows such a strategy the only modules M_ℓ entered are ones where literal ℓ has been set as true by valuation $v_{\exists\forall}$. So player ℓ has already lost in the history of the play, hence transitions to states labeled by ℓ are possible. Since it is still possible to make Eve lose from these states, one can still devise strategy profiles of other players that cooperate only with a given strategy of Eve, thus ensuring that it is not dominated.

On the other hand, if a false literal ℓ is encountered, which will occur if the valuation $v_{\exists\forall}$ does not satisfy ψ or if Eve tries to validate a false part of a disjunction, then the player ℓ has no choice but to make Eve lose, since he must play a strategy of S^1_ℓ. So all strategies that do not correspond to the validation of ϕ are losing even with cooperation of other players, so they are not admissible[4].

Similarly, admissible strategies of Adam, $S^2_{\texttt{Adam}}$, require him to choose, if possible (that means if Eve picked the wrong value for a variable), values for universally quantified variables that makes ψ false.

The iteration of admissible strategies stops after these two steps. Indeed, the only difference from Eve's point of view is the strategies of Adam, which cannot change the result provided Eve plays to satisfy ϕ. Conversely, Adam has no better choice than to try to falsify the formula, even if this case does not arise anymore since strategies of $S^2_{\texttt{Eve}}$ do not make such mistakes.

Therefore the iteratively admissible strategies of $\mathcal{G}_\phi$ are the ones in $S^2_{\texttt{Eve}} \times S^2_{\texttt{Adam}} \times S^1_{x_1} \times S^1_{\neg x_1} \times \cdots$. And for any strategy of $S^2_{\texttt{Eve}}$, the strategy profile such that all literal players chose to go to ℓ whenever possible (*i.e.* whenever they have already lost) is winning for Eve.

Now assume that ϕ is not satisfiable. On the other hand, strategies for Adam are not dominated, $S^2_{\texttt{Adam}}$, are actually winning: Adam can choose to play a valuation such that $v_{\exists\forall} \not\models \psi$. With such a valuation, for any choice of Eve in the disjunctions, a module M_ℓ is eventually encountered where $v_{\exists\forall}(\ell) = 0$, so player ℓ has no choice but to make Eve lose, or equivalently to make Adam win.

Any further elimination is thus useless: any iteratively admissible profile has a strategy for Adam that makes him win, so Eve loses. □

[3] Unless ϕ is the empty formula, in which case the game is only the sink state, thus there are no real strategies.

[4] Since some admissible strategy exists.

As shown in the above proof, the complexity stems from the number of players rather that from the number of necessary iteration needed to reach the set of iteratively admissible strategies:

Corollary 3.11. *Deciding whether there exists a profile σ_P of strategies admissible after* 2 *iterations such that* $\mathrm{Occ}(Out(\sigma_P)) \cap Bad_i = \emptyset$ *is* PSPACE*-hard.*

4 Prefix-independent objectives

Our main result for this section is stated in the following theorem:

Theorem 4.1. *The winning coalition problem with a circuit condition for each player is* PSPACE*-complete.*

PSPACE-hardness follows from PSPACE-hardness of two-player games with circuit conditions. The main idea here is to construct a graph representation of the outcomes of admissible profiles. While the construction also relies on the notion of value, it is more involved than for safety conditions: it is for instance possible that a run for which all histories have value 1 is not winning.

After proving Theorem 4.1, we investigate the special case of Büchi winning conditions, for which we obtain a better complexity:

Theorem 4.2. *The winning coalition problem with Büchi objectives is in* NP $\cap$ coNP*. Moreover, if there exists a polynomial algorithm for solving two-player parity games, then winning coalition problem with Büchi objectives is in* P*.*

We also use the fact that this construction provides automata representing the outcomes of all iteratively admissible profiles to solve the model-checking under admissibility problem with tight bounds:

Theorem 4.3. *The model-checking under admissibility problem is* PSPACE*-complete.*

4.1 Shifting of strategy profiles

In the case of prefix-independent objectives, each player can, at any time change his mind. In fact, it is possible for a strategy to *shift* to another (admissible) strategy.

Definition 4.4 (Shifting)**.** *Given two strategies σ^1 and σ^2, and a history h, we denote by $\sigma^1\left[h \leftarrow \sigma^2\right]$ the strategy σ that follows strategy σ^1 and* shifts *to σ^2 after history h. Formally, given a history h':*

$$\sigma(h') = \begin{cases} \sigma^2(h^{-1}h') & \text{if } h \oslash h' \\ \sigma^1(h') & \text{otherwise} \end{cases}$$

We say that a strategy set S allows shifting*, if for any σ^1 and σ^2 in S, and every history h we have $\sigma^1\left[h \leftarrow \sigma^2\right] \in S$. A rectangular set of profiles allows shifting if all its components do.*

Definition 4.5. *We write $\sigma_i \circ h$ the strategy:*

$$\sigma_i \circ h(h') = \begin{cases} \sigma_i(h \cdot h'_{\geq 1}) & \text{if } h'_0 = \mathrm{last}(h) \\ \sigma_i(h') & \text{otherwise} \end{cases}$$

Strategy $\sigma_i \circ h$ thus plays as if σ_i was played after a history h. However, if the history does not start at the end state of h, this would be inconsistent; in that case $\sigma_i \circ h$ plays like σ_i

Lemma 4.6. *If $\mathcal{S}^n$ allows shifting, h is compatible with a strategy profile $(\sigma_j)_{j\in P}$ of $\mathcal{S}^n$ and σ_i is in $\mathcal{S}_i^{n+1}$ then $\sigma_i \circ h$ is in $\mathcal{S}_i^{n+1}$.*

Proof. Assuming σ_i admissible with respect to $\mathcal{S}^n$, we prove that $\sigma_i \circ h$ is also admissible, i.e. not strictly dominated.

Consider a strategy $\sigma'_i \in \mathcal{S}_i^n$. We show that $\sigma'_i \not\succ_{\mathcal{S}^n} \sigma_i \circ h$, which shows that $\sigma_i \circ h$ is in $\mathcal{S}_i^{n+1}$. First, if $\sigma_i \circ h \succcurlyeq_{\mathcal{S}^n} \sigma'_i$, then clearly $\sigma'_i \not\succ_{\mathcal{S}^n} \sigma_i \circ h$.

Otherwise $\sigma_i \circ h \not\succcurlyeq_{\mathcal{S}^n} \sigma'_i$. We write $\sigma''_i = \sigma_i\,[h \leftarrow \sigma'_i]$ for the strategy profile that plays according to σ_i until history h and then shifts to σ'_i.

Since σ'_i is not weakly dominated by $\sigma_i \circ h$, there is a state s and a profile $\tau_{-i} \in \mathcal{S}_{-i}^n$ such that $Out_s(\tau_{-i}, \sigma'_i) \in \textsc{Win}_i$ and $Out_s(\tau_{-i}, \sigma_i \circ h)$ does not win for i. Note that since σ_i and $\sigma_i \circ h$ are identical except when starting from state $\mathrm{last}(h)$, it must be the case that $s = \mathrm{last}(h)$.

We define the strategy profile $\tau'_{-i} = \sigma_{-i}\,[h \leftarrow \tau_{-i}]$ that follows σ_{-i} and until h then shifts to τ_{-i}. Recall that σ_{-i} is a profile of strategies of $\mathcal{S}_{-i}^n$ such that $h \otimes Out_{h_0}(\sigma_{-i}, \sigma_i)$. We have that $Out_{h_0}(\tau'_{-i}, \sigma_i) = h \cdot Out_{\mathrm{last}(h)}(\tau_{-i}, \sigma_i \circ h)$ is losing for player i, and $Out_{h_0}(\tau'_{-i}, \sigma''_i) = h \cdot Out_{\mathrm{last}(h)}(\tau_{-i}, \sigma'_i)$ is winning for player i, by prefix independence. Moreover $\tau'_{-i} \in \mathcal{S}_{-i}^n$ because $\mathcal{S}^n$ allows shifting by hypothesis. So σ_i does not weakly dominate σ''_i: $\sigma_i \not\succcurlyeq_{\mathcal{S}^n} \sigma''_i$.

Since σ_i is not strictly dominated, in particular $\sigma''_i \not\succ_{\mathcal{S}^n} \sigma_i$. And by the above result $\sigma_i \not\succcurlyeq_{\mathcal{S}^n} \sigma''_i$, so it must be the case that $\sigma''_i \not\succcurlyeq_{\mathcal{S}^n} \sigma_i$. Hence, there is a strategy profile τ_{-i} and a state s' such that $Out_{s'}(\tau_{-i}, \sigma_i)$ is winning for player i and $Out_{s'}(\tau_{-i}, \sigma''_i)$ is losing for player i. Since σ_i and σ''_i are identical except after history h, this means that $s' = h_0$ and that h is a prefix of both $Out_{h_0}(\tau_{-i}, \sigma_i)$ and $Out_{h_0}(\tau_{-i}, \sigma''_i)$. We therefore have that $Out_{\mathrm{last}(h)}(\tau_{-i} \circ h, \sigma_i \circ h)$ is winning for player i but $Out_{\mathrm{last}(h)}(\tau_{-i} \circ h, \sigma''_i \circ h)$ is losing for i from $\mathrm{last}(h)$. Since $\sigma''_i \circ h = \sigma'_i$, this means that σ'_i does not dominate $\sigma_i \circ h$. ☐

Lemma 4.7. *For any integer n, $\mathcal{S}^n$ allows shifting.*

Proof. The proof is by induction over n. For the case $n = 0$, the property obviously holds since all strategies are in $\mathcal{S}^0$. Assuming the property holds for n, we show it holds for $n+1$.

Let σ^1 and σ^2 be two strategies of $\mathcal{S}_i^{n+1}$ and h a history compatible with a profile of $\mathcal{S}^n$. First, if there exists a winning strategy against all profiles of $\mathcal{S}_{-i}^n$, then all strategies of $\mathcal{S}_i^{n+1}$ are winning strategies. In particular it is the case for both σ^1 and σ^2. By prefix-independence, it is also the case for $\sigma^1\left[h \leftarrow \sigma^2\right]$, hence $\sigma^1\left[h \leftarrow \sigma^2\right] \in \mathcal{S}_i^{n+1}$. Therefore we assume that there is no winning strategy against all profiles of $\mathcal{S}_{-i}^n$.

Let $\sigma^3 \in \mathcal{S}_i^n$, we show that $\sigma^3 \not\succ_{\mathcal{S}^n} \sigma^1\left[h \leftarrow \sigma^2\right]$. Without loss of generality, σ^3 can be assumed admissible: $\sigma^3 \in \mathcal{S}_i^{n+1}$. Lemma 4.6 ensures that $\sigma^2 \circ h \in \mathcal{S}_i^{n+1}$ and $\sigma^3 \circ h \in \mathcal{S}_i^{n+1}$. Since they are both admissible though not winning strategies, there exists a profile τ_{-i} such that $\sigma^2 \circ h$ wins while $\sigma^3 \circ h$ loses from $\mathrm{last}(h)$. Since σ^1 and σ^3 are also both admissible, there is also a profile τ'_{-i} that makes σ^1 win and σ^3 lose. Consider now profile $\tau''_{-i} = \tau'_{-i}\,[h \leftarrow \tau_{-i}]$. We show that $\rho = Out_{h_0}(\tau''_{-i}, \sigma^3)$ is losing for player i while $\rho' = Out_{h_0}(\tau''_{-i}, \sigma^1\left[h \leftarrow \sigma^2\right])$ is winning for player i.

If h is not a prefix of the run $Out_{h_0}(\tau'_{-i}, \sigma^3)$, then this losing outcome is exactly ρ. If h is a prefix of $Out_{h_0}(\tau'_{-i}, \sigma^3)$, then $\rho = h \cdot Out(\tau_{-i}, \sigma^3 \circ h)$ which is a losing outcome, by prefix independence.

Similarly, if h is not a prefix of $Out_{h_0}(\tau'_{-i}, \sigma^1)$, then this winning outcome is exactly ρ'. And if h is a prefix of $Out_{h_0}(\tau'_{-i}, \sigma^1)$, then $\rho' = h \cdot Out(\tau_{-i}, \sigma^2 \circ h)$ which is a winning outcome for player i, by prefix independence. ☐

4.2 Characterizing outcomes of admissible strategies

Proposition 4.8. *For prefix-independent objectives, the value depends only on the last state of the history.*

Proof. Let h, h' be two different histories with $\text{last}(h) = \text{last}(h')$. Assume $\text{Val}_i^{n+1}(h) = 1$. Then i has a strategy σ_i starting from that wins against all profiles of $\mathcal{S}_{-i}^n$. Then playing like σ_i after h', namely switching to $\sigma_i \circ h$ which is a strategy of $\mathcal{S}_i^n$ by Lemma 4.6, is still a winning strategy so $\text{Val}_i^{n+1}(h) = 1$.

Assume $\text{Val}_i^{n+1}(h) = -1$ and by contradiction that $\text{Val}_i^{n+1}(h') > -1$. Then there exists a profile $\sigma_P \in \mathcal{S}^n$ such that σ_P wins from h'. So playing like σ_P after h, namely playing $\sigma_P \circ h' \in \mathcal{S}^n$, is also winning w.r.t. Win_i. Therefore it is not the case that $\text{Val}_i^{n+1}(h) = -1$.

The case of $\text{Val}_i^{n+1}(h) = 0$ is then obtained by definition of value. $\square$

Hence we write $\text{Val}_j^n(s)$ instead of $\text{Val}_j^n(h)$ when $\text{last}(h) = s$. Since Val_j^n is here a function from V to $\{-1, 0, 1\}$, it can be extended to runs: $\widehat{\text{Val}}_j^n(\rho)$ is the word $w \in \{-1, 0, 1\}^\omega$ such that $w_k = \text{Val}_j^n(\rho_k)$ for all k.

Lemma 4.9. *For any n, if $s \in V_i$, $s \to s'$ with $\text{Val}_i^n(s) = \text{Val}_i^n(s')$, then for any strategy $\sigma_i \in \mathcal{S}^{n+1}$ that is admissible, the strategy σ_i' defined by*

$$\sigma_i'(s) = s' \qquad \sigma_i'(s \cdot h) = \sigma_i(h) \qquad \sigma_i'(h) = \sigma_i(h) \text{ if } h_0 \neq s$$

is also an admissible strategy of $\mathcal{S}^{n+1}$.

Proof. The case where $\text{Val}_i^n(s) = 1$ (resp. $\text{Val}_i^n(s) = -1$) is trivial since σ_i is a winning (resp. always losing) strategy, then so is σ_i'.

Otherwise, assume $\text{Val}_i^n(s) = \text{Val}_i^n(s') = 0$. Let $\tau_i \in \mathcal{S}_i^n$. We only consider what happens in s, since otherwise σ_i' behaves the same as σ_i. If $\tau_i(s) = s'' \neq s'$, then $\text{Val}_i^n(s'') \leq 0$ so there is a profile that makes τ_i lose from s'' hence from s. Additionally, a profile exists that makes σ_i win from s', hence it makes σ_i' win from s.

If $\tau_i(s) = s'$. Since σ_i is admissible, $\tau_i \circ s \not\succ_{\mathcal{S}^n} \sigma_i$.

- If there is a profile that makes σ_i win but $\tau_i \circ s$ lose, it also makes σ_i' win and τ_i lose.
- Otherwise every profile that makes $\tau_i \circ s$ win also makes σ_i win, hence all profiles that make τ_i win also make σ_i' win. $\square$

Lemma 4.10. *For any integer n and states s, s' such that $s \to s'$, if s is controlled by i then $\text{Val}_i^n(s) \geq \text{Val}_i^n(s')$.*

Proof. Assume s is controlled by i:

- if $\text{Val}_i^n(s') = -1$ then the property is obviously true.
- if $\text{Val}_i^n(s') = 0$ then there exists a strategy profile $\sigma_P \in \mathcal{S}^n$ that makes player i win from s'. We define the strategy σ_i' that plays from s to s' and then follows σ_i:

  $$\sigma_i'(s) = s' \qquad \sigma_i'(s \cdot h) = \sigma_i(h) \qquad \sigma_i'(h) = \sigma_i(h) \text{ if } h_0 \neq s$$

  Strategy σ_i' is a strategy of $\mathcal{S}_i^n$ by Lemma 4.9. The strategy profile $\sigma_{j\neq i}[s \cdot s' \leftarrow \sigma_{j\neq i}]$ is in $\mathcal{S}^n$ because of Lemma 4.7. The outcome of $(\sigma_i', \sigma_{j\neq i}[s \cdot s' \leftarrow \sigma_{j\neq i}])$ is then winning from s by prefix independence. The value of s is therefore at least 0.
- if $\text{Val}_i^n(s') = 1$ then there exists a strategy $\sigma_i \in \mathcal{S}_i^n$ that is winning for player i from s' for all the strategy profiles in $\prod_{i\neq j} \mathcal{S}_j^n$. Consider the strategy σ_i' that plays from s to s' and then follows σ_i. Let $\sigma_{j\neq i}'$ be a strategy profile in $\prod_{i\neq j} \mathcal{S}_j^n$. By Lemma 4.6, $\sigma_{j\neq i}' \circ (s \cdot s')$ is a strategy profile in $\prod_{i\neq j} \mathcal{S}_j^n$. Strategy σ_i is winning against $\sigma_{j\neq i}' \circ (s \cdot s')$ from s' and by prefix independence σ_i' is winning against $\sigma_{j\neq i}'$ from s. Therefore the value of s is 1. $\square$

Lemma 4.11. *For any n, if $s \in V_i$ and $\sigma_i \in \mathcal{S}_i^{n+1}$ then for any history h, $\mathrm{Val}_i^n(s) = \mathrm{Val}_i^n(\sigma_i(h \cdot s))$.*

Proof. By the previous lemma the value cannot increase. If player i plays an admissible strategy $\sigma_i \in \mathcal{S}_i^{n+1}$, we show that it cannot decrease. Let $s' = \sigma_i(h \cdot s)$

- If $\mathrm{Val}_i^n(s) = 1$, then σ_i is a winning strategy. Since there is no such strategy from a state with value $\mathrm{Val}_i^n \leq 0$, $\mathrm{Val}_i^n(s') = 1$.
- If $\mathrm{Val}_i^n(s) = 0$, then there is a profile $\sigma_{-i} \in \mathcal{S}_{-i}^n$ such that $\rho = Out(\sigma_i, \sigma_{-i}) \in \textsc{Win}_i$. Note that $h \cdot s \cdot s' \otimes \rho$. If $\mathrm{Val}_i^n(s') = -1$, there can be no such profile, thus $\mathrm{Val}_i^n(s') > -1$, hence $\mathrm{Val}_i^n(s') = 0$
- If $\mathrm{Val}_i^n(s) = -1$, there can be no lower value so $\mathrm{Val}_i^n(s') = -1$. ◻

Lemma 4.12. *For any state s, if $\rho \in Out_s(\mathcal{S}_{-i}^n, \mathcal{S}_i^{n+1})$ then $\widehat{\mathrm{Val}}_i^n(\rho) \in 0^*1^\omega + 0^\omega + 0^*(-1)^\omega$.*

Proof. Let ρ be the outcome of a strategy profile $\sigma_P \in \mathcal{S}^n$ and $k \geq 0$ be an index.

- If $\mathrm{Val}_i^n(\rho_k) = -1$ and $\mathrm{Val}_i^n(\rho_{k+1}) \geq 0$. There is a strategy profile σ'_P in $\mathcal{S}^n$ that is winning for player i from ρ_{k+1}. By Lemma 4.7 and Lemma 4.6, $\sigma_P \circ (\rho_{\leq k})\,[\rho_{k+1} \leftarrow \sigma'_P]$ is in $\mathcal{S}^n$. It makes player i win from ρ_k. It is a contradiction with the fact that $\mathrm{Val}_i^n(\rho_k) = -1$.
- If $\mathrm{Val}_i^n(\rho_k) = 1$ and $\mathrm{Val}_i^n(\rho_{k+1}) \leq 0$. There is a strategy profile in $\mathcal{S}^n$ that makes $\sigma_i \circ \rho_{\leq k}$ lose from ρ_{k+1}. On the other hand, since $\mathrm{Val}_i^n(\rho_k) = 1$ there is a strategy σ'_i that is winning for player i against any strategy of $\mathcal{S}^n$. Hence $\sigma_i \circ \rho_{\leq k}$ is dominated by σ'_i with respect to $\mathcal{S}^n$. By Lemma 4.6 this means that σ_i is dominated. Therefore ρ is not compatible with a strategy in $\mathcal{S}_i^{n+1}$.

Therefore all the paths compatible with a profile in $\mathcal{S}^n$ and a strategy in $\mathcal{S}_i^{n+1}$ have value of the form $0^*1^\omega + 0^\omega + 0^*(-1)^\omega$. ◻

Now, we characterize outcomes of admissible strategies according to whether they end with value 1, -1, or 0.

4.2.1 Value 1

To be admissible at step n of elimination, from a state of value 1, a strategy has to be winning against all strategies of $\mathcal{S}^n$:

Lemma 4.13. *Let $\rho \in Out_s(\mathcal{S}^n)$ and be such that $\widehat{\mathrm{Val}}_i^n(\rho) \in 0^*1^\omega$. Then $\rho \in \textsc{Win}_i$ if, and only if, $\rho \in Out_s(\mathcal{S}_i^{n+1})$.*

Proof. We construct σ_i that follows ρ and if the history deviates from this outcome, revert to a fixed admissible strategy. Formally, let $\sigma_i^a \in \mathcal{S}_i^{n+1}$, we define σ_i by:

- if h is a prefix of ρ with $\mathrm{last}(h) \in V_i$, then $\sigma_i(h) = \rho_{|h|+1}$, so that we ensure that ρ is compatible with σ_i;
- otherwise, let k be the largest index such that $h_{\leq k}$ is a prefix of ρ, then $\sigma_i(h) = \sigma_i^a(h_{>k})$.

We show that σ_i is not strictly dominated. Let σ'_P be a strategy profile of $\mathcal{S}^n$ whose outcome ρ' is winning for player i but such that $\rho'' = Out(\sigma'_{-i}, \sigma_i)$ is not winning for player i. We show that some profile in $\mathcal{S}_{-i}^n$ makes σ'_i lose. Consider the first index k such that $\rho'_k \neq \rho''_k$. The state ρ'_{k-1} is controlled by player i.

If $\mathrm{Val}_i^n(\rho'_{k-1}) = 0$ then there is a profile $\tau_{-i} \in \mathcal{S}_{-i}^n$ that makes σ'_i lose from ρ'_k. As there is a profile $\tau'_{-i} \in \mathcal{S}_{-i}^n$ that makes σ_i win from ρ''_k, we can combine this two strategy profiles to obtain one winning for σ_i and losing for σ'_i. Namely, the profile τ_{-i}^c defined by

$$\forall j \neq i,\ \tau_j^c = \sigma_j \left[\rho'_k \leftarrow \tau_j\right] \left[\rho''_k \leftarrow \tau'_j\right]$$

is by construction such a profile. Therefore, σ_i is not strictly dominated by σ'_i.

Otherwise, $\mathrm{Val}_i^n(\rho'_{k-1}) = 1$, we show that in fact ρ'' is a winning path which is a contradiction. If $\rho' = \rho$ then it is winning by hypothesis. Otherwise there is some point from which σ_i plays according to σ_i^a.

If this point is before index k. Since $\mathrm{Val}_i^n(\rho_{k-1}) = 1$ there is a strategy of player i winning from ρ_{k-1} against all profile in $\mathcal{S}_{-i}^n$. As σ_i^a is in $\mathcal{S}_i^{n+1}$, so is $\sigma_i^a \circ \rho_{\leq k-1}$ by Lemma 4.6. Therefore $\sigma_i^a \circ \rho_{\leq k-1}$ should be winning from ρ_{k-1} against all profiles in $\mathcal{S}_{-i}^n$. Hence ρ'' is winning for player i, which is a contradiction.

Otherwise, let k' be the index at which σ_i starts playing according to σ_i^a. As $\mathrm{Val}_i^n(\rho_{k-1}) = 1$ and $k' \geq k$, $\mathrm{Val}_i^n(\rho_{k'-1}) = 1$. Moreover $\rho''_{k'}$ is a successor of $\rho_{k'-1}$ which is not controlled by i, so $\mathrm{Val}(\rho''_{k'}) = 1$ by Lemma 4.10. Therefore $\sigma_i^a \circ \rho_{\leq k'-1}$ should be winning from ρ''_{k-1} against all profile in $\mathcal{S}_{-i}^n$. Hence ρ'' is winning for player i, which is a contradiction.

This shows that σ_i is not dominated with respect to $\mathcal{S}^n$.

Reciprocally, assume that $\sigma_i \in \mathcal{S}_i^{n+1}$. Let k be a index such that $\mathrm{Val}_i^n(\rho_k) = 1$. There is a strategy of player i that is winning from ρ_k against all strategies of $\mathcal{S}^n$. By Lemma 4.6, $\sigma_i \circ \rho_{\leq k}$ is also in $\mathcal{S}_i^{n+1}$. Therefore $\sigma_i \circ \rho_{\leq k}$ also has to be winning against all strategies of $\mathcal{S}^n$. As $\rho_{\geq k}$ is compatible both with a profile of $\mathcal{S}_i^n$ and $\sigma_i \circ \rho_{\leq k}$, it is winning for player i. Hence, by prefix independence of the objectives, ρ is winning for player i. □

4.2.2 Value -1

If the run reaches a state of value -1, then, from there, there is no possibility of winning, so any strategy is admissible but the state of value -1 must not have been reached by player i's fault:

Lemma 4.14. *Let $\rho \in Out_s(\mathcal{S}^n)$ be such that $\widehat{\mathrm{Val}}_i^n(\rho) \in 0^*(-1)^\omega$, then $\rho \in Out_s(\mathcal{S}_{-i}^n, \mathcal{S}_i^{n+1})$ if and only if, $\forall k,\ \rho_k \in V_i \Rightarrow \mathrm{Val}_i^n(\rho_{k+1}) = \mathrm{Val}_i^n(\rho_k)$.*

Proof. We construct a strategy σ_i that follows ρ and if the history deviates from this outcome, revert to a fixed admissible strategy, in the same way that we did in Lemma 4.13 for value 1. We define σ_i by:

- if h is a prefix of ρ with $\mathrm{last}(h) \in V_i$, then $\sigma_i(h) = \rho_{|h|+1}$, so that we ensure that ρ is compatible with σ_i;
- otherwise, let k be the biggest index such that $h_{\leq k}$ is a prefix of ρ, then $\sigma_i(h) = \sigma_i^a(h_{>k})$.

We show that σ_i is not strictly dominated. Note that starting from a state $s \neq \rho_0$ means σ_i plays according to an admissible strategy. Hence only the case of plays starting from ρ_0 remain to be considered.

Let σ'_P be a strategy profile of $\mathcal{S}^n$ whose outcome ρ' is winning for player i but such that $\rho'' = Out_{\rho_0}(\sigma'_{-i}, \sigma_i)$ is not winning for player i. Consider the first index k such that $\rho'_k \neq \rho''_k$. The state $\rho'_{k-1} = \rho''_{k-1}$ is controlled by player i. In order to show that $\sigma'_i \not\succ_{\mathcal{S}^n} \sigma_i$, we build a profile that makes σ'_i lose and σ_i win.

If $\mathrm{Val}_i^n(\rho'_{k-1}) = -1$, then ρ' cannot be a winning path, this is a contradiction. Otherwise $\mathrm{Val}_i^n(\rho'_{k-1}) = 0$. Then there is a profile in $\mathcal{S}_{-i}^n$ that makes σ'_i lose from ρ'_k.

Now we find a profile that makes σ_i win from ρ''_k.

- If $\rho''_{\le k-1}$ is not a prefix of ρ, then σ_i plays according to an admissible strategy. As $\mathrm{Val}_i^n(\rho''_{k-1}) = 0$, there is a strategy profile that makes the strategy $\sigma_i \circ \rho''_{\le k}$ win.
- Otherwise, $\rho''_{\le k-1}$ is a prefix of ρ. Since $\rho''_{k-1} = \rho_{k-1}$ is controlled by player i, $\mathrm{Val}_i^n(\rho_k) = \mathrm{Val}_i^n(\rho_{k-1}) = 0$ by hypothesis. Let k' be the largest index such that $\mathrm{Val}_i^n(\rho_{k'}) = 0$. Since $\mathrm{Val}_i^n(\rho_{k'+1}) = -1$, state $\rho_{k'}$ is not controlled by i. From $\rho_{k'}$ there is a strategy profile σ''_P that makes σ_i^a win. Note that profile σ''_{-i} deviates from ρ at that point since $\mathrm{Val}_i^n(\rho_{k'+1}) = -1$: $\sigma''_{-i}(\rho_{\le k'}) \ne \rho_{k'+1}$. As a result, after $\rho_{\le k'}$, strategy σ_i follows σ_i^a. Therefore the outcome produced by σ''_{-i} and σ_i is winning from $\rho_{k'}$. So there is also a profile in $\mathcal{S}^n_{-i}$ that makes σ_i win from ρ''_k. Namely:

$$\sigma'''_j(h) = \begin{cases} \rho_{\ell+1} & \text{if } h = \rho_{\le \ell} \text{ with } \ell < k' \text{ and } \rho_\ell \in V_j \\ \sigma''_j(h) & \text{otherwise} \end{cases}$$

We can combine the strategy profile that makes σ'_i lose with σ'''_{-i} win to obtain one that is winning for σ_i and losing for σ'_i. Hence σ_i is not strictly dominated. ☐

Example 4.15. As an illustration of Lemmata 4.13 and 4.14, consider the left game in Figure 6. Both runs $s_0 \cdot s_1 \cdot s_2 \cdot \top^\omega$ and $s_0 \cdot s_1 \cdot s_2 \cdot \bot^\omega$ are outcomes of non dominated strategies of player 1. Indeed, the play that goes to $\top$ is winning and the value of the path that goes to $\bot$ belongs to $0^*(-1)^\omega$ and the value decreases on a transition by player 2.

4.2.3 Value 0

This case is more involved. From a state of value 0, an admissible strategy of player i should allow a winning run for player i with the help of other players. We write H_i^n for set of states s controlled by a player $j \ne i$ that have at least two successors that *(i)* have value 0 or 1 for player i and *(ii)* have the same value for player j than s after iteration $n-1$. Formally, for $n \ge 0$, the "Help!"-states of player i are defined as:

$$H_i^n = \bigcup_{j \in P \setminus \{i\}} \left\{ s \in V_j \;\middle|\; \begin{array}{l} \exists s', s''.\ s' \ne s'' \wedge s \to s' \wedge s \to s'' \\ \wedge\ \mathrm{Val}_i^n(s') \ge 0 \wedge \mathrm{Val}_i^n(s'') \ge 0 \\ \wedge\ \mathrm{Val}_j^{n-1}(s) = \mathrm{Val}_j^{n-1}(s') = \mathrm{Val}_j^{n-1}(s'') \end{array} \right\}$$

These states have the following property.

Lemma 4.16. *Let $\rho \in Out_s(\mathcal{S}^n)$ be such that $\widehat{\mathrm{Val}}_i^n(\rho) = 0^\omega$.*
Then $\rho \in Out_s(\mathcal{S}_i^{n+1})$ if, and only if, $\rho \in \mathrm{W{\small IN}}_i$ or $\mathrm{Inf}(\rho) \cap H_i^n \ne \emptyset$.

Proof. Assume first that there is only a finite number of visits to H_i^n and that ρ is not winning for player i. Let k be the greatest index such that $\rho_k \in H_i^n$. Let $\sigma_P \in \mathcal{S}^n$ be a profile such that $Out(\sigma_P) = \rho$. We will show that the strategy for player i in this profile, namely σ_i, is strictly dominated with respect to $\mathcal{S}^n$.

First, we show that after some prefix of ρ no profile of $\mathcal{S}^n$ can make σ_i win. Let $\alpha_{-i} \in \mathcal{S}^n_{-i}$ and consider the profile $\sigma'_{-i} = \sigma_{-i}\,[\rho_{\le k+1} \leftarrow \alpha_{-i}]$ that follows σ_P until ρ_{k+1}. Let ρ' the outcome of (σ_i, σ'_{-i}). If $\rho = \rho'$ then σ'_{-i} does not make σ_i win.

Otherwise, consider k' the first index where $\rho_{k'} \ne \rho'_{k'}$, and j the player controlling state $s = \rho_{k'-1} = \rho'_{k'-1}$; we have that $j \ne i$. Since $\sigma'_j \in \mathcal{S}_j^n$, Lemma 4.11 yields that $\mathrm{Val}_j^{n-1}(\rho'_{k'}) = \mathrm{Val}_j^{n-1}(\rho'_{k'-1})$. Similarly, since $\sigma_j \in \mathcal{S}_j^n$, we also have $\mathrm{Val}_j^{n-1}(\rho_{k'}) = \mathrm{Val}_j^{n-1}(\rho_{k'-1})$. Therefore $\mathrm{Val}_j^{n-1}(\rho_{k'}) = \mathrm{Val}_j^{n-1}(\rho'_{k'}) = \mathrm{Val}_j^{n-1}(s)$. Recall that $\mathrm{Val}_i^n(\rho) = 0^\omega$ so $\mathrm{Val}_i^n(\rho_{k'}) = 0$. Since $\rho_{k'-1} \notin H_i^n$, it must be the case that $\mathrm{Val}_i^n(\rho'_{k'}) = -1$. Therefore no strategy profile in $\prod_{j \ne i} \mathcal{S}_j^n$ makes σ_i win from ρ_{k+1}.

(a) $k' \geq k$ (b) $k' < k$

Figure 5: Divergence of outcomes by strategy shifting in the proof of Lemma 4.16.

Secondly, since $\mathrm{Val}_i^n(\rho_k) = 0$ there is a strategy profile σ''_P of $\mathcal{S}^n$ whose outcome is winning for i from ρ_k. Hence $\sigma_i\left[\rho_{\leq k} \leftarrow \sigma''_i\right]$ strictly dominates σ_i, and $\sigma_i \notin \mathcal{S}_i^{n+1}$.

Assume now that either $\rho \in \mathrm{W{\small IN}}_i$ or there is an infinite number of indexes k such that $\rho_k \in H_i^n$. Let α_i be a strategy in $\mathcal{S}_i^{n+1}$. We build σ_i so that it is compatible with ρ and revert to α_i in case of a deviation. Formally, for a history h, such that $\mathrm{last}(h)$ is controlled by player i:

- if $h = \rho_{\leq k}$ for some index k then $\sigma_i(h) = \rho_{k+1}$ (this ensures that σ_i is compatible with ρ);
- otherwise, let k be the largest index such that $\rho_{\leq k}$ is a prefix of h, then $\sigma_i(h) = \alpha_i(h_{\geq k+1})$.

We now show that σ_i is not strictly dominated. Assume that there is a strategy profile $\sigma'_P \in \mathcal{S}^n$ such that $\rho' = Out(\sigma'_{-i}, \sigma'_i) \in \mathrm{W{\small IN}}_i$ and $\rho'' = Out(\sigma'_{-i}, \sigma_i) \notin \mathrm{W{\small IN}}_i$. We show that there is a profile that makes σ_i wins and σ'_i lose, so that σ'_i does not strictly dominate σ_i.

We consider the points where ρ'' diverges from ρ and ρ', respectively. Formally, let k (resp. k') be the largest index such that $\rho''_{\leq k} = \rho_{\leq k}$ (resp. $\rho''_{\leq k'} = \rho'_{\leq k'}$). We distinguish two cases as illustrated in Figure 5.

- If $k' \geq k$, i.e. ρ'' diverges from ρ before diverging from ρ'. Since α_i is not strictly dominated, from ρ''_{k+1} there is a strategy profile σ''_{-i} that makes α_i win and $\sigma'_i \circ \rho''_{\leq k}$ lose. Hence $\sigma'_{-i}\left[\rho''_{\leq k'+1} \leftarrow \sigma''_{-i}\right]$ makes σ_i win but not σ'_i. Therefore σ'_i does not strictly dominates σ_i.
- In the other case, $k' < k$, the point where the two outcomes diverge is along ρ. We have $\rho_{\leq k'} = \rho'_{\leq k'}$. We write $s = \rho'_{k'} = \rho_{k'}$, which is controlled by $j \neq i$, $s' = \rho'_{k'+1}$, and $s'' = \rho_{k'+1} = \sigma_i(\rho_{\leq k'}) = \rho''_{k'+1}$. Notice that $\mathrm{Val}_j^n(s') \leq 0$ because $\mathrm{Val}_i^n(s) = 0$ and using Lemma 4.10. Hence there is a strategy profile σ^ℓ_{-i} in $\mathcal{S}^n$ that makes σ'_i lose from s'.
  - If ρ is a winning outcome for player i, we consider the strategy profile:
  $$\sigma'_{-i}\left[\rho'_{\leq k'+1} \leftarrow \sigma^\ell_{-i}\right]\left[\rho_{\leq k'+1} \leftarrow \sigma_{-i} \circ \rho_{\leq k'+1}\right].$$
  Combined with σ_i, its outcome is ρ and therefore is winning. This strategy profile belongs to $\mathcal{S}^n$ by Lemma 4.7 and makes σ_i win and σ'_i lose, hence σ'_i does not strictly dominates σ_i.
  - Otherwise, there is an infinite number of states in H_i^n along ρ. Let k'' be the index of the first occurrence of H_i^n after the divergence: the smallest $k'' > k'$ such that $\rho_{k''} \in H_i^n$. State $\rho_{k''}$ is controlled by a player $j \neq i$. We consider a strategy profile σ^d_P such that:

* $\sigma_j^d(\rho_{k''}) = s^d$ such that $s^d \neq \rho_{k''+1}$, with $\mathrm{Val}_i^n(s) \geq 0$, and $\mathrm{Val}_j^{n-1}(\rho_{k''}) = \mathrm{Val}_j^{n-1}(s^d)$. This state exists since $\rho_{k''} \in H_i^n$.
* Then from s^d it follows a strategy profile $\sigma''_{-i} \in \mathcal{S}^n_{-i}$ that makes α_i win. This profile exists since $\alpha_i \in \mathcal{S}_i^{n+1}$ and $\mathrm{Val}_i^n(s^d) \geq 0$.

The profile σ_P^d belongs to $\mathcal{S}^n_{-i}$. Indeed, all players but j play according to a strategy of $\mathcal{S}^n_{-i}$ by construction. In the case of player j, it first involves a step that does not change the value Val_j^{n-1}, so by Lemma 4.9, it is also admissible.
We consider the strategy profile:

$$\sigma'_{-i}\left[\rho'_{\leq k'+1} \leftarrow \sigma^{\ell}_{-i}\right]\left[\rho_{\leq k''} \leftarrow \sigma^{d}_{-i}\right].$$

This strategy profile belongs to $\mathcal{S}^n$ by Lemma 4.7 and makes σ_i win and σ'_i lose, hence σ'_i does not strictly dominates σ_i.

This shows that σ_i is not strictly dominated and belongs to $\mathcal{S}_i^{n+1}$. □

Figure 6: Two games. The goal for player 1 is to reach $\top$.

Example 4.17. As an illustration of Lem. 4.16, consider the two games in Figure 6. In the left game, a strategy of player 1 that stays in the loop $(s_0 \cdot s_1)$ forever is strictly dominated: it has no chance of winning, while getting out after k steps can be winning if player 2 helps. In the right game, the strategy that always chooses s_3 from s_1 is admissible. Consider the strategy of player 2 that goes to state $\top$ at the $k+1$-th visit of state s_3, and makes strategies of player 1 lose if s_2 is visited beforehand. This difference is characterized by state $s_3 \in H_1^0$ in the loop.

4.2.4 Bounding the number of iteration phases

We can show that $\mathrm{Val}_i^n(s) = 1 \Rightarrow \mathrm{Val}_i^{n+1}(s) = 1$ as winning strategies are admissible and $\mathcal{S}_{-i}^{n+1} \subseteq \mathcal{S}^n_{-i}$ implies that winning strategies remain winning. Similarly, we can show $\mathrm{Val}_i^n(s) = -1 \Rightarrow \mathrm{Val}_i^n(s) = -1$. So the number of times that the value function changes is bounded by $|P| \cdot |V|$. This allows us to bound the number of iterations necessary to reach the fix-point:

Proposition 4.18. $\mathcal{S}^* = \mathcal{S}^{|P|\cdot|V|}$.

Proof. Since $\mathcal{S}^{n+1} \subset \mathcal{S}^n$, a state that has value $\neq 0$ at a given iteration cannot change its value at a subsequent iteration. Hence there are at most $|P| \cdot |V|$ changes of value during the iteration process. Since admissible strategies depend only on the values (even through the "Help!"-states), the strategy elimination stops when the value stops changing. □

4.2.5 Automata for $Out(\mathcal{S}^n)$

From the above characterization of admissible strategies w.r.t. values of state, we define an automaton that accepts $Out(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})$, and one for $Out(\mathcal{S}^n)$. These constructions rely on values at previous iterations. We start by defining an automaton $\mathcal{A}_i^n$, for $n > 0$, that recognizes[5] $Out(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})$.

- The set of states is V, i.e. the same states as in $\mathcal{G}$;
- From the transitions in $\mathcal{G}$ we remove those controlled by player i that decrease his value. Formally:
$$T = E \setminus \{(s, s') \mid s \in V_i \wedge \mathrm{Val}_i^{n-1}(s) > \mathrm{Val}_i^{n-1}(s')\}.$$
- A run ρ is accepted by $\mathcal{A}_i^n$ if one of the following conditions is satisfied: $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^*(-1)^\omega$; or $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^*1^\omega$ and $\rho \in \textsc{Win}_i$; or $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^\omega$ and $\rho \in \textsc{Win}_i$ or $\rho \in (V^* H_i^{n-1})^\omega$.

Note that the structure of the automaton is a subgame of $\mathcal{G}$. We prove the following lemma that shows the relation between automaton $\mathcal{A}_i^n$ and outcomes of admissible strategies (at step n).

Lemma 4.19. $\mathcal{L}(\mathcal{A}_i^n) \cap Out(\mathcal{S}^{n-1}) = Out(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})$

Proof. Let ρ be the outcome of a strategy profile in $\mathcal{S}^{n-1}$ that is also accepted by $\mathcal{A}_i^n$. Note that proving that $\rho \in Out(\mathcal{S}_i^n)$ suffices to prove that $\rho \in Out(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})$.

There are three different cases, depending on which part of the accepting condition of $\mathcal{A}_i^n$ is fulfilled.

- If $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^*1^\omega$, then $\rho \in \textsc{Win}_i$, so Lemma 4.13 yields that $\rho \in Out(\mathcal{S}_i^n)$.
- If $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^*(-1)^\omega$, the definition of T ensures that for any index k such that $\rho_k \in V_i$, $\widehat{\mathrm{Val}}_i^{n-1}(\rho_k) = \widehat{\mathrm{Val}}_i^{n-1}(\rho_{k+1})$. Thus Lemma 4.14 yields that $\rho \in Out(\mathcal{S}_i^n)$.
- Otherwise $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^\omega$, and $\rho \in \textsc{Win}_i$ or there is an infinite number of visits to H_i^{n-1}. Then Lemma 4.16 allows to conclude.

Reciprocally, let $\rho \in Out(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})$. By monotonicity of the sets of admissible strategies, $\rho \in Out(\mathcal{S}^{n-1})$, so it remains to be proven that $\rho \in \mathcal{L}(\mathcal{A}_i^n)$. Notice that since $\rho \in Out(\mathcal{S}^n)$, $\rho \notin B_i^n$, so ρ is actually a run of $\mathcal{A}_i^n$.

Now by Lemma 4.12, we can consider the three following cases for $\mathrm{Val}_i^{n-1}(\rho)$.

- If $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^*(-1)^\omega$, then $\rho \in \mathcal{L}(\mathcal{A}_i^n)$.
- If $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^*1^\omega$. Let us write $\rho = Out(\sigma_i, \sigma_{-i})$ with $\sigma_i \in \mathcal{S}_i^n$ and $\sigma_{-i} \in \mathcal{S}_{-i}^{n-1}$. Let s be the first state of value 1 encountered in ρ. From s there exists a strategy σ_i^w for player i winning against all other strategy profiles of $\mathcal{S}_{-i}^{n-1}$. Therefore if σ_i is not a winning strategy, $\sigma_i^w \succ_{\mathcal{S}^{n-1}} \sigma_i$, which is a contradiction with the fact that $\sigma_i \in \mathcal{S}_i^n$. Thus σ_i is a winning strategy so $\rho \in \textsc{Win}_i$ and therefore $\rho \in \mathcal{L}(\mathcal{A}_i^n)$.
- Otherwise $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^\omega$ and Lemma 4.16 allows to conclude that either $\rho \in \textsc{Win}_i$ or $\rho \in (V^* H_i^{n-1})^\omega$, therefore that $\rho \in \mathcal{L}(\mathcal{A}_i^n)$. □

[5] Since the transitions of $\mathcal{G}$ bear no label, the "language" of an automaton is here considered to be the set of accepting runs.

We deduce from the preceding lemma the following property:

Lemma 4.20. $Out(\mathcal{S}^{n-1}) \cap \bigcap_{i\in P} \mathcal{L}(\mathcal{A}_i^n) = Out(\mathcal{S}^n)$

Proof.

$$\begin{aligned}
Out(\mathcal{S}^{n-1}) \cap \bigcap_{i\in P} \mathcal{L}(\mathcal{A}_i^n) &= \bigcap_{i\in P} (\mathcal{L}(\mathcal{A}_i^n) \cap Out(\mathcal{S}^{n-1})) \\
&= \bigcap_{i\in P} (Out(\mathcal{S}_i^n) \cap Out(\mathcal{S}^{n-1})) \\
&\qquad\qquad\qquad \text{(by Lemma 4.19)} \\
&= \left(\bigcap_{i\in P} Out(\mathcal{S}_i^n)\right) \cap Out(\mathcal{S}^{n-1}) \\
&= Out(\mathcal{S}^n) \cap Out(\mathcal{S}^{n-1}) \\
Out(\mathcal{S}^{n-1}) \cap \bigcap_{i\in P} \mathcal{L}(\mathcal{A}_i^n) &= Out(\mathcal{S}^n) \text{ because } \mathcal{S}^n \subseteq \mathcal{S}^{n-1}
\end{aligned}$$

□

As $Out(\mathcal{S}^0)$ is the set of all runs over $\mathcal{G}$, the previous lemma allows the construction of $Out(\mathcal{S}^n)$ by recurrence, for any n. Note that the size of the automaton accepting $Out(\mathcal{S}^n)$ does not grow since all automata in the intersection share the same structure as $\mathcal{G}$ (although some edges have been removed). The accepting condition, however, becomes more and more complex. Nonetheless, if for all $j \in P$, $\textsc{Win}_j$ is a circuit condition, the condition of the automaton accepting $Out(\mathcal{S}^n)$ is a Boolean combination of such conditions, thus it is expressible by a Boolean circuit, of polynomial size. For example, condition $\widehat{\mathrm{Val}}_i^{n-1}(\rho) \in 0^*1^\omega$ can be expressed by a circuit that is the disjunction of all states of value 1 at step $n-1$.

4.3 Inductive computation of values

Now, we show how to compute the values of the states for all players. Initially, at "iteration -1", all values are assumed to be 0, and $\mathcal{S}^{-1} = \mathcal{S}$. Computing the values at the next iteration relies on solving two-player zero-sum games with objectives based on outcomes of admissible strategies. For example, in order to decide whether a state s has value 1 for player i at iteration 0, i.e. whether $\mathrm{Val}_i^0(s) = 1$, one must decide whether player i has a winning strategy from s when playing against all other players. This corresponds to the game with winning condition $\textsc{Win}_i$ where vertices of all players but i belong to a single opponent. To decide whether $\mathrm{Val}_i^0(s) > -1$, all players try together to make i win. Therefore this is a one-player game (or emptiness check) on $\mathcal{G}$ with condition $\textsc{Win}_i$. All the other states have value -1.

This idea is extended to subsequent iterations, but the objectives of the games become more complex in order to take into account the previous iterations: the objectives need to enforce that only outcomes of admissible strategies (at previous iterations) are played. Hence the construction relies on the automata $\mathcal{A}_i^n$ built above. Assuming winning conditions for all players are circuit conditions, this yields a polynomial space algorithm to compute the values.

In the sequel, a strategy or an outcome "winning for objective X" is said *X-winning*, as not to confuse with simply *winning*, meaning "winning for objective $\textsc{Win}_i$" (i being usually clear from context).

4.3.1 Characterizing states of value -1

A state s has a value > -1 for player i at step n if there is a strategy profile $\sigma_P \in \mathcal{S}^n$ s.t. $Out_s(\sigma_P) \in \textsc{Win}_i$. This is expressed by: $\exists \sigma_P \in \mathcal{S}.\ \sigma_P \in \mathcal{S}^n \wedge Out_s(\sigma_P) \in \textsc{Win}_i$. This prompts the definition of the following objective: $\Psi_i^n(s) = Out_s(\mathcal{S}^n) \cap \textsc{Win}_i$.

Lemma 4.21. *If $\Psi_i^n(s) \neq \emptyset$, then $\exists \sigma_P \in \mathcal{S}^n : Out_s(\sigma_P) \in \textsc{Win}_i$.*

Therefore, since the set $Out_s(\mathcal{S}^n)$ is expressed as the language of an automaton, and $\textsc{Win}_i$ can clearly be defined as the language of an automaton, testing whether a state has value -1 boils down to emptiness check.

4.3.2 Characterizing states of value 1

A state s has value 1 for player i at step n if he has a strategy σ_i in $\mathcal{S}_i^n$ such that for all strategies σ_{-i} in $\mathcal{S}_{-i}^n$, $Out_s(\sigma_i, \sigma_{-i}) \in \textsc{Win}_i$. This is expressed by the formula:

$$\exists \sigma_i \in \mathcal{S}_i.\ \forall \sigma_{-i} \in \mathcal{S}_{-i}.\ \left(\sigma_i \in \mathcal{S}_i^n \wedge \left(\sigma_{-i} \in \mathcal{S}_{-i}^n \Rightarrow \textsc{Win}_i^s(\sigma_i, \sigma_{-i})\right)\right).$$

This prompts the definition of the objective $\Omega_i^n(s) = Out_s(\mathcal{S}_i^n) \cap (Out_s(\mathcal{S}^n) \Rightarrow \textsc{Win}_i)$. We show that there is indeed a correspondence between this objective Ω_i^n and states s such that $\mathrm{Val}_i^n(s) = 1$.

Lemma 4.22. *If $\sigma_i \in \mathcal{S}_i$ is a $\Omega_i^n(s)$-winning strategy, then there exists $\sigma_i' \in \mathcal{S}_i^n$ such that $\sigma_i(s \cdot h) = \sigma_i'(s \cdot h)$ for any history h.*

Proof. Let $\sigma_i^0 \in \mathcal{S}_i^n$. We show that for runs starting from s, σ_i is admissible. That directly yields the result with $\sigma_i' = \sigma_i^0\,[s \leftarrow \sigma_i]$.

We only consider runs starting in state s. We show that σ_i is not strictly dominated with respect to $\mathcal{S}^{n-1}$. Assume there exists a strategy profile $\sigma_P' \in \mathcal{S}^{n-1}$ whose outcome ρ' is winning for player i, but such that $\rho = Out_s(\sigma_i, \sigma_{-i}')$ is not winning for player i. Let k be the first index where $\rho_k \neq \rho_k'$. By hypothesis that σ_i is Ω_i^n-winning, $\rho \in Out_s(\mathcal{S}_i^n)$ and is the outcome of an admissible strategy $\alpha_i \in \mathcal{S}_i^n$. Note that $\rho = Out_s(\alpha_i, \sigma_{-i}')$ and is losing, while $\rho' = Out_s(\sigma_i', \sigma_{-i}')$ is winning, with $\sigma_{-i}' \in \mathcal{S}_{-i}^{n-1}$, so $\alpha_i \not\succcurlyeq_{\mathcal{S}^{n-1}} \sigma_i'$. By admissibility of α_i, $\sigma_i' \not\succ_{\mathcal{S}^{n-1}} \alpha_i$, therefore $\sigma_i' \not\succcurlyeq_{\mathcal{S}^{n-1}} \alpha_i$. Hence there exists $\sigma_{-i}'' \in \mathcal{S}^{n-1}$ that makes σ_i' lose from ρ_k'.

Now, σ_i is winning for objective Ω_i^n, and since $\rho_{\leq k}$ is compatible with σ_i, it is Ω_i^n-winning from ρ_k. In particular, if all players $j \neq i$ play according to a strategy of $\mathcal{S}_{-i}^n$ it is winning. Since the condition $V^\omega \setminus Out_s(\mathcal{S}_{-i}^n)$ cannot be satisfied, condition $\textsc{Win}_i$ is satisfied. Therefore the strategy profile $\sigma_{-i}'' \in \mathcal{S}^{n-1}$ makes σ_i win. Recall that σ_{-i}'' also makes σ_i' lose from ρ_k'. Hence σ_i is not strictly dominated with respect to $\mathcal{S}_i^{n-1}$ and it belongs to $\mathcal{S}_i^n$. ◻

Proposition 4.23. *A strategy of player i is a strategy of $\mathcal{S}_i^n$ which is winning from state s against all strategies of $\mathcal{S}_{-i}^n$ if, and only if, it is winning for objective $\Omega_i^n(s)$.*

Proof. Assume σ_i is a strategy of $\mathcal{S}_i^n$ which is winning from s against all strategies of $\mathcal{S}_{-i}^n$. Consider a strategy profile σ_{-i} and the run they produce: $\rho = Out_s(\sigma_i, \sigma_{-i})$. Obviously $\rho \in Out_s(\mathcal{S}_i^n)$ because $\sigma_i \in \mathcal{S}_i^n$. If $\rho \in Out_s\left(\mathcal{S}_{-i}^n\right)$, as σ_i is winning against the strategies of $\mathcal{S}_{-i}^n$ this means that ρ is winning for player i. Hence either $\rho \in \textsc{Win}_i$ or $\rho \in V^\omega \setminus Out_s\left(\mathcal{S}_{-i}^n\right)$. So σ_i wins for condition Ω_i^n.

Reciprocally, let σ_i be winning for $\Omega_i^n(s)$. Lemma 4.22 shows that σ_i can be completed as a strategy of $\mathcal{S}_i^n$ when starting from states other than s. We now show that it is winning from s against all strategies of $\mathcal{S}_{-i}^n$. Let $\sigma_{-i} \in \mathcal{S}_{-i}^n$ and $\rho = Out_s(\sigma_i, \sigma_{-i})$. The path ρ is in $Out_s(S_{-i}^n)$, so since σ_i is winning for Ω_i^n, it means that $\rho \in \textsc{Win}_i$. Hence σ_i is winning from s against all strategies of $\mathcal{S}_{-i}^n$. ◻

Notice that if player i takes a transition decreasing his value, we know that it is not playing a winning strategy. Conversely, if a player $j \neq i$ takes a transition that decreases her value then we immediately know that player i wins for objective Ω_i^n by playing an admissible strategy. We can therefore consider these two cases separately. We define the sets B_i^n, C_i^n, $B_i^{\leq n}$, $C_i^{\leq n}$ and $\overline{BC}_i^n$ by:

$$B_i^n = \{\rho \in V^\omega \mid \exists k.\ \rho_k \in V_i, \mathrm{Val}_i^{n-1}(\rho_{k+1}) < \mathrm{Val}_i^{n-1}(\rho_k)\};$$
$$C_i^n = \{\rho \in V^\omega \mid \exists k.\ \rho_k \in V_j, j \neq i, \mathrm{Val}_j^{n-1}(\rho_{k+1}) < \mathrm{Val}_j^{n-1}(\rho_k)\};$$
$$B_i^{\leq n} = \bigcup_{m=1}^{n} B_i^m; \qquad C_i^{\leq n} = \bigcup_{m=1}^{n} C_i^m; \qquad \overline{BC}_i^n = V^\omega \setminus (B_i^{\leq n} \cup C_i^{\leq n}).$$

Lemma 4.24. *For any player i and iteration n, the following implications hold:*

(1) If $\rho \in Out(\mathcal{S}_i^n)$, then $\rho \notin B_i^{\leq n}$.

(2) If $\rho \in Out(\mathcal{S}_{-i}^n)$, then $\rho \notin C_i^{\leq n}$.

(3) If $\rho \in Out(\mathcal{S}^n)$, then $\rho \in \overline{BC}_i^n$.

Proof. Since $Out(\mathcal{S}_i^{n-1}) \subseteq Out(\mathcal{S}_i^n)$, it is sufficient to prove the following.

(1') If $\rho \in Out(\mathcal{S}_i^n)$, then $\rho \notin B_i^n$.

(2') If $\rho \in Out(\mathcal{S}_{-i}^n)$, then $\rho \notin C_i^n$.

Implication (1') holds because of Lemma 4.11. Implication (2') is obtained by applying the same lemma to players $j \neq i$. Implication (3) is a consequence of (1) and (2). ☐

Objective Ω_i^n can be further decomposed with respect to these sets.

Proposition 4.25. *There is a winning strategy for $\Omega_i^n(s)$ if, and only, if there is one for: $\Theta_i^n(s) = C_i^{\leq n} \cup \left(\overline{BC}_i^n \cap \Omega_i^n(s)\right)$.*

Proof. Objective Θ_i^n can be rewritten

$$\Theta_i^n(s) = C_i^{\leq n} \cup \Big((V^\omega \setminus B_i^{\leq n}) \cap (V^\omega \setminus C_i^{\leq n}) \cap \Omega_i^n(s)\Big).$$

Let σ_i be a $\Omega_i^n(s)$-winning strategy. Let $\rho \in Out_s(\sigma_i)$. Note that since σ_i is $\Omega_i^n(s)$-winning, player i never decreases his value in ρ, therefore $\rho \in (V^\omega \setminus B_i^{\leq n})$. Therefore

- either $\rho \in C_i^{\leq n} \subseteq \Theta_i^n$,
- or $\rho \in V^\omega \setminus C_i^{\leq n}$, then $\rho \in (V^\omega \setminus B_i^{\leq n}) \cap (V^\omega \setminus C_i^{\leq n}) \cap \Omega_i^n(s) \subseteq \Theta_i^n(s)$.

Therefore a σ_i is also winning for Θ_i^n.

Reciprocally, if σ_i is winning for $\Theta_i^n(s)$. We build a $\Omega_i^n(s)$-winning strategy σ_i' as follows. Let $\alpha_i \in \mathcal{S}_i^n$ be an admissible strategy for player i. We set σ_i' that plays like σ_i until a player $j \neq i$ decreases his value, then plays like α_i.

Note that σ_i never decreases the value for i before shifting to α_i. Indeed, assume against a profile σ_{-i}, player i decreases his value after a history h (where no other player has decreased his own value). Let σ'_{-i} be a (positional) profile that never decrease a value for $j \neq i$ at any iteration and $\rho = Out_s(\sigma_i, \sigma_{-i}\,[h \leftarrow \sigma_i'])$. Then $\rho \notin C_i^{\leq n}$ while $\rho \in B_i^n \subseteq B_i^{\leq n}$, so $\rho \notin \Theta_i^n$, which is a contradiction with the fact that σ_i is Θ_i^n-winning.

As a result, applying Lemma 4.9 several (but a finite number of) times yields that σ_i' is admissible. So every run produced by σ_i' starting from s is by definition in $Out_s(\mathcal{S}_i^n)$. Also remark that σ_i' is a $\Theta_i^n(s)$-winning strategy.

Let σ_{-i} be a profile and let $\rho = Out_s(\sigma_i', \sigma_{-i})$. If for some $j \neq i$, $\rho \notin Out_s(\mathcal{S}_j^n)$, then $\rho \in \Omega_i^n(s)$ because $\rho \notin Out_s(\mathcal{S}_{-i}^n)$. Otherwise, by Lemma 4.24, $\rho \notin C_i^{\leq n}$ thus $\rho \in \Omega_i^n(s)$. ☐

4.3.3 Using the automata

We can now use the acceptance conditions of automata $(\mathcal{A}_j^m)_{m\leq n, j\in P}$ to rewrite Θ_i^n. Note that the state s is only relevant as the starting point of the runs, we can consider winning condition Θ_i^n regardless of the initial state: $\Theta_i^n = \bigcup_{s\in V} \Theta_i^n(s)$. Remark that we also have $\Theta_i^n(s) = \Theta_i^n \cap (s \cdot V^\omega)$. So, if we expand the definition of Ω_i^n in Θ_i^n, we obtain:

$$\Theta_i^n = C_i^{\leq n} \cup \left(\overline{BC}_i^n \cap \mathit{Out}(\mathcal{S}_i^n) \cap (\mathit{Out}(\mathcal{S}^n) \Rightarrow \textsc{Win}_i)\right).$$

The set $C_i^{\leq n}$ is easily definable by an automaton, so is $\textsc{Win}_i$, hence we just have to construct automata recognizing $\mathit{Out}(\mathcal{S}^n)$ and $\mathit{Out}(\mathcal{S}_i^n) \cap \overline{BC}_i^n$. In the former case, this is done through an intersection of automata $(\mathcal{A}_j^m)_{m\leq n, j\in P}$, as shown in Lemma 4.20 in Section 4.2.5. We now consider the latter case.

Computing $\mathit{Out}(\mathcal{S}_i^n) \cap \overline{BC}_i^n$. The construction for the language $\mathit{Out}(\mathcal{S}_i^n) \cap \overline{BC}_i^n$ is also based on the automaton $\mathcal{A}_i^n$. For this, we show that it can be defined through a combination of $\mathit{Out}(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})$ and of outcomes of admissible strategies at the previous iteration. Namely:

Lemma 4.26.

$$\begin{aligned} \mathit{Out}(\mathcal{S}_i^n) \cap \overline{BC}_i^n \quad = \quad & \mathit{Out}(\mathcal{S}_i^{n-1}) \cap \overline{BC}_i^n \\ & \cap \left(\mathit{Out}(\mathcal{S}^{n-1}) \Rightarrow \mathit{Out}(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})\right). \end{aligned}$$

Proof. $\boxed{\subseteq}$ The inclusion holds because $\mathcal{S}_i^n \subseteq \mathcal{S}_i^{n-1}$ and

$$\mathit{Out}(\mathcal{S}_i^n) \cap \mathit{Out}(\mathcal{S}^{n-1}) = \mathit{Out}(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1}).$$

$\boxed{\supseteq}$ In the other direction, let ρ be such that

$$\rho \in \overline{BC}_i^n \cap \mathit{Out}(\mathcal{S}_i^{n-1}) \cap \left(\mathit{Out}(\mathcal{S}^{n-1}) \Rightarrow \mathit{Out}(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})\right).$$

If ρ is an outcome of $\mathcal{S}^{n-1}$ then it is an outcome of $(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1})$ and therefore in $\mathit{Out}(\mathcal{S}_i^n)$.

Otherwise, ρ is in $\mathit{Out}(\mathcal{S}_i^{n-1}) \setminus \mathit{Out}(\mathcal{S}^{n-1})$. Let α_i be a strategy of $\mathcal{S}_i^n$, we define σ_i that follows ρ and revert to α_i if the history deviates from ρ. Formally, for a history h, such that $\mathrm{last}(h) \in V_i$:

- if there is k such that $h = \rho_{\leq k}$, then $\sigma_i(h) = \rho_{k+1}$; this is to ensure that ρ is an outcome of σ_i;
- otherwise let k be the last index such that $\rho_{\leq k}$ is a prefix of h, then $\sigma_i(h) = \alpha_i(h_{>k})$.

Note that $\mathit{Out}(\sigma_i) \cap B_i^n = \emptyset$. Indeed, the only outcome of σ_i which might not be in $\mathit{Out}(\mathcal{S}_i^n)$ is ρ which does not belong to B_i^n by assumption.

We show that σ_i is in $\mathcal{S}_i^n$. Assume there is a strategy profile $\sigma'_P \in \mathcal{S}^{n-1}$ whose outcome ρ' is winning for player i but such that $\rho'' = \mathit{Out}(\sigma_i, \sigma'_{-i})$ is not. The path ρ'' is different from ρ since it belongs to $\mathit{Out}(\mathcal{S}^{n-1})$.

However, as noted before, $\rho'' \notin B_i^n$. Let k (resp. k') be the first index where ρ'' differs from ρ (resp. ρ'). We distinguish two cases, once again as in Figure 5. If $k' \geq k$, σ_i is playing according to a non-dominated strategy from $\rho_{\leq k+1}$, hence it is not strictly dominated by σ'_i.

Otherwise, $k' < k$. Let $s = \rho''_{k'}$, it is controlled by player i. Let $s' = \sigma'_i(\rho_{\leq k'})$, and $s'' = \sigma_i(\rho_{\leq k'})$. Note that since $\rho \notin B_i^n$, $\mathrm{Val}_i^{n-1}(s) = \mathrm{Val}_i^{n-1}(s'')$. Additionally, if $\mathrm{Val}_i^{n-1}(s) > \mathrm{Val}_i^{n-1}(s')$, then a profile that fares better for σ_i than for σ'_i can easily be found, so $\sigma'_i \not\succ_{\mathcal{S}^{n-1}} \sigma_i$. Therefore we assume $\mathrm{Val}_i^{n-1}(s) = \mathrm{Val}_i^{n-1}(s')$ and distinguish cases according to the three possible values.

- If $\mathrm{Val}_i^{n-1}(s) = -1$, then it is not possible for σ'_P to be winning, hence a contradiction.
- If $\mathrm{Val}_i^{n-1}(s) = 1$. First assume that $\mathrm{Val}_i^{n-1}(\rho''_{k+1}) = 1$. Then, since σ_i plays according to a strategy of $\mathcal{S}_i^n$ after $\rho''_{\geq k+1}$, ρ'' is a winning path. This contradicts our hypothesis.

  Otherwise, consider the first state of $\rho''_{\geq k'}$ that does not have value 1: the smallest $k'' > k'$ such $\mathrm{Val}_i^{n-1}(\rho''_{k''}) < 1$. As a result, and since $\rho'' \notin B_i^n$, $\rho''_{k''} = \rho_{k''} \in V_j$ for some $j \neq i$. Since $\rho \notin C_i^n$, $\mathrm{Val}_j^{n-1}(\rho''_{k''}) = \mathrm{Val}_j^{n-1}(\rho''_{k''-1})$. By Lemma 4.9, from the strategy of $\mathcal{S}_{-i}^{n-1}$ from $\rho''_{k''}$ that makes player i lose, we can construct another one that makes player i lose from $\rho''_{k''-1}$. This contradicts the fact that $\mathrm{Val}_i^{n-1}(\rho''_{k''-1}) = 1$.
- If $\mathrm{Val}_i^{n-1}(s) = 0$. Then one can find a profile that makes σ'_i lose. First assume that $\mathrm{Val}_i^{n-1}(\rho''_{k+1}) \geq 0$. Then a profile can be found that makes σ_i win, since σ_i plays according to an admissible strategy from that point.

  Otherwise, consider the step in which a player $j \neq i$ decreases Val_i^{n-1}: the smallest $k'' > k'$ such $\mathrm{Val}_i^{n-1}(\rho_{k''+1}) = -1$. We have that $\rho_{k''} \in V_j$ for some $j \neq i$. Since $\mathrm{Val}_i^{n-1}(\rho_{k''}) = 0$, there exists a successor state of $\rho_{k''}$ of the same value for player j that is of value ≥ 0 for player i: a state $q \neq \rho_{k''+1}$ such that $\rho_{k''} \to q$, $\mathrm{Val}_i^{n-1}(q) \geq 0$, and $\mathrm{Val}_j^{n-1}(\rho_{k''}) = \mathrm{Val}_j^{n-1}(q)$. Then from q there is a profile $\sigma''_{-i} \in \mathcal{S}_{-i}^{n-1}$ that makes σ_i win, since σ_i plays according to an admissible strategy from that point. Notice that the profile that plays like σ'_{-i} up to $\rho_{k''}$ then goes to q and shifts to σ''_{-i} is still a profile of $\mathcal{S}_{-i}^{n-1}$.

Hence σ_i is admissible and $\rho \in Out(\mathcal{S}_i^n)$, which concludes the proof. □

Lemma 4.27. *$Out(\mathcal{S}_i^n) \cap \overline{BC}_i^n$ is equal to:*
$Out(\mathcal{S}_i^{n-1}) \cap \overline{BC}_i^{n-1} \cap \left(Out(\mathcal{S}^{n-1}) \Rightarrow \mathcal{L}(\mathcal{A}_i^n)\right) \cap (V^\omega \setminus B_i^n) \cap (V^\omega \setminus C_i^n)$.

Proof. Remark that $B_i^{\leq n} = B_i^n \cup B_i^{\leq n-1}$ and $C_i^{\leq n} = C_i^n \cup C_i^{\leq n-1}$, so

$$\begin{aligned}
\overline{BC}_i^n &= V^\omega \setminus (B_i^{\leq n} \cup C_i^{\leq n}) \\
&= V^\omega \setminus (B_i^n \cup B_i^{\leq n-1} \cup C_i^n \cup C_i^{\leq n-1}) \\
&= \left(V^\omega \setminus (B_i^{\leq n-1} \cup C_i^{\leq n-1})\right) \cap (V^\omega \setminus B_i^n) \cap (V^\omega \setminus C_i^n) \\
\overline{BC}_i^n &= \overline{BC}_i^{n-1} \cap (V^\omega \setminus B_i^n) \cap (V^\omega \setminus C_i^n). \qquad (1)
\end{aligned}$$

In addition, $Out(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1}) = Out(\mathcal{S}^{n-1}) \cap \mathcal{L}(\mathcal{A}_i^n)$, so

$$\begin{aligned}
Out(\mathcal{S}^{n-1}) &\Rightarrow Out(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1}) = \\
&\qquad Out(\mathcal{S}^{n-1}) \Rightarrow \left(Out(\mathcal{S}^{n-1}) \cap \mathcal{L}(\mathcal{A}_i^n)\right) \\
Out(\mathcal{S}^{n-1}) &\Rightarrow Out(\mathcal{S}_i^n, \mathcal{S}_{-i}^{n-1}) = Out(\mathcal{S}^{n-1}) \Rightarrow \mathcal{L}(\mathcal{A}_i^n). \qquad (2)
\end{aligned}$$

Rewriting Lemma 4.26 with Equations (1) and (2) yields the result. □

The previous lemma provides a recurrence relation to compute the intersection $Out(\mathcal{S}_i^n) \cap \overline{BC}_i^n$, with base case being all the runs of $\mathcal{G}$.

4.4 Algorithm for circuit conditions

The above construction yields procedures to compute the values, and in turn to solve the winning coalition problem.

Proposition 4.28. *Checking whether $\mathrm{Val}_i^n(s) = 1$ is in* PSPACE.

Proof. We define the following two-player game $\mathcal{G}_i^n$ as follows. $\mathcal{G}_i^n$ has the structure of $\mathcal{G}$, except that the edges corresponding to a player decreasing his own value at any previous iteration have been removed (thus avoiding $B^{\leq n}$ and $C^{\leq n}$ syntactically). The players are `Eve` and `Adam`, the states of `Eve` are V_i and the states of `Adam` are $\bigcup_{j \neq i} V_j$. The winning condition for `Eve` is Θ_i^n, which is a circuit condition.

If `Eve` has a winning strategy starting from state s, then $\mathrm{Val}_i^n(s) = 1$, as shown by Propositions 4.23 and 4.25. Recall that for circuit condition, deciding whether there exists a winning strategy for `Eve` is in PSPACE [15]. □

Proposition 4.29. *Checking whether* $\mathrm{Val}_i^n(s) = -1$ *is in* coNP*.*

Proof. This is done by checking the emptiness of Ψ_i^n when starting from state s. Again, this is a circuit condition on $\mathcal{G}$ where edges decreasing the value of their owner are removed. The emptiness check is in coNP since testing non-emptiness amounts to guessing a valuation satisfying the circuit then guessing a lasso path starting from s and looping in exactly the states of the guessed valuation. □

Proof of Theorem 4.1. Let W, L be the set of players that must win and lose, respectively, in the instance of the winning coalition problem. The winning coalition problem is thus equivalent to the non-emptiness of

$$\Phi = Out(\mathcal{S}^*) \cap \bigcap_{i \in W} \mathrm{W\scriptstyle IN}_i \cap \bigcap_{i \in L} \neg \mathrm{W\scriptstyle IN}_i$$

over $\mathcal{G}$, which is a circuit condition. This check is done in NP.

This however requires the computation of the condition corresponding to $Out(\mathcal{S}^*)$, hence of the values for all intermediate iterations before the fix-point. Each iteration means solving a circuit game, which is done in polynomial space. Moreover, by Proposition 4.18, there are at most $|P| \cdot |V|$ iterations. Thus the overall complexity is in PSPACE. □

4.5 Büchi objectives

In the case where the winning conditions for each players are Büchi objectives, a careful analysis of condition Θ_i^n shows that it can be reformulated as a parity condition. Hence computing the value of a state boils down to solving two-player parity games, which can be done in $\mathsf{UP} \cap \mathsf{coUP}$ (and is suspected to be polynomial). This idea is the basis for the proof of Theorem 4.2, which is detailed in the remainder of the section.

In this section, We assume that each winning condition $\mathrm{W\scriptstyle IN}_i$ is given by a Büchi set F_i. In this special case, we show that the objective Θ_i^n can in fact be written as a parity condition. It follows that if we are given an oracle to solve two-player parity games, we have a polynomial algorithm. Parity games are known to be in $\mathsf{UP} \cap \mathsf{coUP}$, but the question whether a polynomial algorithm exists for parity games has been open for several years [16, 7]. Our algorithm thus works in polynomial time with call to oracles in $\mathsf{NP} \cap \mathsf{coNP}$, hence is also in $\mathsf{NP} \cap \mathsf{coNP}$ [17].

A parity condition is given by a coloring function $\chi : V \to \{0, \ldots, M\}$ with $M \in \mathbb{N}$. Accepted runs with respect to the coloring functions are the ones where the maximal color visited infinitely often is even: $\mathrm{W\scriptstyle IN}_\chi = \{\rho \mid \max(\mathrm{Inf}(\chi(\rho))) \text{ is even}\}$.

4.5.1 Expressing Θ_i as a parity condition.

First note that the acceptance condition of $\mathcal{L}(\mathcal{A}_i^n)$ can be expressed by the following Büchi set, assuming that we remove all the edges that decrease the value of player i:

$$\begin{aligned} K_i^n = & \{s \mid \mathrm{Val}_i^{n-1}(s) = -1\} \cup \{s \mid \mathrm{Val}_i^{n-1}(s) = 1 \wedge s \in F_i\} \\ & \cup \{s \mid \mathrm{Val}_i^{n-1}(s) = 0 \wedge s \in (F_i \cup H_i^{n-1})\} \end{aligned}$$

Note that $F_i \subseteq K_i^n$ for every n.

$Out(\mathcal{S}^0)$ is accepted by the automaton with the same structure as $\mathcal{G}$ and where the Büchi set is V. Since $Out(\mathcal{S}^{n-1}) \cap \bigcap_{i \in P} \mathcal{L}(\mathcal{A}_i^n) = Out(\mathcal{S}^n)$ by Lemma 4.20, $Out(\mathcal{S}^n)$ is recognized by an automaton whose acceptance condition is a conjunction of $n \times |P|$ Büchi conditions. By taking a product of the game with an automaton of size $n \times |P|$, a condition $Out(\mathcal{S}^n)$ can be expressed as a single Büchi set that we write D^m.

Recall that condition Θ_i^n is

$$C_i^{\leq n} \cup \left(\overline{BC}_i^n \cap Out(\mathcal{S}_i^n) \cap (Out(\mathcal{S}^n) \Rightarrow \text{WIN}_i)\right).$$

We isolate the prefix-independent part of Θ_i^n by defining

$$\Gamma_i^n = Out(\mathcal{S}_i^n) \cap (Out(\mathcal{S}^n) \Rightarrow \text{WIN}_i)$$

which, by unfolding the recurrence relation of Lemma 4.27, can be rewritten:

$$\Gamma_i^n = \bigcap_{m=0}^{n-1} \left(Out(\mathcal{S}^m) \Rightarrow \mathcal{L}(\mathcal{A}_i^{m+1})\right) \cap (Out(\mathcal{S}^n) \Rightarrow \text{WIN}_i)$$

Note that the consequence of each implications can be strengthened by the premise:

$$\Gamma_i^n = \bigcap_{m=0}^{n-1} \left(Out(\mathcal{S}^m) \Rightarrow \left(\mathcal{L}(\mathcal{A}_i^{m+1}) \cap Out(\mathcal{S}^m)\right)\right) \qquad \cap (Out(\mathcal{S}^n) \Rightarrow \text{WIN}_i)$$

Also remark that the "prefix-dependent" part of Θ_i^n can be expressed by either removing edges (to avoid sets B_i) or adding an additional winning state (for sets C_i).

Each set $Out(\mathcal{S}^m)$ is expressed as a conjunction of $m \times |P|$ Büchi conditions. Similarly, condition $\left(\mathcal{L}(\mathcal{A}_i^{m+1}) \cap Out(\mathcal{S}^m)\right)$ is a conjunction of $m \times |P| + 1$ Büchi conditions. Therefore, in order to consider all these different conditions on the same game, it can be assumed that $\mathcal{G}$ is synchronized with a product of size[6] $O(n^2 \cdot |P|)$. Then each condition $Out(\mathcal{S}^m)$ can be expressed as a single Büchi set that we write D^m. Also, each condition $\left(\mathcal{L}(\mathcal{A}_i^{m+1}) \cap Out(\mathcal{S}^m)\right)$ can be expressed as a single Büchi set that we write E_i^{m+1}.

Notice that we have a "chain" of inclusion of the various objectives. For each index m and player i:

$$Out(\mathcal{S}^m) \cap \mathcal{L}(\mathcal{A}_i^{m+1}) \subseteq Out(\mathcal{S}^m) \subseteq Out(\mathcal{S}^{m-1}) \cap \mathcal{L}(\mathcal{A}_i^m). \tag{3}$$

Hence, if in an implication $Out(\mathcal{S}^m) \Rightarrow \left(Out(\mathcal{S}^m) \cap \mathcal{L}(\mathcal{A}_i^{m+1})\right)$ the left operand is satisfied, then all such implications are satisfied for indexes $k < m$. And if the right operand is satisfied, then it is the case for indexes $k \leq m$. The condition Γ_i^n can then be defined by the parity condition given by the following coloring function χ_i^n:

- if $s \in F_i$ then $\chi_i^n(s) = 2n + 2$
- otherwise $\chi_i^n(s)$ is the maximum value between 0,

  $$2 \cdot \max\{m \mid s \in E_i^m\}, \qquad \text{and} \qquad 2 \cdot \max\{m \mid s \in D^m\} + 1,$$

  with the convention that $\max \emptyset = -\infty$.

Lemma 4.30. *For all index n and player i, $\rho \in \Gamma_i^n$ if, and only if, ρ satisfies the parity condition χ_i^n.*

[6] Precisely $|P| \cdot (n-1) \cdot (n-2) + n$.

Proof. Assume $\rho \in \Gamma_i^n$. Let $m \leq n$ be the largest index such that $\rho \in Out(\mathcal{S}^m)$.

If $m = n$ then $\rho \in \textsc{Win}_i$, therefore F_i is visited infinitely often by ρ. The highest color visited infinitely often is thus $2n+2$ and ρ satisfies condition χ_i^n.

If $m < n$ then ρ is not in $Out(\mathcal{S}^{m+1})$ and therefore, for $m' > m$, not in $Out(\mathcal{S}^{m'})$. Hence ρ does not visit infinitely often odd colors greater than $2m+2$. Moreover, since $\rho \in \Gamma_i^n$, $\rho \in Out(\mathcal{S}^m) \cap \mathcal{L}(\mathcal{A}_i^{m+1})$ hence it visits infinitely often states of E_i^{m+1}, which have an even color of (at least) $2m+2$. This means that ρ satisfies condition χ_i^n.

Reciprocally, assume that ρ satisfies χ_i^n. Let c be the highest color that is reached infinitely often.

If $c = 2n+2$ then F_i is visited infinitely often, therefore $\rho \in \textsc{Win}_i$. Moreover, for every $m \leq n$, $F_i \subseteq K_i^m$ so K_i^m is also visited infinitely often, hence $\rho \in \mathcal{L}(\mathcal{A}_i^{m+1}) \subseteq Out(\mathcal{S}^m) \Rightarrow \big(Out(\mathcal{S}^m) \cap \mathcal{L}(\mathcal{A}_i^{m+1})\big)$, therefore $\rho \in \Gamma_i^n$.

Otherwise $c = 2m+2$ with $m < n$. Then set E_i^{m+1} is visited infinitely often while, for every $m' > m$, the set $D^{m'}$ is not. So for $m' > m$, ρ is not in $Out(\mathcal{S}^{m'})$, hence all implications are satisfied of index $> m$. Note that it is also the case for $m' = n$, so $\rho \in Out(\mathcal{S}^n) \Rightarrow \textsc{Win}_i$.

Moreover, since $\rho \in Out(\mathcal{S}^m) \cap \mathcal{L}(\mathcal{A}_i^{m+1})$, for all indexes $m' \leq m$, Equation (3) yields that $\rho \in Out(\mathcal{S}^{m'}) \cap \mathcal{L}(\mathcal{A}_i^{m'+1}$. So the implication $Out(\mathcal{S}^{m'}) \Rightarrow \Big(Out(\mathcal{S}^{m'}) \cap \mathcal{L}(\mathcal{A}_i^{m'+1})\Big)$ is satisfied. Therefore $\rho \in \Gamma_i^n$. $\square$

4.5.2 Solving the winning coalition problem for Büchi objectives.

We showed that all objectives encountered in the computation of values boil down to parity objectives. This yields the following proof:

Proof of Theorem 4.2. By reformulating Θ_i^n as a parity condition, checking whether the value of a state is 1 amounts to solving a two player game with parity condition. This is known to be in $\mathsf{UP} \cap \mathsf{coUP}$, although suspected to be in P. Similarly, condition Ψ_i^n is a conjunction of Büchi objectives, hence solving its emptiness – which means deciding whether the value of a state is -1 – is in P.

Let n_0 be the index where the sets of admissible strategies stabilize (we have $n_0 \leq |P| \cdot |V|$). Let W, L be the set of players that should win and lose, respectively. Solving the winning coalition problem amounts to solving emptiness of

$$\Phi = Out(\mathcal{S}^{n_0}) \cap \bigcap_{i \in W} \textsc{Win}_i \cap \bigcap_{i \in L} \neg\textsc{Win}_i.$$

As the conjunction of $n_0 + |W|$ Büchi conditions and a coBüchi condition, it can be expressed as a parity condition with 3 colors. $\square$

4.6 Model-checking under admissibility

This section is devoted to the proof of Theorem 4.3.

Given a LTL formula ϕ, the model-checking under admissibility problem is equivalent to the emptiness of $\Phi = \mathcal{L}(\neg\phi) \cap Out(\mathcal{S}^*)$. It was proven in [18] that the language $\mathcal{L}(\neg\phi)$ can be represented by a Büchi deterministic automaton $\mathcal{A}$ with set of states Q such that:

- the size of each state of $\mathcal{A}$ is polynomial in $|\phi|$;
- it can be checked if a state is initial in space polynomial in $|\phi|$;
- it can be checked if a state is in F in space polynomial in $|\phi|$;

- each transition of $\mathcal{A}$ can be checked in space polynomial in $|\phi|$.

Lemma 4.31. *Let $\mathcal{A}$ be an automaton, then for any run ρ of the automaton, there exists π and τ such that:*

- *$\rho' = \pi \cdot \tau^\omega$ is a run of $\mathcal{A}$;*
- *$\mathrm{Occ}(\rho') = \mathrm{Occ}(\rho)$;*
- *$\mathrm{Inf}(\rho') = \mathrm{Inf}(\rho)$;*
- *$|\pi| + |\tau| \leq 2 \cdot n^2$ where n is the number of states of $\mathcal{A}$.*

The proof can be seen as a particular case of Lemma 2.2 in [19].

Proof. We inductively construct a history $\pi = \pi_0\pi_1 \dots \pi_n$ that is not too long and visits precisely those states that are visited by ρ.

The initial state is $\pi_0 = \rho_0$. Then assume we have constructed $\pi_{\leq k} = \pi_0 \dots \pi_k$ which visits exactly the same states as $\rho_{\leq k'}$ for some k'. If all the states of ρ have been visited in $\pi_{\leq k}$ then the construction is over. Otherwise there is an index i such that ρ_i does not appear in $\pi_{\leq k}$. The smallest such i is therefore the next *target*: define $t(\pi_{\leq k}) = \min\{i \mid \forall j \leq k.\ \pi_j \neq \rho_i\}$. Now consider the occurrence of the current state π_k that is the closest to the target in ρ: $c(\pi_{\leq k}) = \max\{i < t(\pi_{\leq k}) \mid \pi_k = \rho_i\}$. Run π emulates what happens at that position by choosing $\pi_{k+1} = \rho_{c(\pi_{\leq k})+1}$. Then π_{k+1} is either the target $\rho_{t(\pi_{\leq k})}$, or a state that has already been seen before in $\pi_{\leq k}$, in which case the resulting $\pi_{\leq k+1}$ visits exactly the same states as $\rho_{\leq c(\pi_{\leq k})+1}$.

At each step, either the number $|\mathrm{Occ}(\rho) \setminus \mathrm{Occ}(\pi_{\leq k})\,|$ of remaining targets strictly decreases, or the number of remaining targets is constant but the distance $t(\pi_{\leq k}) - c(\pi_{\leq k})$ to the next target strictly decreases. Therefore the construction terminates. Moreover, notice that between two targets the same state is never visited twice, and only states that have already been visited, or the target, are visited. As the number of targets is bounded by n, we obtain that the length of the path π is bounded by $1 + |n| \cdot (|n| - 1)/2$. In addition, states of π are always picked from ρ, ensuring that at each step $\mathrm{Occ}(\pi_{\leq k}) \subseteq \mathrm{Occ}(\rho)$.

Using similar ideas, we now inductively construct $\tau = \tau_0\tau_1 \dots \tau_m$, which visits precisely those states which are seen infinitely often along ρ, and which is not too long. Let ℓ be the least index after which the states visited by ρ are visited infinitely often: $\ell = \min\{i \mid \forall j \geq i.\ \rho_j \in \mathrm{Inf}(\rho)\}$. The run $\rho_{\geq \ell}$ is such that $\mathrm{Inf}(\rho_{\geq \ell}) = \mathrm{Inf}(\rho)$. Run τ can be built in the same way as π above, but for play $\rho_{\geq \ell}$. As a by-product, we also get $c(\tau_{\leq k})$, for $k < m$.

For $\pi \cdot \tau^\omega$ to be a real run, π must be joined to τ, and τ must be joined to itself. The last state of π must be the first state of τ (and similarly the last state of τ must also be its first state). This possibly requires appending some more states to π and τ: the target of π and τ is set to be τ_0, and the same construction as previously is applied. As before, the added states belong to ρ (resp. $\rho_{\geq \ell}$), ensuring that $\mathrm{Occ}(\pi) = \mathrm{Occ}(\rho)$ (resp. $\mathrm{Inf}(\tau) = \mathrm{Inf}(\rho)$). The total length of the resulting paths π and τ is bounded by $1 + (n-1) \cdot (n+2)/2$ which less than n^2. ☐

Lemma 4.32. *There is a word in Φ if and only if there are histories π and τ of length bounded by $(|\mathcal{G}| \times |\mathcal{A}|)^2$, such that $\pi \cdot \tau^\omega$ is a path in the product $\mathcal{G} \times \mathcal{A}$, τ^ω satisfies the circuit condition defining Out($\mathcal{S}^*$), and τ visits an accepting state of $\mathcal{A}$.*

Proof. By applying the preceding lemma to the product automaton $\mathcal{G} \times A$. ☐

MCunderADM($\mathcal{G}_0, \phi$)::= **begin**
  Compute $(\mathcal{G}, \psi)$ as automaton with circuit condition for $Out(\mathcal{S}^*(\mathcal{G}_0))$; `// see Section 4.2.5.`
  Compute the machine representing the Büchi automaton $\mathcal{A}$ accepting $\mathcal{L}(\neg\phi)$;
  $\ell := 0$;
  Nondeterministically set $s \in V$ and $q \in Q$ an initial state of $\mathcal{A}$;
  $In\pi := true$;
  while $In\pi \wedge \ell \leq (|\mathcal{G}| \times |\mathcal{A}|)^2$ **do**
    Guess s' such that $s \to s'$; $s := s'$;
    Guess q' such that $q \xrightarrow{s'} q'$; $q := q'$;
    Guess a boolean value for $In\pi$; $\ell := \ell + 1$;
  if $In\pi$ **then return** *false* ; `// ℓ reached its bound`
  else
    $t := s$; $p := q$; $\ell := 0$; $X := \emptyset$; $b := false$;
    while $\neg((s,q) = (t,p) \wedge X \vDash \psi \wedge b) \wedge \ell \leq (|\mathcal{G}| \times |\mathcal{A}|)^2$ **do**
      Guess s' such that $s \to s'$; $s := s'$; $X := X \cup \{s'\}$;
      Guess q' such that $q \xrightarrow{s'} q'$; $q := q'$; $b := b \vee (q' \in F)$;
      $\ell := \ell + 1$;
    if $(s,q) = (t,p) \wedge X \vDash \psi \wedge b$ **then return** *true*;
    else return *false* ; `// ℓ reached its bound`

Algorithm 1: PSPACE algorithm for the model-checking under admissibility problem.

From this lemma we deduce the non-deterministic polynomial space Algorithm 1 to decide the model-checking under admissibility problem.

The algorithm looks for π and τ satisfying the conditions of Lemma 4.32. It remembers the length ℓ of the path guessed so far. This length is bounded[7] by $(|\mathcal{G}| \times |\mathcal{A}|)^2$, which is exponential in the input, hence can be stored in polynomial space. The algorithm also stores the current states of both $\mathcal{G}$ and A (s and q, respectively) and the last states of $\mathcal{G}$ and A at the end of π – therefore the beginning of τ – (t and p, respectively). Since there is only an exponential number of states in $\mathcal{A}$ (and a polynomial one in $\mathcal{G}$), coding a state can be done in polynomial space. In order to check that the winning conditions are satisfied, the algorithm remembers the set X of states of $\mathcal{G}$ that have been visited (by τ), and the Boolean b telling whether an accepting state of $\mathcal{A}$ as been visited (again, by τ). In the end, the fact that τ is actually a looping path is also checked by testing $(s,q) = (t,p)$. Note that the paths π and τ in $\mathcal{G}$ must respect the local conditions (*e.g.* a player does not take an edge that decreases his value).

The correctness of the algorithm follows by Lemma 4.32, it terminates and use only polynomial space. Hence the model-checking under admissibility problem is in PSPACE.

PSPACE-hardness derives from the PSPACE-hardness of LTL model-checking [10]. This concludes the proof of Theorem 4.3.

[7] If the bound is reached, the algorithm rejects the run.

5 Example: a metro system

5.1 The model of the system

We illustrate the use of model-checking under admissibility over the following metro system. The metro track is composed of a directed ring divided in n slots (one can assume for example that even-numbered slots represent a station, while odd-numbered slots represent a section of track between two stations). There are p trains numbered 1 to p on the track, initially at positions $p-1, \dots, 0$ (hence train number 1 starts in front of the others (see Figure 7). In the sequel we study the case of a track of length 6 with two trains t_1 and t_2.

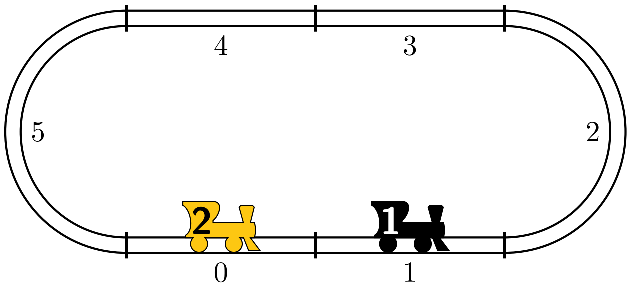


Figure 7: Metro track with $n = 6$ and $p = 2$, at their initial position $(1, 0)$.

At each step, all the trains successively declare whether they want to advance or not. Once everyone has chosen, all trains try to move synchronously. However, there is an evil environment *env* that can prevent trains from moving (for example by activating an alarm on the section the train is trying to move in). That means some trains that wanted to move may in fact remain in their position. Nevertheless, all other trains must comply with their original choice, modeling the fact that trains cannot communicate[8] in real-time. Additionally, if at any point two trains are on the same track section, they collide and the game stops. (Part of) the game graph for this protocol is depicted in Figure 8. In this game, the configuration of the track is given by the pair of positions for t_1 and t_2, and their respective wish to move or stay in place by the symbols "$\rightarrow$" (advance) or "$\downarrow$" (stay in place).

The objective of both trains is to loop infinitely often, which can be expressed by a generalized Büchi winning condition[9] requiring visiting infinitely often both sections 0 and 1. The objective of the environment is to give rise to a collision: since collision are sink states it is equivalent to a Büchi winning condition over the set of collision states. Globally, we are interested in knowing whether the LTL formula $\psi_{\neg\text{coll}}$ that expresses that no collision ever happens is satisfied under admissibility. Remark that here:

- no player has a winning strategy alone;
- the formula $\psi_{\neg\text{coll}}$ is not verified on all paths of the model;
- that the players are not *a priori* trying to satisfy $\psi_{\neg\text{coll}}$: indeed, the environment's objective is to negate $\psi_{\neg\text{coll}}$.

In the sequel we will show that the mechanism of admissibility enforces the satisfaction of $\psi_{\neg\text{coll}}$, although each train cannot guarantee that its objective will be fulfilled and $\psi_{\neg\text{coll}}$ is not an explicit requirement in the objective of any player.

[8] Although some kind of communication permits the trains to make their choice knowing in which sections the other trains are and what choice has been made by the trains with lower indices.

[9] This condition can therefore be expressed by a Büchi condition on a graph with memory of visits to 0 or 1. It can also be expressed by a circuit condition encoding formula $\bigvee(0, \_) \wedge \bigvee(1, \_)$.

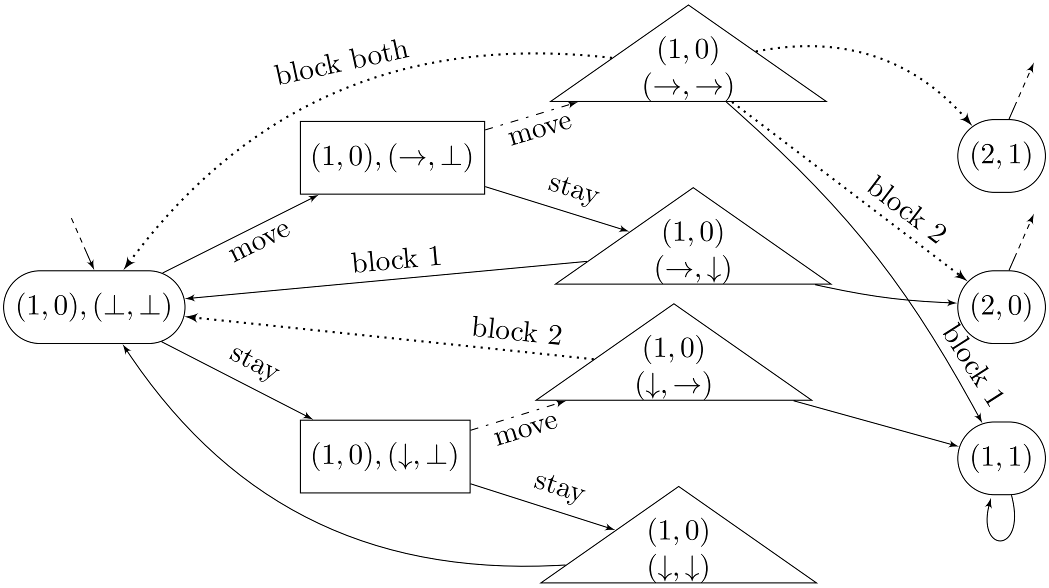


Figure 8: Part of the game played by the trains and the environment when trying to move from the original position. Round states belong to train t_1, square ones to train t_2, and triangular ones to the environment. Dotted edges are eliminated in the first iteration. Dashed ones in the second iteration.

5.2 Strategy elimination

5.2.1 Value computation for first iteration

At the first iteration, players for both trains never have a winning strategy for their winning condition that requires infinitely many visits to both positions 0 and 1: they cannot prevent, for example, the environment from always blocking their movement. Note that one train can also make the other one lose on its own, by choosing to always stay in the same position. On the other hand, in all states but the one with a collision, *e.g.* state $(1, 1)$, both trains may achieve their goal if the other players cooperate. Hence the value of states for both train players is 0 except in collision states, where it is -1.

For the environment player, however, there are states where he can force a collision, and hence has a winning strategy: whenever train 2 is right behind train 1 and train 2 has decided to move, which happens for example in states $(1, 0), (\rightarrow, \rightarrow)$ and $(1, 0), (\downarrow, \rightarrow)$ in Figure 8. These states therefore have value 1 for the *env*. Additionally, from collision states the environment has already won, hence their value is also 1. From other states, if trains chose to never move, for example, *env* will not achieve its goal. And they can also chose to reach a winning state for *env*, hence the value from other states is 0.

5.2.2 Strategy elimination for first iteration

From the values computed above, we first see that *env* eliminates all strategies that do not go to a collision state whenever it is possible: this amounts to removing some edge that originate from winning state, for example the dotted edges stemming from states $(1, 0), (\rightarrow, \rightarrow)$ and $(1, 0), (\downarrow, \rightarrow)$ in Figure 8.

Note that the strategy of *env* that always blocks train 1 if he is the only train wishing to move (in state $(1, 0), (\rightarrow, \downarrow)$), which may end up losing for *env* if both players repeat their choice, is not eliminated, since this play traverses infinitely often state $(1, 0), (\rightarrow, \perp)$, which is a “Help!”-state for player *env*.

Players t_1 eliminates for example the strategy that always stays in the same position, because it is always losing with respect to his objective. However, although it is intuitively not a good choice for t_1 to always choose to "stay" in position 1, the play

$$\Big((0,1),(\perp,\perp) \xrightarrow{stay} (1,0),(\downarrow,\perp) \xrightarrow{stay} (1,0),(\downarrow,\downarrow) \longrightarrow (0,1),(\perp,\perp)\Big)^{\omega}$$

is an outcome of an admissible strategy. For example, consider the strategy of t_1 that tries to move only if the path

$$(0,1),(\perp,\perp) \xrightarrow{stay} (1,0),(\downarrow,\perp) \xrightarrow{move} (1,0),(\downarrow,\rightarrow) \xrightarrow{block2} (0,1),(\perp,\perp)$$

is taken first[10]. Against a strategy of t_2 that always chooses not to move, the aforementioned loop is the outcome of the game. This is a consequence of the fact that $(1,0),(\downarrow,\perp)$ is a "Help!"-state for player t_1.

Similarly, all states of t_1 are "Help!"-states for player t_2.

As a result, the outcomes of strategies admissible after the first phase of elimination are all paths that never use the dotted transitions of player *env*.

5.2.3 Second iteration

At the second iteration, the value for player *env* do not change. On the other hand, the states where player *env* can force a collision are now losing for players t_1 and t_2, and hence now have value -1.

As a result, player t_2 will never play the transitions that lead to these states (the dashed transitions in Figure 8). However, the "Help!"-states for player t_1 and t_2 are the same, hence the outcomes of strategies admissible after two iterations remains unchanged (except for these local conditions).

5.2.4 Third and fourth iteration

At these iterations the values do not change. Nonetheless, some "Help!"-states have to be removed. For example, state $(1,0),(\downarrow,\perp)$ belonging to t_2 is not a "Help!"-state of t_1 anymore. Hence any admissible strategy must, from $(0,1),(\perp,\perp)$, try to move infinitely often.

Neither values nor sets of admissible strategies change at the fourth iteration, meaning the set $Out(\mathcal{S}^*)$ has been computed.

5.2.5 Analysis of $Out(\mathcal{S}^*)$

In this set, one can see that whenever t_2 is right behind t_1:

- t_2 never tries to move;
- t_1 infinitely tries to move;

As a result, one can see there can be no collision, hence the objective $\psi_{\neg\text{coll}}$ is satisfied. Hence, the model-checking under admissibility problem answers positively for $\psi_{\neg\text{coll}}$, while the ("classical") model-checking problem for the same formula answers negatively.

This also shows for example that the winning coalition problem with $env \in W$ answers negatively: there is no admissible profile that makes player *env* win.

On the other hand, it cannot be assured that t_1 and t_2 will win for their objective: the environment can choose to always block their movement, preventing them from looping infinitely often on the track.

[10]One can show that this strategy is indeed admissible.

6 Weak objectives

In this section, we assume that the winning conditions are weak circuit condition. We will show the following theorem:

Theorem 6.1. *The winning coalition problem for weak circuit conditions is* PSPACE*-complete.*

Proposition 6.2. *For weak circuit conditions, the value of a history only depends on the set* $\mathrm{Occ}(h)$ *of states that have been visited and on* $s = \mathrm{last}(h)$ *the last state of the history.*

Proof. A weak circuit condition can be transformed into a prefix-independent one by remembering which states have already been visited, this is a consequence of the fact that for prefix-independent objectives the value depends only on the last state of the history. ☐

We will thus simply write $\mathrm{Val}_j^n(R, s)$ for $\mathrm{Val}_j^n(h)$ where $R = \mathrm{Occ}(h)$ and $s = \mathrm{last}(h)$. The core of the argument consists in proving that the value for given R and s can be computed in polynomial space.

Lemma 6.3. *For every iteration n, player i, state s, and set of states R, the value of* $\mathrm{Val}_j^n(R, s)$ *can be computed in polynomial space.*

Proof. We define a recursive procedure $\mathsf{Proc}(R, n)$ which computes the value $\mathrm{Val}_i^n(R, s)$ for each player i and state s. This procedure relies on the conditions Θ_i^n (and Ψ_i^n) from Section 4.3, but require them to maintain R as the set of states visited so far. Namely,

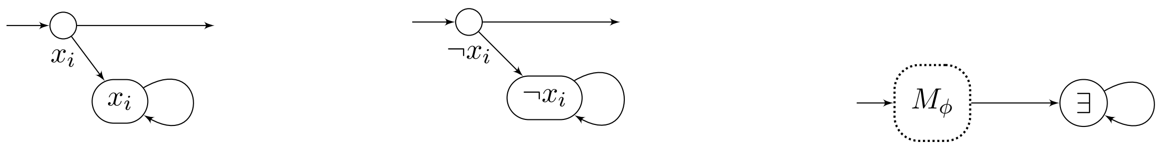
$$\Theta_i^n(R) = \Theta_i^n \cap \mathrm{Occ}^{-1}(R) \qquad \text{and} \qquad \Psi_i^n(R) = \Psi_i^n \cap \mathrm{Occ}^{-1}(R)\,.$$

Given R and n, $\mathsf{Proc}(R, n)$ works as follows.

- Compute the condition $\Theta_i^n(R)$. In that case $\textsc{Win}_i$ is a trivial condition (either *true* if R satisfies $\textsc{Win}_i$ or *false* otherwise) so we can use the polynomial space algorithm of Section 4. Note that the computation of this condition may require computing value of some state for some $n' < n$, this will be done by recursive call to $\mathsf{Proc}(R, n')$.
- Do the same thing for $\Psi_i^n(R)$.
- For each transition $s \to s'$ with $s \in R$ and $s' \notin R$ we compute $\mathrm{Val}_i^n(R', s')$ with $R' = R \cup \{s'\}$. This is done by a recursive call to $\mathsf{Proc}(R', n)$; note that R' strictly contains R.
- If from s, there is a strategy of player i such that any outcome either:
  - stays in R and is winning for $\Theta_i^n(R)$;
  - or the first state reached outside of R has value 1;

  then s has value 1.
- If there is no strategy profile that satisfies $\Phi_i^n(R)$ and no profile whose outcome from s leaves R for $s' \notin R$ with $\mathrm{Val}_i^n(R \cup s', s') \geq 0$, then s has value -1.
- In the other cases s has value 0.

At each recursive call either n decreases or the size of R strictly increases, therefore the height of the stack of recursive call is bounded by $n + |V|$. The computation, ignoring recursive calls, can be done in polynomial space. Thus the global procedure works in polynomial space. ☐



(a) M_{x_i} (b) $M_{\neg x_i}$ (c) $\mathcal{G}_\phi$

Figure 9: Modules for the definition of the game $\mathcal{G}_\phi$. A label y *inside* a state q denotes that $q \in \mathit{Good}_y$; a label y *below* a state q denotes that $q \in V_y$. Note that `Eve` is abbreviated $\exists$.

Proof of Theorem 6.1. Let W and L be the sets of players that should win and lose, respectively. We guess a lasso path ρ such that $\mathrm{Occ}(\rho) = R$ such that for $i \in W$, $R \vDash \varphi_i$ and for $i \in L$, $R \nvDash \varphi_i$, where for $i \in P$, φ_i is the formula over V defined by the circuit for the winning condition of i.

Let n_0 be the iteration after which the values do not change, and assume we have, with $\mathsf{Proc}(R, n_0)$, computed the values for all players and states.

We check that ρ is the outcome of an admissible profile. In order to do that, ρ is divided into $\rho = \rho_1 \cdot \rho_2^\omega$. Note that we can choose $\mathrm{Occ}(\rho_1) = R$, $|\rho_1| \leq |V|^2$ (since a visit to a state of R may require a path of length at most $|V|$), and $|\rho_2| \leq |V|$ (we have also $\mathrm{Occ}(\rho_2) \subseteq R$). We first check that in ρ_1, no player ever lower its own value.

We then check that ρ_2^ω, the looping part of ρ, is part of an admissible profile that stays in R, i.e.:

$$\rho_2^\omega \in \mathit{Out}(\mathcal{S}^{n_0}) \cap \mathrm{Occ}^{-1}(R)$$

where $\mathit{Out}(\mathcal{S}^{n_0})$ is computed as in Section 4.3 with trivial objectives for all players:

$$\mathit{Out}(\mathcal{S}^{n_0}) = \bigcap_{n=0}^{n_0} \bigcap_{i \in P} \mathcal{L}(\mathcal{A}_i^n).$$

In particular, no player can decrease its own value. Note that the definition of the winning condition of these automata requires the values of states at previous iterations, which is provided by calls to $\mathsf{Proc}(R, n)$.

PSPACE-hardness is given by Lemma 3.9. □

In section 3, the winning coalition problem was proved PSPACE-hard even for the special case of safety conditions. The problem is in fact also PSPACE-hard for reachability conditions.

Lemma 6.4. *The winning coalition problem for reachability conditions is* PSPACE*-hard, even for sets of players such that* $|W| = 1$ *and* $L = \emptyset$.

Proof. This is essentially the same proof than for safety, we will therefore only insist on the differences.

- If $\phi = x_i$ we define the module M_ϕ in which player x_i has a choice between winning or letting the game continue.
- If $\phi = \neg x_i$ the construction is similar, with player $\neg x_i$ replacing x_i.
- The other modules are defined in the same way than in Lemma 3.9.

The game $\mathcal{G}_\phi$ is obtained by directing the remaining outgoing transitions to a winning state for `Eve`.

Given a history h of the game, we write $\gamma(h) = \{i \in P \mid \exists k.\ h_k \in Good_i\}$ the set of players who already won in that history. We associate to such a set of player a valuation v_γ such that:

$$v_\gamma(x_i) = \begin{cases} 1 & \text{if } x_i \in \gamma \\ 0 & \text{if } \neg x_i \in \gamma \wedge x_i \notin \gamma \\ \text{undefined} & \text{otherwise} \end{cases}$$

If ℓ is a literal, the strategies of $\mathcal{S}^1_\ell$ are exactly the one such that if $\ell \notin \gamma(h)$ and $\text{last}(h) \in V_\ell$, take the transition to the state in $Good_\ell$.

Now `Eve` should only visit a state controlled by ℓ if it has already been visited in the past. We are in the same situation than in the proof of Lemma 3.9, so the remainder of the proof is exactly the same. $\square$

7 Conclusion

We presented techniques to determine the set of outcomes of iteratively admissible, yielding algorithms to decide several problems on the set of strategies that survive the elimination process.

Future work also include investigating the quantitative setting, where players cannot just win or lose, but obtain a reward, in keeping with the original presentation of this concept for matrix games.

References

[1] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. of POPL'89*, pages 179–190, 1989.

[2] John Nash. Equilibrium points in n-person games. *Proc. of the National Academy of Sciences*, 36(1):48–49, 1950.

[3] Martin J. Osborne and Ariel Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[4] Robert Aumann. Agreeing to disagree. *Annals of Statistics*, 4(6):1236–1239, 1976.

[5] Dietmar Berwanger. Admissibility in infinite games. In Wolfgang Thomas and Pascal Weil, editors, *Proc. of STACS'07*, volume 4393 of *LNCS*, pages 188–199. Springer, February 2007.

[6] Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Strategy logic. *Information and Computation*, 208(6):677–693, 2010.

[7] Marcin Jurdziński. Deciding the winner in parity games is in UP and in co-UP. *Information Processing Letters*, 68(3):119–124, 1998.

[8] Marcin Jurdziński, Mike Paterson, and Uri Zwick. A deterministic subexponential algorithm for solving parity games. In *Proc of SODA'06*, pages 117–123. ACM Press, January 2006.

[9] Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science (FoCS'77)*, pages 46–57. IEEE Computer Society, October 1977.

[10] Aravinda P. Sistla and Edmund M. Clarke. The complexity of propositional linear temporal logics. *Journal of the ACM*, 32(3):733–749, July 1985.

[11] Fabio Mogavero, Aniello Murano, and Moshe Y. Vardi. Reasoning about strategies. In Kamal Lodaya and Meena Mahajan, editors, *Proc. of FSTTCS'10*, volume 8 of *LIPIcs*, pages 133–144. Schloss Dagstuhl - LZI, December 2010.

[12] Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Concurrent games with ordered objectives. In Lars Birkedal, editor, *Proc. of FoSSaCS'12*, volume 7213 of *LNCS*, pages 301–315. Springer, March 2012.

[13] Miroslav Klimoš, Kim Guldstrand Larsen, Filip Stefanak, and Jeppe Thaarup. Nash equilibria in concurrent priced games. In *Proc. of LATA'12*, volume 7183 of *LNCS*. Springer, 2012.

[14] Emmanuel Filiot, Tristan Le Gall, and Jean-François Raskin. Iterated regret minimization in game graphs. In *Proc. of MFCS'10*, pages 342–354, 2010.

[15] Paul Hunter. *Complexity and Infinite Games on Finite Graphs*. PhD thesis, Computer Laboratory, University of Cambridge, 2007.

[16] E.A. Emerson and C.S. Jutla. Tree automata, mu-calculus and determinacy. In *Proc. of FCS'91*, pages 368–377. IEEE, 1991.

[17] Gilles Brassard. A note on the complexity of cryptography. *IEEE Transactions on Information Theory*, 25(2):232–233, March 1979.

[18] A. Prasad Sistla, Moshe Y. Vardi, and Pierre Wolper. The complementation problem for Büchi automata with appplications to temporal logic. *Theorethical Computer Science*, 49:217–237, 1987.

[19] Romain Brenguier. *Équilibres de Nash dans les Jeux Concurrents – Application aux Jeux Temporisés*. Thèse de doctorat, Laboratoire Spécification et Vérification, ENS Cachan, France, November 2012.