The complexity of Rational Synthesis
- Full version -

Rodica Condurache[1,3], Emmanuel Filiot[1], Raffaella Gentilini[2], and Jean-François Raskin[1]

[1] Université Libre de Bruxellse, Computer Science Department, Brussels
[2] University of Perugia, Dept. of Mathematics and Computer Science, Perugia
[3] Université Paris Est, LACL(EA 4219), UPEC, 94010 Créteil Cedex

- 17 Feb. 2016 -

Abstract. In this paper, we study the computational complexity of the cooperative and non-cooperative rational synthesis problems, as introduced by Kupferman, Vardi and co-authors in recent papers for general LTL objectives. We investigate these problems on multiplayer turn-based games played on graphs, and provide complexity results for the classical omega-regular objectives. Most of these complexity results are tight and shed light on how to solve those problems optimally.

1 Introduction

In this paper, we study the computational complexity of the rational synthesis problem as introduced in [15, 19]. Rational synthesis uses $k + 1$ player non-zero sum games to formalize the problem of synthesising a system (modeled by Player 0) that is executed in an environment made of several components (modeled by Players $1, ..., k$). The behaviour of the components composing the environment is assumed to be rational, and not necessarily fully antagonistic as in the classical two player zero-sum setting, see e.g. [23]. Rationality of the environment is modelled by assuming that the components behave according to a Nash equilibrium. Rational synthesis has been introduced in [15, 19] in two different settings.

In the first setting, called *cooperative rational synthesis* [15], the environment cooperates with the system in the sense that its components agree to play a Nash equilibrium that is winning for Player 0 (if it exists). In other words, in the cooperative setting, one assumes that once a Nash equilibrium winning for Player 0 is proposed, all the players will adhere to the suggested strategies.

In the second setting, called *non-cooperative rational synthesis* [19], the components of the environment may follow any strategy, providing it is a Nash equilibrium. In this setting, one has to output (if it exists) a strategy σ_0 for the system which has to be winning against all the possible strategy profiles that include σ_0 for Player 0 and which are Nash equilibria.

The main contribution of the original papers is to propose and to motivate the definitions above. The only computational complexity results given in those papers are as follows: the cooperative and non-cooperative rational synthesis problems are 2EXPTIME-C for specifications expressed in linear temporal logic (LTL), thus matching exactly the complexity of classical zero-sum two-player LTL synthesis [21]. The upper bound is obtained by reductions to the satisfiability problem of formulas in Strategy Logic [20] (SL). The reduction to SL and the use of LTL specifications does not allow one to understand finely the computational complexity aspects of solving the underlying n player non-zero sum games.

Contributions To better understand the computational complexity of the rational synthesis problems and how to manipulate their underlying games algorithmically, we consider variants of those problems for games played on turn-based graph structures for reachability, safety, Büchi, coBüchi, parity, Rabin, Streett and Muller objectives. We also study the computational complexity of solving those games when the number of players is fixed. This parameterised analysis makes sense as the number of components forming the environment may be limited in practical applications. The results we obtain are summarized in Table 1.

	Cooperative		Non-Cooperative	
	Unfixed k	Fixed k	Unfixed k	Fixed k
Safety	NP-c	PTIME-c	PSPACE-c	PTIME-c
Reachability	NP-c	PTIME-c	PSPACE-c	PTIME-c
Büchi	PTIME-c[25]	PTIME-c[25]	PSPACE-c	PTIME-c
co-Büchi	NP-c[25]	PTIME-c	PSPACE-c	PTIME-c
Parity	NP-c[25]	$UP \cap co-UP$, parity-h	EXPTIME, PSPACE-h	PSPACE, NP-h, coNP-h
Streett	NP-c [25]	NP [25], NP-hard	EXPTIME,PSPACE-h	PSPACE-c
Rabin	P^{NP}, NP-h, coNP-h	P^{NP}, coNP-h	EXPTIME, PSPACE-h	PSPACE-c
Muller	PSPACE-c	PSPACE-c	EXPTIME, PSPACE-h	PSPACE-c
LTL	2EXPTIME-C[15]	2EXPTIME-C[15]	2EXPTIME-C[19]	2EXPTIME-C[19]

Table 1: Complexity of rational synthesis for k players.

On the positive side, our results show that for a *fixed* number of players, for objectives that admit a polynomial time solution in the two-player zero-sum case (reachability, safety, Büchi and coBüchi), cooperative and non-cooperative rational synthesis can be solved in PTIME. On the negative side, for rich omega regular objectives defined by parity, Rabin, or Streett objective, the complexity increases. First, games with parity objectives cannot be solved in polynomial time unless PTIME equals NP while it is conjectured that this result does not hold for two-player zero sum parity games. Second, games with Rabin or Streett objectives are PSPACE-C for the non-cooperative setting while they have solution in nondeterministic polynomial time for their zero-sum two player versions. When the number of players is not fixed, the complexity is usually substantially higher than for the two-player zero-sum case. For example, non-cooperative rational synthesis is PSPACE-H for all objectives, so even for safety objectives.

Cooperative rational synthesis is a particular case of the more general problem of checking the existence of a constrained Nash equilibrium in a multiplayer game, where the strategy of Player 0 is required to be winning. The complexity of constrained Nash equilibria has been studied by Ummels in [25] for some classes of objectives, based on a characterisation of Nash equilibria by means of LTL formulas to be checked on the game arena. This directly gives us upper-bounds for cooperative synthesis and Büchi, coBüchi, parity and Streett objectives. For the other objectives, we extend this characterization. Solutions to the *non-cooperative* case are much more involved and are based on a fine tuned application of tree automata techniques. This is a central contribution of our paper. In particular, our tree automata have exponential size but we show how to test their emptiness in PSPACE to obtain optimal algorithms for Streett, Rabin and Muller objectives and fixed number of players.

The tree automata that we construct not only allow us to test the existence of solution to the non-cooperative rational synthesis problem but also to symbolically represent all the strategies for the system that are solutions. This set is thus regular and can be manipulated with automata-based techniques. Also, it should be clear that those techniques are amenable to symbolic implementations when the game structure is given with binary decision diagrams. This is important as it shows that our techniques pave the way to implementations that have proven useful and efficient by the computer aided verification community, and implemented in tools like nuSMV [13] for example. To obtain lower-bounds, we had to design several original and intricate reductions that explain cleanly why some of those problems are intractable.

Related works Non-zero games for synthesis are gaining attention recently, see e.g. [5] for a survey of recent results. *Secure equilibria* were introduced for two players in [10] and their potential for synthesis was demonstrated in [9]. Secure equilibria are refinement of Nash equilibria [24]. *Doomsday equilibria* extend secure equilibria to the n player case, and their complexity complexity is studied in [8]. Subgame perfect equilibria, that also refines Nash equilibra, were first studied in [24, 25]. To model rationality of players, the notion of *admissible strategy* is used in [4, 14] instead of the notion of Nash equilibria, and the computational complexity of related decision problems is studied in [7]. Synthesis rules for reactive systems based on admissibility are studied in [6]. All those works consider games played on a game structure with classical omega-regular objectives and provide tight complexity results for almost all the relevant synthesis problems. This is not the case for *cooperative* and *non-cooperative rational synthesis* for which only the complexity for specifications given in LTL was known [15, 19]. This paper provides algorithms and computational complexity results for cooperative and non-cooperative rational synthesis that allows us to better understand the complexity picture of non-zero sum games played on graphs with omega-regular objectives.

Structure of the paper In Sect. 2, we recall the definition of the cooperative and non-cooperative synthesis problem as introduced in [15, 19], together with the game structure variant and objectives that we study here. Sect. 3 provides lower and upper complexity bounds for the cooperative rational synthesis problem. Sect. 4 provides results for the non-cooperative variant. Sect. 5 summarizes complexity results when the number of players is fixed.

2 Preliminaries

2.1 Trees and Tree Automata

Let Λ be a set of directions and Σ be an alphabet. A Σ-labeled Λ-tree is a mapping $t : \Lambda^* \to \Sigma$. Its set of nodes is Λ^* and the empty word ϵ is the root. For every $x \in \Lambda^*$ and $c \in \Lambda$, the node $xc \in \Lambda^*$ is called the *successor* of x. A *branch* is an infinite sequence of directions $\pi \in \Lambda^\omega$. Given a tree t and a node x, the *subtree* of t at node x is a mapping $t^x : \Lambda^* \to \Sigma$ such that $t^x(y) = t(xy)$ for all $y \in \Lambda^*$.

Tree automata A finite nondeterministic tree automaton over Σ-labeled Λ-trees is a tuple $\mathcal{T} = (Q, Q_0, \delta, \alpha)$ where Q is the set of states, Q_0 is the set of initial states, $\alpha \subseteq Q^\omega$ is the accepting condition and δ is the transition relation of the form $\delta : Q \times \Sigma \to 2^{\Lambda \to Q}$, i.e., it maps any pair of states and labels to a set of mappings from directions to states (states sent the children of the current node). A *run* of $\mathcal{T}$ on a tree t is Q-labeled D-tree $r : D^* \to Q$ such that $r(\epsilon) \in Q_0$ and for all $h \in D^*$, all $d \in D$, the mapping $d \in D \mapsto r(hd) \in Q$ is in $\delta(r(h), t(h))$. The image of a branch $\pi = \lambda_1\lambda_2 \cdots \in \Lambda^\omega$ by r is the word in Q^ω defined by $r(\epsilon)r(\lambda_1)r(\lambda_1\lambda_2)\ldots$. With respect to the accepting condition $\alpha \subseteq Q^\omega$, r is accepting if all its branches are in α, and the *language* of $\mathcal{T}$ is the set $\mathcal{L}_\alpha(\mathcal{T})$ of trees for which there exists an accepting run.

We say that a tree automaton $\mathcal{T}$ is *deterministic* if the transition relation is of the form $\delta : Q \times \Sigma \to D \to Q$, i.e., it maps any pair of states and labels to one mappings from directions to states. In this case, we equivalently say that the transition relation is of the form $\delta : Q \times \Sigma \times D \to Q$. Also, a tree automaton is a *safety* automaton if the winning condition consists on all the sequences in Q^ω that avoid a certain set S of states, i.e., $\alpha = Q^\omega \backslash Q^* S Q^\omega$

2.2 Multiplayer Games and Rational Synthesis

Multiplayer Games Let $k \in \mathbb{N}$. A *multiplayer arena* ($k+1$-players arena) is a tuple $\mathcal{A} = \langle \Omega, V, (V_i)_{i\in\Omega}, E, v_0\rangle$, where $\Omega = \{0, 1, ..., k\}$ is a finite set of players, (V, E) is a finite directed graph whose vertices are called *states*, $v_0 \in V$ is the initial state and $(V_i)_{i\in\Omega}$ is a partition of V where V_i is the set of states controlled by Player $i \in \Omega$. A *play* in $\mathcal{A}$ starts in the initial state v_0 and proceeds in rounds. At each round, the player controlling the current state chooses the next position according to E. Wlog we assume that each vertex has a successor by E and that player's rounds are ordered according to their index[4], i.e.

[4] Otherwise we just add a polynomial number of extra intermediate states and the winning objectives considered in this paper can be modified accordingly.

$E \subseteq \bigcup_{i \in \Omega} V_i \times V_{i+1 \bmod k}$. Formally, a play $\pi = u_0 u_1 \ldots$ is an infinite path in V^ω such that $u_0 = v_0$ and $(v_i, v_{i+1}) \in E$ for each $i \geqslant 0$. The prefix (or history) of π up to v_n is written $\pi[{:}n]$ and its last state $\pi(n)$. We denote by $\sqsubset$ the prefix relation. We let $\mathsf{Plays}(\mathcal{A})$ stand for the set of plays, and $\mathsf{Prefs}(\mathcal{A})$ for its closure under $\sqsubset$. Finally, for $\pi \in V^\omega$, we write $\inf(\pi)$ for the set of states occurring infinitely many times in π and $\pi\restriction_{V_i}$ for the restriction of a play only to the states of Player i.

A *strategy* of Player $i \in \Omega$ in $\mathcal{A}$ is a total function $\sigma_i : V^* V_i \mapsto V$ s.t. for all $x \in V^*$, for all $v \in V_i$, $(v, \sigma_i(xv)) \in E$. Note that as rounds are ordered, σ_i has type $V^* V_i \mapsto V_{i+1 \bmod k}$. A play π is *consistent* with σ_i if $\pi(n+1) = \sigma_i(\pi[{:}n])$ for all $n \geqslant 0$ s.t. $\pi(n) \in V_i$. The outcome of σ_i is the set of plays $out(\sigma_i) \subseteq \mathsf{Plays}(\mathcal{A})$ that are consistent with σ_i. Given $h \in V^*$, we define $\sigma_i|_h$ as $\sigma_i|_h(h') = \sigma_i(hh')$ for all $h' \in V^* V_i$. A *winning objective* (or just objective) is a set $\mathcal{O} \subseteq V^\omega$. A Player i's strategy σ_i is *winning* for $\mathcal{O}$ if $out(\sigma_i) \subseteq \mathcal{O}$. In this paper, we consider the following classical ω-regular objectives [**?**]:

- *Safety*: Given the set $S \subseteq V$ called the set of safe states, $\mathsf{Safe}(S) = \{\pi \in V^\omega \mid \forall n \geqslant 0 : \pi(n) \in S\}$.
- *Reachability*: Given the set $T \subseteq V$ called the set of target states, $\mathsf{Reach}(T) = \{\pi \in V^\omega \mid \exists n \geqslant 0 : \pi(n) \in T\} = \overline{\mathsf{Safe}(\bar{T})}$.
- *Büchi*: $\mathsf{Buchi}(F)$ is the set of sequences in which some state in $F \subseteq V$ occurs infinitely many times, i.e. $\mathsf{Buchi}(F) = \{\pi \in V^\omega \mid \inf(\pi) \cap F \neq \varnothing\}$.
- *co-Bučhi*: $\mathsf{coBuchi}(F)$ is the set of sequences in which all states of $F \subseteq V$ occurs finitely many times, i.e. $\mathsf{coBuchi}(F) = \{\pi \in V^\omega \mid \inf(\pi) \cap F = \varnothing\} = \overline{\mathsf{Buchi}(F)}$.
- *Streett*: Given a set $\Psi \subseteq 2^V \times 2^V$, the Streett condition for Ψ is the set of infinite sequences $\pi \in V^\omega$ such that for all pairs $(L, R) \in \Psi$ such that $\pi(k) \in L$ for infinitely many $k \in \omega$, it is the case that $\pi(k) \in R$ for infinitely many $k \in \omega$, i.e. $\mathsf{Streett}(\Psi) = \{\pi \in V^\omega \mid \forall (L, R) \in \Psi, (\inf(\pi) \cap L \neq \varnothing) \implies (\inf(\pi) \cap R \neq \varnothing)\}$.
- *Rabin*: Given a set $\Psi \subseteq 2^V \times 2^V$, the Rabin condition for Ψ is the set of infinite sequences $\pi \in V^\omega$ such that there is a pair $(L, R) \in \Psi$ such that $\pi(k) \in L$ for infinitely many $k \in \omega$ and $\pi(k) \in R$ for finitely many $k \in \omega$, i.e. $\mathsf{Rabin}(\Psi) = \{\pi \in V^\omega \mid \forall (L, R) \in \Psi, (\inf(\pi) \cap L \neq \varnothing) \text{ and } (\inf(\pi) \cap R = \varnothing)\} = \overline{\mathsf{Streett}(\psi)}$.
- *Parity*: Given a function $p : V \mapsto \omega$, called a priority function, $\mathsf{Parity}(p)$ is the set of infinite plays $\pi \in V^\omega$ such that the least number occurring infinitely often in $p(\pi)$ is even, i.e. $\mathsf{Parity}(p) = \{\pi \in V^\omega \mid \min\{p(\pi(n)) \mid n \geqslant 0\} \text{ is even}\}$.
- *Muller*: Given a Boolean formula μ over the set of states V, the Muller condition for μ is the set of infinite sequences $\pi \in V^\omega$ such that the set of states appearing infinitely often in π satisfies μ, i.e., $\mathsf{Muller}(\mu) = \{\pi \in V^\omega \mid inf(\pi) \models \mu\}$

Note that the Büchi and co-Büchi conditions are Parity conditions with two priorities and a Büchi (resp. co-Büchi) condition F is also a Streett condition (V, F) (resp. $(F, \varnothing)$) or a Parity condition $p : V \rightarrow \{0, 1\}$ with $p(v) = 0$ if $v \in F$ and $p(v) = 1$ otherwise (resp. $p : V \rightarrow \{1, 2\}$ with $p(v) = 1$ if $v \in F$ and $p(v) = 2$ otherwise).

A *multiplayer game* is a pair $\mathcal{G} = \langle \mathcal{A}, (\mathcal{O}_i)_{i \in \Omega} \rangle$, where $(\mathcal{O}_i)_{i \in \Omega}$ is the tuple of objectives for each Player $i \in \Omega$. The notations Plays and Prefs carries over naturally to $\mathcal{G}$ by considering its underlying arena. For $X \in \{\mathsf{Reach}, \mathsf{Safe}, \mathsf{Buchi}, \mathsf{coBuchi}, \mathsf{Street}, \mathsf{Rabin}, \mathsf{Parity}, \mathsf{Muller}\}$, a multiplayer X-game is a multiplayer game where each player has an X-objective. For a strategy σ_i, $i \in \Omega$, we denote by $\mathcal{G}[\sigma_i]$ the (possible infinite) game obtained from $\mathcal{G}$ in which Player i plays the strategy σ_i.

Nash equilibria A (pure) *strategy profile* $\bar{\sigma}$ in $\mathcal{G} = \langle \mathcal{A}, (\mathcal{O}_i)_{i \in \Omega}$ is a tuple $\bar{\sigma} = (\sigma_i)_{i \in \Omega}$, where σ_i is a strategy for player $i \in \Omega$. The *outcome* of a strategy profile $\bar{\sigma}$, written $out(\bar{\sigma})$ is the play consistent with each $\sigma_i, i \in \Omega$ (it always exists and is unique). Given a strategy profile $\bar{\sigma}$ and a strategy τ for $i \in \Omega$, we write $(\bar{\sigma}_{-i}, \tau)$ for the strategy profile obtained by replacing σ_i with τ in $\bar{\sigma}$. Given winning objectives $(\mathcal{O}_i)_{i \in \Omega}$ for each player, the *payoff* of a strategy profile $\bar{\sigma}$ is the vector $pay(\bar{\sigma}) \in \{0,1\}^n$ defined by $pay(\bar{\sigma})[i] = 1$ if and only if $out(\bar{\sigma}) \in \mathcal{O}_i$. We write $pay_i(\bar{\sigma})$ for Player i's payoff $pay(\bar{\sigma})$. Payoffs are compared by the pairwise natural order on their bits, denoted by $\leqslant$, i.e., $pay(\bar{\sigma}) \leqslant pay(\bar{\beta})$ if $pay_i(\bar{\sigma}) \leqslant pay_i(\bar{\beta})$ for all $i \in \Omega$.

A strategy profile $\bar{\sigma} = (\sigma_i)_{i \in \Omega}$ is called a *Nash equilibrium* of the multiplayer game $\mathcal{G}$ if $pay(\bar{\sigma}_{-i}, \tau) \leqslant pay(\bar{\sigma})$ for all players $i \in \Omega$ and all strategies τ of i. Thus, intuitively, in a Nash equilibrium no player can improve his payoff by (unilaterally) switching to a different strategy. We say that a strategy profile $\bar{\sigma} = (\sigma_i)_{i \in \Omega}$ is a 0-*fixed* Nash equilibrium if $pay(\bar{\sigma}_{-i}, \tau) \leqslant pay(\bar{\sigma})$ for all players $i \in \Omega \backslash \{0\}$ and all strategies τ of i. In other words, it is a Nash equilibrium in which player 0 cannot change his strategy. Observe that any Nash equilibrium $(\sigma_i)_{i \in \Omega}$ is a 0-fixed equilibrium, but the converse may not hold.

Let Σ_i be the set of all the possible strategies of i and $\mathcal{O}$ a winning objective for Player i. We denote by W_i the set of *winning states* (also called *winning set*) for Player i and objective $\mathcal{O}$, i.e. the set of states v such that if the game initially starts in state v, then Player i has a strategy to win his objective.

2.3 Rational Synthesis

Rational synthesis aims at finding a winning strategy for the system (Player 0) against an environment composed by several other systems (Players $1, \ldots, k$) that have a rational behavior. Here rationality is modeled by assuming that the players behave according to a Nash equilibrium. Rational synthesis has been introduced in [15, 19] in two different settings. In the first setting, called *cooperative rational synthesis* [15], the environment cooperates with the system in the sense that its components agree to play a Nash equilibrium that is winning for Player 0 (if it exists). In other words, in the cooperative setting, one assumes that once a Nash equilibrium winning for Player 0 is proposed, all the agents will adhere to the suggested strategies.

In the second setting, called *non-cooperative rational synthesis* [19], the components of the environment may follow any strategy profile, providing it is a Nash equilibrium. In this setting, one has to output (if it exists) a strategy σ_0 for the system which has to be winning against any possible strategy profile that includes σ_0 for Player 0 and which is a Nash equilibrium. Formally,

Definition 1 (Rational Synthesis Problems). *The cooperative and non-cooperative rational synthesis problems ask, given as input an $(n+1)$-player game $\mathcal{G}$ with winning objectives $(\mathcal{O}_i)_{i\in\Omega}$, the following questions according to the two settings:*

cooperative: *Is there a 0-fixed Nash equilibrium $\bar{\sigma}$ such that $pay_0(\bar{\sigma}) = 1$?*

non-cooperative: *Is there a strategy σ_0 for Player 0 such that for any 0-fixed Nash equilibrium $\bar{\sigma} = \langle \sigma_0, \dots, \sigma_k \rangle$, we have $pay_0(\bar{\sigma}) = 1$?*

Example 1. As an example, consider the two-player game arena of Figure 1 in which Player 0 owns round states and Player 1 square states, with the reachability objectives given by the set $R_0 = \{2\}$ and $R_1 = \{3\}$. Consider the Player 0's strategies σ_0 which consists in looping forever in state 2, and σ_0' which eventually goes to state 3.

Let Player 1 cooperate by playing the strategy σ_1 that goes to state 2 (making Player 0 win). Both strategy profiles $\langle \sigma_0, \sigma_1 \rangle$ and $\langle \sigma_0', \sigma_1 \rangle$ are solutions to the cooperative setting: for the first strategy profile Player 1 loses but cannot get better payoff by deviating, and for the second one Player 1 wins. Strategy σ_0 is not a solution to the non-cooperative setting, because Player 1 could stay forever in state 1 (according to a strategy σ_1'): The profile $\langle \sigma_0, \sigma_1' \rangle$ is a 0-fixed NE because even by deviating and going to state 2 Player 1 would still lose, and it is losing for Player 0. However, σ_0' is a solution to the non-cooperative setting: The only 0-fixed NE in that case are when Player 1 eventually move to state 2, making him and Player 0 win.

We may refer to Player 0 as the *system* and to the other players as the *environment*. It is shown in [15] and [19] that both cooperative and non-cooperative rational synthesis problems are 2EXPTIME-COMPLETE when the winning objectives are defined by LTL formulas. In general, the synthesis problem also asks to *synthesise* (i.e.*construct*) such solution if it exists. The existence problem is sometimes referred to as the *realisability problem*. All our algorithms also solve the synthesis problem.

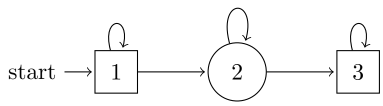


Fig. 1: Example for rational synthesis

3 General characterization of 0-fixed Nash Equilibria

In this section, we give a general method to solve the rational cooperative and non-cooperative synthesis problems. It is based on effective characterizations of the existence of 0-fixed Nash equilibria in multiplayer games with safety, reachability, or tail objectives, through the existence of a path of the arena satisfying certain properties. These properties will be expressed by LTL formulas.

Linear-Time Temporal Logic on Game Arenas We do not recall here the semantics of LTL (we refer the reader to [3] for instance, for an overview of LTL and its semantics), but we rather make explicit in which context we will use it. In particular, we will use LTL to express properties of infinite paths in a game arena $\mathcal{A} = \langle \Omega, V, (V_i)_{i\in\Omega}, E, v_0\rangle$. In addition to Boolean connectives, we write $\mathcal{U}$, $\Box$, $\Diamond$ to denote the until, always and eventually temporal operators. Given a state $s \in V$, we view s has an atomic proposition, true in s, and false otherwise. Given $S \subseteq V$, we may freely use S in an LTL formula, where it stands for the formula $\bigvee_{s\in S} s$. Therefore, we may write, for instance, $\Box\neg S$, to denote the set of infinite paths in $(V\backslash S)^\omega$. We denote by LTL($\mathcal{A}$) the set of LTL formulas over the set of atomic propositions V, and for a game $\mathcal{G}$ whose underlying arena is $\mathcal{A}$, LTL($\mathcal{G}$) stands for LTL($\mathcal{A}$). A set $\mathcal{O} \subseteq V^\omega$ is *definable* in LTL($\mathcal{A}$) if there exists an LTL($\mathcal{A}$) formula ϕ such that for all $\pi \in V^\omega$, $\pi \models \phi$ iff $\pi \in \mathcal{O}$. In [**?**] similar formulas were given for similar tale objectives (see Corollary[26]).

LTL characterization of 0-fixed Nash equilibria For all the winning objectives considered in this paper, we characterize the existence of a 0-fixed Nash equilibria in a game by the existence of a path satisfying some LTL formula, that depends on the winning objectives. For tail objectives, we give a generic way of constructing such an LTL formula. An objective $\mathcal{O} \subseteq V^\omega$ is *tail* if for all $\pi_1 \in V^*$ and $\pi_2 \in V^\omega$, $\pi_1\pi_2 \in \mathcal{O}$ iff $\pi_2 \in \mathcal{O}$. In other words, a path is winning iff one of its (infinite) suffix is. Büchi, coBüchi, parity, Streett, Rabin and Muller objectives are all tail.

Let $\mathcal{G} = \langle \mathcal{A}, (\mathcal{O}_i)_{0\leqslant i\leqslant k}\rangle$ be a $k+1$-player game. Let $(W_i)_{0\leqslant i\leqslant k}$ be the winning sets for the objectives $(\mathcal{O}_i)_{0\leqslant i\leqslant k}$, and V be the set of states of $\mathcal{A}$. We define an LTL[$\mathcal{G}$]-formula $\phi_{0\text{Nash}}$ that characterizes the existence of a 0-fixed Nash equilibrium in $\mathcal{G}$. It is defined as follows:

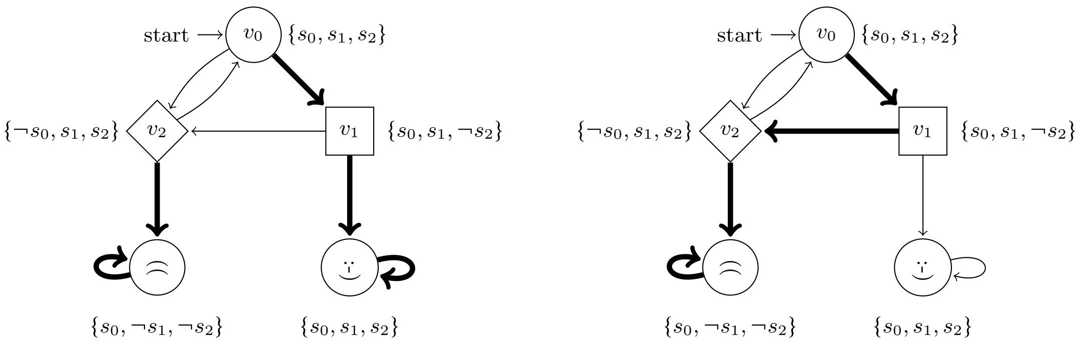
$$\phi^{\mathcal{G}}_{0\text{Nash}} = \begin{cases} \bigwedge_{i=1}^k ((\neg W_i^{\mathcal{G}}\ \mathcal{U}\ \neg S_i) \vee \Box S_i) & \text{if } \mathcal{O}_i \text{ are safety objectives of the form} \\ & \mathcal{O}_i = \mathsf{Safe}(S_i) \text{ for some } S_i \subseteq V \\ \bigwedge_{i=1}^k \neg\varphi_i \rightarrow \Box\neg W_i^{\mathcal{G}} & \text{if } \mathcal{O}_i \text{ are either all reachability or all tail} \\ & \text{objectives definable by an LTL}[\mathcal{G}] \text{ formula } \varphi_i \end{cases}$$

The formula $\phi_{0\text{Nash}}$ characterises 0-fixed Nash equilibria in the following sense:

Lemma 1 (Characterization of 0-fixed Nash Equilibria). *Let $\mathcal{G}$ be a multiplayer game with either all safety, all reachability, or all tail objectives, definable in LTL[$\mathcal{G}$]. Then, the following hold:*

1. *For all $\pi \in \mathsf{Plays}(\mathcal{G})$, if $\pi \models \phi^{\mathcal{G}}_{0Nash}$, then there exists a 0-fixed Nash equilibrium $\bar{\sigma}$ in $\mathcal{G}$ such that $out(\bar{\sigma}) = \pi$,*
2. *For all 0-fixed Nash equilibrium $\bar{\sigma}$ in $\mathcal{G}$, $out(\bar{\sigma}) \models \phi^{\mathcal{G}}_{0Nash}$.*

Before proceeding to the proof of Lemma 1, we illustrate this characterization on an example of safety game.



(a) Path consistent with 0-fixed NE

(b) π not consistent with a 0-fixed NE

Fig. 2: A safety game

Example 2. Consider the game in Fig. 2, played by three agents that control round (Player 0), square (Player 1) and diamond (Player 2) states respectively. The objective the players is to stay in their safe regions, denoted by the labels s_0, s_1, s_2 (e.g. state v_1 is safe for Player 0 and 1 and unsafe for Player 2). Then, the winning sets for the three players are $W_0 = \{☺, ☹\}$, $W_1 = \{☺, v_1\}$ and $W_2 = \{☺\}$.

First, consider the path $\pi = v_0 v_1 (☺)^\omega$ that satisfies $\bigwedge_{i=1}^{2}((\neg W_i \ \mathcal{U} \ \neg S_i) \vee \Box S_i)$. We can build a 0-fixed Nash equilibrium $\bar{\sigma}$ that is represented in Fig. 2a with bold arrows which has as outcome the path π. On the other hand, consider the path $\pi' = v_0 v_1 v_2 (☹)^\omega$, in bold in Fig.2b. It does not satisfy $\bigwedge_{i=1}^{2}((\neg W_i \ \mathcal{U} \ \neg S_i) \vee \Box S_i)$. Suppose that π is the outcome of a strategy profile $\bar{\sigma}$, then $\bar{\sigma}$ is not a 0-fixed Nash equilibrium. Indeed, Player 1 reaches for the first time an unsafe state (☹), after visiting v_1, which is in his winning region. Therefore, Player 1 would better deviate and go to state ☺.

Proof (Proof of Lemma 1). Statement 1*, safety objectives* The strategy profile $\bar{\sigma}$ is intuitively defined as follows: as long as the current history is a prefix of π, then the players play according to π. If at some point, some player, say Player i, decides to deviate from π, ending up in a state s, then if $s \notin W_i$, all the players but Player i punish him by playing a strategy that will make him lose, otherwise, they play any strategy. Let us give some arguments to justify that it is a 0-fixed equilibrium. The outcome of $\bar{\sigma}$ is π, and if a player wins along π, then he has no incentive to deviate. If some player, say Player i, loses along π, then suppose that he decides eventually to deviate from π: either he has already lost before deviating and therefore his deviation is useless, or he deviates to a state v before visiting an unsafe state for the first time, but in that case, since $\pi \models \neg W_i \ \mathcal{U} \ \neg S_i$, we have $v \notin W_i$ (otherwise the previous state would be winning), and all the other players retaliate, making his deviation useless, again.

Let us define $\bar{\sigma}$ formally. For a state $s \notin W_i$, we let $(\mathsf{ret}_j^{s,i})_{j \neq i}$ a retaliating profile, i.e., a profile a strategies for the players $j \neq i$ that make Player i lose. We also pick a profile of strategies $(\beta_0, \ldots, \beta_k)$. Then, we define σ_j as follows, for $x \in V^* V_j$:

$$\bar{\sigma}_j(x) = \begin{cases} \pi[l+1] & \text{if } x = v_0 v_1 ... v_l \text{ is a prefix of } \pi \\ \mathsf{ret}_j^{s,i}(s x_2) & \text{if condition (1) is satisfied} \\ \beta_j(x) & \text{otherwise} \end{cases}$$

where condition (1) requires that x can be decomposed into $x = x_1 s x_2$ such that $x_1 \in V^* V_i$, x_1 is a prefix of π, $s \notin W_i$, $x_1 s$ is not a prefix of π (meaning that Player i has deviated to a losing state).

Clearly, we have $out(\bar{\sigma}) = \pi$. We claim that $\bar{\sigma}$ is a 0-fixed Nash equilibrium. Towards a contradiction, we suppose that some player i ($1 \leqslant i \leqslant k$) loses, and can win by playing another strategy σ_i' (when the other players stick to their strategies $\bar{\sigma}_{-i}$). Then necessarily, $out(\bar{\sigma}_{-i}, \sigma_i')$ deviates from π after some prefix x_1 of π such that $x_1 \in V^\omega V_i$. Let $s \in V$ and $y \in V^\omega$ such that $out(\bar{\sigma}_{-i}, \sigma_i') = x_1 s y$. Then, if $s \notin W_i$, the other players retaliate making $out(\bar{\sigma}_{-i}, \sigma_i')$ losing for Player i, which is a contradiction. Therefore $s \in W_i$. Let s' be the last state of x_1. Then $s' \in W_i$ since $s' \in V_i$ and it has a successor in W_i (s). Now, we consider two cases: (i) suppose that some state of x_1 is unsafe for Player i, then it contradicts the fact that $out(\bar{\sigma}_{-i}, \sigma_i')$ is winning for Player i; (ii) if $x_1 \in (S_i)^*$, then since π is losing for Player i, there is an unsafe state for Player i that occurs after the prefix x_1, contradicting the fact that $\pi \models \neg W_i \ \mathcal{U} \ \neg S_i$, since the last state of x_1 is in W_i. In all cases, we have found a contradiction, showing that such a strategy σ_i' cannot exist.

Statement (2)*, safety objectives* Assume that some player, say i for $1 \leqslant i \leqslant k$ does not win, i.e. $out(\bar{\sigma}) \models \Diamond \neg S_i$. Towards a contradiction, assume that $out(\bar{\sigma}) \not\models \neg W_i \mathcal{U} \neg S_i$. Consider the first occurrence j of state satisfying $\neg S_i$ in $out(\bar{\sigma})$, i.e. $j = argmin\{j \mid out(\bar{\sigma})[j] \notin S_i\}$ (it exists since $out(\bar{\sigma}) \models \Diamond \neg S_i$). Clearly, there exists a position $0 \leqslant t < j$ such that $out(\sigma)[t] \in W_i$ (otherwise $out(\bar{\sigma})$ would satisfy $\neg W_i \mathcal{U} \neg S_i$). At that position, Player i could have deviated and apply a winning strategy, thus getting a strictly better payoff, contradicting the fact that $\bar{\sigma}$ is a 0-fixed Nash equilibrium.

Statements (1) *and* (2)*, reachability and tail objectives* The proofs of these two statements are very similar to that of safety objectives. The only difference here is that the objectives are either all reachability or all tail, and therefore one has to make sure that on π, the players that lose never visit their winning region, because if it is so, they would have an incentive to deviate: indeed, the satisfaction of their winning objective would be independent from the prefix up to a visit to their winning region. For statement (1), the profile of strategies $\bar{\sigma}$ is: follow the path π as long as the play stays in π, and the first time the play deviates (say Player i deviates from π), then if π is losing for Player i, then apply from that point on a retaliating strategy (as a coalition of all the players $j \neq i$), otherwise apply any strategy. If π is not winning for Player i, the retaliating strategy exists by definition of $\phi_{0\text{Nash}}$, since the first position after the deviation would not be in W_i.

Conversely, any 0-fixed NE $\bar{\sigma}$ satisfies $\phi_{0\text{Nash}}$. Indeed, if it is not the case, then there exists some player that satisfies $\neg\phi_i$ and $\Diamond W_i$. When reaching its winning region, this player would better apply a winning strategy and strictly increase his payoff. ∎

As a consequence of Lemma 1, we also get a characterization of Nash equilibria in multiplayer games, for safety, reachability and tail objectives. The case of tail objectives was already covered in [25]. We give this result since it might be of independent interest for the reader.

Let $\mathcal{G} = \langle \mathcal{A}, (\mathcal{O}_i)_{0 \leqslant i \leqslant k} \rangle$ be a $k+1$-player game. Let $(W_i)_{0 \leqslant i \leqslant k}$ be the winning sets for the objectives $(\mathcal{O}_i)_{0 \leqslant i \leqslant k}$, and V be the set of states of $\mathcal{A}$. We define an LTL[$\mathcal{G}$]-formula ϕ_{Nash} as follows:

$$\phi_{\text{Nash}} = \begin{cases} \bigwedge_{i=0}^{k}((\neg W_i \,\mathcal{U}\, \neg S_i) \vee \Box S_i) & \text{if } \mathcal{O}_i \text{ are safety objectives of the form} \\ & \mathcal{O}_i = \mathsf{Safe}(S_i) \text{ for some } S_i \subseteq V \\ \bigwedge_{i=0}^{k} \neg\varphi_i \rightarrow \Box\neg W_i & \text{if } \mathcal{O}_i \text{ are either all reachability or all tail} \\ & \text{objectives definable by an LTL}[\mathcal{G}] \text{ formula } \varphi_i \end{cases}$$

The following characterization of Nash equilibria was given in [25] for tail objectives only. We extend it to safety and reachability objectives.

Corollary 1 (Characterization of Nash Equilibria ([25] for tail objectives)). *Let $\mathcal{G}$ be a multiplayer game with either all safety, all reachability, or all tail objectives, definable in LTL[$\mathcal{G}$]. Then, the following hold:*

1. *For all $\pi \in \mathsf{Plays}(\mathcal{G})$, if $\pi \models \phi_{Nash}$, then there exists a Nash equilibrium $\bar{\sigma}$ in $\mathcal{G}$ such that $out(\bar{\sigma}) = \pi$,*
2. *For all Nash equilibrium $\bar{\sigma}$ in $\mathcal{G}$, $out(\bar{\sigma}) \models \phi_{Nash}$.*

Proof. Let $\mathcal{G} = \langle \mathcal{A}, (\mathcal{O}_i)_{0 \leqslant i \leqslant k} \rangle$ where the set of states V is partitioned into $V_0, V_1, \ldots, V_k$. We define the $k+2$-player game $\mathcal{G}' = \langle \mathcal{A}', (\mathcal{O}'_i)_{0 \leqslant i \leqslant k+1} \rangle$ where $\mathcal{A}'$ is the $k+2$-game arena obtained from $\mathcal{A}$ by increasing the index of each player by one (Player i becomes Player $i+1$), and by adding a new Player 0, who owns no states, i.e. $V' = V$, $V'_0 = \varnothing$ and $V'_i = V_{i-1}$ for all $1 \leqslant i \leqslant k+1$. The structure (transition relation) of $\mathcal{A}$ is kept. Player 0 in $\mathcal{G}'$ has the trivial objective V^ω, and $\mathcal{O}'_i = \mathcal{O}_{i-1}$ for all $1 \leqslant i \leqslant k+1$. Then, there exists a 0-fixed Nash equilibria in $\mathcal{G}'$ iff there exists a Nash equilibria in $\mathcal{G}$. Then, it suffices to apply Lemma 1 to get the result. Note that the trivial objective is a tail objective, but can be seen as a reachability objective where all states are target states, as well as a safety objective where all states are safe. ∎

4 Cooperative Rational Synthesis Problem(CRSP)

General solution to cooperative rational synthesis Lemma 1 allows us to give a generic procedure to solve the cooperative rational synthesis problem, which is based on the following direct consequence of Lemma 1:

Lemma 2. *Let $\mathcal{G}$ be a $k+1$-player game with either all safety, all reachability, or all tail objectives, definable in LTL[$\mathcal{G}$] by formulas $(\varphi_i)_{0 \leqslant i \leqslant k}$. There is a solution to the cooperative synthesis problem iff there exists a path $\pi \in \mathsf{Plays}(\mathcal{G})$ such that $\pi \models \phi_{0Nash} \wedge \varphi_0$.*

Then, in order to solve the cooperative synthesis problem, it suffices to compute the winning sets W_i, for $i = 1, \ldots, k$, and to model-check the formula $\phi_{0\text{Nash}} \wedge \varphi_0$ against the game arena underlying $\mathcal{G}$. Depending on the winning objectives, the formula $\phi_{0\text{Nash}} \wedge \varphi_0$ may have different forms, which may impact the complexity of model-checking it. One objective of this paper is to give tight complexity bounds for the model-checking of this formula and, thus, to the cooperative rational synthesis problem.

4.1 Safety games

In the case of safety condition, the characterization of a 0-fixed Nash equilibrium intuitively expresses the fact that either Player i always stays in its safe set of states, or it is the case that he loses by eventually reaching a unsafe state, but he couldn't play better, i.e., the play didn't pass through a state from which he has a winning strategy.

Based on Lemma 2, we can provide an algorithm to solve the cooperative rational synthesis problem for safety games. It suffices to model-check the LTL formula $\Box S_0 \wedge \bigwedge_{i=1}^{k}((\neg W_i \mathcal{U} \neg S_i) \vee \Box S_i)$ against the game arena. We show it can be done in NP, based on the following property: if a path satisfies of the game arena satisfies the formula, then there is a lasso path xy^ω satisfying it, such that x and y have polynomial length. Then, the nondeterministic algorithm solving the synthesis problem simply guesses such a path and verifies, in polynomial time, that it satisfies the desired property.

Lemma 3. *The cooperative rational synthesis problem for multiplayer safety games is in NP.*

Proof. To solve the NP membership of this problem, it suffices to check the existence of a path in the game arena that satisfies the LTL formula $\varphi = \Box S_0 \wedge \bigwedge_{i=1}^{k}((\neg W_i \mathcal{U} \neg S_i) \vee \Box S_i)$.

First, it is well-known that two-player safety games can be solved in polynomial time, and therefore the winning sets W_i can be computed in polynomial time.

Then, given a lasso path $\pi = xy^\omega$, it can be checked in polynomial time (in $|x|$ and $|y|$ and the size of the game arena) whether $\pi \models \varphi$. Indeed, viewing the sets W_i and S_i as atomic propositions, one can easily construct a 5-state automaton equivalent to each of the subformula $((\neg W_i \mathcal{U} \neg S_i) \vee \Box S_i)$, for which checking the acceptance of xy^ω can be done in polynomial time.

It remains to show that we can bound the length of x and y polynomially.

Let $\pi \in V^\omega$ be a path satisfying φ. For each $i \in \{1, \ldots, k\}$, we consider the first occurrence of an unsafe state of Player i in π, and decompose π according to these positions as follows. Formally, π is decomposed as $\pi = x_1 v_{P_1} x_2 v_{P_2} \ldots x_l v_{P_l} x_{l+1}$ such that for all $j \in \{1, \ldots, l\}$, $P_j \subseteq \{1, \ldots, k\}$ and v_{P_l} is the first occurrence of a state which is unsafe for all the players in P_l (P_l is maximal for that property).

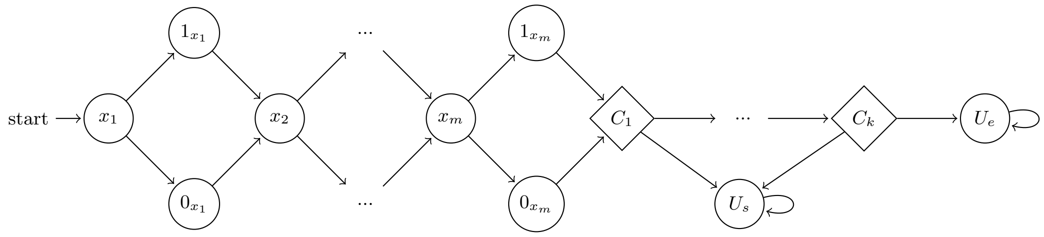


Fig. 3: Cooperative Safety: Reduction from 3-SAT

First, we remove the cycles in all x_j, $j \in \{1, \ldots, l\}$, leading to a new path of the form $x'_1 v_{P_1} x'_2 v_{P_2} ... x'_l v_{P_l} x_{l+1}$ where the x'_j are loop-free. This preserves the satisfaction of φ, i.e. $\pi' \models \varphi$. Indeed, by doing so, the subformula $\Box S_0$ is still satisfied, and for all $i \in \{1, \ldots, k\}$, if $\Box S_i$ was satisfied by π, then it is still satisfied in π'. If $\neg W_i \mathcal{U} \neg S_i$ was satisfied in π, then by the choice of our decomposition, removing the cycles still preserve the existence of an unsafe state for Player i in π', and all the states before its first occurrence in π' satisfies $\neg W_i$.

Second, we modify x_{l+1} into a short lasso path $x'_{l+1}(x'')^\omega$, where x'' is a simple loop, and x'_{l+1} is loop-free. This can be done by taking x'_{l+1} to be shortest prefix of x_{l+1} to a state v that repeats in the future, and to take x'' has any loop from v to v, shortened into a simple loop by removing all inner-cycles. All these operations preserve the properties of satisfying $\Box S_i$ for all $i \in \{0, \ldots, k\}$.

Then, we set $x = x'_1 v_{P_1} x'_2 v_{P_2} ... x'_l v_{P_l} x'_{l+1}$ and $y = x''$. Then, $xy^\omega \models \varphi$, and $|xy| \leqslant n(k+2)$, concluding the proof. ∎

Lemma 4. *The cooperative rational synthesis problem for multiplayer safety games is NP-hard.*

Proof. The proof for the NP-hardness of this problem is done by reduction from $3SAT$. Given a Boolean formula $\varphi = C_1 \wedge ... \wedge C_k$ in conjunctive normal form where each clause has at most three literals, we construct a $(k+1)$-player safety game $\mathcal{G}_\varphi$ as follows (its arena is depicted in Fig.3): Let $X = \{x_1, ..., x_m\}$ be the set of variables that appear in φ. The game $\mathcal{G}_\varphi$ has three states $x, 1_x$ and 0_x controlled by the system (Player 0) for all variables $x \in X$. The two latter states correspond to the two possible truth values of x. For all $i \in \{1, \ldots, m-1\}$, there are edges from each x_i to both 1_{x_i} and 0_{x_i} and from 1_{x_i} and 0_{x_i} to x_{i+1}. There is a state C_j controlled by Player j for each clause in φ and two states U_s and U_e (which will be unsafe for the system and the environment respectively). We add edges from 1_{x_m} and 0_{x_m} to C_1, from C_j to C_{j+1} for all $1 \leqslant j < k$, and from every C_j to U_s and from C_k to U_e. In Fig. 3 the system plays the round states and each environment Player i plays the diamond state C_i.

We let V be the set of states (vertexes) of $\mathcal{G}_\varphi$, x_1 being the initial one. All states but U_s are safe for Player 0, i.e. $S_0 = V \backslash \{U_s\}$. The unsafe states for Player $j \neq 0$ are U_e, as well as the state 0_x if $\neg x$ appears in C_j, and the state 1_x if x appears in C_j, i.e. $S_j = V \backslash (\{0_x \mid \neg x \in C_j\} \cup \{1_x \mid x \in C_j\} \cup \{U_e\})$.

Let us now prove the correctness of this reduction, i.e. φ is satisfiable iff there is a 0-fixed strategy Nash equilibrium winning for Player 0 in $\mathcal{G}_\varphi$. Suppose first that φ is satisfiable by a valuation $\nu = X \to \{0,1\}$ of its variables. The strategy σ_0 of the system is then to choose the truth values of the literals according to this valuation: choose 1_x if $\nu(x) = 1$, and 0_x otherwise. By doing so, all the players of the environment visit at least one unsafe state before reaching C_1. Indeed, let $j \in \{1, \dots, k\}$. Since C_j is satisfied by ν, there is a litteral ℓ in C_j such that $\nu(\ell) = 1$. If $\ell = x$ for some $x \in X$, then 1_x is unsafe for Player j, but that is exactly the choice of Player 0 to go to 1_x (and similarly when $\ell = \neg x$). After reaching C_1, the choices of the players $j \neq 0$ are to go down to C_k and then to U_e. This profile is winning for Player 0, and losing for all the other players. They have no incentive to deviate since they have already lost before making any choice. Therefore, it is a 0-fixed Nash equilibrium.

Conversely, if there is a solution for the cooperative synthesis problem, the only way to obtain a Nash equilibrium σ winning for the system is to make all the players j $1 \leqslant j \leqslant k$ lose before reaching C_1. Indeed, if σ is winning for the system, then $out(\sigma)$ eventually reaches U_e, which is losing for the environment. In order to prevent the deviation of the environment to U_s (which is safe for the environment), it is necessary that all the players but Player 0 has lost before reaching C_1. By definition of their sets of unsafe states, the only way to make them lose before reaching C_1 is a to chose a valuation that satisfies the formula, if it exists. ■

As a consequence of Lemma 3 and 4, one gets:

Theorem 1. *The cooperative rational synthesis problem for multiplayer safety games is NP-complete.*

4.2 Reachability games

In this section, we prove the NP-completness of the cooperative rational synthesis problem with reachability objectives. We provide a similar algorithm as in the case of safety objectives. However, observe that unlike the case of two-player zero-sum games, there is no duality between reachability and safety, and no natural reduction from reachability to safety.

Based on Lemma 2, in order to solve the rational cooperative synthesis problem in reachability games, it suffices to have a procedure that test the existence of the path in the game that satisfies $\Diamond R_0 \wedge \left(\bigwedge_{i=1}^{k} \Box\neg R_i \to \Box\neg W_i\right)$. We prove that such a formula has short (polynomial length) lasso witnesses, and therefore we obtain an NP procedure for the synthesis problem.

Lemma 5. *The cooperative rational synthesis problem for multiplayer reachability games is in NP.*

Proof. We start by showing that each path π such that $\pi \models \varphi$ where:

$$\varphi = \Diamond R_0 \wedge \big(\bigwedge_{i=1}^{k} \Box\neg R_i \rightarrow \Box\neg W_i\big)$$

can be shortened into a lasso path π^* such that $\pi^* \models \varphi$ and π^* can be decomposed into a prefix of length at most nk followed by a simple loop. In fact, let L be the maximal set of players such that $\pi \models \bigwedge_{i \in L} \Diamond R_i$. Then, it is sufficient to enucleate the first occurrences of states in R_i for all $i \in L$ along π, and eliminate all the cycles between these occurrences. This leads to a new path π' where each player $i \in L$ accomplishes its reachability objective in at most $n|L|$ steps and such that $\pi' \models \bigwedge_{i \notin L} \Box\neg W_i$. Let j be the smallest position in π' such that each player $i \in L$ has accomplished its reachability objective, i.e. $j = min\{\ell \geqslant 0 \mid \pi'[: \ell] \models \bigwedge_{i \in L} \Diamond R_i\}$. Then, π^* is obtained from π' by considering the first cycle (reduced to a simple cycle) appearing from the vertex $\pi'(j)$ on.

Therefore, the NP algorithm works as follows: guess a lasso-path of length at most $n(k + 1)$, check whether it fulfils φ, and use it to build a winning strategy that uses as much memory as the length of the path. This is correct by the small lasso property proved before, and Lemma 2. ■

We finally show that the cooperative rational synthesis problem for reachability games is NP-hard. The proof is by reduction from 3-SAT, and is a slight modification of the reduction for the safety objectives (Lemma 4).

In this latter reduction, remind that given a 3-SAT formula $\varphi = C_1 \wedge \cdots \wedge C_k$ in conjunctive normal form, we build a $k + 1$-reachability game $\mathcal{G}_\varphi$ over the arena of Figure 3. For safety, one actually shows that φ is satisfiable if and only if $\mathcal{G}_\varphi$ admits a 0-fixed NE where all the players composing the environment lose, i.e. reach the complement of their safety sets $S_1, \ldots S_k$. As for the system, avoiding his only forbidden state U_s amounts to reach U_e (since in $\mathcal{G}_\varphi$, U_s and U_e are the only two states that are eventually reached). Therefore, there is a solution for the safety sets $\{V \backslash \{U_s\}, (S_i)_{i=1\ldots k}\}$ iff there is a solution for the reachability sets $\{\{U_e\}, (\bar{S}_i)_{i=1\ldots k}\}$, which shows the following lemma.

Lemma 6. *The cooperative rational synthesis problem for reachability games is NP-hard.*

As a consequence of Lemma 5 and Lemma 6, we obtain the following theorem:

Theorem 2. *The cooperative rational synthesis problem for reachability games is NP-complete.*

4.3 Büchi and ω-regular winning conditions

In the following, we treat the cooperative rational synthesis problem in the case of ω-regular objectives, in particular for Büchi, co-Büchi, Streett and Parity objectives. For these objectives, we can rely on results shown by Ummels in [25], following the next remark.

Remark 1. In [25] is studied the complexity of finding a Nash equilibrium $\bar{\sigma}$ in a k-player game with Büchi, co-Büchi, Streett and Parity objectives, such that $x \leqslant pay(\bar{\sigma}) \leqslant y$, for two given threshold $x, y \in \{0,1\}^k$. The cooperative synthesis problem reduces to this setting, by taking $x = (1, 0, \ldots, 0)$ and $y = (1, 1, \ldots, 1)$. Note that in [25], the threshold Nash equilibria problem was not studied for safety and reachability and Muller conditions. For Rabin conditions, there is a remark in the conclusions of [25] that a P^{NP} complexity can be obtained.

Based on Remark 1 and the results of [25], one obtains the upper-bound of following theorem:

Theorem 3. *The cooperative rational synthesis problem for multiplayer games is:*

- *in PTime for Büchi objectives,*
- *NP-complete for co-Büchi, Streett and parity objectives.*

Proof. As we have said, the upper bounds are direct consequences of the results of [25] and remark 1. Let us show the NP lower bound for co-Büchi, Streett and parity objectives.

Co-Büchi objectives. It is shown in [25][Theorem 15] that the problem of finding a Nash equilibrium co-Büchi multiplayer games with a payoff between the thresholds $(1, 0, 0, ..., 0)$ and $(1, 1, 1, ..., 1)$ is NP-hard. The result follows from Remark 1.

Parity. It follows directly from these two facts: (i) the problem is NP-hard for co-Büchi objectives, (ii) a co-Büchi objective given by a set of states F can be equivalently expressed by the priority function p_F such that $p_F(v) = 1$ if $v \in F$, and 2 otherwise.

Streett. As for parity, a Streett condition can easily express a co-Büchi condition F, by taking the set of pairs $\{(F, \varnothing)\}$. The result follows from the NP-hardness of co-Büchi objectives. ■

Rabin games Let consider the $k+1$-player Rabin game $\mathcal{G} = \langle \mathcal{A}, (\mathsf{Rabin}(\psi_i))_{0 \leqslant i \leqslant k} \rangle$ where each Player i has the objective $\psi_i = \{(L_1, R_1), ..., (L_{m_i}, R_{m_i})\}$. Then, based on Lemma 2 and the fact that the Rabin condition ψ_i can be equivalently expressed by the $LTL[\mathcal{G}]$ formula $\varphi_i = \bigvee_{j=1}^{n_i} (\Box\Diamond L_{ij} \wedge \Diamond\Box\neg R_{ij})$, solving the cooperative rational synthesis in Rabin games is equivalent to find a path satisfying the formula

$$\phi_{0\text{Nash}}^{\mathcal{G}} \wedge \varphi_0 \equiv \bigvee_{j=1}^{n_0} (\Box\Diamond L_{0j} \wedge \Diamond\Box\neg R_{0j}) \wedge \bigwedge_{i=1}^{k} \Big(\Box\neg W_i \vee \bigvee_{j=1}^{n_i} (\Box\Diamond L_{ij} \wedge \Diamond\Box\neg R_{ij})\Big)$$

.

Theorem 4. *The cooperative rational synthesis problem for multiplayer Rabin games is in P^{NP}.*

Proof. We first show that given the sets W_i for $1 \leqslant i \leqslant k$, each path $\pi = xy^\omega$ such that $\pi \models \phi_{0\text{Nash}}^{\mathcal{G}} \wedge \varphi_0$ can be shortened into a lasso path $\pi' = x'(y')^\omega$ such that $\pi' \models \phi_{0\text{Nash}}^{\mathcal{G}} \wedge \varphi_0$ and $|x'y'| \leqslant n^2 + nk$.

First, we mark as red node the first occurrence along xy of a state in W_i for every player not satisfying φ_i. We also mark as red node the first occurrence along y of states in L_{ij} and R_{ij}(if any) for every player and every pair in the Rabin condition. Note that along x at most k red nodes are marked and along y we have at most n states since we marked the first occurrence of a state and the game arena has n states.

Then, by removing all the loops in x and y that don't contain red nodes we obtain x' and y' such that $|x'| \leqslant nk$ and $|y'| \leqslant n^2$. Note that the property $\pi' \models \phi^{\mathcal{G}}_{0\text{Nash}} \wedge \varphi_0$ also holds since we didn't remove key nodes(red nodes on y) from π.

Then the P^{NP} algorithm will run as follows. First, it guesses a path $\pi = xy^\omega$ of polynomial length (as we saw $|xy| \leqslant n^2 + nk$ is enough) and mark the states in xy by W_i or $\neg W_i$ by checking in NP time if $s \in W_i$ for each state. This can be done in P^{NP} time. Then, in polynomial time check if $\pi \models \phi^{\mathcal{G}}_{0\text{Nash}} \wedge \varphi_0$ by checking if for each player is a pair (L_{ij}, R_{ij}) such that L_{ij} appears in y but not R_{ij}. If not, W_i should not be a label of a state in xy. If it is, the algorithm rejects. ■

Theorem 5. *The cooperative rational synthesis problem for multiplayer Rabin games is NP-hard and co-NP-hard.*

Proof. As in the case of Parity and Streett games, the NP-hardness comes directly from the fact that we can easily express a co-Büchi condition F into a Rabin condition $\psi = \{(V, F)\}$ where V is the set of states of the game arena.

To show the co-NP-hardness, we reduce from the two-players zero-sum Rabin games which are NP-hard. Let $\mathcal{G} = \langle V, V_A, V_B, E, v_0, \psi\rangle$ be such a game where the protagonist (Player A) has the Rabin objective ψ. We construct the game $\mathcal{G}'$ by considering a copy of $\mathcal{G}$ together with two extra states v and v' and transitions from v to both v' and the initial state of $\mathcal{G}$ and a self loop on v'. Then, Player 0 controls the states belonging to Player B in $\mathcal{G}$ and Player 1 controls the states belonging to Player A in $\mathcal{G}$ together with the newly introduced states v and v'.

Formally, $\mathcal{G}' = \langle V', V_0, V_1, E', v'_0, \psi_0, \psi_1\rangle$ where $V' = V \cup \{v, v'\}$, $V_0 = V_B$, $V_1 = V_A \cup \{v, v'\}$, $E' = E \cup \{(v, v_0), (v, v'), (v', v')\}$, $v'_0 = v$ and the objectives of the two players are defined as $\psi_0 = \{(\{v'\}, \varnothing)\}$ and $\psi_1 = \psi$. That is, Player 0 wins if the game goes in the state v' and Player 1 wins if it is satisfied the winning condition of the protagonist in the game $\mathcal{G}$.

We claim that there is a solution to the rational synthesis problem in $\mathcal{G}'$ iff there is no winning strategy for Player A in $\mathcal{G}$. Indeed, if there is a solution to the rational synthesis, there is a 0-fixed Nash equilibrium (σ_0, σ_1) winning for Player 0. The only possibility that this happens is if $\sigma_1(v) = v'$ in which case Player 1 looses. But since (σ_0, σ_1) is a 0-fixed Nash equilibrium, for any other strategy σ'_1 s.t. $\sigma'_1(v) = v_0$, $out(\sigma_0, \sigma'_1)$ doesn't satisfy $\mathsf{Rabin}(\psi_1)$. That is, there is a strategy σ_B for Player B in $\mathcal{G}$ such that $\forall \sigma_A$ a strategy pf Player A, $out(\sigma_A, \sigma_B) \not\models \mathsf{Rabin}(\psi)$ which means that Player A has no winning strategy in $\mathcal{G}$.

Suppose now that there is no solution to the rational synthesis. It means that there is no 0-fixed Nash equilibrium (σ_0, σ_1) such that $out(\sigma_0, \sigma_1)$ satisfies $\mathsf{Rabin}(\psi_0)$. That is, whatever strategy σ_0 chooses Player 0, Player 1 prefers to go in the copy of $\mathcal{G}$ where he has a strategy to win. That is, Player A has a winning strategy σ_A in $\mathcal{G}$ to ensure $\mathsf{Rabin}(\psi)$.

Muller games Let $\mathcal{G} = \langle \mathcal{A}, \mathsf{Muller}(\mu_i)_{i \in \Omega} \rangle$ be a multiplayer Muller game with winning condition for Player i given as the boolean formula $\mu_i = l_1\ Op_1\ l_2\ Op_2\ ...\ l_{m_i}$ where $Op_j \in \{\wedge, \vee\}$ for all $1 \leqslant j < m_i$ and each literal l_j is either a state $v_j \in V$ or its negation $\neg v_j$.

Let define the LTL formula φ_i from μ_i by replacing each v_j by the subformula $\Box\Diamond v_j$. Then, we claim that for any path π, we have that π satisfies $\mathsf{Muller}(\mu_i)$ iff $\pi \models \varphi_i$. Intuitively, it holds since whenever a $v \in inf(\pi)$, it satisfies both the Muller condition v and the LTL formula $\Box\Diamond v$. And if $v \notin inf(\pi)$, both Muller condition $\neg v$ and LTL formula $\Diamond\Box\neg v \equiv \neg\Box\Diamond v$ are satisfied by π.

Then, in the cooperative setting, using the characterization of 0-fixed Nash equilibria for ω-regular objectives, the problem is to decide the existence of a path satisfying the LTL formula $\varphi = \varphi_0 \vee \bigwedge_{i=1}^{k} (\varphi_i \vee \Box\neg W_i)$ where φ_i, $0 \leqslant i \leqslant k$ is defined as above. The formula φ us an LTL formula in the fragment $\mathfrak{B}(\mathcal{L}_{\Box\Diamond}(\mathcal{P}) \cup \mathcal{L}_{\Diamond,\wedge}(\mathcal{P}))$ where $\mathcal{P}$ corresponds to the atomic propositions associated to the states of $\mathcal{G}$ and to each W_i. For this fragment of LTL, it is shown in [1] that the solving a two-player game with the protagonist having the LTL objective is in PSPACE and therefore also the cooperative rational synthesis for Muller objectives is.

For the PSPACE-hardness, we reduce from the problem of solving two-players zero-sum Muller games with Muller objective μ that is well known to be PSPACE-hard. In the newly constructed game keep the game arena and set the objective of Player 0 to be μ and the objective of Player 1 to be $\neg\mu$. Then, it is obvious that there is a 0-fixed Nash equilibrium (σ_0, σ_1) winning for Player 0 iff there is a winning strategy σ_0 for the Player 0 in the zero-sum two-player game.

5 Non-Cooperative Rational Synthesis Problem(NCRSP)

In this section, we study the complexity of non-cooperative rational synthesis problem (when the number of players is not fixed). In this setting the environment may not co-operate with the system, and may play (rationally) any strategy profile providing it is a 0-fixed Nash equilibrium.

In the cooperative setting, in the cases where we could not rely on existing results [25], namely reachability and safety objectives, we get our upper bounds via a reduction to a model-checking problem. In the non-cooperative setting, we cannot rely on existing results.

In Lemma 1, we have characterised 0-fixed NE by means of an LTL formula $\phi^{\mathcal{G}}_{0\mathrm{Nash}}$. This allowed us to solve cooperative rational synthesis problem by model-checking against the game $\mathcal{G}$, the formula $\phi^{\mathcal{G}}_{0\text{Nash}} \wedge \varphi_0$, where φ_0 is Player 0's objective. It is tempting to think that non-cooperative rational synthesis reduces to a two-player zero-sum game between Player 0, whose objective is $\phi^{\mathcal{G}}_{0\text{Nash}} \rightarrow \varphi_0$, and the coalition of the other players. However, the three state game arena from Example 1 shows that this is not true in general. Indeed, in this example there is a solution to non-cooperative rational synthesis problem, but no solution to the two-player game with objective $(\Box\bar{R}_1 \rightarrow \Box\bar{W}_1) \rightarrow \Diamond R_0$. Since $W_1 = \{3\}$, whatever the strategy of Player 0 is, if Player 1 stays in state 1 forever, the path $\pi = (1)^\omega$ satisfies $(\Box\bar{R}_1 \rightarrow \Box\bar{W}_1)$ but not $\Diamond R_0$ and therefore Player 0 loses. The intrinsic reason why the reduction to two-player games is incorrect lies in the definition of non-cooperative rational synthesis problem: once a Player 0's strategy σ_0 is fixed, only 0-fixed NE with respect to σ_0 are considered, while the formula $\phi_{0\text{Nash}}$ can be satisfied by paths which are outcomes of some 0-fixed NE, fixed for a different strategy of Player 0.

Intuitive solution Let fix a strategy σ_0 that we represent as a tree t_{σ_0} and use *tree automata* to define the set of strategies that are solutions to the non-cooperative rational synthesis. The emptiness of tree automata is then checked by solving a two-player zero-sum game, whose complexity is carefully analyzed for all the winning conditions considered in the paper.

Strategy trees and good deviations Let $\mathcal{A}$ be a $k+1$-players arena with set of states V and let $\sigma_0 : V^*V_0 \rightarrow V$ be a strategy of Player 0 in $\mathcal{A}$. We explain how σ_0 is encoded as a tree. The labels are in the set $\Sigma = V \cup \{*_i \mid 1 \leqslant i \leqslant k\} \cup \{\#\}$ and the set of directions is V. Therefore, any node of the tree is an history h in the game arena $\mathcal{A}$. Then, if $h = \epsilon$ (root node), we set its label to $\#$. Otherwise, it is of the form $h = h'v$, then there are two cases:

(i) if $v \in V_0$, then $t_{\sigma_0}(h) = \sigma_0(h)$,
(ii) if $v \in V_i$ for $i \neq 0$, then $t_{\sigma_0}(h) = *_i$ (only the turn i is encoded)

Intuitively, the "letter" $*_i$ in the strategy tree encodes the fact that Player i could do any choice in the turn-based (k+1)-player game $\mathcal{G}$. We denote by T_0 the set of strategy trees t_{σ_0}. Note that not all Σ-labeled V-tree is a strategy tree.

We now want to characterize the strategy trees t_{σ_0} s.t. σ_0 is a solution to non-cooperative rational synthesis problem in a game $\mathcal{G} = \langle \mathcal{A}, (\mathcal{O}_i)_{i\in\Omega}\rangle$ with either all safety, all reachability, or all tail objectives. The strategy tree t_{σ_0} is a solution of the problem if for all branches π, either it is winning for Player 0, or π doesn't correspond to a 0-fixed Bash equilibrium and there is a player that could deviate and win considering the system plays the strategy σ_0. That is,

$$\pi \models \phi^{\mathcal{G}[\sigma_0]}_{\text{Nash}} \rightarrow \alpha_0$$

The branch π is not the outcome of a 0-fixed NE iff some player loses ($\pi \notin \mathcal{O}_i$ for some $i \neq 0$) and there is a prefix h from which Player i has a winning strategy against all other players (and the strategy σ_0). We call the history h a good deviation point. Formally, h is a *good deviation point* if there exists $i \in \{1, \dots, k\}$ s.t. $\pi \notin \mathcal{O}_i$ and there is a strategy σ_i for

Player i s.t. for all strategies $(\sigma_j)_{i\in\{1,...,k\}\setminus\{i\}}$, $h.out(\sigma_0|_h,...,\sigma_{i-1}|_h,\sigma_i|_h,\sigma_{i+1}|_h,...,\sigma_k|_h) \in \mathcal{O}_i$. A branch $\pi \in V^\omega$ has a *good deviation* if some of its prefix h is a good deviation point. Let us denote by $\mathsf{NCRSP}(\mathcal{G})$ the set of strategy trees t_{σ_0} such that σ_0 is a solution to the NCRSP in $\mathcal{G}$. Then:

Lemma 7. *For all strategies σ_0 of Player 0, $t_{\sigma_0} \in \mathsf{NCRSP}(\mathcal{G})$ iff for all branches π of t_{σ_0} compatible with σ_0, either $\pi \in \mathcal{O}_0$ or π has a good deviation.*

Proof. First, let prove the implication from left to right and consider $t_{\sigma_0} \in \mathsf{NCRSP}(\mathcal{G})$ and a branch π such that $\pi \notin \mathcal{O}_0$. Then, since t_{σ_0} is a solution to $\mathsf{NCRSP}(\mathcal{G})$, for all $\sigma_1,...,\sigma_k$ such that $out(\sigma_0,\sigma_1,...,\sigma_k) = \pi$, $\langle\sigma_0,\sigma_1,...,\sigma_k\rangle$ is not a 0-fixed Nash equilibrium.

Let $\sigma_1,...,\sigma_k$ such that all the players follow the path π and punish the Player i that deviates from it by playing the worst strategy profile for him (they play the retaliating strategies $\mathsf{ret}_j^{s,i}$ against Player i from the state s to which he deviates). That is, each Player j plays σ_j defined as

$$\sigma_j(x) = \begin{cases} \pi[l+1] & \text{if } x = v_0v_1...v_l \text{ is a prefix of } \pi \\ \mathsf{ret}_j^{s,i}(sx_2) & \text{if condition (1) is satisfied} \\ \beta_j(x) & \text{otherwise} \end{cases}$$

where β_j is an arbitrary strategy of Player j and condition (1) requires that x can be decomposed into $x = x_1sx_2$ such that $x_1 \in V^*V_i$, $s \notin W_i^{\mathcal{G}[\sigma_0]}$, x_1 is a prefix of π, x_1s is not a prefix of π (meaning that Player i has deviated to a losing state when the strategy σ_0 is fixed, i.e., Players $1,...,j-1,j+1,...,k$ have a strategy to make him lose under σ_0).

Clearly, $out(\sigma_0,\sigma_1,...,\sigma_k) = \pi$ and therefore by hypothesis, $\langle\sigma_0,\sigma_1,...,\sigma_k\rangle$ is not a 0-fixed Nash equilibrium. Hence, there is a player i that prefers to deviate from π and has a strategy σ_i' such that $out(\langle\sigma_0,\sigma_1,...,\sigma_{i-1},\sigma_i',\sigma_{i+1},...,\sigma_k\rangle) \in \mathcal{O}_i$. From the construction of the strategy profile, Player i chooses to deviate to a state in which he has a winning strategy when Player 0 plays σ_0. Let $h \in V^*V_i$ be the prefix of π after which Player i deviates. Then, for all strategies $\tilde{\sigma}_1,...\tilde{\sigma}_{i-1},\tilde{\sigma}_{i+1},...,\tilde{\sigma}_k$ for the players $1,...,i-1,i+1,...,k$, we have that $h.out(\sigma_0|_h,...,\tilde{\sigma}_{i-1}|_h,\sigma_i'|_h,\tilde{\sigma}_{i+1}|_h,...,\tilde{\sigma}_k|_h) \in \mathcal{O}_i$ which means that there is a good deviation for Player i and then π has a good deviation h.

In the other direction, let take the strategy treet_{σ_0} and π a branch of t_{σ_0} compatible with σ_0 s.t. $\pi \notin \mathcal{O}_0$. Then, there is a good deviation from π for a Player i that loses in π. That is, Player i has a strategy σ_i' such that he wins by deviating from π at a position j against any strategy profile that follows π. That is, for all $\sigma_1,...,\sigma_k$ s.t. $out(\langle\sigma_0,\sigma_1,...,\sigma_k\rangle) = \pi$, $\langle\sigma_0,\sigma_1,...,\sigma_k\rangle$ is not a 0-fixed NE since Player i can deviate and win. Therefore, $t_{\sigma_0} \in \mathsf{NCRSP}(\mathcal{G})$.

The equivalence is straightforward for the branches $\pi \in \mathcal{O}_0$. ∎

Lemma 8. *Let $\mathcal{G} = \langle\mathcal{A},\mathcal{O} = (\mathcal{O}_i)_{i\in\Omega}\rangle$ be a $k+1$-player game with n vertices. One can construct a non-deterministic tree automaton $\mathcal{T}_\mathcal{G}$ (with an exponential number of states in k, and polynomial in n) with an accepting condition α such that $\mathcal{L}_\alpha(\mathcal{T}_\mathcal{G}) = \mathsf{NCRSP}(\mathcal{G})$.*

Moreover, for all runs r of $\mathcal{T}_{\mathcal{G}}$, for all branches π of r, the number of states appearing in π is polynomial in n and k.

The nondeterministic tree automaton $\mathcal{T}_{\mathcal{G}}$ is obtained as a product of two automata. First, we construct a deterministic safety automaton $\mathcal{C}_{\mathcal{G}}$ that checks that an accepting tree t_{σ_0} is a proper encoding of a strategy σ_0 in the turn-based game $\mathcal{G} = \langle \Omega, V = V_0 \uplus ... \uplus V_n, E, v_0, (\mathcal{O}_i)_{i \in \Omega} \rangle$. Then, we construct a nondeterministic tree automaton $\mathcal{U}$ that is assumed to run on proper encodings of strategies and checks that it corresponds to a solution to the NCRSP. Details on the construction of the two automata are given in the following.

Automaton $\mathcal{C}_{\mathcal{G}}$. The deterministic safety automaton $\mathcal{C}_{\mathcal{G}}$ accepts trees t that are proper strategy trees encoding a strategy σ_0 of Player 0. This automaton is polynomial in the size of the game and keeps the information about the last direction taken in the tree t and depending on the player that controls it, checks if the tree is correctly labeled.

Formally, the automaton $\mathcal{C}_{\mathcal{G}}$ is formally defined as $\mathcal{C}_{\mathcal{G}} = (Q_{\mathcal{C}}, q_0^{\mathcal{C}}, \delta_{\mathcal{C}}, \alpha_{\mathcal{C}})$ where $Q_{\mathcal{C}} = V \cup \{\bot, q_0^{\mathcal{C}}\}$ and the transition relation defined as $\delta_{\mathcal{C}}(q_0^{\mathcal{C}}, \#, v_0) = v_0$, $\delta_{\mathcal{C}}(q_0^{\mathcal{C}}, l, d) = \bot$ for any direction $d \in V$ if $l \neq \#$, $\delta_{\mathcal{C}}(q \neq q_0^{\mathcal{C}}, \#, d) = \bot$, $\delta_{\mathcal{C}}(\bot, l, d) = \bot$ and

$$\delta_{\mathcal{C}}(v, l, d) = \begin{cases} d & \text{if } (v \in V_0 \text{ and } (v, l) \in E) \\ & \quad \text{or } (v \in V_{i \neq 0} \text{ and } l = *_i) \\ \bot & \text{otherwise} \end{cases}$$

Then, the acceptance condition on $\mathcal{C}_{\mathcal{G}}$ is $\alpha_{\mathcal{C}} = \{\eta \in (Q_{\mathcal{C}} \backslash \{\bot\})^{\omega}\}$. Note that by the construction of the automaton, this is a safety condition that asks to avoid $\bot$ which appears when being the turn of Player 0(state $v \in V_0$), the letter l we read is not a valid choice of him($(v, l) \notin E$). Also, the automaton $\mathcal{C}_{\mathcal{G}}$ is deterministic.

Having the definition if the automaton $\mathcal{C}_{\mathcal{G}}$ in following we construct the automaton $\mathcal{U}$ assuming it only runs on proper tree encodings of strategies.

Automaton $\mathcal{U}$. The construction of $\mathcal{U}$ is based on Lemma 7. For each branch, it will check that it belongs to $\mathcal{O}_0$ or it will guess a prefix and check it is a good deviation. To guess good deviations, the automaton $\mathcal{U}$ has to guess subtrees in which at least one player has a winning strategy. This information is stored in a set $W \subseteq \Omega$, with the following semantics: if $\mathcal{U}$ is in some state with set W at some node $h \in V^*$ and $i \in W$, then Player i has a winning strategy in the subtree rooted at h. The set of players for which a good deviation has been guessed is stored in a set $D \subseteq \Omega$, with the following semantics: if $\mathcal{U}$ is in some state with set D and $i \in D$, at some node $h \in V^*$, then some prefix of h is a good deviation.

The information in D is monotonic. Whenever $i \in D$ in a state, then $i \in D$ in all the successor states. In addition, it is updated by adding players in D depending on the updates of W. The information on W is maintained as follows: at some node $hv \in V^*$, if $i \in W$ and $v \in V_i$, then $\mathcal{U}$ non-deterministically chooses a strategy move for the Player

i and send W to one of the successor of v (and $W\backslash\{i\}$ in the other ones). If $i \notin W$ and $v \in V_i$, there are two possibilities. First, if $i \in D$ means that a deviation was guessed before and then W is sent to all successors. Otherwise, if $i \notin D$, there was not guessed a good deviation point before. Then either the current node h (owned by Player i) is not guessed to be a good deviation point and D is sent to all successors, or it is guessed to be a good deviation for Player i and then $D \cup \{i\}$ (and W) are sent to all successors but one in which is sent D and $W \cup \{i\}$. If $v \notin V_i$, $\mathcal{U}$ keeps $i \in W$ in all successors of v.

Formally, $\mathcal{U} = (Q_{\mathcal{U}}, q_0^{\mathcal{U}}, \delta_{\mathcal{U}}, \alpha_{\mathcal{U}})$ where the set of states is $Q_{\mathcal{U}} \subseteq \{q_0^{\mathcal{U}}, \top\} \cup 2^{\Omega} \times 2^{\Omega} \times V$ Intuitively, a state $q = (W, D, v)$ stores information about the set W of agents that need a winning strategy from the current node, the set D of agents that may deviate to win and the last direction taken.

To define the transition relation, we will define functions mapping directions to states. If we do not define them for some directions d, it means that d is mapped to $\top$. Then, considering a state $q = (W, D, v)$, the transition relation $\delta_{\mathcal{U}}$ is defined as:

- $\delta_{\mathcal{U}}(q_0^{\mathcal{U}}, \#) = \{\rho_0\}$ where $\rho_0(v_0) = (\varnothing, \varnothing, v_0)$
- $\delta_{\mathcal{U}}(q, v_1) = \{\rho_1\}$ where $\rho_1(v_1)((W, D, v_1), v_1)$, $v_1 \in V_1$,
- $\delta_{\mathcal{U}}(\top, l) = \{\rho_2\}$ where $\rho_2(d) = \top$, for all $d \in V$ and $l \neq \#$
- $$\delta_{\mathcal{U}}(q, *_{i \neq 0}) = \begin{cases} \{\rho\} & \text{if } i \in D \\ & \text{where } \rho(v') = (W, D, v') \text{ for all } (v, v') \in E \\ \{\rho_{v'} \mid (v, v') \in E\} & \text{if } i \in W \\ & \text{where } \rho_{v'}(v') = (W, D, v') \\ & \text{and } \rho_{v'}(v'') = (W\backslash\{i\}, D \cup \{i\}, v'') \text{ for all } v'' \neq v'. \\ \{\rho\} \cup \{\lambda \mid (v, v') \in E\} & \text{if } i \notin W \text{ and } i \notin D \\ & \text{where } \lambda_{v'}(v') = (W \cup \{i\}, D, v') \\ & \text{and } \lambda_{v'}(v'') = (W, D \cup \{i\}, v') \forall v'' \neq v' \end{cases}$$

Along a path of a run of $\mathcal{U}$, there are monotonicity properties for the W and D-components of the states. Indeed, by construction, $\mathcal{U}$ never removes a player from D. For W, a player i can be removed (case 3) but then it is added to D and, once a player belongs to D, it can never be added to W again. It is correct since for a history h, if one guesses that Player i has a winning strategy from history hv, then i is added to D for all successors hv' ($v' \neq v$) and there is no need to guess again later on a good deviation for Player i in the subtrees rooted at the nodes hv', and therefore no need to add i in W again. Therefore along a path η of a run, there is only a polynomial number of different components D and W, and they necessarily stabilize eventually, to a set that we denote by $\lim_D(\eta)$ and $\lim_W(\eta)$. This monotonic behavior is crucial for complexity.

Finally, the winning condition for reachability and tail objectives asks that on each path of the accepting run, either Player 0 wins, or there is a player that looses but he belongs to the set D eventually (therefore in the past he could have deviate and win). In the same time, the automaton checks that the players that belong to W after it stabilizes (that pretend to win along the path), indeed win by checking that the projection on the directions belong to $\mathcal{O}_i$. As for the safety condition, the winning condition asks for a winning state before the unsafe state of Player i. Formally,if we denote by $\mathrm{IRuns}(\mathcal{U})$ the set of images of branches of runs of $\mathcal{U}$, and by $\eta|_V$ the V-projection of any $\eta \in (2^{\Omega} \times 2^{\Omega} \times V)^{\omega}$, we have:

$$\alpha_{\mathcal{U}} = Q^*\{\top\}^{\omega} \cup (\{\eta \in \mathrm{IRuns}(\mathcal{U}) \cap (Q\backslash\{\top\})^{\omega} \mid \eta|_V \in \mathcal{O}_0 \vee \bigvee_{i=1}^{k} \big(\eta|_V \notin \mathcal{O}_i \wedge \varphi_{\exists dev}(i,\eta)\big)\} \cap$$
$$\cap \{\eta \in \mathrm{TRuns}(\mathcal{U}) \cap (Q\backslash\{\top\})^{\omega} \mid \bigwedge_{i \in \lim_W(\eta)} \eta|_V \in \mathcal{O}_i\})$$

where the formula $\varphi_{\exists dev}(i,\eta)$ says that there is a good deviation for Player i. That is, $\varphi_{\exists dev}(i,\eta) = \exists p \geqslant 0$ s.t. $i \in \eta|_D[p]$ in the case of tail objectives or $\varphi_{\exists dev}(i,\eta) = \exists p \geqslant 0$ s.t. $i \in \eta|_D[p]$ and $\forall r \leqslant p, \eta|_V \in S_i$ for safety condition $\mathsf{Safe}(S_i)$.

Note that for different particular winning conditions, we may need to add more information on states of the automaton in order to check the satisfaction of the winning condition of the players and therefore slightly modify the transition relation. For example, in the case of Safety conditions, we may need a set of players that already lost and ask that the deviation is made before losing.

Automaton $\mathcal{T}_{\mathcal{G}}$. Then, as mentioned before, the tree automaton $\mathcal{T}_{\mathcal{G}}$ with the accepting condition α such that $\mathcal{L}_{\alpha}(\mathcal{T}_{\mathcal{G}}) = \mathsf{NCRSP}(\mathcal{G})$ is defined as the product of the two automata $\mathcal{C}_{\mathcal{G}}$ and $\mathcal{U}$. Formally, the automaton $\mathcal{T}_{\mathcal{G}} = \langle Q, \{q_0\}, \delta, \alpha\rangle$ has states in $Q \subseteq (2^{\Omega} \times 2^{\Omega} \times V) \cup \{\bot, (q_0^{\mathcal{U}}, q_0^{\mathcal{C}})\} \cup (\{\top\} \times V)$, $q_0 = (q_0^{\mathcal{U}}, q_0^{\mathcal{C}})$ is the initial state and the transition relation for $l \in \Sigma$ is defined by

- $\delta((q_0^{\mathcal{U}}, q_0^{\mathcal{C}}), l) = \begin{cases} \{\rho_p\} & \text{with } \rho_p(d) = \bot \text{ if } \delta_{\mathcal{C}}(q_0^{\mathcal{C}}, l) = \rho_p \\ \{\rho_0\} & \text{if } \delta_{\mathcal{C}}(q_0^{\mathcal{C}}, l, d) = d \text{ and } \delta_{\mathcal{T}}(q_0^{\mathcal{T}}, l) = \{\rho_0\} \end{cases}$
- $\delta(\bot, l) = \{\rho_p\}$
- $\delta((\top, v), l) = \begin{cases} \{\rho_t\} & \text{with } \rho_t(v') = (\top, v') \text{ if } \delta_{\mathcal{C}}(v, v') = v' \text{ for } v' \in V \\ \{\rho_p\} & \text{if } \delta_{\mathcal{C}}(v) = \rho_p \end{cases}$
- $\delta(W, D, v, l) = \begin{cases} \delta_{\mathcal{U}}(W, D, v) & \text{if } \delta_{\mathcal{C}}(v, v') = v' \text{ for all } v' \in V \\ \{\rho_p\} & \text{otherwise} \end{cases}$

Remark 2. Note that on each branch of a run there are still only a polynomial number in the size of the initial game $\mathcal{G}$ of different states of the automaton since $\mathcal{T}_{\mathcal{G}}$ is the product of $\mathcal{U}$ with a deterministic safety tree automaton of polynomial size.

Finally, the acceptance condition for the automaton $\mathcal{T}_{\mathcal{G}}$ is essence the condition $\alpha_{\mathcal{U}}$ but also asks to avoid states $\bot$ that are reached in $\mathcal{C}_{\mathcal{G}}$ if the tree to accept is not a proper encoding of a strategy σ_0. That is,

$$\alpha = Q^*(\{\top\} \times V)^\omega \cup \Big\{\eta \in \mathrm{IRuns}(\mathcal{T}_\mathcal{G}) \cap \{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega \mid \Big(\eta|_V \in \mathcal{O}_0 \vee \bigvee_{i=1}^{k} \big(\eta|_V \notin \mathcal{O}_i \wedge \varphi_{\exists dev}(i,\eta)\big)\Big) \wedge \bigwedge_{i \in \lim_W(\eta)} \eta|_V \in \mathcal{O}_i\Big\}$$

Lemma 9. *Let take a path $\pi \in (Q \cap \{q_0\}(2^\Omega \times 2^\Omega \times V))^\omega$ of a run in $\mathcal{T}_\mathcal{G}$. Then, each loop on π has only one value on states for the sets W and D.*

Proof. Let take a path $\pi = xq'yq'z$ of a run in $\mathcal{T}_\mathcal{G}$. Because of the definition of $\delta_\mathcal{U}$, $\pi' = xq'(yq')^\omega$ is also a valid path of a run in $\mathcal{T}_\mathcal{G}$. Suppose that there are two consecutive states in y such that a player is removed/added from/to W in the second state compared to the previous one. Then, there are also two consecutive states in y such that it is added/removed to/from W. This contradicts the fact that π' is a valid path of a run in $\mathcal{T}_\mathcal{G}$ since we could do more than one addition of a player to W. The, there is no change on W on a loop. Also, because of the monotonicity of D, we prove that the value of D remains unchanged along a cycle using the same argument. ∎

From tree automata to two-player games. We now study the complexity of testing emptiness of the language defined by $\mathcal{T}_\mathcal{G}$, for the different winning objectives of this paper. Classically, non-deterministic tree automata emptiness is reduced to solving a two-player zero sum game between Eve, who constructs a tree and a run on this tree, and Adam, whose goal is to prove that the run is non-accepting, by choosing directions in the tree and falsifying the acceptance condition.

Formally, remind that the alphabet is $\Sigma = V \cup \{*_i \mid 1 \leqslant i \leqslant k\} \cup \{\bot\}$ and for a function $f : V \to Q$, we denote by $\mathsf{Range}(f)$ its range. We construct a zero-sum two-player game $\mathcal{G}_\mathcal{T} = \langle V_E, V_A, E', q_0, \mathcal{O}\rangle$ where $V_E = Q$, $V_A = \{\mathsf{Range}(f) \mid \exists q \in Q, l \in \Sigma, f \in \delta(q,l)\}$. Then, the transition relations is defined for all $q \in Q$, all $P \in V_A$, by $(q,P) \in E'$ if there exists $l \in \Sigma$ and $f \in \delta(q,l)$ s.t. $P = \mathsf{Range}(f)$, and $(P,q) \in E'$ if $q \in P$. In other words, to go from q to P, Eve chooses a symbol α and a function $f : V \to Q$ in $\delta(q,\alpha)$. Then, Adam chooses a direction in V, but since he wants to construct a sequence of states not in α, one only needs to remember $\mathsf{Range}(f)$. Adam then picks a state in that set. Finally, Eve's objective is then the set $\mathcal{O} = \{\pi = v_1w_1v_2w_2\cdots \in (V_EV_A)^\omega \mid v_1v_2\cdots \in \alpha\}$.

Proposition 1. *Eve has a winning strategy in $\mathcal{G}_\mathcal{T}$ iff $\mathcal{L}_\alpha(\mathcal{T}_\mathcal{G}) \neq \varnothing$.*

Complexity of solving two-player games. The game $\mathcal{G}_\mathcal{T}$ has linear size in the size of $\mathcal{T}_\mathcal{G}$. A precise analysis of the time complexity of solving $\mathcal{G}_\mathcal{T}$ gives upper bounds to non-cooperative rational synthesis problem.

For safety, reachability, Büchi and co-Büchi winning objectives, we exploit the monotonicity of the sets W and D (the fact that only a polynomial number in k of different sets W and D can be met along a play), to show that if Eve can win the game $\mathcal{G}_\mathcal{T}$, then she can win in a polynomial number of steps (in the size of the original game $\mathcal{G}$), in the sense that she wins iff she can enforce, in a polynomial number of steps, to visit a state q she has already visited and which forms a *good cycle* (the notion of good cycle depends on the winning condition of $\mathcal{G}_{\mathcal{T}}$). In other words, $\mathcal{G}_{\mathcal{T}}$ reduces to a finite duration game with a polynomial number of steps (this kind of reduction is known as *first-cycle game* in the literature [2]). This game is not constructed explicitly, but solved on-the-fly by a PTime alternating algorithm. This gives a PSpace upper-bound for NCRSP.

For Muller conditions, we the polynomial reduction to first-cycle game doesn't work. Therefore, we transform $\mathcal{G}_{\mathcal{T}}$ into a two-player zero-sum parity game with an exponential number of states but a polynomial number of priorities, which can be solved in Exptime (in the size of $\mathcal{G}$). This reduction is based on the *Last Appearance Record (LAR)* [16, 26], which allows us to identify states in V appearing infinitely often. More details on the exact complexity for each type of winning condition are given in the following.

5.1 Safety

In the case of safety objectives $(\mathsf{Safe}(S_i))_{i\in\Omega}$ the winning condition in the game $\mathcal{G}_{\mathcal{T}}$ can be checked by keeping an extra set of players $I \subseteq \Omega$ with the following semantics: if the play is in a state (q, I) and some history $h = q_1 w_1 q_2 w_2 ... q_l$ and $i \in I$, then Player i lost the play, i.e., there is a position $s \leqslant l$ s.t. $q_s|_V \notin S_i$.

Initially, $I = \varnothing$ and it is updated as follows. If a player i belongs to I, then $i \in I$ also in the successor nodes. Otherwise, whenever the game goes in a state $q = (W, D, v)$ such that $v \notin S_i$ for some $i \in \Omega$, then $i \in I$. Then, if eventually there is a player $i \in W \cap I$, the only next state of the game is $\perp$(losing state for Eve). The last situation appears when it is made a wrong guess for a good deviation for some player. Then, if the play never go to the node $\perp$ we are sure that all the players from the set W win. We don't need to keep the information about the players in I if a state in $\{\perp\} \cup (\{\top\} \times V)$ is reached.

Formally, the game $\mathcal{G}_{\mathcal{T}}$ is as follows: $V_E = (Q\backslash(\{\perp\} \cup (\{\top\} \times V))) \times 2^{\Omega} \cup \{\perp\} \cup (\{\top\} \times V)$ and $V_A = \{\mathsf{Range}(f) \mid \exists q \in Q, l \in \Sigma, f \in \delta(q, l)\} \times 2^{\Omega}$ and the transition relation as in the general construction of the game(since I is deterministically updated). Then, because of the monotonicity of the sets W, D and I, Eve's winning condition simplifies to a Büchi condition. $\mathcal{O} = \mathsf{Buchi}(F^S)$ where

$$F^S = (\{\top\} \times V) \cup \{(W, D, I, v) \in Q \mid 0 \notin I\} \cup \{(W, D, I, v) \in Q \mid D \cap I \neq \varnothing\}$$

Intuitively, the first set corresponds to the branches of the tree t_{σ_0} that don't correspond to plays in $\mathcal{G}$ compatible with the strategy σ_0. For the plays compatible to σ_0, the Büchi condition asks that Player 0 never belongs to the set I(therefore wins) or there is a player i for which was guessed a good deviation but loses in the current play ($i \in D \cap I$).

Definition 2. *Given the two-players zero-sum game $\mathcal{G}_{\mathcal{T}}$, we define the first cycle two-player zero-sum game $\mathcal{G}^f_{\mathcal{T}}$ over the same game arena as $\mathcal{G}_{\mathcal{T}}$ where each play ends after the first cycle on Eve's states. Then, a play $\pi = xqyq$ in $\mathcal{G}^f_{\mathcal{T}}$ is winning for Eve if either $q \in \{\top\} \times V$ or $q = (W, D, I, v)$ such that either $0 \notin I$ or $D \cap I \neq \varnothing$.*

Note that because of the monotonicity of I and D, this means that either all the Eve's states are such that $0 \notin I$ or there is some player i that lost but he had a profitable deviation.

Lemma 10. *All the plays of the game $\mathcal{G}_{\mathcal{T}}^{f}$ are of polynomial length in the size of the initial game $\mathcal{G}$.*

Proof. Since D and I are monotone, there are at most $|\Omega| + 1$ different values that they can take on a path of $\mathcal{G}_{\mathcal{T}}$. Also, in the set W we can have at most one addition and one removal for each player $i \in \Omega$ and hence $2|\Omega| + 1$ different values for W. Therefore, along a play π there are at most $r = 1 + (2|\Omega| + 1) \cdot (|\Omega| + 1)^2 \cdot |V|$ different states. The, since all the plays in $\mathcal{G}_{\mathcal{T}}^{f}$ stop after the first cycle, the length of each play is of at most $r + 1$ states since there is only one state that appears twice. Therefore, all plays in $\mathcal{G}_{\mathcal{T}}^{f}$ have polynomial length in Ω and V of the initial play $\mathcal{G}$. ∎

Proposition 2. *Eve has a winning strategy in the game $\mathcal{G}_{\mathcal{T}}$ iff she has a winning strategy in the first cycle game $\mathcal{G}_{\mathcal{T}}^{f}$.*

Proof. From right to left, if Eve has a winning strategy σ_E^f in $\mathcal{G}_{\mathcal{T}}^{f}$, for all σ_A^f a strategy for Adam, $out(\sigma_E^f, \sigma_A^f) = xqyq$ either is such that $q \in \{\top\} \times V$ or $q = (W, D, I, v)$ s.t. ($0 \notin I$ or $I \cap D \neq \varnothing$).

We define now the strategy σ_E of Eve in $\mathcal{G}_{\mathcal{T}}$ as $\sigma_E(hq) = \sigma_E^f(h'q)$ s.t. h' is h from which are removed all cycles and prove that σ_E is winning for Eve in $\mathcal{G}_{\mathcal{T}}$. Let π be a play compatible with σ_E. Then, by the definition of σ_E, we can decompose π in $\pi = \pi_1\pi_2\pi_3...$ such that each π_j is a suffix of a play π'_j compatible with σ_E^f in $\mathcal{G}_{\mathcal{T}}^{f}$. If all π_j on π satisfy $0 \notin I$ on the last state (resp. if it belongs to $\{\top\} \times V$), then also π will satisfy $\Box(0 \notin I)$(because I is monotone)(resp. $\pi|_{V_E} \in Q^*(\{\top\} \times V)^\omega$) and then Eve wins. Otherwise, if there is j such that π_j ends in a state $q = (W, D, I, v)$ s.t. $I \cap D \neq \varnothing$, because of the monotonicity of I and D(Lemma 9), all the states of Eve in the continuations of the game will satisfy $I \cap D \neq \varnothing$ and then Eve wins.

Now, from left to right, if there is no winning strategy for Eve in $\mathcal{G}_{\mathcal{T}}^{f}$, by determinacy, there is a winning strategy σ_A^f for Adam such that $\forall \sigma_E^f$ of Eve, either $out(\sigma_E^f, \sigma_A^f)$ contains $\perp$ (has a suffix in $(\{\perp\})^*$) or it doesn't contain $\perp$, but $out(\sigma_E^f, \sigma_A^f) = xqyq$ such that $q = (W, D, I, v)$ with $0 \in I$ and $I \cap D = \varnothing$.

Let σ_A be the strategy of Adam in $\mathcal{G}_{\mathcal{T}}$ defined as $\sigma_A(hq) = \sigma_A^f(h'q)$ where h' is h from which are removed all cycles. We prove that σ_A is winning for Adam in the game $\mathcal{G}_{\mathcal{T}}$.

Let π be a play compatible with σ_A. By definition of σ_A, we can decompose π in $\pi = \pi_1\pi_2\pi_3...$ such that each π_j is a suffix of a play π'_j compatible with σ_A^f in $\mathcal{G}_{\mathcal{T}}^{f}$. If all π_j are such that they don't contain $\perp$ but they end in a state $q = (W, D, I, v)$ such that $0 \in I$ and $I \cap D = \varnothing$, because of the monotonicity of I and D(Lemma 9), $0 \in I$ in all states of Eve in $\pi_{j'>j}$ on π and since $I \cap D = \varnothing$, it means that all the states of π will satisfy it and therefore $I \cap D = \varnothing$ appears a finite number of times which means that Adam wins. Otherwise, if there is a π_j that ends in $\perp$, then by definition of the game arena(induced by the transition relation in $\mathcal{T}_L$) all $\pi_{j'>j}$ have Eve's states equal to $\perp$ which is again winning for Adam since they visit a finite number of times states in F^S. ∎

Theorem 6. *Deciding the existence of a solution for the non-cooperative synthesis in multiplayer Safety games is in* PSPACE.

Proof. Thanks to Lemma 10 and Proposition 2, to decide the existence of a solution for the non-cooperative synthesis in multiplayer Safety games $\mathcal{G}$ is equivalent to solve the two-player zero-sum finite game $\mathcal{G}^f_{\mathcal{T}}$ that has all the plays of polynomial size in the size of the game $\mathcal{G}$. This can be done in PSPACE using an alternating Turing machine running in PTIME. ∎

5.2 Reachability

For the reachability objectives $(\mathsf{Reach}(R_i))_{i\in\Omega}$, we have the same approach as in the case of safety objectives but with a new meaning for the newly introduced set. In this case, we keep a set $J \subseteq \Omega$ with the following semantics: if the play is in a state (q, J) at some history $h = q_1w_1q_2w_2...q_l$ and $i \in J$, then Player i won in the current play, i.e., there is a position $s \leqslant l$ s.t. $q_s|_V \in R_i$.

Initially, $J = \varnothing$ and it is updated as follows. Whenever a player belongs to the set J, this remains true for the successor nodes. Otherwise, whenever the game goes in a state $q = (W, D, v)$ such that $v \in R_i$ for some $i \in \Omega$, then i is added to the set J. Note that along a play, the set J is monotone since there are only additions of new players.

The formal definition of the game is the same as in the case of Safety objectives, but with the later semantics for the introduced set of players. Then, the Eve's winning condition translates to the Büchi objective $\mathcal{O} = \mathsf{Buchi}(F^R)$ where

$$F^R = (\{\top\} \times V) \cup \{(W, D, J, v) \in Q \mid W \subseteq J \text{ and } (0 \in J \text{ or } D\backslash J \neq \varnothing)\}$$

Since the sets J and D are monotone and also W is establishing after at most $2k$ changes, a play π satisfies the winning condition $\mathcal{O}$ iff $\pi \models \Diamond\Box(W \subseteq J \wedge (0 \in J \vee D\backslash J \neq \varnothing))$. Then, we define the first cycle game as follows:

Definition 3. *Given a two-player zero-sum game $\mathcal{G}_{\mathcal{T}}$, we define the first cycle two-player zero-sum game $\mathcal{G}^f_{\mathcal{T}}$ over the same game arena as $\mathcal{G}_{\mathcal{T}}$ where each play ends after the first cycle. Then, a play $\pi = xqyq$ in $\mathcal{G}^f_{\mathcal{T}}$ is winning for Eve if either $q \in \{\top\} \times V$ or $q = (W, D, J, v)$ such that $W \subseteq J$ and either $0 \in J$ or $D\backslash J \neq \varnothing$.*

Lemma 11. *All the plays of $\mathcal{G}^f_{\mathcal{T}}$ are of polynomial length in the size of the initial game $\mathcal{G}$.*

Proof. The argument is the same as in the case of Safety games since the set L is replaced by the set J in the Reachability case having the same monotonic property. ∎

Proposition 3. *Eve has a winning strategy in the game $\mathcal{G}_{\mathcal{T}}$ iff she has a winning strategy in the first cycle game $\mathcal{G}^f_{\mathcal{T}}$.*

Proof. From right to left, let take a winning strategy σ_E^f of Eve in the game $\mathcal{G}_{\mathcal{T}}^f$. We define Eve's the strategy σ_E in $\mathcal{G}_{\mathcal{T}}$ as $\sigma_E(hq) = \sigma_E^f(h'q)$ where h' is obtained from h by removing all the loops. We prove that σ_E is winning for Eve in $\mathcal{G}_{\mathcal{T}}$.

Let π be a play compatible with σ_E. By the definition of σ_E, we can decompose π in $\pi = \pi_1\pi_2\pi_3...$ s.t. π_j is a suffix of a play π'_j in $\mathcal{G}_{\mathcal{T}}^f$ (π'_j is obtained from $\pi_1\pi_2...\pi_j$ by removing all the cycles from $\pi_1\pi_2...\pi_{j-1}$). Since σ_E^f is winning for Eve, the last state of all π_j are either in $\{\top\} \times V$ or are s.t. $W \subseteq J$ and ($0 \in J$ or $D\backslash J \neq \varnothing$). Therefore we see infinitely often states from F^R and Eve wins in $\mathcal{G}_{\mathcal{T}}$.

In the other direction, if Eve doesn't have a winning strategy in $\mathcal{G}_{\mathcal{T}}^f$, by determinacy, there is a winning strategy σ_A^f for Adam in $\mathcal{G}_{\mathcal{T}}^f$ such that $\forall \sigma_E^f$ of Eve, $out(\sigma_E^f, \sigma_A^f) = \pi'$ such that either π' contains $\perp$ (there is a suffix in $(\{\perp\})^\omega$) or $\pi' = aqyq$ s.t. $q = (W, D, J, v)$ with $W \nsubseteq J$ or ($0 \notin J$ and $D\backslash J = \varnothing$).

Now we define the strategy σ_A for Adam in $\mathcal{G}_{\mathcal{T}}$ as $\sigma_A(hq) = \sigma_A^f(h'q)$ where h' is obtained from h by removing all the cycles and prove that it is winning for Adam in $\mathcal{G}_{\mathcal{T}}$.

Let π be a play in $\mathcal{G}_{\mathcal{T}}$ compatible with σ_A. From the definition of σ_A, we can decompose it as $\pi = \pi_1\pi_2\pi_3...$ where each π_j is the suffix of a play π'_j in $\mathcal{G}_{\mathcal{T}}^f$.

If there is one π_j that contains a $\perp$ in one position, then all the following states equal $\perp$ by the definition of $\delta_{\mathcal{T}}$ and then Adam wins. Now, if there is no state equal to $\perp$ on π, then since σ_A^f is winning in $\mathcal{G}_{\mathcal{T}}^f$, all π_j end in a state that either satisfy $W \nsubseteq J$ or ($0 \notin J$ and $D\backslash J = \varnothing$). Suppose by contradiction that there are two π_{j_1} and π_{j_2} such that appear infinitely often in π and π_{j_1} ends in a state q_1 with $W \nsubseteq J$ and π_{j_2} ends in a state q_2 that satisfies $W \subseteq J$ and $0 \notin J$ and $D\backslash J = \varnothing$. Also, if we take the plays $\pi'_{j_1} = x_1q_1y_1q_1$ and $\pi'_{j_2} = x_2q_2y_2q_2$ whose suffixes π_{j_1} and π_{j_2} are, then x_1q_1 is a prefix of x_2q_2 or vice versa(otherwise they don't both appear infinitely often on π). Then, between q_1 and q_2 all the players that belong to $W\backslash J$ have to be added(if x_1q_1 is a prefix of x_2q_2) or remover (otherwise) from W. But in Lemma 9 we saw that this is not possible. Therefore, on π either all but a finite number of π_j are ending on a state satisfying $W \nsubseteq J$, or all but a finite number of π_j satisfy $0 \notin J$ and $D\backslash J = \varnothing$ on the last state. In addition, from the definition of $\delta_{\mathcal{T}}$, from one position on, the values of W,D and J are unchanged. Therefore, From one position on, all states on π satisfy either $W \notin J$ or ($0 \notin J$ and $D\backslash J \neq \varnothing$) and in both cases Adam wins. ■

Theorem 7. *Deciding the existence of a solution for the non-cooperative synthesis problem in multiplayer Reachability games is in* PSPACE.

Proof. The result is thanks to Lemma 11 and Proposition 3 and the fact that the finite duration game can be solved in PSPACE using an alternating Turing machine running in PTIME. ■

5.3 Büchi

Consider that the objective of Player i are given as Büchi sets $F_i \subseteq V$, $0 \leqslant i \leqslant k$. Therefore, a sequence $v_0v_1v_2... \in V^\omega$ belongs to $\mathcal{O}_i$ iff it satisfies the $LTL[\mathcal{G}]$ formula $\Box\Diamond F_i$ where F_i

is an atomic proposition true in a state v iff $v \in F_i$. Then, the winning condition for Eve in the game $\mathcal{G}_{\mathcal{T}}$ is $\mathcal{O} = \{\pi = q_1 w_1 q_2 w_2 \cdots \in (V_E V_A)^\omega \mid \pi\restriction_{V_E} = q_1 q_2 \cdots \in \alpha\}$ where

$$\alpha = Q^*(\{\top\} \times V)^\omega \cup \Big\{\eta \in \mathrm{IRuns}(\mathcal{T}_{\mathcal{G}}) \cap \{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega \mid \\ \Big(\eta|_V \models \Box\Diamond F_0 \vee \bigvee_{i=1}^{k} \big(\eta|_V \not\models \Box\Diamond F_i \wedge i \in \lim_D(\eta)\big)\Big) \wedge \bigwedge_{i \in \lim_W(\eta)} \eta|_V \models \Box\Diamond F_i\Big\}$$

In order to check the satisfaction of α along the plays of the game $\mathcal{G}_{\mathcal{T}}$, we introduce two counters $c_W \in \Omega \cup \{-1\}$ and $c_D \in \Omega \cup \{-1\}$ in the states of the states of the game help to monitor the appearance of states in F_i that make the formula $\varphi_W = \bigwedge_{i \in \lim_W(\eta)} \eta|_V \models \Box\Diamond F_i$ and $\varphi_D = \bigvee_{i=1}^{k} \big(\eta|_V \not\models \Box\Diamond F_i \wedge i \in \lim_D(\eta)\big)$ true. The goal in using this counters is to write the formulas φ_W and φ_D as Büchi and respectively co-Büchi conditions. Intuitively, whenever c_W or c_D equal to i means that it is expected a state belonging to F_i.

In order to correctly update the counters, we need to keep in the states of Adam also the last previous state belonging to Eve. Note that by doing this, the size of the game remains exponential in the size of the initial game and the number of different states along a play remains polynomial in the size of initial game.

Formally, given the game $\mathcal{G}_{\mathcal{T}} = (V_E, V_A, E', q_0, \mathcal{O})$, we define the new game $\tilde{\mathcal{G}}_{\mathcal{T}} = (\tilde{V}_E, \tilde{V}_A, \tilde{E}, \tilde{q}_0, \tilde{\mathcal{O}})$ with $\tilde{q}_0 = (q_0, -1, -1)$, $\tilde{V}_E = V_E \times (\Omega \cup \{-1\}) \times (\Omega \cup \{-1\})$, $\tilde{V}_A = V_A \times V_E \times (\Omega \cup \{-1\}) \times (\Omega \cup \{-1\})$ and the transition relation is the smallest set $\tilde{E}$ such that

- $((q_E, c_W, c_D), (q_A, q_E, c_W, c_D)) \in \tilde{E}$ iff $(q_E, q_A) \in E'$ for $q_E \in V_E$ and $q_A \in V_A$
- $((q_A, q_E, c_W, c_D), (q'_E, c'_W, c'_D)) \in \tilde{E}$ iff $(q_A, q'_E) \in E'$ where

$$c'_W = \begin{cases} -1 & \text{if } q_E = \perp \text{ or } q_E \in \{\top\} \times V \\ & \quad \text{or } q'_E = (W', D', v') \text{ s.t. } W' = \varnothing \\ min\{(c_W + l) \bmod k \in W' \mid l > 0\} & \text{if } q_E = (W, D, v), q'_E = (W', D', v') \\ & \text{s.t. } W' \neq \varnothing \wedge (v \in F_{c_W} \vee c_D \in W' \backslash W \vee W = \varnothing) \\ c_W & \text{otherwise} \end{cases}$$

and

$$c'_D = \begin{cases} -1 & \text{if } q_E = \perp \text{ or } q_E \in \{\top\} \times V \\ & \quad \text{or } q'_E = (W', D', v') \text{ s.t. } D' = \varnothing \\ min\{(c_D + l) \bmod k \in D' \mid l > 0\} & \text{if } q_E = (W, D, v), q'_E = (W', D', v') \\ & \text{s.t. } D' \neq \varnothing \text{ and } (v \in F_{c_D} \text{ or } D = \varnothing) \\ c_D & \text{otherwise} \end{cases}$$

Note that a play $\pi \in (\tilde{V}_E, \tilde{V}_A)^\omega$ is in $\mathsf{Plays}(\tilde{\mathcal{G}}_{\mathcal{T}})$ iff π' obtained from π by projecting away c_W and c_D (and q_E from Adam's nodes) is in $\mathsf{Plays}(\mathcal{G}_{\mathcal{T}})$. Intuitively, the role of the counters c_W and c_D wait for the first occurrence of a state such that $v \in F_{c_W}$ and $v \in F_{c_D}$ respectively. If $q_E = \perp$ or $W = \varnothing$ (or $D = \varnothing$), then $c_W = -1$ ($c_D = -1$ resp.).

Lemma 12. *For a play π in $\tilde{\mathcal{G}}_\mathcal{T}$, if $\pi|_{V_E} \in \{q_0\}\{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega$ and $W_p = lim_W(\pi|_{V_E})$, then*

$$\pi \models \bigwedge_{i \in W_p} \Box\Diamond F_i \text{ iff } \pi \models \Box\Diamond H_w$$

$$\text{where } H_w = \{(W, D, v, c_W, c_D) \in \tilde{V}_E \mid W = \varnothing \vee (v \in F_{c_W} \wedge c_W = min\{i \in W\})\}$$

Proof. Note that since we considered $\pi|_{V_E} \in \{q_0\}\{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega$, states in $\{\top\} \times V$ are not reached by π. Let first treat the case $W_p = \varnothing$, then $\pi \models true$ and also $\pi \models \Box\Diamond H_w$ since the only set W visited infinitely often is $\varnothing$.

Now, consider $W_p \neq \varnothing$. If $\pi \models \bigwedge_{i \in W_p} \Box\Diamond F_i$, then $inf(\pi|_{c_W}) = W_p$ by the construction of the game $\tilde{\mathcal{G}}_\mathcal{T}$(whenever a final state is reached, the counter c_W is increased to the next player in W). Then, we see an infinite number of times final states of the "smallest" player in W, i.e., states in which $v \in F_{c_W}$ and $c_W = min\{i \in W\}$ and then $\pi \models \Box\Diamond H_w$.

In the other direction, if we see an infinite number of times states with $v \in F_{c_W}$ and $c_W = min\{i \in W\}$, because of the construction of the game, once we reach a final state with $v \in F_{c_W}$ the counter c_W is increased to the next player in W and so on. Therefore, between tho states having $v \in F_{c_W}$ and $c_W = min\{i \in W\}$, the projection on the direction visits all the states F_i where $i \in W$. Then, since W is stabilizes to W_p and since we visit an infinite number of times final states with $v \in F_{c_W}$ where $c_W = min\{i \in W_p\}$, it means that we visit infinitely often the final states of all players in $\lim_W(\pi)$ and therefore $\pi \models \bigwedge_{i \in W_p} \Box\Diamond F_1$. ■

Lemma 13. *For a play π in $\tilde{\mathcal{G}}_\mathcal{T}$, if $\pi|_{V_E} \in \{q_0\}\{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega$ and $D_p = lim_D(\pi|_{V_E})$, then*

$$\pi \models \bigvee_{i \in D_p} \Diamond\Box\neg F_i \text{ iff } \pi \models \Diamond\Box\neg H_d$$

$$\text{where } H_d = \{(W, D, v, c_W, c_D) \in V_E \mid D = \varnothing \vee (v \in F_{c_D} \wedge c_D = min\{i \in D\})\}$$

Proof. The proof is similar to the proof of the previous Lemma. Indeed, if $D_p = \varnothing$, then by the monotonicity of D, all the states along the play are such that $D = \varnothing$ and then both $\pi \models \bigvee_{i \in D_{p+1}} \Diamond\Box\neg F_i$ and $\pi \models \Diamond\Box\neg H_d$ are false.

If $D_p \neq \varnothing$, and $\pi \models \bigvee_{i \in D_p} \Diamond\Box\neg F_i$, then there is a player that sees finitely often F_i. Therefore, from the construction of the game $\tilde{\mathcal{G}}_\mathcal{T}$, there is a player that blocks the cycling through all the values in D for the counter c_D and for that player, there are eventually seen only non-final states. That is, there are not seen infinitely often states in which $v \in F_{c_D}$ and $c_D = min\{i \in D\}$ and therefore $\pi \models \Diamond\Box\neg H_d$.

In the other direction, if π visits finitely often states in which $v \in F_{c_D}$ and $c_D = min\{i \in D\}$, from the definition of the game $\tilde{\mathcal{G}}_\mathcal{T}$, either F_{c_D} is seen a finite number of times along π, or there is a $i \in D$ that blocks the cycling of c_D through all the values in D. Therefore, there is a player $i \in D$ such that $\pi \models \Diamond\Box\neg F_i$. ■

Using Lemmas 12 and 13, if we note $H_\top = \{q \in \tilde{V}_E \mid q|_{V_E} \in \{\top\} \times V\}$ and $H_0 = \{(W, D, v, c_W, c_D) \in V_E \mid v \in F_0\}$ we can rewrite Eve's winning condition in the game $\tilde{\mathcal{G}}_\mathcal{T}$ as

$$\tilde{\mathcal{O}} = \{\pi \in (\tilde{V}_E\tilde{V}_A)^\omega \mid \pi \models \Diamond\Box H_\top \vee \big((\Box\Diamond H_0 \vee \Diamond\Box\neg H_d) \wedge \Box\Diamond H_w\big)\}$$

Note that by asking to see infinitely often H_w, there are also avoided the states having $q =\perp$.

The above formula $\Diamond\Box H_\top \vee \big((\Box\Diamond H_0 \vee \Diamond\Box\neg H_d) \wedge \Box\Diamond H_w\big)$ is equivalent to $\Diamond\Box H_\top \vee (\Box\Diamond H_0 \wedge \Box\Diamond H_w) \vee (\Diamond\Box H_d \wedge \Box\Diamond H_w)$. Then, to be able to check if a path satisfies $\Box\Diamond H_0 \wedge \Box\Diamond H_w$, we need to introduce a counter $b \in \{0, 1\}$ in the states of the game $\tilde{\mathcal{G}}_\mathcal{T}$ as follows.

Definition 4. *We define a game* $\hat{\mathcal{G}}_\mathcal{T} = (\hat{V}_E = \tilde{V}_E \times \{0, 1\}, \hat{V}_A = \tilde{V}_A \times \{0, 1\}, (\tilde{q}_0, 0), \hat{E}, \hat{\mathcal{O}})$ *where*

- $((q, b), (q', b')) \in \hat{E}$ *iff* $(q, q') \in \tilde{E}$ *and* $b' = \begin{cases} 0 & \text{if } b = 1 \text{ and } q \in H_w \\ 1 & \text{if } b = 0 \text{ and } q \in H_0 \\ b & \text{else} \end{cases}$
- $\hat{\mathcal{O}} = \{\pi \in (\hat{V}_E\hat{V}_A)^\omega \mid \pi \models \Diamond\Box H'_\top \vee \Box\Diamond H'_0 \vee (\Diamond\Box H'_d \wedge \Box\Diamond H'_w)\}$ *where* $H'_\top = H_\top \times \{0, 1\}$, $H'_0 = H_0 \times \{0\}$, $H'_d = H_d \times \{0, 1\}$ *and* $H'_w = H_w \times \{0, 1\}$.

Note that form the way of defining the transition relation in $\hat{\mathcal{G}}_\mathcal{T}$, the updates of the counter b are deterministic. Therefore, for each path π in $\tilde{\mathcal{G}}_\mathcal{T}$, there is a unique corresponding path π' in $\hat{\mathcal{G}}_\mathcal{T}$ s.t. by projecting away the counter b from π' we obtain the path π.

Lemma 14. *Let* $\pi' \in \mathsf{Plays}(\hat{\mathcal{G}}_\mathcal{T})$ *and* $\pi \in \mathsf{Plays}(\tilde{\mathcal{G}}_\mathcal{T})$ *obtained from* π' *by projecting away the counter* b. *Then*

$$\pi \models \Diamond\Box H_\top \vee \big((\Box\Diamond H_0 \vee \Diamond\Box\neg H_d) \wedge \Box\Diamond H_w\big) \text{ iff } \pi' \models \Diamond\Box H'_\top \vee \Box\Diamond H'_0 \vee (\Diamond\Box\neg H'_d \wedge \Box\Diamond H'_w)$$

Proof. If $\pi \models \Box\Diamond H_0 \wedge \Box\Diamond H_w$, then from the definition of $\hat{\mathcal{G}}_\mathcal{T}$, the path π' will contain infinitely many changes of the value of the counter b by reaching alternatively states in $H_0 \times \{0\}$ and $H_w \times \{1\}$. Therefore, $\pi' \models H'_0$.

If $\pi \not\models \Box\Diamond H_0 \wedge \Box\Diamond H_w$ but $\pi \models \Diamond\Box\neg H_d \wedge \Box\Diamond H_w$, it means that there is a position from which the counter b remains unchanged along π' and $\pi' \models \Diamond\Box\neg H'_d \wedge \Box\Diamond H'_w$. Also, if $\pi \models \Diamond\Box H_\top$, it means that eventually a state containing $\top$ is reached and the game remains in $H_\top$. Then, from the construction of $\hat{\mathcal{G}}_\mathcal{T}$, the same property holds along π' and therefore $\pi' \models \Diamond\Box H'_\top$.

In the other direction, if $\pi' \models \Box\Diamond H'_0$, from the definition of the transition relation in the game $\hat{\mathcal{G}}_\mathcal{T}$, π' has to visit infinitely often both $H_0 \times \{0\}$ and $H_w \times \{1\}$ and therefore $\pi \models \Box\Diamond H_0 \wedge \Box\Diamond H_w$.

If $\pi' \models \Diamond\Box\neg H'_d \wedge \Box\Diamond H'_w$, by projecting away the counter b, the play $\pi \models \Diamond\Box\neg H_d \wedge \Box\Diamond H_w$ since $H'_d = H_d \times \{0, 1\}$ and $H'_w = H_w \times \{0, 1\}$. The same argument works if $\pi \models \Diamond\Box H'_\top$. ■

Parity game Further, we express Eve's winning condition $\hat{\mathcal{O}} = \{\pi \in \mathsf{Plays}(\hat{\mathcal{G}}_{\mathcal{T}}) \mid \pi \models \Diamond\Box H'_{\top} \vee \Box\Diamond H'_0 \vee (\Diamond\Box\neg H'_d \wedge \Box\Diamond H'_w)\}$ using a parity condition where the priority function pr defined as follows:

$$pr(q \in \hat{V}_E) = \begin{cases} 0 & \text{if } q \notin H'_{\top} \wedge q \in H'_0 \\ 1 & \text{if } q \notin H'_{\top} \wedge q \notin H'_0 \wedge q \in H'_d \\ 2 & \text{if } q \notin H'_{\top} \wedge q \notin H'_0 \wedge q \notin H'_d \wedge q \in H'_w \\ 3 & \text{if } q \notin H'_{\top} \wedge q \notin H'_0 \wedge q \notin H'_d \wedge q \notin H'_w \\ 4 & \text{if } q \in H'_{\top} \end{cases}$$

For states belonging to Adam, we just put priority $pr(q \in \hat{V}_A) = 6$, so that they have no influence.

Lemma 15. *Let* $\pi \in \mathsf{Plays}(\hat{\mathcal{G}}_{\mathcal{T}})$. *Then*

$$\pi \models \Diamond\Box H'_{\top} \vee \Box\Diamond H'_0 \vee (\Diamond\Box\neg H'_d \wedge \Box\Diamond H'_w) \textit{ iff } min\{pr(q)) \mid q \in inf(\pi)\} \textit{ is even}$$

Proof. If $\pi \models \Diamond\Box H'_{\top}$, then eventually the game remains in the set $H'_{\top}$ and the only priority appearing infinitely often along π is 4 which is even. Otherwise, If $\pi \models \Box\Diamond H'_0$, then H'_0 is visited infinitely often $min\{pr(q) \mid q \in inf(\pi)\} = 0$ which is even.

If $\pi \not\models \Diamond\Box H'_{\top} \vee \Box\Diamond H'_0$ but $\pi \models \Diamond\Box\neg H'_d \wedge \Box\Diamond H'_w$, then from a position on H'_d is never visited and we see infinitely often H'_w along π. This means that $min\{pr(q) \mid q \in inf(\pi)\} = 2$ which is even.

In the other direction, if $min\{pr(q) \mid q \in inf(\pi)\} = 0$, then $\pi \models \Box\Diamond H'_0$. Otherwise, if $min\{pr(q) \mid q \in inf(\pi)\} = 2$, it means that π visits an infinite number of times H'_w and only a finite number of times H'_d (otherwise $min\{pr(q) \mid q \in inf(\pi)\} = 1$). That is, $\pi \models \Diamond\Box\neg H'_d \wedge \Box\Diamond H'_w$. Finally, if $min\{pr(q) \mid q \in inf(\pi)\} = 4$, it means that eventually only states in $H'_{\top}$ are visited (otherwise the smallest priority appearing infinitely often is not 4) and therefore $\pi \models \Diamond\Box H'_{\top}$. ∎

Definition 5. *Given a two-player zero-sum parity game $\hat{\mathcal{G}}_{\mathcal{T}}$ with the priority function pr, we define the first cycle two-player zero-sum game $\mathcal{G}^f_{\mathcal{T}}$ over the same game arena as $\hat{\mathcal{G}}_{\mathcal{T}}$ where each play ends after the first cycle. Then, a play $\pi = xqyq$ in $\mathcal{G}^f_{\mathcal{T}}$ is winning for Eve iff $min\{pr(yq[j]) \mid 0 \leqslant j < |yq|\}$ is even.*

Lemma 16. *All the plays in $\mathcal{G}^f_{\mathcal{T}}$ are of polynomial length in the size of the initial game $\mathcal{G}$.*

Proof. It follows from the monotonicity of D and quasi-monotonicity of W and the fact that $1 \leqslant c_W, c_D \leqslant k$ and $b \in \{0, 1\}$. ∎

Proposition 4. *Eve has a winning strategy in the game $\hat{\mathcal{G}}_{\mathcal{T}}$ iff she has a winning strategy in the first cycle game $\mathcal{G}^f_{\mathcal{T}}$.*

Proof. From right to left, if Eve has a winning strategy σ_E^f in $\mathcal{G}_\mathcal{T}^f$, then for all strategies σ_A^f of Adam, $out(\sigma_E^f, \sigma_A^f) = xqyq$ is such that $min\{pr(yq[j]) \mid 0 \leqslant j < |yq|\}$ is even. We define Eve's strategy σ_E in $\hat{\mathcal{G}}_\mathcal{T}$ as $\sigma_E(hq) = \sigma^f(h'q)$ where h' is obtained from h by removing all the loops. We prove that σ_E is winning for Eve in $\hat{\mathcal{G}}_\mathcal{T}$.

Let π be a play compatible with σ_E. By the definition of σ_E, we can decompose π in $\pi = \pi_1\pi_2\pi_3...$ s.t. π_j is a suffix of a play π'_j in $\mathcal{G}_\mathcal{T}^f$ compatible with σ_E^f. Moreover, there is a decomposition of the suffixes π_j such that by reordering the resulting fragments of all suffixes appearing infinitely often, we obtain an infinite sequence of loops being suffixes of plays in $\mathcal{G}_\mathcal{T}^f$ compatible with σ_E^f preceded by a finite prefix. Then, since σ_E^f is winning in $\mathcal{G}_\mathcal{T}^f$, all the loops have the minimum priority even and therefore the minimum priority appearing infinitely often in π is even and π is winning for Eve.

On the other direction, if there is no winning strategy for Eve in $\mathcal{G}_\mathcal{T}^f$, by determinacy, there is a winning strategy σ_A^f for Adam such that $\forall \sigma_E^f$, $out(\sigma_E^f, \sigma_A^f) = xqyq$ is such that $min\{pr(yq[j]) \mid 0 \leqslant j < |yq|\}$ is odd. Let σ_A be the strategy of Adam in $\hat{\mathcal{G}}_\mathcal{T}$ defined as $\sigma_A(hq) = \sigma_A^f(h'q)$ where h' is obtained from h by removing all cycles. We prove that σ_A is winning for Adam in $\hat{\mathcal{G}}_\mathcal{T}$.

Let π be a play compatible with σ_A. Doing the same reasoning as before, we can decompose π and rearrange the components such that we obtain an infinite sequence of loops being suffixes of plays in $\mathcal{G}_\mathcal{T}^f$ compatible with σ_A^f preceded by a finite prefix. Then, since σ_A^f is winning in $\mathcal{G}_\mathcal{T}^f$ for Adam, all the loops have the minimum priority odd and then the priority that appears infinitely often in π is odd. Therefore, π is winning for Adam in $\hat{\mathcal{G}}_\mathcal{T}$. ∎

Theorem 8. *Deciding the existence of a solution for the non-cooperative synthesis problem in multiplayer Büchi games is in* PSPACE.

Proof. It follows directly from Lemma 16 and Proposition 4 and the fact that the finite duration game $\mathcal{G}_\mathcal{T}^f$ can be solved in PSPACE using an alternating Turing machine running in PTIME. ∎

5.4 Co-Büchi

Let consider now the case when the winning conditions for each player i is given as a Co-Büchi set $F_i \subseteq V$. Then, the winning condition for Eve in the game $\mathcal{G}_\mathcal{T}$ is $\mathcal{O} = \{\pi = q_1w_1q_2w_2\cdots \in (V_EV_A)^\omega \mid \pi\restriction_{V_E} = q_1q_2\cdots \in \alpha\}$ where

$$\alpha = Q^*(\{\top\} \times V)^\omega \cup \Big\{\eta \in \mathrm{IRuns}(\mathcal{T}_\mathcal{G}) \cap \{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega \mid \Big(\eta|_V \models \Diamond\Box\neg F_0 \vee \bigvee_{i=1}^{k} \big(\eta|_v \not\models \Diamond\Box\neg F_i \wedge i \in \lim_D(\eta)\big)\Big) \wedge \bigwedge_{i\in\lim_W(\eta)} \eta|_V \models \Diamond\Box\neg F_i)\Big\}$$

Lemma 17. *For a play π in $\mathcal{G}_\mathcal{T}$, if $\pi\restriction_{V_E} \in \{q_0\}\{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega$ and $W_p = lim_W(\pi\restriction_{V_E})$, then*

$$\pi\restriction_{V_E} \models \bigwedge_{i\in W_p} \Diamond\Box\neg F_i \textit{ iff } \pi\restriction_{V_E} \models \Diamond\Box\neg H_w$$
$$\textit{where } H_w = \{(W, D, v) \in V_E \mid v \in \bigcup_{i\in W} F_i\}.$$

Proof. This holds because all the states that appear an infinite number of times along π have $W = W_p$(W stabilizes along a play) and visiting a finite number of sets a finite number of time is equivalent to visiting their union a finite number of times. ∎

Lemma 18. *For a play π in $\mathcal{G}_\mathcal{T}$, if $\pi\restriction_{V_E} \in \{q_0\}\{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega$ and $D_p = lim_D(\pi\restriction_{V_E})$, then*

$$\pi\restriction_{V_E} \models \bigvee_{i\in D_p} \Box\Diamond F_i \textit{ iff } \pi\restriction_{V_E} \models \Box\Diamond H_d$$
$$\textit{where } H_d = \{(W, D, v) \in V_E \mid v \in \bigcup_{i\in D} F_i\}.$$

Proof. This holds because all the states that appear an infinite number of times along π have $D = D_p$(D stabilizes along a play) and visiting one set among $F_1, ..., F_r$ an infinite number of times is equivalent to visiting their union an infinite number of times. ∎

Let now $H_0 = \{(W, D, v) \in V_E \mid v \in F_0\}$ and $H_\top = \{q \in V_E \mid q \in \{\top\} \times V\}$. Then, Using Lemmas 17 and 18, we get that Eve's winning condition is equivalent to

$$\mathcal{O} = \{\pi \in (V_E V_A)^\omega \mid \pi\restriction_{V_E} \models \Diamond\Box H_\top \vee \big((\Diamond\Box\neg H_0 \vee \Box\Diamond H_d) \wedge \Diamond\Box\neg H_w\big)\}$$

Let $I = H_0 \cup H_w$. Then the formula $\Diamond\Box H_\top \vee \big((\Diamond\Box\neg H_0 \vee \Box\Diamond H_d) \wedge \Diamond\Box\neg H_w\big)$ is equivalent to $\Diamond\Box H_\top \vee \Diamond\Box\neg I \vee (\Box\Diamond H_d \wedge \Diamond\Box\neg H_w)$. Further, from the construction, a play cannot alternate states in $H_\top$ and I (once in $H_\top$, all the future states are in the same set). Therefore, we can define the set $J = H_\top \cup (V_E \backslash I)$ and equivalently write $\Diamond\Box J$ instead of $\Diamond\Box H_\top \vee \Diamond\Box\neg I$.

Definition 6. *Given the two player game $\mathcal{G}_\mathcal{T}$, we define Eve's winning condition as a parity condition with the priority function $pr : (V_E \cup V_A) \to \{1, .., 6\}$ with*

$$pr(q \in V_E) = \begin{cases} 1 & \textit{if } q \notin J \wedge q \in H_w \\ 2 & \textit{if } q \notin J \wedge q \in H_d \wedge q \notin H_w \\ 3 & \textit{if } q \notin J \wedge q \notin H_d \wedge q \notin H_w \\ 4 & \textit{if } q \in J \end{cases}$$

For states belonging to Adam, we just put priority $pr(q \in V_A) = 6$, so that they have no influence.

Lemma 19. *Let $\pi \in \mathsf{Plays}(\mathcal{G}_\mathcal{T})$. Then,*

$$\pi\restriction_{V_E} \models \Diamond\Box H_\top \vee \big((\Diamond\Box\neg H_0 \vee \Box\Diamond H_d) \wedge \Diamond\Box\neg H_w\big) \textit{ iff } min\{pr(q) \mid q \in inf(\pi)\} \textit{ is even}$$

Proof. If $\pi\restriction_{V_E} \models \Diamond\Box H_\top$, then $\pi\restriction_{V_E} \models \Diamond\Box J$ and therefore from a position on we see only states in J which means that $min\{pr(q) \mid q \in inf(\pi)\} = 4$ which is even. If $\pi \models$

$\Diamond\Box\neg H_0 \wedge \Diamond\Box\neg H_w$, then $\pi \models \Diamond\Box\neg I$ and then $\pi\restriction_{V_E} \models \Diamond\Box J$ which means as before $min\{pr(q) \mid q \in inf(\pi)\} = 4$.

Otherwise, if $\pi \models \Box\Diamond H_d \wedge \Diamond\Box\neg H_w$, then from a point on H_w doesn't appear in π and H_d appears infinitely often which means that $min\{pr(q) \mid q \in inf(\pi)\} = 2$ which is even.

In the other direction, if $min\{pr(q) \mid q \in inf(\pi)\} = 4$,because of the construction, it means that from a position on, either appears only states in $H_\top$ or the set $I = H_0 \cup H_w$ doesn't appear in π(otherwise $min\{pr(q) \mid q \in inf(\pi)\} < 4$) and therefore $\pi \models \Diamond\Box H_\top \vee \Diamond\Box\neg H_0 \wedge \Diamond\Box\neg H_w$.

If $min\{pr(q) \mid q \in inf(\pi)\} = 2$, it means that H_d appears infinitely often along π and H_w appears a finite number of times(otherwise $min\{pr(q) \mid q \in inf(\pi)\} = 1$). Therefore, $\pi \models \Box\Diamond H_d \wedge \Diamond\Box\neg H_w$. ■

Now, having the two player parity game $\mathcal{G}'_\mathcal{T}$ over the same game arena as $\mathcal{G}_\mathcal{T}$ and objective $\mathsf{Parity}(pr)$, we define the first cycle game $\mathcal{G}^f_\mathcal{T}$ ans we did in Definition 5 in the case of Buchi games whose plays have polynomial length in the size of the initial game $\mathcal{G}$ and solve it in alternating PTIME.

Theorem 9. *Deciding the existence of a solution for the non-cooperative synthesis problem in the multiplayer co-Büchi games is in* PSPACE

5.5 Muller

We now study the complexity of solving $\mathcal{G}_\mathcal{T}$ when the original game $\mathcal{G}$ has Muller conditions $\mathsf{Muller}(\mu_i)$ for the $k+1$ players. For Muller conditions, he winning condition for Eve in the game $\mathcal{G}_\mathcal{T}$ is $\mathcal{G}_\mathcal{T}$ is $\mathcal{O} = \{\pi = q_1 w_1 q_2 w_2 \cdots \in (V_E V_A)^\omega \mid \pi\restriction_{V_E} = q_1 q_2 \cdots \in \alpha\}$ where

$$\alpha = Q^*(\{\top\} \times V)^\omega \cup \Big\{\eta \in \mathrm{IRuns}(\mathcal{T}_\mathcal{G}) \cap \{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega \mid$$
$$\Big(\eta|_V \in \mathsf{Muller}(\mu_0) \vee \bigvee_{i=1}^{k} \big(\eta|_V \notin \mathsf{Muller}(\mu_i) \wedge i \in \lim_D(\eta)\big)\Big) \wedge \bigwedge_{i \in \lim_W(\eta)} \eta|_V \in \mathsf{Muller}(\mu_i))\Big\}$$

We transform $\mathcal{G}_\mathcal{U}$ into a two-player zero-sum parity game with an exponential number of states but a polynomial number of priorities, which can be solved in $\mathsf{Exptime}$ (in the size of $\mathcal{G}$). This reduction is based on the *Last Appearance Record (LAR)* [16, 26], which allows us to identify states in V appearing infinitely often.

LAR For the given set of states V, we define the deterministic transition system LAR_V that records the most recent states in V that appeared along an execution. We let $P(V)$ the set of permutations of V, which we denote by words of length n over alphabet V such that each element of V appears exactly once. We define a deterministic finite automaton $LAR_V = (P(V) \times \{0, \ldots, |V|-1\}, (m_0, h_0), \to)$, $m_0 = v_1 \ldots v_n$ and $h_0 = 1$, and $(m, h) \xrightarrow{v} (x_1 x_2 v, |x_1|)$ where $m = x_1 v x_2$ for some $x_1, x_2 \in V^*$.

Let (m, h) be a state of LAR_V. Then, h is called the *hit*, representing the position from which the last state v is taken and moved to the back, and the states after position

h on in m are the most recent states v seen along the path, called *recent states*. Then, let $\xi = v_0 v_1 v_2 ...$ be an infinite sequence of states in V. A path in LAR_V on ξ is a infinite sequence $\tau(\xi) = (m_0, h_0)(m_1, h_1)(m_2, h_2)...$ such that $(m_0, h_0) \in M$ and $\forall j \geqslant 1, (m_j, h_j) = \delta((m_{j-1}, h_{j-1}), v_{j-1})$. Let h_{min} be the smallest hit appearing infinitely often along $\tau(\xi)$. Then, the set of vertexes v in m situated after position h_{min} is always the same from some point on and is equal to $inf(\xi)$, i.e., the sequence of subsets $(\{m_i[r] \mid r \geqslant h^{\pi}_{min}\})_{i \geqslant 0}$ eventually stabilizes to $\inf(\xi)$.

Parity game Now we can define the parity game $\tilde{\mathcal{G}}_{\mathcal{T}}$ by taking the product of $\mathcal{G}_{\mathcal{T}}$ and LAR_V as follows.

Definition 7. *Given the two-players zero-sum game $\mathcal{G}_{\mathcal{T}} = (V_E, V_A, q_0, E', \mathcal{O})$ with $V_{\mathcal{T}} = V_E \uplus V_A$ and the deterministic transition system LAR_V defined as above, we define the parity game $\tilde{\mathcal{G}}_{\mathcal{T}} = (\tilde{V} = \tilde{V}_E \uplus \tilde{V}_A, \tilde{q}_0, \tilde{E}, pr)$ where $\tilde{V} = V_{\mathcal{T}} \times M$, $\tilde{q}_0 = (q'_0, (m_0, h_0))$ and the set E' is defined by*

- $((q_E, (m, h)), (q_A, (m, h))) \in E'$ *iff* $(q_E, q_A) \in E$
- $((q_A, (m, h)), (q_E, (m', h'))) \in E'$ *iff* $\begin{cases} (q_A, q_E) \in E \\ (m', h') = \delta((m, h), q_E|_V) \text{ if } q_E \notin \{\bot\} \cup (\{\top\} \times V) \\ (m', h') = (m, h) \text{ if } q_E \in \{\bot\} \cup (\{\top\} \times V) \end{cases}$

Finally, the priority function $pr : \tilde{V} \to \{0, ..., 2|V| + 2\}$ is defined as follows: $pr(\bot, m, h) = 1$, $pr(q_\top, m, h) = 0$ for $q_\top \in \{\top\} \times V$ and

$$pr((W, D, v), (m, h)) = \begin{cases} 2h & \text{if } \forall i \in W \{m[r] \mid r \geqslant h\} \models \mu_i \text{ and} \\ & (\{m[r] \mid r \geqslant h\} \models \mu_0 \text{ or } \exists i \in D \text{ s.t. } \{m[r] \mid r \geqslant h\} \models \neg\mu_i) \\ 2h + 1 & \text{else} \end{cases}$$

For states whose first component belongs to Adam, we just put priority $2|V| + 2$, so that they have no influence.

Let π a play in $\mathcal{G}_{\mathcal{T}}$. Note that according to the definition above, there is a unique play π' in $\tilde{\mathcal{G}}_{\mathcal{T}}$ such that by projecting away the LAR construction along π', we obtain the play π. Also, the LAR component changes only on states belonging to Eve which helps verifying the winning condition $\mathcal{O}$.

Lemma 20. *Let π a play in $\mathcal{G}_{\mathcal{T}}$ and the corresponding play π' in $\tilde{\mathcal{G}}_{\mathcal{T}}$. Then,*

$$\pi \in \mathcal{O} \textit{ iff } \pi' \in \mathsf{Parity}(pr)$$

Proof. Let h_{min} be the smallest hit appearing infinitely often along π'. As remarked before, $\{m[r] \mid r \geqslant h_{min}\} = \inf(\pi' \upharpoonright_{V_E}|_V) = \inf(\pi \upharpoonright_{V_E}|_V)$.

Let $\pi \in \mathsf{Muller}(\mu_0)$. This is equivalent to $\{m[r] \mid r \geqslant h_{min}\} \models \mu_0$. If $D_p = \lim_D(\pi \upharpoonright_{V_E})$, the fact that $\exists i \in D_p$ s.t. $\inf(\pi \upharpoonright_{V_E}|_V) \notin \mathsf{Muller}(\mu_i)$, since $\pi \upharpoonright_{V_E}|_D = \pi' \upharpoonright_{\tilde{V}_E}|_D$ and $\{m[r] \mid r \geqslant h_{min}\} = \inf(\pi' \upharpoonright_{V_E}|_V) = \inf(\pi \upharpoonright_{V_E}|_V)$, is equivalent with $\{m[r] \mid r \geqslant h_{min}\} \not\models \mu_i$.

Also, considering $W_p = \lim_W(\pi\restriction_{V_E})$, the property that $\forall i \in W_p, \pi\restriction_{V_E}|_V \in \mathsf{Muller}(\mu_i)$ translates to $\inf(\pi\restriction_{V_E}|_V) = \{m[r] \mid r \geqslant h_{min}\} \models \mu_i$.

From the above, $\pi \in \mathcal{O}$ iff the smallest priority appearing infinitely often when hitting h_{min} is $2h_{min}$ which is even and therefore $\pi' \in \mathsf{Parity}(pr)$. ∎

Theorem 10. *The non-cooperative multiplayer Muller rational synthesis problem is in* EXPTIME.

Proof. The complexity comes from the fact that the game $\tilde{\mathcal{G}}_{\mathcal{T}}$ is a two-player Parity game with exponential number of states, but with polynomial number of priorities which can be solved in EXPTIME since parity games can be solved in PTime in the number of states and exponential in the number of priorities [18, 22], so proving the result. ∎

5.6 Lower Bounds for the Non-Cooperative Setting

We finally provide some lower bounds to the complexity of the non-cooperative rational synthesis problem. Clearly, given an objective $\mathcal{O} \in \{\mathsf{Reach}, \mathsf{Safe}, \dots\}$ the corresponding non-cooperative rational synthesis problem is at least as hard as the 0-sum two-players game with the same objective[5]. We show that indeed, for each objective $\mathcal{O} \in \{\mathsf{Reach}, \mathsf{Safe}, \mathsf{Buchi}, \mathsf{coBuchi}, \mathsf{Street}, \mathsf{Rabin}, \mathsf{Parity}, \mathsf{Muller}\}$, a PSPACE lower bound applies to the corresponding non-cooperative rational synthesis problem. The result is obtained by reduction from the quantified boolean formula (QBF) problem.

Theorem 11. *For each* $\mathcal{X} \in \{\mathsf{Reach}, \mathsf{Safe}, \mathsf{Buchi}, \mathsf{coBuchi}, \mathsf{Street}, \mathsf{Rabin}, \mathsf{Parity}, \mathsf{Muller}\}$*, the non-cooperative rational synthesis problem in multiplayer* $\mathcal{X}$*-games is* PSPACE-H.

Proof. By reduction from QBF. Let $\psi = \exists x_1 \forall x_2 ... \exists x_m \gamma(x_1, x_2, ..., x_m)$ be a QBF in 3CNF with k clauses $C_1, C_2, ..., C_k$.

Given $\mathcal{X} \in \{\mathsf{Reach}, \mathsf{Safe}, \mathsf{Buchi}, \mathsf{coBuchi}, \mathsf{Street}, \mathsf{Rabin}, \mathsf{Parity}, \mathsf{Muller}\}$, we build a multiplayers $\mathcal{X}$-game $\mathcal{G}_\psi$ such that ψ is true if and only if $\mathcal{G}_\psi$ admits a solution to the non-cooperative rational synthesis problem. The game $\mathcal{G}_\psi$ involves $2m + 2$ players $\Omega = \{A, B, P_{10}, P_{11}, P_{20}, P_{21}, \dots, P_{m0}, P_{m1}\}$. Intuitively, player A (the system) controls the existential variables, while player B (first player of the environment) controls the universal ones. More precisely, $\mathcal{G}_\psi$ is played on the arena $\mathcal{A}_\psi$ obtained as follows (cfr. Figure 4, where the round nodes are owned by Player A, the diamond ones by player B, and the rectangular ones by players $P_{10}, P_{11} \dots, P_{m0}, P_{m1}$ as specified below.).

For each existential (resp. universal) variable x_i the arena $\mathcal{A}_\psi$ contains a node x_i controlled by the system (resp. by player B). For each node $x_i, 1 \leqslant i < m$, the arena $\mathcal{A}_\psi$ contains the edges $(x_i, 0_{x_i}), (x_i, 1_{x_i}), (0_{x_i}, x_{i+1}), (1_{x_i}, x_{i+1})$, where the vertex 0_{x_i} (resp. 1_{x_i}) intuitively represents the value $val(x_i) = 1$ (resp. $val(x_i) = 0$) for the variable x_i.

[5] In fact, given the zero-sum two-player game $\mathcal{G}$ where Player 0 has the objective γ, it is sufficient to consider a non-zero-sum game on the same arena where Player 0 (the system) has objective γ and Player 1 (the environment) wins in any case.

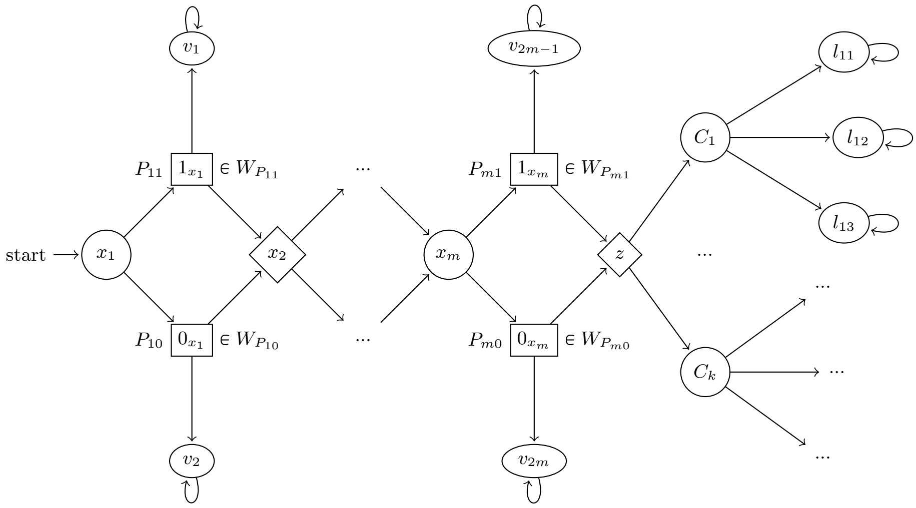


Fig. 4: Non-cooperative Büchi: Reduction from QBF

For each $1 \leqslant i \leqslant m$, the value-node 1_{x_i} (resp. 0_{x_i}) is controlled by player P_{i1} (resp. P_{i0}) and has a further edge leading to the self-loop over the node v_{2i-1} (resp. v_{2i}), owned by the system. The value nodes $1_{x_m}, 0_{x_m}$ (for the last variable x_m) are then connected to a vertex z controlled by player B, where intuitively player B can choose a clause (i.e. an edge $(z, C_i), 1 \leqslant i \leqslant k$, out from z). Each clause-node C_i, controlled by the system, has three outgoing edges toward the terminal nodes (with self-loops) l_{i1}, l_{i2}, l_{i3}, one for each literal in C_i.

Given the arena described above for the $\mathcal{X}$-game $\mathcal{G}_\psi$, the objectives of players are properly designed so that the following conditions are satisfied:

(i) Given v_i, where $1 \leqslant i \leqslant 2m$, each lasso-path ending up into v_i is winning for each player in the game.
(ii) Given l_{ij}, where $1 \leqslant i \leqslant k$ and $1 \leqslant i \leqslant 3$, each lasso-path ending up into l_{ij} is winning for each player in the game but the system (i.e. player A) and the player P_{hb}, where:

$$(l_{ij} = x_h \wedge b = 1) \vee (l_{ij} = \neg x_h \wedge b = 0)$$

Note that condition (i) implies that for each $1 \leqslant i \leqslant m$ and $b \in \{0, 1\}$, the vertex b_{x_i} belongs to the winning region $W_{P_{ib}}$ of player P_{ib} (since it is controlled by P_{ib} and leads the play to a lasso-path ending up either into v_{2i} or into v_{2i-1}, winning for P_{ib}).

We claim that the formula ψ is true iff there is a solution for the non-cooperative rational synthesis problem in the multiplayer $\mathcal{X}$-game $\mathcal{G}_\psi$.

Assume that ψ is true. Then, the existential player has a winning strategy in the QBF game associated to ψ. Player A (the system) can play in $\mathcal{G}_\psi$ according to such a strategy up to the node z, ensuring a configuration of variables such that all the clauses are satisfied. Then, from each clause node C_i, $1 \leqslant i \leqslant k$, player A can choose one literal l_{ij}, $1 \leqslant j \leqslant 3$, that makes true C_i and go to the corresponding node l_{ij}. Each path π on $\mathcal{G}_\psi$ in the outcome of such a strategy for player A is either winnng for player A (since it does not reach z, i.e. is a lasso-path to some v_i, where $1 \leqslant i \leqslant 2m$) or it ends up into a node l_{ij} such that: either $l_{ij} = x_h$ and π passed trough 1_{x_h} or $l_{ij} = \neg x_h$ and π passed trough 0_{x_h} (i.e. either player P_{h1} respectively player P_{h0} doesn't play in NE since he looses but passed trough his winning region).

Otherwise, assume that ψ is false. Then, the universal player has a winning strategy σ in the QBF game associated to ψ. Consider a strategy profile (for the environment) where player B plays according to σ and each player P_{ib}, for $1 \leqslant i \leqslant m, b \in \{0, 1\}$, plays to the next variable-node (or to z). Once in z, player B can choose a clause C_i that is false according to the instantiation of variables along the path followed so far. Therefore, for any choice of the system from C_i, the play will be loosing for the system and in NE for each player of the environment. Indeed, let l_{ij} be the choice of player A from C_i. Then, there is an index h such that $l_{ij} = x_h$ or $l_{ij} = \neg x_h$. In the first case player P_{h1} looses but he could not avoid it (since the play did not pass trough 1_{x_h} and he never played) and each other player in the environment wins. In the second case P_{h0} looses but he could not avoid it (since the play did not pass trough 0_{x_h} and he never played) and each other player in the environment wins.

To conclude the proof, we just need to show that the objectives of the players in the $\mathcal{X}$-game $\mathcal{G}_\psi$ can be defined in order to satisfy the conditions (i) and (ii) above, for each $\mathcal{X} \in \{\mathsf{Reach}, \mathsf{Safe}, \mathsf{Buchi}, \mathsf{coBuchi}, \mathsf{Street}, \mathsf{Rabin}, \mathsf{Parity}, \mathsf{Muller}\}$, .

- $\mathcal{X} = \mathsf{Reach}$. It is sufficient to define the reachability objective for each player as follows: $R_A = \{v_i \mid 1 \leqslant i \leqslant 2m\}$, $R_B = V$, and for each $h \in \{1, \ldots, m\}$, the reachability objective of P_{h1} is $R_{P_{h1}} = \{v_i \mid 1 \leqslant i \leqslant 2m\} \cup \{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3 \text{ and } l_{ij} \neq x_h\}$ and the one for P_{h0} is $R_{P_{h0}} = \{v_i \mid 1 \leqslant i \leqslant 2m\} \cup \{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3 \text{ and } l_{ij} \neq \neg x_h\}$.
- $\mathcal{X} = \mathsf{Safe}$. It is sufficient to define the safety objective for each player as follows. $S_A = V \backslash \{(l_{ij} \mid 1 \leqslant i \leqslant k \wedge 1 \leqslant j \leqslant 3\}$, $S_B = V$, and for each $h \in \{1, \ldots, m\}$: $S_{P_{h1}} = V \backslash \{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3 \text{ and } l_{ij} = x_h\}$ and $S_{P_{h0}} = V \backslash \{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3 \text{ and } l_{ij} = \neg x_h\}$
- $\mathcal{X} = \mathsf{Büchi}$. It is sufficient define the Büchi objectives of the players as follows: $F_A = \{v_i \mid 1 \leqslant i \leqslant 2m\}$, $F_B = V$, and for each $h \in \{1, \ldots, m\}$: $F_{P_{h1}} = V \backslash \{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3 \text{ and } l_{ij} = x_h\}$ and $F_{P_{h0}} = V \backslash \{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3 \text{ and } l_{ij} = \neg x_h\}$
- $\mathcal{X} = \mathsf{co\text{-}Büchi}$. It is sufficient define the co-Büchi objectives of the players as follows. The co-Büchi objective of the system is $\{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3\}$. The co-Büchi objective of Player B is $\varnothing$. For each $h \in \{1, \ldots, m\}$, the co-Büchi objective of P_{h1} is $\{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3 \text{ and } l_{ij} = x_h\}$ and the objective of P_{h0} is $\{l_{ij} \mid 1 \leqslant i \leqslant k,\, 1 \leqslant j \leqslant 3 \text{ and } l_{ij} = \neg x_h\}$
- $\mathcal{X} \in \{\mathsf{parity}, \mathsf{Street}, \mathsf{Rabin}\}$. The PSPACE-hardness for the non-cooperative rational synthesis for parity, Streett and Rabin comes directly from the fact that we can easily express any Büchi condition as a parity, Streett or Rabin condition.

– The PSPACE-h for the non-cooperative strategy synthesis problem for Muller games follows from the fact that 0-sum two player Muller games are PSPACE-h (we could clearly also define proper Muller objectives in $\mathcal{G}_\psi$ is order to satisfy conditions (i),(ii)).

■

Theorem 12. *The non-cooperative rational synthesis problem is* PSPACE-C *in multiplayers $\mathcal{X}$-games, $\mathcal{X} \in \{\mathsf{Safe}, \mathsf{Reach}, \mathsf{Buchi}, \mathsf{coBuchi}\}$. It is* PSPACE-H *in multiplayers $\mathcal{X}$-games, $\mathcal{X} \in \{\mathsf{Parity}, \mathsf{Street}, \mathsf{Rabin}, \mathsf{Muller}\}$)*

6 Fixed number of players

Until now, we considered the general case where the number of agents consisting the environment is not fixed. In the following, we restrict the rational synthesis problem to the particular case when the number of players is fixed (let say k players) and study the complexity of solving the rational synthesis problem in both cooperative and non-cooperative cases.

6.1 k-fixed Cooperative Setting

Upper Bounds The following theorems prove the upper bounds to k-fixed CRSP provided in the second column of Table 1. In particular, Theorem 13 provides PTIME procedures to solve the k-fixed CRSP w.r.t. safety, reachability, Büchi and coBüchi objectives. Theorem 14 provides a UP∩ coUP algorithm for parity k-fixed CRSP.

Theorem 13. *The k-fixed* CRSP *w.r.t. safety, reachability, Büchi and coBüchi objectives is in* PTIME*.*

Proof. As seen in the proof of Lemma 2, there is a solution for cooperative rational synthesis iff there is a path π such that $\pi \models \varphi$ where $\varphi = \varphi_0 \wedge \phi^{\mathcal{G}}_{0\text{Nash}}$.

Given the above, the PTIME algorithm for the winning conditions $(\mathsf{X_i})_\mathsf{i}$ first labels in polynomial time each node in the winning region W_i of each Player i, $0 \leqslant i \leqslant k$, by W_i. Also, for the winning conditions of each player, we label in polynomial time nodes belonging to S_i (resp. R_i and F_i) with the corresponding atomic proposition v_{S_i} (resp. v_{R_i} and v_{F_i}). Note that since the number of players is fixed, also the number of atomic propositions introduces is and the formula $\varphi = \varphi_0 \wedge \phi^{\mathcal{G}}_{0\text{Nash}}$ becomes a constant formula(depends only on the number of players).

Then, to check the existence of a path such that $\pi \models \varphi$, we build a constant size Büchi word automaton $\mathcal{B}_\varphi$(since the LTL formula φ is constant for k constant), take the product with the game arena and check in polynomial time the emptiness of the resulting automaton.

■

Theorem 14. *The k-fixed* CRSP *w.r.t. parity objectives is in* UP∩ coUP*.*

Proof. Given $0 \leqslant i \leqslant k$, let $p_i : V \to \{0, ..., 2n\}$ be the priority function for Player i, where $n = |V|$. We need to provide a UP∩coUP algorithm to check if $\mathcal{G}$ admits a path $\pi \models \phi$, where $\phi = parity(p_0) \wedge \bigwedge_{1 \leqslant i \leqslant k}(parity(p_i) \vee \Box\neg W_i)$ and $parity(p_i) = \bigvee_{j=0}^{(k_i-1)/2}(\Box\Diamond C_{2j} \wedge \bigwedge_{p<2j} \Diamond\Box\neg C_p)$ encodes the winning condition for Player i, where C_j is an atomic proposition corresponding to color j. First, notice that if $\mathcal{G}$ admits a path π such that $\pi \models \phi$, then $\mathcal{G}$ admits a path $\pi^* = \pi_1^*\pi_2^*$ such that $\pi^* \models \phi$, $|\pi_1^*| \leqslant n$ and π_2^* is a loop of size $(k+2) \cdot n$. In fact, given $\pi \models \phi$ we can build π^* as follows. If $\pi \models \Box\neg W_i$ for each $1 \leqslant i \leqslant k$, then π^* can be obtained by cutting π as soon as the first node repeats on it. Otherwise, for each $1 \leqslant i \leqslant k$ such that $\pi \models parity(p_i)$, let $m_i \in \{0 \ldots n\}$ be the least priority w.r.t. p_i occurring infinitely often on π. For each node v, label v by the vector $\bar{a} = (a_0 \ldots, a_k)$, where for each $0 \leqslant i \leqslant k$, $\bar{a}[i] = m_i$ if $p_i(v) = m_i$, and $\bar{a}[i] = \bot$ otherwise. Given m_0, there is a vertex u that appears infinitely often on π and is assigned infinitely often a label that have m_0 (rather than $\bot$) at index 0. Pick the first occurrence of such an u and color it by green. Repeat the above procedure for each $1 \leqslant i \leqslant k$ such that $\pi \models parity(p_i)$ (starting from the last green node) in order to recover a green node on π for each $0 \leqslant i \leqslant k$ such that $\pi \models parity(p_i)$. Once detected the last green node, cut the remaining path as soon as you find a further occurrence of u. Therefore, you obtain a path $\pi' = \pi_1'\pi_2'$, where π_2' is a loop (from u to u) witnessing that $\pi' \models \phi$. Removing each simple loop on π_1' as well as on each subpath of π_2' withouth green nodes lead to a path $\pi^* = \pi_1^*\pi_2^*$ such that $\pi^* \models \phi$, $|\pi_1^*| \leqslant n$, and π_2^* is a loop of size $(k+2) \cdot n$.

Given the above premises, it is sufficient to design UP∩ coUP algorithm to check if $\mathcal{G}$ admits a path $\pi^* \models \phi$, where $\pi^* = \pi_1^*\pi_2^*$ such that $|\pi_1^*| \leqslant n$ and and π_2^* is a loop of size $(k+2) \cdot n$. The UP algorithm works as follows. For each node v in $\mathcal{G}$, for each $1 \leqslant i \leqslant k$ guess if $v \in W_i$ or $v \notin W_i$. Verify the guess applying the corresponding UP algorithm. If the guess was incorrect, then reject immediately. Otherwise check in NLOGSPACE if $\mathcal{G}$ contains a path $\pi^* \models \phi$, where $\pi^* = \pi_1^*\pi_2^*$ such that $\pi^* \models \phi$, $|\pi_1^*| \leqslant n$ and π_2^* is a loop of size $(k+2) \cdot n$. This is possible by guessing on-the-fly a path π^* and a node u on it where the loop should start, while mantaining (1) for each $0 \leqslant i \leqslant k$, the minimum priority seen along the loop w.r.t p_i (2) for each $0 \leqslant i \leqslant k$, a bit to check if $\Box\neg W_i$ along π^* (2) the lenghts of π_1^*, π_2^* and (3) the node u witnessin that π_2^* is a loop. Infact, since k is a fixed constant, the priorities are bounded by n, and the length of the path is polynomial w.r.t. the size of the graph, the amount of space required is logarithmic w.r.t. the size of the input graph.

A coUP algorithm needs to verify in UP if $\forall\pi.\neg(\pi \models \phi)$. This can be done as follows. For each node v in $\mathcal{G}$, for each $1 \leqslant i \leqslant k$ guess if $v \in W_i$ or $v \notin W_i$. Verify the guess applying the corresponding UP algorithm. If the guess was incorrect, then reject immediately. Otherwise, verify in coNLOGSPACE if $\forall\pi.\neg(\pi \models \phi)$. This amounts to check in NLOGSPACE if $\exists\pi.\pi \models \phi$ that can be done as above. ■

Lower Bounds The following Theorem 15 provides a reduction from two players zero-sum games to k-fixed CRSP, that allows to infer the lower bounds on k-fixed CRSP given in the second column of table 1.

Theorem 15. *Let* $\mathcal{X} \in \{\mathsf{Safety}, \mathsf{Reachability}, \mathsf{Buchi}, \mathsf{coBuchi}, \mathsf{Parity}, \mathsf{Streett}, \mathsf{Rabin}, \mathsf{Muller}\}$. *Given a two-player zero-sum game between player A and B with an objective of type $\mathcal{X}$ for player A, we can construct a multiplayer game with objective of type $\mathcal{X}$ with two players $\Omega = \{0, 1\}$ such that player A does not have a winning strategy in the zero-sum game if and only if the multiplayer game is a positive instance of the CRSP problem.*

Proof. Let $\mathcal{G}$ be a two-players zero-sum game where the protagonist (Player A) has the objective ψ, and so Player B has objective $\neg\psi$. We construct the 2-players CRSP game $\mathcal{G}'$ by considering a copy of $\mathcal{G}$ and two fresh states v and w. The state v is the initial state of $\mathcal{G}'$ and has a transition to the initial state of $\mathcal{G}$ and a transition to w, which is equipped with a self-loop. The environment (Player 1) controls v, w and the states belonging to Player A in $\mathcal{G}$, while the system (Player 0) controls the states belonging to Player B in $\mathcal{G}$. For the winning conditions, Player 0 wins only if the play gets into w (and stays that forever), while the objective of the environment is ψ (i.e. the objective of Player A in $\mathcal{G}$).

$\mathcal{G}'$ is a positive instance of the CRSP problem iff Player 1 playing edge $v \to w$ is a NE. But clearly Player 1 does not have an incentive to deviate if and only if Player A does not have a winning strategy in $\mathcal{G}$ for forcing ψ. ■

Therefore, we obtain:

Corollary 2. *For parity objectives, k-fixed* CRSP *is in* UP$\cap$ coUP *and parity-hard. For Street objectives, k-fixed* CRSP *is* NP-c. *For Rabin objectives k-fixed* CRSP *is* P^{NP} *and* coNP-h. *Finally, for Muller objectives k-fixed CRSP is* Pspace-c.

Proof. The result for parity follows directly from Theorem 14 and Theorem 15. For Street objectives, the upper bound follows from [25], while the lower bound follows from Theorem 15. For Rabin objectives, the upper bound follows from Theorem 4, while the lower bound follows from Theorem 15. The lower bound for Muller games follows from Theorem 15, while the upper bound was already true for an unfixed number of players. ■

Note that a gap remains open for Rabin k-fixed CRSP. In fact, we do not have a coNP algorithm for such a problem. Rather, we only have a P^{NP} procedure to solve it and we do not know whether Rabin k-fixed CRSP is NP-h.

6.2 K-Fixed non-Cooperative setting

We finally prove the upper bounds and the lowe bounds to the complexity of k-fixed NCRSP, reported in the last column of Table 1.

Upper Bounds to k-fixed NCRSP For $\mathcal{O} \in \{\mathsf{Safety}, \mathsf{Reachability}, \mathsf{Buchi}, \mathsf{coBuchi}\}$, a polynomial uppe bound applies, as shown in the following Theorem 16.

Theorem 16. *The problem of deciding the existence of a solution for the non-cooperative rational synthesis for a k-fixed number of players in Safety, Reachability, Büchi and co-Büchi games can be solved in* PTIME.

Proof. In the case of a fixed number of players k, we obtain the polynomial size two-player zero-sum game $\mathcal{G}_{\mathcal{T}}$ and a fixed objective ϕ, where $\phi \in \{\varphi_s, \varphi_r, \varphi_c, \varphi_b\}$ are the formulas characterizing the winning objectives in the case of Safety, Reachability, Büchi and co-Büchi games. That is, $\varphi_s = \Box\Diamond F^S$, $\varphi_r = \Diamond\Box F^R$, $\varphi_b = \Diamond\Box H_\top \vee \big((\Box\Diamond H_0 \vee \Diamond\Box\neg H_d) \wedge \Box\Diamond H_w\big)$, and $\varphi_c = \Diamond\Box H_\top \vee \big((\Diamond\Box\neg H_0 \vee \Box\Diamond H_d) \wedge \Diamond\Box\neg H_w\big)$.

First, we can label in polynomial time the nodes of the game $\mathcal{G}_{\mathcal{T}}$ with atomic propositions atomic propositions F^S(for safety), F^R(for reachability), $H_\top$, H_0, H_d and H_w(for Buchi and co-Buchi respectively defined according the considered condition). Each node is labeled with the atomic proposition corresponding to the set it belongs.

Then, since the formula ϕ is constant over the newly introduced atomic propositions, we get a constant size automaton $\mathcal{A}_\phi$ equivalent to the LTL formula ϕ and by taking the product $\mathcal{A}_\phi \times \mathcal{G}_{\mathcal{T}}$ we obtain a Büchi game that can be solved in polynomial time[12]. ■

The procedure outlined within the proof of Theorem 16 does not yield a polynomial upper bound for the remaining objectives considered in this paper. However, we show that Muller k-fixed NCRSP can be solved in PSPACE (cfr. Theorem 17). This entails a PSPACE upper bound also for k-fixed NCRPS w.r.t. $\mathcal{O} \in \{\mathsf{Parity}, \mathsf{Streett}, \mathsf{Rabin}\}$.

Theorem 17. *The problem of deciding the existence of a solution for the non-cooperative rational synthesis on k-players Müller games, where k is a fixed constant, is in* PSPACE.

Proof. For a fixed number of players k, the game $\mathcal{G}_{\mathcal{T}}$ has size polynomial in the size of the initial game $\mathcal{G}$. Moreover, the objective of Eve in $\mathcal{G}_{\mathcal{T}}$ is equivalent to a Muller condition μ that is polynomial in the size of the game, as we show below. The thesis follows from the fact that $\mathcal{G}$ two-players zero-sum Muller games can be solved in PSPACE.

To conclude the proof, we show how to transform Eve's objective $\mathcal{O}$ when each player has an implicit Muller condition μ_i into an unique equivalent implicit Muller objective μ. Note that we can ignore the states belonging to Adam and define the objective μ only on Eve's states.

First, for each tuple (W, D, v), we consider an atomic proposition $x_{W,D,v}$. Note that since the number of players is fixed, the state space of the game $\mathcal{G}_{\mathcal{T}}$ is polynomial and so is the size of the set of newly introduced atomic propositions. Then, let take $\eta \in \mathrm{IRuns}(\mathcal{T}_{\mathcal{G}}) \cap \{q_0\}(2^\Omega \times 2^\Omega \times V)^\omega$ and the condition $\eta|_V \in \mathsf{Muller}(\mu_0)$. Since the sets W and D stabilize along η, we can equivalently write it as $\eta \in \mathsf{Muller}(\mu_0')$ where

$$\mu_0' = \mu_0[v \leftarrow \bigvee_{W,D} x_{W,D,v}]$$

is the boolean formula where each state v is replaced by a disjunction for all W and D of $x_{W,D,v}$. Further, the condition $\gamma_i = (\eta|_V \notin \mathsf{Muller}(\mu_i) \wedge i \in \lim_D(\eta))$ asks that the Player i belongs to D from a position on, and the Muller condition μ_i is not satisfied. Using again the monotonicity of the sets W and D, we can rewrite the condition γ_i as a Muller condition

$$\mu_i^D = \bigwedge_{\substack{D \subseteq \Omega \\ i \in D}} \Big((\bigvee_{W,v} x_{W,D,v}) \to \neg \mu_i[v \leftarrow \bigvee_{W} x_{W,D,v}] \Big)$$

Intuitively, the formula says that for the set D that appears infinitely often (after stabilization) that contains i, the formula $\neg\mu_i$ holds for some W (that is also fixed after some steps). Similarly, we take the condition $\bigwedge_{i \in \lim_W(\eta)} \eta|_V \in \mathsf{Muller}(\mu_i)$ and write the equivalent Muller condition

$$\mu^W = \bigwedge_{W \subseteq \Omega} \Big((\bigvee_{D,v} x_{W,D,v}) \to \bigwedge_{i \in W} \mu_i[v \leftarrow \bigvee_{D} x_{W,D,v}] \Big)$$

The formula says that for the set W that appears infinitely often, for all the players in this set, the Muller condition μ_i holds for some D.

Finally, the condition that $\eta \in Q^*(\{\top\} \times V)^\omega$ can be expressed using an atomic proposition $x_\top$ that is true only in the states belonging to $\{\top\} \times V$ as $\mu_\top = x_\top$ since once η goes outside $\{\top\} \times V$, it goes to $\perp$ and all the following states equal $\perp$. Therefore, the objective of Eve in the game $\mathcal{G}_\mathcal{T}$ is equivalent to the Muller condition $\mathcal{O} = \mathsf{Muller}(\mu)$ with

$$\mu = \mu_\top \vee \big((\mu_0' \vee \bigvee_{i=1}^{k} \mu_i^D) \wedge \mu^W \big)$$

■

Corollary 3. *The problem of deciding the existence of a solution for the non-cooperative rational synthesis for a k-fixed number of players in Parity, Street and Rabin games is in* PSPACE.

Lower Bounds to *k*-fixed NCRSP We start to note that the reduction from QBF to general NCRSP provided in Theorem 11 does not apply to the case of a fixed number of players, as it requires a number of components for the environment that is linear in the numebe of variables of the given QBF. Clearly, k-fixed Muller NCRSP is PSPACE-H by reduction from the corresponding two player zero-sum games.

The lower bounds for parity k-fixed NCRSP reported in Table 1 have been obtained by reduction from the generalized parity games considered in [11], where the objective is a disjunction (dually, a conjunction) of parity conditions. In particular, we have proven that NCRSP is NP-H (cfr. Theorem 18) on 3-players parity games, and coNP-H (cfr. Theorem 19) on 4-players parity game.

Finally, as listed in Table 1, we could provide a PSPACE lower bound also to Street and Rabin k-fixed NCRSP. This is done in two steps: First a reduction from QBF to zero-sum

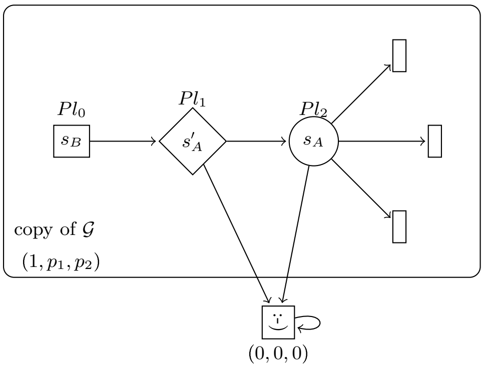


Fig. 5: k-fixed Non-cooperative Parity: NP-hardness

two players Muller games is provided (cfr. the proof of Theorem 20), similar to the one given in [17]. Then, the latter is reduced to a Street (resp. Rabin, cfr. Theorem 21) NCRSP with two players.

Theorem 18. *The problem of deciding the existence of a solution for the non-cooperative Parity synthesis problem in a 3-player game arena is* NP-H.

Proof. We prove the theorem by reduction from the two-player zero-sum game $\mathcal{G} = (V = V_A \uplus V_B, E, v_0, \mathcal{O}_A = parity(p_1) \wedge parity(p_2))$ where player A(protagonist) has as objective an outcome satisfying a conjunction of two parity objectives p_1 and p_2. In [11] was proven that computing the winning region for the antagonist is NP-hard.

W.l.o.g. we consider that the game $\mathcal{G}$ is turn-based and that the initial state belongs to player A. Intuitively, the new 3-player parity game consists in a modified copy of $\mathcal{G}$ by duplicating the states of Player A and adding an extra sink state called $\ddot{\smile}$ where are players are happy with priority function equal to 0.

We define formally the 3-player parity game $\mathcal{G}' = (V' = V_0 \uplus V_1 \uplus V_2, E', v_0, p'_0, p'_1, p'_2)$ where $V_0 = V_B \cup \{\ddot{\smile}\}$, $V_1 = \{v' \mid v \in V_A\}$, $V_2 = V_A$ and E' is defined as the smaller set such that for all $(v_A, v_B) \in E$, then $(v_A, v_B) \in E'$ and for all $(v_B, v_A) \in E$, we have $(v_B, v'_A) \in E'$ and $(v'_A, v_A) \in E'$ and for all $v \in V_1 \cup V_2$, $(v, \ddot{\smile}) \in E'$. A sketch of the game arena is depicted in Figure 5.

Then, we define the parity functions as

- $p'_0(v) = 1$ for all $v \neq \ddot{\smile}$ and $p'_0(\ddot{\smile}) = 0$;
- $p'_1(v) = p_1(v)$ for all $v \in V_B \cup V_A$, $p'_1(v') = p_1(v)$ for $v \in V_A$ and $p'_1(\ddot{\smile}) = 0$;
- $p'_2(v) = p_2(v)$ for all $v \in V_B \cup V_A$, $p'_2(v') = p_2(v)$ for $v \in V_A$ and $p'_2(\ddot{\smile}) = 0$;

We claim that Player A has a winning strategy in $\mathcal{G}$ iff there is no solution to the synthesis problem in the game $\mathcal{G}'$. Indeed, if there is a strategy σ_A in $\mathcal{G}$ such that $parity(p_1) \wedge parity(p_2)$ holds on all $\pi \in out(\sigma_A)$, in $\mathcal{G}'$ Player 2 can play σ_2 defined as $\sigma_2(h) = \sigma_A(h')$ where h' is the restriction of h on the states in $V_A \cup V_B$. Then, for all

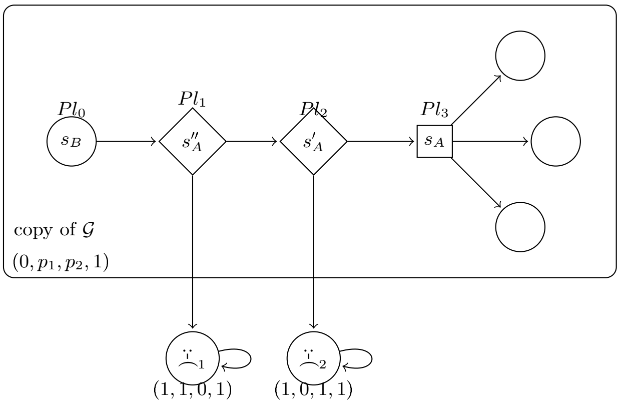


Fig. 6: k-fixed Non-cooperative Parity: co-NP-hardness

σ_0, there is a NE $(\sigma_0, \sigma_1, \sigma_2)$ such that the game stays in the copy of $\mathcal{G}$. Therefore, the system(Player 0) loses and there is no solution for the synthesis problem.

Otherwise, if there is no strategy σ_A to ensure $parity(p_1) \wedge parity(p_2)$ on all the paths compatible with it, it means that there is a strategy σ_B s.t. $\forall \pi \in out(\sigma_B)$, $\pi \models \overline{parity(p_1)} \vee \overline{parity(p_2)}$. That is, there is a strategy σ_0 for Player 0 s.t. at least one of the players 1 and 2 wants to deviate to ☺. Let take $(\sigma_0, \sigma_1, \sigma_2)$ a strategy profile where $\sigma_1(v') = v$ and $\sigma_2(v) \neq$ ☺. If $out(\sigma_0, \sigma_1, \sigma_2) \models \overline{parity(p_1)}$, this is not a NE because Player 1 looses and prefers to go to ☺. Otherwise, if $out(\sigma_0, \sigma_1, \sigma_2) \models \overline{parity(p_2)}$, player 2 looses and prefers ☺ instead of staying in the copy of $\mathcal{G}$. Therefore, all the NE are such that their outcome reaches ☺ and Player 0 wins. This means that σ_0 is a solution to the rational synthesis problem. ■

Theorem 19. *The problem of deciding the existence of a solution for the non-cooperative Parity synthesis problem in a 4-player game arena is co-NP-hard.*

Proof. The proof is done by reducing from two-player zero-sum games where the objective of the protagonist is a conjunction of two parity objectives p_1 and p_2. For this games, in [11] is proven that the protagonist has a winning strategy from a given state is co-NP hard.

The 4-player game is obtained from the game $\mathcal{G}$ by making two extra copies of each node of Player B and adding two extra states $☹_1$ and $☹_2$. We define a 4-player parity game $\mathcal{G}' = (V' = V_0 \uplus V_1 \uplus V_2 \uplus V_3, E', v_0, p'_0, p'_1, p'_2, p'_3)$ where $V_0 = V_A \cup \{☹_1, ☹_2\}$, $V_1 = \{v'' \mid v \in V_B\}$, $V_2 = \{v' \mid v \in V_B\}$, $V_3 = V_B$ and E' is the smaller set such that for all $(v_A, v_B) \in E$, then $\{(v_A, v''_B), (v''_B, v'_B), (v'_B, v_B)\} \subseteq E'$, for all $(v_B, v_A) \in E$, also $(v_B, v_A) \in E'$ and for all $v \in V_1$, $(v, ☹_1) \in E'$ and for all $v \in V_2$, $(v, ☹_2) \in E'$. A sketch of the game arena is depicted in Figure 6.

Then, the parity functions are defined as

– $p'_0(v) = 0$ for all $v \notin \{☹_1, ☹_2\}$ and $p'_0(☹_1) = p'_0(☹_2) = 1$

- $p'_1(v) = p_1(v)$ for all $v \in V_A$, $p'_1(v'') = p'_1(v') = p'_1(v) = p_1(v)$ for all $v \in V_B$, $p'_1(\frownie_1) = 1$ and $p'_1(\frownie_2) = 0$.
- $p'_2(v) = p_2(v)$ for all $v \in V_A$, $p'_2(v'') = p'_2(v') = p'_2(v) = p_2(v)$ for all $v \in V_B$, $p'_2(\frownie_1) = 0$ and $p'_2(\frownie_2) = 1$.
- $p'_3(v) = 1$ for all $v \in V'$.

We claim that there is a winning strategy σ_A for player A in $\mathcal{G}$ iff there is a solution for the rational synthesis problem in $\mathcal{G}'$. If there is a strategy σ_A to satisfy $parity(p_1) \wedge parity(p_2)$, it means that there is a strategy σ_0 for Player 0 defined as $\sigma_0(h) = \sigma_A(h')$ where h' is the restriction of h on the states in $V_A \cup V_E$ s.t. for any strategy σ_3 of Player 3, both Player 1 and Player 2 prefer to play in the copy of $\mathcal{G}$ since they win and in $\frownie_i$ Player i loses. Therefore, all NE have as outputs plays in $\mathcal{G}$ and then Player 0 wins and σ_0 is a solution for the rational synthesis problem.

On the other way, if there is a solution for the rational synthesis problem, all the NE have outputs in the copy of $\mathcal{G}$ which means that $(\sigma_0, \sigma_1, \sigma_2, \sigma_3)$ where $\sigma_1(v'') = v'$ and $\sigma_2(v') = v$ are the only NE. This means that both $parity(p'_1)$ and $parity(p'_2)$ are satisfied for any strategy σ_3. That is, there is a strategy σ_A defined as $\sigma_A(h') = \sigma_0(h)$ where h' is the restriction of h on the states in $V_A \cup V_E$(note there is only one such h by the definition of $\mathcal{G}'$) in $\mathcal{G}$ s.t. for all σ_B, $out(\sigma_A, \sigma_B) \models parity(p_1) \wedge parity(p_2)$. ■

Theorem 20. *The problem of deciding the existence of a solution for the non-cooperative Streett synthesis problem in a 2-player game arena is* PSPACE-H.

Proof. By reduction from QBF. Let $\phi = Q_k x_k \dots \forall x_1 \exists x_0 \gamma$ be a quantified boolean formula in disjunctive normal form, where the quantifiers are strictly alternating. The proof will proceed as follows. First, we build a two-players zero-sum Muller game $\mathcal{G}_\phi$ such that Player 0 has a winning strategy in $\mathcal{G}_\phi$ if and only if ϕ is true. Then, we use $\mathcal{G}_\phi$ to build a non-cooperative Street strategy synthesis two-players game $\mathcal{G}^*_\phi$, such that the system wins if and only if ϕ is false.

Let us first define the two-players zero-sum Muller game $\mathcal{G}_\phi$. Let $\phi = Q_k x_k \dots \forall x_1 \exists x_0 \gamma$ be a QBF formula in disjunctive normal form, where ϕ is a disjunction of the clauses $C_0, \dots, C_m$ over the literals $\{x_0, \neg x_0, \dots, x_k, \neg x_k\}$. Given ϕ, the two-players zero-sum Muller game $\mathcal{G}_\phi = (\langle V_0, V_1 \rangle, E, v_0, \mathcal{O}_0 \subseteq V^\omega)$ is defined as follows:

- $V_0 = \{\phi\} \cup \{x, \neg x \mid x \text{ is a variable appearing in } \phi\}$
- $V_1 = \{C_0, \dots, C_m\}$, the set of clauses in ϕ
- $v_0 = \phi$
- E is given by:
  - for each $0 \leqslant i \leqslant m$, $(\phi, C_i) \in E$
  - if $C_i = \ell_0 \wedge \ell_1 \wedge \ell_2$, then $(C_i, \ell_0) \in E$, $(C_i, \ell_1) \in E$, $(C_i, \ell_2) \in E$
  - for each $0 \leqslant i \leqslant k$, $(x_i, \phi) \in E$, $(\neg x_i, \phi) \in E$
- Given a path $\pi \in V^\omega$, let $i(\pi)$ be the index $0 \leqslant i(\pi) \leqslant k$ such that:
  - either $x_{i(\pi)}$ or $\neg x_{i(\pi)}$ is seen infinitely often on π

- for all $i(\pi) < j \leqslant k$, both x_j and $\neg x_j$ are seen finitely often on π

In other words, if we refer to the set of literals $\{x_i, \neg x_1\}$ as literals of level i, then $i(\pi)$ is the index of the last level of literals (counting the levels from 0 to k) visited infinitely often in π. Note that $i(\pi)$ is well defined since, by definition of E, each infinite path of $\mathcal{G}_\phi$ contains at least one literal that repeats infinitely often.

The winning condition $\mathcal{O}_0 \subseteq V^\omega$ states that the set of winning plays for Player 0 is given by:

$$\begin{aligned} \mathcal{O}_0 = \{\pi \mid i(\pi) \text{ is odd} \wedge x_{i(\pi)}, \neg x_{i(\pi)} \in \mathsf{inf}(\pi)\} \cup \\ \{\pi \mid i(\pi) \text{ is even } \wedge (x_{i(\pi)} \notin \mathsf{inf}(\pi) \vee \neg x_{i(\pi)} \notin \mathsf{inf}(\pi)\} \end{aligned} \tag{1}$$

where $\mathsf{inf}(\pi)$ is the set of nodes that appear infinitely often on the path π.

In other words, Player 0 wins the play π if and only if:

- either the index of the last level of literals visited infinitely often is odd (i.e. $i(\pi)$ is odd) and both $x_{i(\pi)}$ and $\neg x_{i(\pi)}$ are visited infinitely often, or
- the index of the last level of literals visited infinitely often is even (i.e. $i(\pi)$ is even), but only one literal in $\{x_{i(\pi)}, \neg x_{i(\pi)}\}$ appears infinitely often in π.

We show that $\mathcal{O}_0$ can be written as a combination of a Street and a Rabin condition, i.e. $\mathcal{O}_0 = S \wedge R$ where S (resp. R) is a Street (resp. Rabin) condition. Given $0 \leqslant i \leqslant k$, denote by $L_{j>i}$ the set of literals $L_{j>i} = \{x_j, \neg x_j \mid j > i\}$. Then:

$$S = \bigwedge_{\text{i odd}} (\{x_i\}, \{\neg x_i\} \cup L_{j>i}) \wedge (\{\neg x_i\}, \{x_i\} \cup L_{j>i}) \tag{2}$$

$$R = \bigvee_{\text{i odd}} (\{x_i, \neg x_i\}, L_{j>i}) \vee \bigvee_{\text{i even}} (\{x_i\}, \{\neg x_i\} \cup L_{j>i}) \vee (\{\neg x_i\}, \{x_i\} \cup L_{j>i}) \tag{3}$$

Namely, S states that for each odd level i, if you see x_i (resp. $\neg x_i$) infinitely often in π, then either $i(\pi) > i$ (i.e. i is not the last level visited) or $i(\pi) = i$ (i.e. the last level visited is odd) and both literals at the odd level $i(\pi) = i$ are seen infinitely often on π. The Rabin condition R instead states that either the last level visited is even and only one between $x_{i(\pi)}$ and $\neg x_{i(\pi)}$ is seen infinitely often, or otherwise the last level is odd (and the condition S takes care of its properties).

Given the above definition of the 0-sum Muller game $\mathcal{G}_\phi$, we are now ready to prove that $\phi = Q_k x_k \ldots \forall x_1 \exists x_0 \gamma$ is true if and only if Player 0 wins $\mathcal{G}_\phi$. In particular, we will proceed by induction on k. Note that if x_0 does not appear in ϕ, we can add the clause $x_0 \wedge \neg x_0$ withouth changing the truth value of ϕ.

Base Case By the idempotency of $\vee$ and $\wedge$ and assuming that ϕ is closed, ϕ is logically equivalent to one of the following forms:

1. $\phi = \exists x_0(x_0)$ or $\exists x_0(\neg x_0)$. In this case, the arena consists of four vertices $\{\phi, C_0, x_0, \neg x_0\}$. If $\phi = \exists x_0(x_0)$, then $\neg x_0$ is isolated, otherwise x_0 is isolated. Therefore, $\mathcal{G}_\phi$ contains only one cycle winning for Player 0.

2. $\phi = \exists x_0(x_0 \vee \neg x_0)$. $\mathcal{G}_\phi$ consists of five vertices $\{\phi, C_0, C_1, x_0, \neg x_0\}$. Player 0 wins by choosing always C_0 from ϕ.
3. $\phi = \exists x_0(x_0 \wedge \neg x_0)$. $\mathcal{G}_\phi$ consists of four vertices $\{\phi, C_0, x_0, \neg x_0\}$. Player 0 can only play to $C_0 = x_0 \wedge \neg x_0$ from ϕ. Player 1 wins by choosing alternatively x_0 and $\neg x_0$ from C_0.

Inductive Case By inductive hypothesis, we know that if ϕ has $k-1$ quantifiers and is closed, than Player 0 has a winning strategy if and only if ϕ is true. To prove the inductive step for k quantifiers we use the following lemma that shows how subgames correspond to restricted subformulae. First, let us introduce some notation. Given $v \in V, U \subseteq V$ and $i \in \{0,1\}$, we denote by $\mathsf{Avoid}_i(U, v)$ the subset of U from which Player i has a strategy to avoid vertex v withoth leaving U.

Lemma 21. *If* $\phi = \mathsf{Q}x.\gamma$ *and* $\gamma[x \to \mathsf{true}]$ *does not simplify to either* true *or* false*, then* $\mathsf{Avoid}_1(\mathsf{Avoid}_0(V, \neg x), x)$ *induces a subgame of* $\mathcal{G}_\phi$ *that is isomorphic to* $\mathcal{G}_{\phi[x \to \mathsf{true}]}$ *. Dually, if* $\gamma[x \to \mathsf{false}]$ *does not simplify to either* true *or* false*, then* $\mathsf{Avoid}_1(\mathsf{Avoid}_0(V, x), \neg x)$ *induces a subgame of* $\mathcal{G}_\phi$ *that is isomorphic to* $\mathcal{G}_{\phi[x \to \mathsf{false}]}$ *.*

Proof. $\gamma[x \to \mathsf{true}]$ consists of the clauses of γ that do not contain $\neg x$, say $c_1, \ldots, c_p$ with all the occurrences of x removed. The arena of the game $\mathcal{G}_{\gamma[x \to \mathsf{true}]}$ consists therefore of an initial vertex, one vertex for each clause $c_1, \ldots, c_p$ and one vertex for each literal different from $x, \neg x$. The edges are the same of $\mathcal{G}_\gamma$ restricted to the above set of vertices. We show that the graph induced by $\mathsf{Avoid}_1(\mathsf{avoid}_0(V, \neg x), x)$ is isomorphic to the arena of $\mathcal{G}_{\gamma[x \to \mathsf{true}]}$. The set of vertices $U = \mathsf{Avoid}_0(V, \neg x)$ is given by V minus the set of clauses c containing $\neg x$ and the vertex $\neg x$. Note that C is not empty since $\gamma[x \to \mathsf{true}]$ does not simplify to false. The set of vertices $W = \mathsf{Avoid}_1(U)$ is then obtained by removing from U the only vertex x (note that Player 1 has more than one choice from each clause since $\gamma[x \to \mathsf{true}]$ does not evaluate to true). Therefore, W precisely consists of the initial vertex, one node for each clause not containing $\neg x$ n γ and a node for each literal different from $x, \neg x$. Hence, the graph induced by $\mathsf{Avoid}_1(\mathsf{avoid}_0(V, \neg x), x)$ is isomorphic to $\mathcal{G}_{\gamma[x \to \mathsf{true}]}$. The proof of the case $\gamma[x \to \mathsf{false}]$ is symmetric. ■

Given the above lemma, we are now ready to deal with the inductive step. We consider two cases, depending on wether the variable x in $\phi = \mathsf{Q}x.\gamma$ is quantified universally or existentially.

1. $\phi = \exists x.\gamma$. If ϕ is true, then there is a value $v \in \{0,1\}$ such that $\gamma[x \to v]$ is true. Assume $v = 1$ is such a value. The, Player 0 plays in $\mathsf{Avoid}_0(V, \neg x)$ trying to reach infinitely often x. If Player 1 at some point prevents him to reach x (from that point of the game over) then the game gets confined in $\mathsf{Avoid}_1(\mathsf{avoid}_0(V, \neg x), x)$ in which Player 0 has a strategy to win. The subcase where $\gamma[x \to 1]$ is symmetric.
   If ϕ is false, then $\gamma[x \to 0]$ (resp $\gamma[x \to 1]$) is false and Player 1 can use the following strategy to win. Indefinitely, alternatively try to reach first x (while avoiding $\neg x$), and

then try to reach $\neg x$ (while avoiding x). If at any point the opponent prevents him to reach his current objective, the game gets confined in $\mathsf{Avoid}_1(\mathsf{avoid}_0(V, \neg x), x)$ or $\mathsf{Avoid}_1(\mathsf{avoid}_0(V, x), \neg x)$ in which Player 1 has a winning strategy.

2. $\phi = \forall x.\gamma$. If ϕ is true, Player 0 can adopt the following strategy to win. He will try alternatively to reach x (while avoiding $\neg x$) and then reach $\neg x$ while avoiding x. If at any point of the game Player 1 prevents Player 0 to reach its target then the game gets confined into $\mathsf{Avoid}_1(\mathsf{avoid}_0(V, \neg x), x)$ or $\mathsf{Avoid}_1(\mathsf{avoid}_0(V, x), \neg x)$ where Player 0 has a winning strategy. The subcase where $\phi = \forall x.\gamma$ is false is symmetric to the subcase where $\phi = \exists x.\gamma$ is true seen above.

Note that, intuitively, Player 1 fixes the value of a variable each time he forbid Player 0 to reach (from that point on of the play) its target literal. Whenever the value of a variable—say x_i—is fixed by Player 1 in this way, the play prooceds into a inner layer of variables, i.e. into the arena of a subgame that contains only literals of levels less than i.

Resuming, we have now proven that the QBF formula $\phi = Q_k x_k \ldots \forall x_1 \exists x_0 \gamma$ (in DNF) is true if and only if Player 0 wins the 0-sum Muller game $\mathcal{G}_\phi$, in which the objective of Player 0 $W_0 = S \wedge R$ is a combination of a Street condition S and a Rabin condition R. Given $\mathcal{G}_\phi$, consider now the following non-cooperative 2-players Street game $\mathcal{G}^*_\phi$:

- the arena of $\mathcal{G}^*_\phi$ is exactly the same of $\mathcal{G}_\phi$
- Player 1 is the system
- the environment is composed by the only Player 0
- the objective of the system is the Street condition $\neg R$
- the objective of the only component in the environment is the Street condition S

We show that ϕ is false if and only if there is a solution to the non-cooperative strategy synthesi problem $\mathcal{G}^*_\phi$. Before prooceding in such a proof, note that $\neg R \rightarrow S$. In fact:

$$\neg R = \bigwedge_{\text{i even}} (\{x_i\}, \{\neg x_i\} \cup L_{j>i}) \wedge (\{\neg x_i\}, \{x_i\} \cup L_{j>i}) \wedge \bigwedge_{\text{i odd}} (\{x_i, \neg x_i\}, L_{j>i}) \qquad (4)$$

Hence $\neg R$ states that the last level visited infinitely often is even, that implies S. Given the above observation, we prooced to prove that ϕ is false if and only if there is a solution to the non-cooperative strategy synthesis problem $\mathcal{G}^*_\phi$. There are two cases to consider. If ϕ is true, then the environment can ensure $S \wedge R$, i.e. he has a strategy to guarantee that he accomplishes his objective, while the system does not.

In the other case, suppose that ϕ is false. Then the system has a strategy to ensure $\neg S \vee \neg R$. Since $\neg R \rightarrow S$, either the environment looses, or it holds $\neg R$ and both the players win. The environment has always the possibility to cooperate to establish $\neg R$ by e.g. always choosing the clause[6] $x_0 \wedge \neg x_0$. ∎

Theorem 21. *The problem of deciding the existence of a solution for the non-cooperative Rabin synthesis problem in a 2-player game arena is* PSPACE-H.

[6] W.l.o.g. we can assume that the only clause containg x_0 in ϕ is $x_0 \wedge \neg x_0$. In fact, if this is not the case we can leadϕ to such a form by renaming each valriable x_i to x_{i+2} and adding the clause $x_0 \wedge \neg x_0$

Proof. By reduction from QBF. Let $\phi = Q_k x_k \ldots \forall x_1 \exists x_0 \gamma$ be a quantified boolean formula in disjunctive normal form, and consider the equivalent QBF:

$$\phi' = Q_k x_k \ldots \forall x_1 \exists x_0 \forall y_1 \exists y_0 ((y_1 \wedge \gamma) \vee (\neg y_1 \wedge \gamma))$$

Let ϕ'' be the formula obtained from ϕ' by first renaming each variable x_i, $i = 0 \ldots k$, to x_{i+2}, and each variable y_i, $i \in \{0, 1\}$, to x_i, and then normalizing the resulting formula in DNF. Let $\mathcal{G}_{\phi''}$ be the 2-players 0-sum Muller game such that Player 0 has a winning strategy in $\mathcal{G}_{\phi''}$ if and only if ϕ'' is true, built according to the procedure shown within the proof of Theorem 20. Given $\mathcal{G}_{\phi''}$, consider the following non-cooperative 2-players Rabin game $\mathcal{G}^*_{\phi''}$:

- the arena of $\mathcal{G}^*_{\phi''}$ is exactly the same of $\mathcal{G}_{\phi''}$
- Player 1 is the system
- the environment is composed by the only Player 0
- the objective of the system is the Rabin condition $\neg S$
- the objective of the only component in the environment is the Rabin condition R

We show that ϕ'' is false if and only if there is a solution to the non-cooperative strategy synthesi problem $\mathcal{G}^*_{\phi''}$. There are two cases to consider:

1. In the first case, assume that ϕ'' is true. Then, the environment can ensure $S \wedge R$ that is a NE where he wins while the system looses.
2. In the second case, suppose that ϕ'' is false. Then, the system has a strategy to ensure $\neg S \vee \neg R$. We claim that such a strategy (c.f.r. the proof of Theorem 20) is indeed a solution to the non-cooperative Rabin strategy synthesis problem on $\mathcal{G}^*_{\phi''}$. In fact, the environment can win if he cooperates with the system to establish $\neg S$, i.e. if he cooperate to let the last level of variables visited to be odd. The environment can effectively force the last level visited to be odd by opposing to the system a strategy that forbid the system to reach its target literal, leading the play to be confined within inner and inner layers (of literals), until the objective of the system is to reach (only one) literal of level 1—say e.g. x_1. At that point, the environment simply let the system to pursue its objective by choosing only clauses with the literal x_1 (that appears in ϕ'' by construction).

■

7 Conclusion

In this paper, we have studied the complexity of rational synthesis in both the cooperative and non-cooperative settings, and depending on whether the number of players is fixed or not. Our results are summarised in Table 1. Rationality of the environment is modeled by assuming that the players composing it play a Nash equilibrium. Interesting directions for future work would be to assume other notions of rationality, e.g. secure equilibria [10], doomsday equilibria [8], subgame perfect equilibria [24, 25], or admissible strategies [4, 14].

References

1. R. Alur and S. La Torre. Deterministic generators and games for ltl fragments. *ACM Trans. Comput. Logic*, 5(1):1–25, Jan. 2004.
2. B. Aminof and S. Rubin. First cycle games. In *Proceedings 2nd International Workshop on Strategic Reasoning, SR 2014, Grenoble, France, April 5-6, 2014.*, pages 83–90, 2014.
3. C. Baier and J.-P. Katoen. *Principles of model checking.* MIT Press, 2008.
4. D. Berwanger. Admissibility in infinite games. In *STACS 2007, 24th Annual Symposium on Theoretical Aspects of Computer Science, Aachen, Germany, February 22-24, 2007, Proceedings*, volume 4393 of *Lecture Notes in Computer Science*, pages 188–199. Springer, 2007.
5. R. Brenguier, L. Clemente, P. Hunter, G. A. Pérez, M. Randour, J. Raskin, O. Sankur, and M. Sassolas. Non-zero sum games for reactive synthesis. *CoRR*, abs/1512.05568, 2015.
6. R. Brenguier, J. Raskin, and O. Sankur. Assume-admissible synthesis. In *26th International Conference on Concurrency Theory, CONCUR 2015, Madrid, Spain, September 1.4, 2015*, volume 42 of *LIPIcs*, pages 100–113. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.
7. R. Brenguier, J. Raskin, and M. Sassolas. The complexity of admissibility in omega-regular games. In *Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014*, pages 23:1–23:10. ACM, 2014.
8. K. Chatterjee, L. Doyen, E. Filiot, and J. Raskin. Doomsday equilibria for omega-regular games. In *Verification, Model Checking, and Abstract Interpretation - 15th International Conference, VMCAI 2014, San Diego, CA, USA, January 19-21, 2014, Proceedings*, volume 8318 of *Lecture Notes in Computer Science*, pages 78–97. Springer, 2014.
9. K. Chatterjee and T. A. Henzinger. Assume-guarantee synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems, 13th International Conference, TACAS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007 Braga, Portugal, March 24 - April 1, 2007, Proceedings*, volume 4424 of *Lecture Notes in Computer Science*, pages 261–275. Springer, 2007.
10. K. Chatterjee, T. A. Henzinger, and M. Jurdzinski. Games with secure equilibria. *Theor. Comput. Sci.*, 365(1-2):67–82, 2006.
11. K. Chatterjee, T. A. Henzinger, and N. Piterman. Generalized parity games. In *Proceedings of the 10th International Conference on Foundations of Software Science and Computational Structures*, FOSSACS'07, pages 153–167, Berlin, Heidelberg, 2007. Springer-Verlag.
12. K. Chatterjee, T. A. Henzinger, and N. Piterman. Algorithms for büchi games. *CoRR*, abs/0805.2620, 2008.
13. A. Cimatti, E. M. Clarke, E. Giunchiglia, F. Giunchiglia, M. Pistore, M. Roveri, R. Sebastiani, and A. Tacchella. Nusmv 2: An opensource tool for symbolic model checking. In *Computer Aided Verification, 14th International Conference, CAV 2002,Copenhagen, Denmark, July 27-31, 2002, Proceedings*, volume 2404 of *Lecture Notes in Computer Science*, pages 359–364. Springer, 2002.
14. M. Faella. Admissible strategies in infinite games over graphs. In *Mathematical Foundations of Computer Science 2009, 34th International Symposium, MFCS 2009, Novy Smokovec, High Tatras, Slovakia, August 24-28, 2009. Proceedings*, volume 5734 of *Lecture Notes in Computer Science*, pages 307–318. Springer, 2009.
15. D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. *CoRR*, abs/0907.3019, 2009.
16. Y. Gurevich and L. Harrington. Trees, automata, and games. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 60–65, New York, NY, USA, 1982. ACM.
17. P. Hunter and A. Dawar. Complexity bounds for regular games. In *Proceedings of the 30th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, Lecture Notes in Computer Science, pages 495–506, Berlin, Heidelberg, 2005. Springer-Verlag.
18. M. Jurdzinski, M. Paterson, and U. Zwick. A deterministic subexponential algorithm for solving parity games. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2006, Miami, Florida, USA, January 22-26, 2006*, pages 117–123, 2006.

19. O. Kupferman, G. Perelli, and M. Y. Vardi. Synthesis with rational environments. In *Multi-Agent Systems - 12th European Conference, EUMAS 2014, Prague, Czech Republic, December 18-19, 2014, Revised Selected Papers*, pages 219–235, 2014.
20. F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. What makes atl* decidable? A decidable fragment of strategy logic. In *CONCUR 2012 - Concurrency Theory - 23rd International Conference, CONCUR 2012, Newcastle upon Tyne, UK, September 4-7, 2012. Proceedings*, volume 7454 of *Lecture Notes in Computer Science*, pages 193–208. Springer, 2012.
21. A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *POPL*, pages 179–190. ACM Press, 1989.
22. S. Schewe. Solving parity games in big steps. In *FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, 27th International Conference, New Delhi, India, December 12-14, 2007, Proceedings*, volume 4855 of *Lecture Notes in Computer Science*, pages 449–460. Springer, 2007.
23. W. Thomas. On the synthesis of strategies in infinite games. In *STACS*, pages 1–13, 1995.
24. M. Ummels. Rational behaviour and strategy construction in infinite multiplayer games. In *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, volume 4337 of *Lecture Notes in Computer Science*, pages 212–223. Springer, 2006.
25. M. Ummels. The complexity of nash equilibria in infinite multiplayer games. In *Foundations of Software Science and Computational Structures, 11th International Conference, FOSSACS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29 - April 6, 2008. Proceedings*, volume 4962 of *Lecture Notes in Computer Science*, pages 20–34. Springer, 2008.
26. W. Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998.