The Complexity of Rational Synthesis

Rodica Condurache[1], Emmanuel Filiot[*2], Raffaella Gentilini[†3], and Jean-François Raskin[‡4]

1 **Université Libre de Bruxelles, Computer Science Department, Brussels, Belgium, and**
Université Paris Est, LACL(EA 4219), UPEC, Créteil Cedex, France
2 **Université Libre de Bruxelles, Computer Science Department, Brussels, Belgium**
3 **University of Perugia, Dept. of Mathematics and Computer Science, Perugia, Italy**
2 **Université Libre de Bruxelles, Computer Science Department, Brussels, Belgium**

Abstract

We study the computational complexity of the cooperative and non-cooperative rational synthesis problems, as introduced by Kupferman, Vardi and co-authors. We provide tight results for most of the classical omega-regular objectives, and show how to solve those problems optimally.

1998 ACM Subject Classification B.6.3 Automatic Synthesis, F.2 Analysis of algorithms and problem complexity, F.1.1 Automata

Keywords and phrases Non-zero sum games, reactive synthesis, omega-regular objectives

Digital Object Identifier 10.4230/LIPIcs.ICALP.2016.121

1 Introduction

Rational synthesis [14, 19] asks to synthesize a system that is executed in an environment made of several components that are assumed to be rational, and not fully antagonistic as in the classical two player zero-sum setting [23]. Rationality of the environment is modeled by assuming that the components behave according to a Nash equilibrium (NE). Rational synthesis has been introduced in [14, 19] in two different settings.

In the first setting, called *cooperative rational synthesis* [14], the environment cooperates with the system: components of the environment agree to play a NE that is winning for Player 0 (if it exists). In the second setting, called *non-cooperative rational synthesis* [19], the components of the environment may follow any strategy, provided that it is a NE. 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 NE.

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

* E. Filiot is research associate at FNRS. This work was partially supported by ARC project *Transform* (French speaking community of Belgium) and Belgian FNRS PDR project *Flare*.
† R. Gentilini acknowledges the support of *Fondi Ricerca Base 2015* (Automi e Teoria dei Giochi per la Verifica) and InDAM-GNCS.
‡ J.-F. Raskin was supported by ERC Starting Grant inVEST (nr. 279499).

43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016).
Editors: Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi;
Article No. 121; pp. 121:1–121:15

Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Complexity of rational synthesis for k players. Full proofs can be found in [12].

	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
coBüchi	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[14]	2ExpTime-c[14]	2ExpTime-c[19]	2ExpTime-c[19]

classical zero-sum two-player LTL synthesis [21]. The upper bound is obtained by reductions to the satisfiability problem of formulas in Strategy Logic (SL) [20]. 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 solve 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 for unfixed and fixed number of players. The fixed number of players case is interesting as the number of components forming the environment may be limited to a few in practical applications. Our results are summarized in Table 1.

On the positive side, our results show that for a *fixed* number of players, and 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 special case of constrained NE (Player 0 has to be winning). The complexity of constrained NE has been studied in [25] for some classes of objectives: this gives us upper-bounds for cooperative synthesis and Büchi, coBüchi, parity and Streett objectives. For the other objectives, we show how to extend the approach proposed in [25]. Solutions to the *non-cooperative* case are much more involved and 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 our algorithms 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 CAV community (see e.g. tools like nuSMV [11]).

To obtain lower-bounds, we design original and informative reductions.

Related work. Non-zero sum games for synthesis are gaining attention recently, see e.g. [4] for a survey. *Secure equilibria* were introduced in [9] and their potential for synthesis was demonstrated in [8]. Secure equilibria is a refinement of NE [24]. *Doomsday equilibria* extend secure equilibria to the n player case [7]. Subgame perfect equilibria, that also refines NE, were first studied in [24, 25]. To model rationality of players, the notion of *admissible strategy* is used in [3, 13] instead of the notion of NE, and computational aspects are studied in [6], potential for synthesis is studied in [5]. All those works consider games played on a game structure with classical ω-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 [14, 19]. This paper provides algorithms and precise computational complexity results.

Structure of the paper. In Sect. 2, we recall the definition of the cooperative and non-cooperative synthesis problem as introduced in [14, 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. Due to the lack of space, we provide sketches of proof of our results in this paper and all the detailed proofs can be downloaded at the following address: [12].

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 $E \subseteq \bigcup_{i\in\Omega} V_i \times V_{i+1 \bmod k}$.[1]. Formally, a play $\pi = v_0 v_1 \dots$ is an infinite path in V^ω such that v_0 is the initial state 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, by $\mathsf{Plays}(\mathcal{A})$ the set of plays, and by $\mathsf{Prefs}(\mathcal{A})$ for its set of finite prefixes. For $\pi \in V^\omega$, $\inf(\pi)$ is the set of states occurring infinitely many times in π.

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

[1] Wlog. we assume that each vertex has a successor by E and that player's rounds are ordered according to their index. Otherwise we just add a polynomial number of extra intermediate states and the winning objectives considered in this paper can be modified accordingly.

a set $\mathcal{O} \subseteq V^\omega$. It is *tail* if it is closed under removing prefixes. A Player i's strategy σ_i is *winning*[2] for $\mathcal{O}$ if $out(\sigma_i) \subseteq \mathcal{O}$. We consider the classical classes of objectives [17]:

- *Safety/Reachability*: Given a set $S \subseteq V$ of safe states, $\mathsf{Safe}(S) = \{\pi \in V^\omega \mid \forall n \geqslant 0 : \pi(n) \in S\}$ and given a set T of target states, $\mathsf{Reach}(T) = \{\pi \in V^\omega \mid \exists n \geqslant 0 : \pi(n) \in T\} = \overline{\mathsf{Safe}(\overline{T})}$.
- *Büchi/coBüchi*: Given a set $F \subseteq V$, $\mathsf{Buchi}(F) = \{\pi \in V^\omega \mid \inf(\pi) \cap F \neq \varnothing\}$ and $\mathsf{coBuchi}(F) = \{\pi \in V^\omega \mid \inf(\pi) \cap F = \varnothing\} = \overline{\mathsf{Buchi}(F)}$.
- *Streett/Rabin*: Given a set $\Psi \subseteq 2^V \times 2^V$, $\mathsf{Streett}(\Psi) = \bigcap_{(L,R)\in\Psi}(\mathsf{coBuchi}(L) \cup \mathsf{Buchi}(R))$ and $\mathsf{Rabin}(\Psi) = \bigcup_{(L,R)\in\Psi}(\mathsf{Buchi}(L) \cap \mathsf{coBuchi}(R)) = \overline{\mathsf{Streett}(\Psi)}$.
- *Parity*: For a priority mapping $p{:}V \to \mathbb{N}$, $\mathsf{Parity}(p) = \{\pi \in V^\omega \mid \min\{p(v) | v \in \inf(\pi)\} \text{ is even}\}$.
- *Muller*: Given a Boolean formula μ over V, $\mathsf{Muller}(\mu) = \{\pi \in V^\omega \mid \inf(\pi) \models \mu\}$.

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 $i \in \Omega$. For all class of objectives X, we say that $\mathcal{G}$ is a multiplayer X-game if all objectives $\mathcal{O}_i$ are in X. The notations Plays and Prefs carries over naturally to $\mathcal{G}$ by considering its underlying arena. For $v \in V$, one denotes by $\mathcal{G}[v]$ the game $\mathcal{G}$ whose initial state is replaced by v (winning objectives are unchanged). A state $v \in V$ is *winning* for Player i if he has a winning strategy in $\mathcal{G}[v]$, and one denotes by $W_i^{\mathcal{G}}$ (or just W_i) the set of *winning states* of Player i, also called the *winning set* of Player i.

Nash Equilibria. A *strategy profile* $\bar{\sigma}$ in a multiplayer game $\mathcal{G} = \langle \mathcal{A}, (\mathcal{O}_i)_{i\in\Omega} \rangle$ is a tuple $\bar{\sigma} = (\sigma_i)_{i\in\Omega}$ of strategies, one for each player. The *outcome* $out(\bar{\sigma})$ of $\bar{\sigma}$ is the only play consistent with all strategies σ_i (it always exists and is unique). Given a strategy τ for Player i, we write $(\bar{\sigma}_{-i}, \tau)$ for the strategy profile obtained by replacing σ_i with τ in $\bar{\sigma}$. For winning objectives $(\mathcal{O}_i)_{i\in\Omega}$ for each player, the *payoff* of a strategy profile $\bar{\sigma}$ is the vector $pay(\bar{\sigma}) \in \{0,1\}^{\{k+1\}}$ defined by $pay(\bar{\sigma})[i] = 1$ iff $out(\bar{\sigma}) \in \mathcal{O}_i$. We write $pay_i(\bar{\sigma})$ for Player i's payoff in $pay(\bar{\sigma})$. 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* (NE) if no player can improve his payoff by (unilaterally) switching to a different strategy, i.e. for all players $i \in \Omega$ and all strategies τ of Player i, $pay(\bar{\sigma}_{-i}, \tau) \leqslant pay(\bar{\sigma})$. We say that $\bar{\sigma}$ is a 0-*fixed* Nash equilibrium (0NE) 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 is not allowed to deviate. Any NE is 0-fixed, but the converse may not hold.

Cooperative and non-cooperative rational synthesis. Rational synthesis aims at finding a winning strategy for the system (Player 0) against an environment composed of several components (Players 1 to k) that are assumed to play rationally. Rationality of the environment is modeled by NE, and following [14, 19], we consider two settings, depending on whether the environment cooperates or not: The cooperative and non-cooperative rational synthesis problems (CRSP and NCRSP resp.) ask, given as input a $(k+1)$-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 every 0-fixed Nash equilibrium $\bar{\sigma} = \langle \sigma_0, \ldots, \sigma_k \rangle$, we have $pay_0(\bar{\sigma}) = 1$?

[2] Here we implicitly consider a two-player zero-sum game in which Player i has objective $\mathcal{O}$ and plays against all the other players in $\Omega \backslash \{i\}$ who have objective $\overline{\mathcal{O}}$.

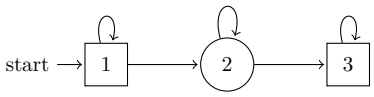

Figure 1 Example for rational synthesis.

▶ **Example 1.** Consider Figure 1 in which Player 0 owns round states and Player 1 square states, with the reachability objectives $R_0 = \{2\}$ and $R_1 = \{3\}$. Consider the Player 0's strategies σ_0 which loops forever in state 2, and σ_0' which eventually moves to state 3.

Let Player 1 cooperate with σ_1 that moves 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 later one Player 1 wins. Strategy σ_0 is not a solution to the non-cooperative setting: 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.

In [14, 19], both CRSP and NCRSP are shown 2EXPTIME-COMPLETE when the winning objectives are defined by LTL formulas, through a reduction to strategy logic. In this paper, we refine this complexity result for particular kinds of winning objectives. In general, the synthesis problem also asks to *synthesise* (i.e. construct) a solution when it exists: Our algorithms also solve the synthesis problem.

3 Cooperative Rational Synthesis Problem (CRSP)

We establish here complexity bounds for CRSP for unfixed number of players. First, some results are obtained as special cases of constrained NE problems [25]. The constrained NE problem asks to decide, given a $k+1$-player game $\mathcal{G} = \langle \mathcal{A}, (\mathcal{O}_i)_{i \in \Omega} \rangle$, and for each player i, a lower bound $l_i \in \{0,1\}$ and an upper bound $u_i \in \{0,1\}$ such that $l_i \leqslant u_i$, whether there exists a NE $\overline{\sigma}$ s.t. $l_i \leqslant pay_i(\overline{\sigma}) \leqslant u_i$ for all $i \in \Omega$. CSRP is a special case of this problem, by setting $l_0 = u_0 = 1$, $l_i = 0$ and $u_i = 1$ for all $i \in \Omega \backslash \{0\}$. The constrained NE problem is known to be in PTIME for Büchi objectives, and in NP for co-Büchi, Streett and parity objectives [25]. So one immediately gets the upper bounds of Table 1 for these measures (and unfixed k).

To establish the remaining upper bounds, we characterize NE by means of LTL formulas. We extend the technique used in [25] for tail objectives to safety and reachability.

Generic solution to cooperative rational synthesis. Syntax and semantics of LTL can be found in [2]. Let V be the set of vertices of $\mathcal{G}$. We use LTL formulas to express properties of infinite paths of $\mathcal{G}$, where we take V as set of atomic propositions. In particular, $s \in V$ is true in s, and false otherwise. For $S \subseteq V$, the formula S is a shortcut for $\bigvee_{s \in S} s$, and LTL($\mathcal{G}$) denotes the set of LTL formulas over V. Let $(W_i^{\mathcal{G}})_{0 \leqslant i \leqslant k}$ be the winning sets of each player and $b \in \{0,1\}$. We define an LTL[$\mathcal{G}$]-formula $\phi_{b\text{Nash}}^{\mathcal{G}}$ that will characterize NE ($b = 1$) and 0-NE ($b = 0$):

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

Assume that $b = 1$, and consider the formula for safety objectives. Intuitively, it says that for all players $i \in \{0, \dots, k\}$, either Player i always stays safe, or if eventually he visits an unsafe position, then he should never visit a winning position until he meets an unsafe position for the first time. This is because otherwise he could apply a winning strategy and satisfy his own objective, and therefore has some incentive to deviate. As announced:

▶ **Proposition 2** (Characterization of 0-fixed NE and NE ([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 $\pi \in \mathit{Plays}(\mathcal{G})$, if $\pi \models \phi^{\mathcal{G}}_{0Nash}$ (resp. $\pi \models \phi^{\mathcal{G}}_{1Nash}$), then there exists a 0-fixed NE (resp. NE) $\bar{\sigma}$ in $\mathcal{G}$ such that $out(\bar{\sigma}) = \pi$,*
2. *For all 0-fixed NE (resp. NE) $\bar{\sigma}$ in $\mathcal{G}$, $out(\bar{\sigma}) \models \phi^{\mathcal{G}}_{0Nash}$ (resp. $out(\bar{\sigma}) \models \phi^{\mathcal{G}}_{1Nash}$).*

Based on the latter proposition, it is not difficult to design a procedure to decide CRSP: first, compute the winning sets W_i, and then check whether the game $\mathcal{G}$ contains a path which satisfies the formula $\phi^{\mathcal{G}}_{0\text{Nash}} \wedge \phi_0$, where ϕ_0 is the objective of Player 0, expressed in LTL[$\mathcal{G}$]. To establish precise upper bounds, one needs to consider the complexity of computing the winning sets, and then the complexity of model-checking these particular LTL formulas. Due to lack of space, we cannot cover all the cases, but let us briefly expose the case of safety conditions. For safety, it is well-known that computing the sets W_i can be done in PTime. Then, we prove a short witness property: if there exists a path satisfying the formula $\phi^{\mathcal{G}}_{0\text{Nash}} \wedge \phi_0$, then there exists a lasso path uv^ω such that u and v have polynomial length. Therefore, it suffices to guess such a lasso path, and to check that it satisfies the formula $\phi^{\mathcal{G}}_{0\text{Nash}} \wedge \phi_0$, which again can be done in PTime. This yields an NP algorithm for safety CRSP. Similar arguments apply for reachability.

Lower bounds. In [25] was shown the NP-hardness of the threshold NE problem for coBüchi objectives and thresholds $l_0 = u_0 = 1$, $l_i = 0$ and $u_i = 1$, $\forall i \in \{1, \dots, k\}$. Therefore, one immediately gets the NP lower bounds of Table 1 for coBüchi objectives, Streett, Rabin and parity objectives, which can (polynomially) express coBüchi objectives. For the other cases, we provide lower bounds with genuine reductions. For Muller objectives, we show that it is already PSpace-hard even for two players and the precise complexity results for a fixed number of players are discussed in the last section of the paper. We finish the section by showing the NP-hardness proof for the safety case (which was not considered in [25]).

▶ **Lemma 3.** *CRSP for multiplayer safety objectives is NP-hard.*

Proof. By reduction from $3SAT$. Given a set of clauses $S = \{C_1, \dots, C_k\}$, one constructs the $(k+1)$-player safety game of Fig. 2. Up to vertex C_1, the previous states are controlled by Player 0, and each state C_i is controlled by Player i. All states but U_s are safe for Player 0. For all $i \in \{1, \dots, k\}$, U_e is unsafe for Player i, 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. If S is satisfiable by some valuation ν, then Player 0 chooses, in state x, the successor $\nu(x)_x$. That way, Player 1 to k have visited an unsafe state before reaching C_1, and have no incentive to deviate if their strategy is to go to U_e. Conversely, if there is a solution to CRSP, necessarily the game end up in U_e, as Player 0 must be winning. It means that before reaching the states C_i all safety conditions for Players 1 to k have been violated, otherwise going to U_s could be a profitable deviation. In other words, for all clauses i, there exists a literal ℓ in C_i such that 1_x (resp. 0_x) has been visited if $\ell = x$ (resp. if $\ell = \neg x$). So the truth values chosen by Player 0 satisfies S.

◀

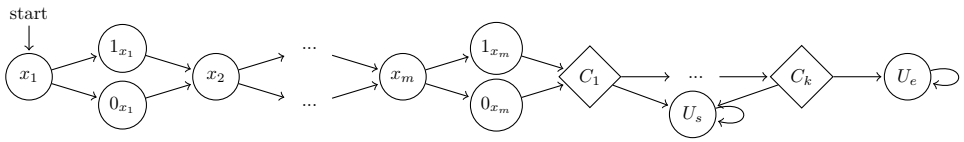

Figure 2 Cooperative Safety: Reduction from 3-SAT.

4 Non-Cooperative Rational Synthesis Problem (NCRSP)

We study here the complexity of NCRSP for unfixed number of players. In this setting, cooperation of the environment is not assumed, and so the system has to win against all 0-fixed NE. In Prop. 2, we have characterised 0-fixed NE by means of an LTL formula $\phi^{\mathcal{G}}_{0\text{Nash}}$. This allowed us to solve CRSP via a reduction to model-checking. It is tempting to think that NCRSP 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, Example 1 shows that this is not true in general. Indeed, in this example there is a solution to NCRSP, 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 NCRSP: 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 other 0-fixed NE, i.e. for a different strategy of Player 0.

The non-cooperative case is more involved and requires tree automata based techniques: we see strategies σ_0 as trees t_{σ_0}, and use *tree automata* to define the set of solutions to NCRSP. Testing existence of a solution then reduces to tree automata non-emptiness.

Strategy trees and good deviations. Let Λ be a finite set of directions and Σ be an alphabet. A Σ-labeled Λ-tree is a mapping $t : \Lambda^* \rightarrow \Sigma$. Its set of nodes is Λ^*. Let $\mathcal{A}$ be a $k+1$-player 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 $\Lambda = V$. Therefore, any node of the tree is a history h. 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). We denote by T_0 the set of strategy trees t_{σ_0}. Note that each Σ-labeled V-tree represents a partial function from V^* to Σ, which may not be a strategy, because it is not total and may not be consistent with the edge relation E of the arena. A *branch* in t_{σ_0} is an infinite sequence of directions $\pi \in V^\omega$. It is *compatible* with σ_0 if for all finite prefixes h of π whose last state is in V_0, $h.t_{\sigma_0}(h)$ is a prefix of π.

We now want to characterize the strategy trees t_{σ_0} s.t. σ_0 is a solution to NCRSP 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. Consider a branch π of t_{σ_0} compatible with σ_0: It is not the outcome of a σ_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 $i \in \{1, \dots, k\} = \Omega\backslash\{0\}$ s.t. $\pi \notin \mathcal{O}_i$ and there is a strategy σ_i for Player i s.t. for all strategies $(\sigma_j)_{i\in\{1,\dots,k\}\backslash\{i\}}$, $h.out(\sigma_0|_h, ..., \sigma_{i-1}|_h, \sigma_i|_h, \sigma_{i+1}|_h, ..., \sigma_k|_h) \in \mathcal{O}_i$. 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 4.** *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.*

Reduction to tree-automata emptiness. Based on Lemma 4, we construct a non-deterministic automaton defining $\mathsf{NCRSP}(\mathcal{G})$. A *non-deterministic tree automaton* $\mathcal{T}$ over Σ-labeled Λ-trees is a tuple (Q, Q_0, δ) where Q is a finite set of states, $Q_0 \subseteq Q$ is a finite set of initial states, and δ is a 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 Λ-tree $r : \Lambda^* \to Q$ such that $r(\epsilon) \in Q_0$ and for all $h \in \Lambda^*$, all $d \in \Lambda$, the mapping $d \in \Lambda \mapsto r(hd) \in Q$ 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 an accepting condition $\alpha \subseteq Q^\omega$, r is accepting if the images of all its branches are in α, and the *language* of $\mathcal{T}$ is the set $\mathcal{L}_\alpha(T)$ of trees for which there exists an accepting run.

▶ **Lemma 5.** *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 $\mathcal{T}_\mathcal{A}$ (with an exponential number of states in k, and polynomial in $|V|$) with an accepting condition $\alpha_\mathcal{A}(\mathcal{O})$ such that $\mathcal{L}_{\alpha_\mathcal{A}(\mathcal{O})}(\mathcal{T}_\mathcal{A}) = \mathsf{NCRSP}(\mathcal{G})$. Moreover, for all runs r of $\mathcal{T}_\mathcal{A}$, for all branches π of r, the number of states appearing in π is polynomial in $|V|$ and k.*

Proof. We sketch the construction of $\mathcal{T}_\mathcal{A}$ and the definition of $\alpha_\mathcal{A}(\mathcal{O})$. First, it is not difficult to make sure that $\mathcal{T}_\mathcal{A}$ only accepts trees that are strategy trees: It has to remember the last direction v taken and make sure that if $v \in V_0$, the current node is labeled by some $v' \in V$ s.t. $(v, v') \in E$, and otherwise by the symbol $*_i$ if $v \in V_i$. This requires only a polynomial number of states. Therefore in the following, we assume that $\mathcal{T}_\mathcal{A}$ only runs on proper tree encodings t_{σ_0} of strategies σ_0.

The construction of $\mathcal{T}_\mathcal{A}$ is based on Lemma 4: For each branch of t_{σ_0}, it will check that either it is not compatible with σ_0, or it belongs to $\mathcal{O}_0$, or it will guess a prefix and check it is a good deviation. To guess good deviations, $\mathcal{T}_\mathcal{A}$ has to guess subtrees in which players have a winning strategy. This information is stored in a set $W \subseteq \Omega$, with the following semantics: if $\mathcal{T}_\mathcal{A}$ is in some state with set W at some node $h \in V^*$ and $i \in W$, then Player i has a winning strategy from h against σ_0 and any strategy of the players in $\Omega\backslash\{0, i\}$ for objective $\mathcal{O}_i$. 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 $\mathcal{T}_\mathcal{A}$ 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 on W is maintained as follows: at some node $hv \in V^*$, if $i \in W$ and $v \in V_i$, then $\mathcal{T}_\mathcal{A}$ non-deterministically send W to one of the successor of v (and $W\backslash\{i\}$ in the other ones) and if $v \notin V_i$, $\mathcal{T}_\mathcal{A}$ sends W in all successors of v. The information D is monotonic: either the current node h (owned by Player i) is not guessed to be a good deviation for any player and D is sent to all successors, or it is guessed to be a good deviation for Player i, $i \notin D$, and then $D \cup \{i\}$ (and W) is sent to all successors but one in which is sent D and $W \cup \{i\}$. This monotonic behavior is crucial for obtaining algorithms with optimal worst-case complexities.

Formally, $\mathcal{T}_\mathcal{A} = (Q, \{q_0\}, \delta)$ with $Q = \{q_0, \top\} \cup (2^\Omega \times 2^\Omega \times V)$. Then, we have $\delta(q_0, \#) = \{\rho_0\}$ where $\rho_0(v_0) = (\varnothing, \varnothing, v_0)$ and $\rho_0(v) = \bot$ for all $v \neq v_0$. For all $q = (W, D, v) \in Q$ such that $v \in V_0$ and all $v' \in V$, $\delta(q, v') = \{\rho_{v'}\}$ where $\rho_{v'}(v') = (W, D, v')$ and $\rho_{v'}(v'') = \bot$ for all $v'' \neq v'$ (the latter case correspond to directions v'' which are not compatible with the strategy). For all $v \neq \#$, $\delta(\top, v) = \{\rho_\top\}$ where $\rho_\top(v') = \top$ for all $v' \in V$. Then, for a state $q = (W, D, v)$ and a label $*_i$ ($i \neq 0$), we consider four cases:

1. $i \in D \cap W$: Such a state will never be reachable by construction.
2. $i \in D \cap \overline{W}$: In this case, we just propagate the information D and W. I.e. $\delta(q, *_i) = \{\rho\}$ s.t. $\rho(v') = (W, D, v')$ for all $(v, v') \in E$.
3. $i \in W \cap \overline{D}$: In this case, one has to check that Player i has a winning strategy in some successor of v', which is guessed non-deterministically, and to which the W information is sent. I.e. $\delta(q, *_i) = \{\rho_{v'} \mid (v, v') \in E\}$ where $\rho_{v'}(v') = (W, D, v')$ and $\rho_{v'}(v'') = (W \backslash \{i\}, D \cup \{i\}, v')$ for all $v'' \neq v'$.
4. $i \in \overline{W} \cap \overline{D}$: In this case, either we do not guess anything, or we guess that Player i has a good deviation, and update the sets W and D accordingly. I.e. $\delta(q, *_i) = \{\rho\} \cup \{\gamma_{v'} \mid (v, v') \in E\}$ where $\gamma_{v'}(v') = (W \cup \{i\}, D, v')$ and $\gamma_{v'}(v'') = (W, D \cup \{i\}, v')$ for all $v'' \neq v'$.

Along a path of a run of $\mathcal{T}_{\mathcal{A}}$, there are monotonicity properties for the W and D-components of the states. Indeed, by construction, $\mathcal{T}_{\mathcal{A}}$ 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 $\lim_W(\eta)$.

Finally, the accepting condition $\alpha_{\mathcal{A}}(\mathcal{O})$ asks that on each path of the accepting run, either it is of the form $Q^*\{\bot\}^\omega$, or Player 0 wins, or there is a player that loses but belongs to some D eventually. For safety objectives, we also have to add the constraint that the losing player belongs to D before visiting an unsafe state. Additionally, the accepting condition also expresses constraints on the W components: each player $i \in \lim_W(\eta)$ wins. Formally, if we denote by $\mathrm{IRuns}(\mathcal{T}_{\mathcal{A}})$ the set of images of branches of runs of $\mathcal{T}_{\mathcal{A}}$, and by $\eta|_V$ the V-projection of any $\eta \in (Q\backslash\{\bot\})^\omega$, we have:

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

where the formula $\varphi_{\exists dev}(i, \eta)$ says that there is a good deviation for Player i. That is, if $D_0 D_1 \ldots$ is the sequence of D-components in η, then $\varphi_{\exists dev}(i, \eta) = i \in \lim_D(\eta)$ for tail or reachability objectives, and $\varphi_{\exists dev}(i, \eta) = \exists p \geqslant 0, i \in D_p \wedge \forall p' \leqslant p, \eta|_V[p] \in S_i$ for safety conditions $\mathsf{Safe}(S_i)$. Details can be found in the technical report. ◀

Tree automata emptiness. We now study the complexity of testing non-emptiness of the languages $\mathcal{L}_{\alpha_{\mathcal{A}}(\mathcal{O})}(\mathcal{T}_{\mathcal{A}})$ for the objectives $\mathcal{O}$ 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 $\mathcal{G}' = \langle V_E, V_A, E', q_0 \rangle$ where $V_E = Q$, $V_A = \{\mathsf{Range}(f) \mid \exists q \in Q, \alpha \in \Sigma, f \in \delta(q, \alpha)\}$. Then, the transition relations is defined for all $q \in Q$, all $P \in V_A$, by $(q, P) \in E'$ if there exists $\alpha \in \Sigma$ and $f \in \delta(q, \alpha)$ s.t. $P = \mathsf{Range}(f)$, and $(P, q) \in E'$ if $q \in P$. That is, 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 $\alpha_{\mathcal{A}}(\mathcal{O})$, one only

needs to remember $\mathsf{Range}(f)$. Adam then picks a state in that set. Eve's objective is the set $\{q_1w_1q_2w_2\cdots \in (V_EV_A)^\omega \mid q_1q_2\cdots \in \alpha_{\mathcal{A}}(\mathcal{O})\}$. It can be shown that Eve has a winning strategy in $\mathcal{G}'$ iff $\mathcal{L}_{\alpha_{\mathcal{A}}(\mathcal{O})}(\mathcal{T}_{\mathcal{A}}) \neq \varnothing$. By a fine analysis of solving this game, we obtain:

▶ **Proposition 6.** *Let $\mathcal{G} = \langle \mathcal{A}, \mathcal{O} = (\mathcal{O}_i)_{i\in\Omega}\rangle$ be a multiplayer game. Non-emptiness of $\mathcal{L}_{\alpha_{\mathcal{A}}(\mathcal{O})}(\mathcal{T}_{\mathcal{A}})$ can be checked in* PSpace *for $\mathcal{O} \in \{\mathsf{Safety}, \mathsf{Reachability}, \mathsf{Büchi}, \mathsf{coBüchi}\}$, and in* ExpTime *for $\mathcal{O} \in \{\mathsf{Parity}, \mathsf{Streett}, \mathsf{Rabin}, \mathsf{Muller}\}$.*

Proof. It amounts to study the complexity of solving $\mathcal{G}'$ for the objectives of this paper. Note that the arena of $\mathcal{G}'$ has linear size in the size of $\mathcal{T}_{\mathcal{A}}$. For safety, reachability, Büchi and coBüchi winning objectives, we reduce the problem of solving $\mathcal{G}'$ to a finite duration reachability tree game of exponential size, whose duration is polynomial (in the size of the original arena of $\mathcal{G}$). This reduction exploits the monotonicity of the sets W and D, and the fact that the game can be stopped once a cycle has been formed. It is similar in spirit to the technique of *first-cycle game* from [1] but the winning condition of $\mathcal{G}'$ do not fall in the general hypothesis of [1] under which infinite duration games reduce to first-cycle games. Our finite duration tree game, though of exponential size, is not constructed explicitly but solved on-the-fly by a PTime alternating algorithm. This gives a PSpace upper-bound for NCRSP.

We now study the complexity of solving $\mathcal{G}'$ when the original game $\mathcal{G}$ has Muller conditions $\mathsf{Muller}(\mu_i)$ for the $k+1$ players. We transform $\mathcal{G}'$ 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)* [15, 26], which allows us to identify states in V appearing infinitely often. Formally, V is assumed to be linearly ordered, i.e. $V = \{v_1, \ldots, v_n\}$. We let $P(V)$ the set of permutations of V, which we represent by words of length n over V with pairwise different letters. 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 $(m, h) \xrightarrow{v} (x_1x_2v, |x_1|)$ where $m = x_1vx_2 \in P(V)$ for some $x_1, x_2 \in V^*$ and $v \in V$. Positions h are called *hits*. LAR_V has the following property: take a sequence $\pi \in V^\omega$ and the associated sequence of LAR_V's states $\ell = (m_0, h_0)(m_1, h_1)\ldots$, let h^π_{min} be the smallest hit appearing infinitely often in ℓ, then the sequence of subsets $(\{m_i[r] \mid r \geqslant h^\pi_{min}\})_{i\geqslant 0}$ eventually equals $\inf(\pi)$.

Remind that $Q = \{q_0, \top\} \cup (2^\Omega \times 2^\Omega \times V)$, therefore each sequence of states not in $Q^*\{\top\}^\omega$ also gives the sequence of visited vertices of $\mathcal{G}$. We take the product of $\mathcal{G}'$ with LAR_V to add LAR information to the game $\mathcal{G}'$, and transform Eve's winning condition in $\mathcal{G}'$ into a parity condition pr in this product as follows: one uses $2|V|+2$ priorities and, for product states with q_0 or $\top$ as first component, the priority is 0, otherwise for states of the form $((W, D, v), (m, h))$, we let:

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

For states whose first component belongs to Adam, we just put priority $2|V|+2$, so that they have no influence. That way we obtain a parity game $\mathcal{G}_{par}$ with an exponential number of states (in $\mathcal{G}$) but a polynomial number of priorities, in which Eve has a winning strategy iff she has a winning strategy in $\mathcal{G}'$. Finally, parity games can be solved in PTime in the number of states and exponential in the number of priorities [18, 22]. ◀

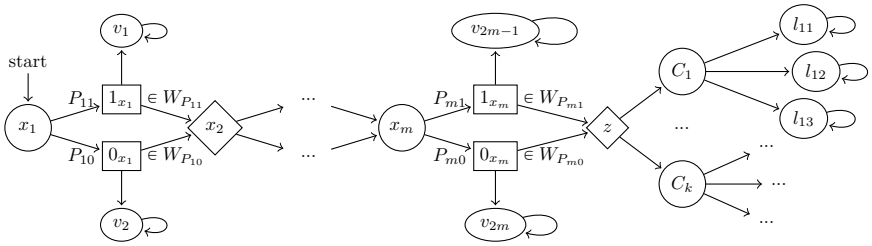

Figure 3 The arena $\mathcal{A}_\psi$ used to show that NCRSP is PSPACE-h.

▶ **Theorem 7.** *For multiplayer games, NCRSP is in* ExpTime *for objectives of type* $X \in \{\mathsf{Parity}, \mathsf{Streett}, \mathsf{Rabin}, \mathsf{Muller}\}$ *and in* PSpace *if* $X \in \{\mathsf{Safety}, \mathsf{Reachability}, \mathsf{Büchi}, \mathsf{coBüchi}\}$. *It is* PSpace-hard *for all objectives.*

Proof. The upper-bounds are consequence of Lemma 4, Lemma 5, and Proposition 6. Let us establish the lower bounds. Our proof is a reduction from QBF which uniformly works for all the types of objective. 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 $C_1, C_2, ..., C_k$. We build a $2m + 2$ players game $\mathcal{G}_\psi$, with players $\Omega = \{A, B, P_{10}, P_{11}, P_{20}, P_{21}, \ldots, P_{m0}, P_{m1}\}$, the system being played by player A, such that ψ is true if and only if $\mathcal{G}_\psi$ admits a solution to the NCRSP. The arena $\mathcal{A}_\psi$ of the game is depicted in Figure 3.

For each existential (resp. universal) variable x_i in ψ, the arena $\mathcal{A}_\psi$ contains a rounded (resp. diamond) state x_i controlled by Player A (resp. player B). For each node $x_i, 1 \leqslant i < m$, $\mathcal{A}_\psi$ contains the edges $(x_i, 0_{x_i}), (x_i, 1_{x_i})$, as well as the edges $(0_{x_i}, x_{i+1}), (1_{x_i}, x_{i+1})$, choices of edges in those states naturally encode valuations of the variables of the QBF formula.

For each $1 \leqslant i \leqslant m$, the rectangle state 1_{x_i} (resp. 0_{x_i}) is controlled by player P_{i1} (resp. P_{i0}) and has an additional edge leading to the self-loop over the state v_{2i-1} (resp. v_{2i}). 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. From there Player B can choose a clause, i.e. an edge $(z, C_i), 1 \leqslant i \leqslant k$. Finally, each state associated to a clause C_i is controlled by Player A and 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 X-game $\mathcal{G}_\psi$, it is easy to phrase with the different types of objectives, the following winning conditions:

1. All paths that end in a state labeled with v_i $(1 \leqslant i \leqslant 2m)$ are winning for all the players.
2. All paths that end in a state labeled with l_{ij} $(1 \leqslant i \leqslant k, 1 \leqslant i \leqslant 3)$ are winning for all the players but Player A and Player P_{hb} such that $(l_{ij} = x_h \wedge b = 1) \vee (l_{ij} = \neg x_h \wedge b = 0)$.

Let us establish the correctness of our reduction. Assume that ψ is true. Then, the existential player has a winning strategy in the QBF game associated to ψ, i.e. he can choose the value of existentially quantified variables x_i as a function of the values given by the universal player to the universally quantified variables x_j, with $j \leqslant i$, so that ψ evaluates to true. We claim that, if Player A plays in the game arena according to such a winning strategy of the existential player, then Player A wins his objective whenever the other players play a NE. Indeed, there are two possibilities: either the game ends in a looping state labeled with v_i, and all players win (including Player A) or the game reaches z where Player B chooses a clause. As the strategy played by Player B encodes a winning strategy of the existential player in the QBF game, we know that each clause is satisfied, so Player A can choose a literal that evaluates to true in the clause. With such a choice, he makes sure that the player associated to this literal is losing while the play has visited a state in which this player has

decided to continue the game instead of going to a looping state labeled with v_i. This means that this player has a profitable deviation in the profile associated to the outcome of the game and so this outcome of the game is not a NE.

Otherwise, ψ is false. Then the universal player has a winning strategy in the QBF game associated to ψ. Consider a strategy profile where Player B plays according to this strategy and each Player P_{ib}, $1 \leqslant i \leqslant m, b \in \{0,1\}$, plays to continue the game and avoid looping states labelled by v_i. Then the game reaches state z, and 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 Player A from C_i, the play will be winning for all the players with the exception of Player A, and the player associated to the literal that has been chosen by Player A in the clause. This outcome is part of a NE as the later player cannot improve on his payoff by deviating as in the current profile, the play does not visit the rectangle state associated to the literal that evaluates to false, and so he has no available deviation at all. ◀

5 Fixed number of players

Several instances of the rational synthesis problems become tractable for fixed number of players. The number of players is a natural parameter to study: in practical applications, the number of components composing the environment may be limited to a few.

Cooperative Setting. We provide a generic reduction for the lower bounds of Table 1.

▶ **Lemma 8.** *Let $X \in \{\mathsf{Safety}, \mathsf{Reachability}, \mathsf{Büchi}, \mathsf{coBüchi}, \mathsf{Parity}, \mathsf{Streett}, \mathsf{Rabin}, \mathsf{Muller}\}$. Given a two-player zero-sum game between players A and B with an objective of type X for Player A, we can construct a multiplayer game with objective of type 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 two-players CRSP $\mathcal{G}'$ 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 there 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 iff Player A does not have a winning strategy in $\mathcal{G}$ for forcing ψ. ◀

As an example, we get NP-hardness for the CRSP problem with Streett objectives because two-player zero-sum Streett games are coNP-hard.

The upper bounds on k-fixed CRSP for $X \in \{\mathsf{Reach}, \mathsf{Safe}, \mathsf{Büchi}, \mathsf{coBüchi}, \mathsf{Parity}, \mathsf{Rabin}\}$ listed in Table 1 can be obtained with the following procedure. First, compute the winning sets W_i for each of the $k+1$ players using the classical algorithms to solve two-player zero-sum games (w.r.t. the corresponding objective) and label the arena with the information $(W_i)_{i\in\Omega}$ (seen as atomic propositions). Then, check whether there is a path π such that $\pi \models \varphi_0 \wedge \phi^{\mathcal{G}}_{0\text{Nash}}$ (witnessing a solution to CRSP, by Proposition 2). The first step can be done in polynomial time for $X \in \{\mathsf{Reach}, \mathsf{Safe}, \mathsf{Büchi}, \mathsf{coBüchi}\}$, in UP$\cap$coUP for parity

conditions and in P^{NP} for Rabin conditions (as checking whether a state belongs to W_i is in NP). Due to the assumption that k is a fixed constant, the second step can be done in PTime for $X \in \{\mathsf{Reach}, \mathsf{Safe}, \mathsf{Büchi}, \mathsf{coBüchi}, \mathsf{Parity}\}$ and in NP for Rabin conditions. For Streett k-fixed CRSP, an NP upper bound was given in [25]. For Muller objectives, by Lemma 8, the problem is PSpace-hard and the PSpace upper bound follows from the unfixed case.

Non-Cooperative Setting. Results are summarized in the last column of Table 1. We start by the justification of the lower bounds. First, it should be clear that deciding the winner in a zero-sum two-player game with objective of type X is a special case of the NCRSP problem. Indeed, assume Player A has objective ψ in a game $\mathcal{G}$. Then, if we give objective ψ to Player 0 on the same game arena and declare all plays winning for Player 1, it is easy to see that this is a positive instance to the NCRSP problem iff Player A has a winning strategy in $\mathcal{G}$ for ψ. So this explains all the lower bounds but for $X \in \{\mathsf{Parity}, \mathsf{Streett}, \mathsf{Rabin}\}$. In the long version of this work [12], we provide a PSpace lower bound also to Streett and Rabin k-fixed NCRSP. This is done in two steps: first a reduction from QBF to zero-sum two-player Muller games is provided, similar to the one given in [16]. Then, the latter is reduced to a Street (resp. Rabin) NCRSP with two players. Finally, the lower bounds for parity k-fixed NCRSP reported in Table 1 have been obtained by reduction from the generalized parity games considered in [10], where the objective is a disjunction (dually, a conjunction) of parity conditions. In particular, we have proven that NCRSP on 3-players (resp. 4-players) parity game is NP (resp. coNP)-hard.

For $X \in \{\mathsf{Safety}, \mathsf{Reachability}, \mathsf{Büchi}, \mathsf{coBüchi}\}$, if the number of players in the given NCRSP is a fixed constant k, then the two-player game construction to test tree automata emptiness in Section 4 yields a polynomial size two-player zero-sum game $\mathcal{G}'$, where the formula characterizing the winning condition has constant size. The latter can be converted into an equivalent deterministic Büchi tree automaton of polynomial size, whose product with $\mathcal{G}'$ gives a Büchi game, solvable in PTime. Finally, if $X \in \{\mathsf{Parity}, \mathsf{Streett}, \mathsf{Rabin}, \mathsf{Muller}\}$, the corresponding k-fixed NCRSP can be reduced to a polynomial-size two-player game, with a 0-sum Muller acceptance condition defined by a boolean formula of polynomial size. Hence, the PSpace upper bound listed in Table 1 applies to k-fixed NCRSP with the above objectives.

References

1. Benjamin Aminof and Sasha Rubin. First cycle games. In *Proceedings 2nd International Workshop on Strategic Reasoning, SR 2014, Grenoble, France, April 5-6, 2014.*, pages 83–90, 2014. doi:10.4204/EPTCS.146.11.
2. Christel Baier and Joost-Pieter Katoen. *Principles of model checking*. MIT Press, 2008.
3. Dietmar 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.
4. Romain Brenguier, Lorenzo Clemente, Paul Hunter, Guillermo A. Pérez, Mickael Randour, Jean-François Raskin, Ocan Sankur, and Mathieu Sassolas. Non-zero sum games for reactive synthesis. *CoRR*, abs/1512.05568, 2015. URL: http://arxiv.org/abs/1512.05568.
5. Romain Brenguier, Jean-François Raskin, and Ocan 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. doi:10.4230/LIPIcs.CONCUR.2015.100.

6 Romain Brenguier, Jean-François Raskin, and Mathieu 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.

7 Krishnendu Chatterjee, Laurent Doyen, Emmanuel Filiot, and Jean-François 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.

8 Krishnendu Chatterjee and Thomas 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.

9 Krishnendu Chatterjee, Thomas A. Henzinger, and Marcin Jurdzinski. Games with secure equilibria. *Theor. Comput. Sci.*, 365(1-2):67–82, 2006. doi:10.1016/j.tcs.2006.07.032.

10 Krishnendu Chatterjee, Thomas A. Henzinger, and Nir 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. URL: http://dl.acm.org/citation.cfm?id=1760037.1760051.

11 Alessandro Cimatti, Edmund M. Clarke, Enrico Giunchiglia, Fausto Giunchiglia, Marco Pistore, Marco Roveri, Roberto Sebastiani, and Armando Tacchella. Nusmv 2: An open-source 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.

12 Rodica Condurache, Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The complexity of rational synthesis – full version. Available at: http://lacl.fr/~rbozianu/papers/fullRationalSynthesis.pdf, 2016.

13 Marco 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.

14 Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational synthesis. *CoRR*, abs/0907.3019, 2009. URL: http://arxiv.org/abs/0907.3019.

15 Yuri Gurevich and Leo 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. doi:10.1145/800070.802177.

16 Paul Hunter and Anuj 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.

17 M. Jurdzinski. Algorithms for solving parity games. In *Lectures in Game Theory for Computer Scientists*, pages 74–98. Cambridge University Press, 2011.

18 Marcin Jurdzinski, Mike Paterson, and Uri 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. URL: http://dl.acm.org/citation.cfm?id=1109557.1109571.

19 Orna Kupferman, Giuseppe Perelli, and Moshe 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. doi:10.1007/978-3-319-17130-2_15.

20 Fabio Mogavero, Aniello Murano, Giuseppe Perelli, and Moshe 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 Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In *POPL*, pages 179–190. ACM Press, 1989. doi:10.1145/75277.75293.

22 Sven 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. doi:10.1007/978-3-540-77050-3_37.

23 Wolfgang Thomas. On the synthesis of strategies in infinite games. In *STACS*, pages 1–13, 1995. doi:10.1007/3-540-59042-0_57.

24 Michael 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 Michael 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. doi:10.1007/978-3-540-78499-9_3.

26 Wieslaw Zielonka. Infinite games on finitely coloured graphs with applications to automata on infinite trees. *Theor. Comput. Sci.*, 200(1-2):135–183, 1998. doi:10.1016/S0304-3975(98)00009-7.