

# The Complexity of Rational Synthesis for Concurrent Games

**Rodica Condurache**

Computer Science Department, “A.I.Cuza” University, Iași  
700483, ROMANIA  
rodica.b.condurache@gmail.com

**Youssef Oualhadj**

Université Paris Est Créteil, LACL(EA 4219), UPEC  
94010 Créteil Cedex, France  
youssef.oualhadj@lacl.fr

**Nicolas Troquard**

The KRDB Research Centre, Free University of Bozen-Bolzano  
I-39100 Bozen-Bolzano BZ, Italy  
nicolas.troquard@unibz.it

---

## Abstract

In this paper, we investigate the rational synthesis problem for concurrent game structures for a variety of objectives ranging from reachability to Muller condition. We propose a new algorithm that establishes the decidability of the non cooperative rational synthesis problem that relies solely on game theoretic techniques as opposed to previous approaches that are logic based. Given an instance of the rational synthesis problem, we construct a zero-sum turn-based game that can be adapted to each one of the class of objectives. We obtain new complexity results. In particular, we show that in the cases of reachability, safety, Büchi, and co-Büchi objectives the problem is in PSPACE, providing a tight upper-bound to the PSPACE-hardness already established for turn-based games. In the case of Muller objective the problem is in EXPTIME. We also obtain positive results when we assume a fixed number of agents, in which case the problem falls into PTIME for reachability, safety, Büchi, and co-Büchi objectives.

**2012 ACM Subject Classification** Theory of computation → Algorithmic game theory, Theory of computation → Solution concepts in game theory

**Keywords and phrases** Synthesis, concurrent games, Nash equilibria

**Digital Object Identifier** 10.4230/LIPIcs.CONCUR.2018.38

**Related Version** A full version of the paper is available at <https://arxiv.org/abs/1707.06936>.

**Funding** The second author is supported by the *RFSI DIM McSYS* project.

**Acknowledgements** The first author did part of this work when she was member of Université Paris Est Créteil, LACL, UPEC.

## 1 Introduction

The synthesis problem aims at automatically designing a program from a given specification. Several applications for this formal problem can be found in the design of *interactive systems* i.e., systems interacting with an environment. From a formal point of view, the synthesis



© Rodica Condurache, Youssef Oualhadj, and Nicolas Troquard;  
licensed under Creative Commons License CC-BY

29th International Conference on Concurrency Theory (CONCUR 2018).

Editors: Sven Schewe and Lijun Zhang; Article No. 38; pp. 38:1–38:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

problem is traditionally modelled as a zero-sum turn-based game. The system and the environment are modeled by two players with opposite interest. The goal of the system is the desired specification. Hence, a *strategy* that allows the system to achieve its goal against any behavior of the environment is a winning strategy and is exactly the program to synthesize.

For a time, the described approach was the standard in the realm of controller synthesis. However, due to the variety of systems to model, such a pessimistic view is not always the most faithful one. For instance, consider a system that consists of a server and  $n$  clients. Assuming that all the agents have opposite interests is not a realistic assumption. Indeed, from a design perspective, the purpose of the server is to handle the incoming requests. On the other hand, each client is only concerned with its own request and wants it granted. None of the agents involved in the described interaction have antagonistic purposes. The setting of *non-zero-sum games* was proposed as model with more realistic assumptions.

In a non zero-sum game, each agent is equipped with a personal objective and the system is just a regular agent in the game. The agents interact together aiming at achieving the *best outcome*. The best outcome in this setting is often formalized by the concept of Nash equilibria. Unfortunately, a solution in this setting offers no guarantee that a specification for a given agent is achieved, and in a synthesis context one wants to enforce a specification for one or a subset of the agents.

The *rational synthesis problem* was introduced as a generalization of the synthesis problem to environment with multiple agents [4]. It aims at synthesizing a Nash equilibrium such that the induced behavior satisfies a given specification. This vision enjoys nice algorithmic properties since it matches the complexity bound of the classical synthesis problem. Later on, yet another version of the problem was proposed where the agents are rational but not cooperative [6, 7]. In the former formalization, the specification is guaranteed as long as the agents agree to behave according to the chosen equilibrium. But anything can happen if not, in particular they can play another equilibrium that does not satisfy the specification. In the Non Cooperative Rational Synthesis (NCRSP), the system has to ensure that the specification holds in any equilibrium (c.f., Section 3 for a formal definition and Figure 1a for an example). A solution for both problems was presented for specifications expressed in Linear Temporal Logic (LTL). The proposed solution relies on the fact that the problem can be expressed in a decidable fragment of a logic called *Strategy Logic*. The presented algorithm runs in 2-EXPTIME. While expressing the problem in a decidable fragment of Strategy Logic gives an immediate solution, it could also hide a great deal of structural properties. Such properties could be exploited in a hope of designing faster algorithms for less expressive objectives. In particular, specifications such as reachability, liveness, fairness, *etc.*

In [3], the first author took part in a piece of work where they considered this very problem for specific objectives such as reachability, safety, Büchi, *etc* in a turn-based interaction model. They established complexity bounds for each objective.

In this paper we consider the problem of non-cooperative rational synthesis with concurrent interactions. We address this problem for a variety of objectives and give exact complexity bounds relying exclusively on techniques inspired by the theory of zero-sum games. The concurrency between agents raises a formal challenge to overcome as the techniques used in [3] do not directly extend. Intuitively, when the interaction is turn-based, one can construct a tree automaton that accepts solutions for the rational synthesis problem. The nodes of an accepted tree are exactly the vertices of the game. This helps a lot in dealing with deviations but cannot be used in concurrent games.

In Section 3, we present an alternative algorithm that solves the general problem for LTL specification. This algorithm constructs a zero-sum turn-based game. This fresh game is played between *Constructor* who tries to construct a solution and *Spoiler* who tries to falsify

the proposed solution. We then show in Section 5 how to use this algorithm to solve the NCRSP for reachability, safety, Büchi, co-Büchi, and Muller conditions. We also observe that we match the complexity results for the NCRSP in turn-based games.

## 2 Preliminaries

### 2.1 Concurrent Game Structures

► **Definition 1.** A game structure is defined as a tuple  $\mathcal{G} = (\text{St}, s_0, \text{Agt}, (\text{Act}_i)_{i \in \text{Agt}}, \text{Tab})$ , where  $\text{St}$  is the set of states in the game,  $s_0$  is the initial state,  $\text{Agt} = \{0, 1, \dots, n\}$  is the set of agents,  $\text{Act}_i$  is the set of actions of Agent  $i$ ,  $\text{Tab} : \text{St} \times \prod_{i \in \text{Agt}} \text{Act}_i \rightarrow \text{St}$  is the transition table.

► **Remark.** Note that we consider game structures that are complete and deterministic. That is, from each state  $s$  and any tuple of actions  $\bar{a} \in \prod_{i \in \text{Agt}} \text{Act}_i$ , there is exactly one successor state  $s'$ .

► **Definition 2.** A *play* in the game structure is a sequence of states and actions profile  $\rho = s_0 \bar{a}_0 s_1 \bar{a}_1 s_2 \bar{a}_2 \dots$  in  $(\text{St} \times \prod_{i \in \text{Agt}} \text{Act}_i)^\omega$  such that  $s_0$  is the initial state and for all  $j \geq 0$ ,  $s_{j+1} = \text{Tab}(s_j, \bar{a}_j)$ .

Throughout the paper, for every word  $w$ , over any alphabet, we denote by  $w[j]$  the  $j + 1$ -th letter, and we denote by  $w[0..j]$  the prefix of  $w$  of size  $j + 1$ .

By  $\rho \upharpoonright_{\text{St}}$  we mean the projection of  $\rho$  over  $\text{St}$ , and  $\text{Plays}(\mathcal{G})$  is the set of all the plays in the game structure  $\mathcal{G}$ . We call history any finite sequence in  $\text{St} \times (\prod_{i \in \text{Agt}} \text{Act}_i \text{St})^*$ . For a history  $h$ , we denote by  $h \upharpoonright_{\text{St}}$  its projection over  $\text{St}$ , and by  $\text{Last}_{\text{St}}(h)$  the last element of  $h \upharpoonright_{\text{St}}$ . We denote by  $\text{Hist}$  the set of all the histories.

In this paper we allow agents to see the actions played between states. Therefore, they behave depending on the past sequence of states and tuples of actions.

► **Definition 3 (Strategy and strategy profile).** A *strategy* for Agent  $i$  is a mapping  $\sigma_i : \text{St} \times (\prod_{i \in \text{Agt}} \text{Act}_i \text{St})^* \rightarrow \text{Act}_i$ .

A *strategy profile* is defined as a tuple of strategies  $\bar{\sigma} = \langle \sigma_0, \sigma_1, \dots, \sigma_n \rangle$  and by  $\bar{\sigma}[i]$  we denote the strategy of  $i$ -th position (of Agent  $i$ ).

Also,  $\bar{\sigma}_{-i}$  is the partial strategy profile obtained from the strategy profile  $\bar{\sigma}$  from which the strategy of Agent  $i$  is ignored. The tuple of strategies  $\langle \bar{\sigma}_{-i}, \sigma'_i \rangle$  is obtained from the tuple  $\bar{\sigma}$  by substituting Agent  $i$ 's strategy with  $\sigma'_i$ .

Once a strategy profile is chosen it induces a play  $\rho$ . We say that a play  $\rho = s_0 \bar{a}_0 s_1 \bar{a}_1 s_2 \bar{a}_2 \dots$  in  $(\text{St} \times \prod_{i \in \text{Agt}} \text{Act}_i)^\omega$  is *compatible* with a strategy  $\sigma_i$  of Agent  $i$  if for every prefix of  $\rho[0..2k]$  with  $k \geq 0$ , we have  $\sigma(\rho[0..2k]) = \bar{a}_k(i)$ , where  $\bar{a}_k(i)$  is the action of Agent  $i$  in the vector  $\bar{a}_k$ .

We denote by  $\text{Plays}(\sigma_i)$  the set of all the plays that are compatible with the strategy  $\sigma_i$  for Agent  $i$ .  $\text{Hist}(\sigma_i)$  is the set of all the histories that are compatible with  $\sigma_i$ . The outcome of an interaction between agents following a certain strategy profile  $\bar{\sigma}$  defines a unique play in the game structure  $\mathcal{G}$  denoted  $\text{Out}(\bar{\sigma})$ . It is the unique play in  $\mathcal{G}$  compatible with all the strategies in the profile  $\bar{\sigma}$  which is an infinite sequence over  $(\text{St} \times \prod_{i \in \text{Agt}} \text{Act}_i)$ .

### 2.2 Payoff and Solution Concepts

Each Agent  $i \in \text{Agt}$  has an objective expressed as a set  $\text{Obj}_i$  of infinite sequences of states in  $\mathcal{G}$ . As defined before, a play  $\rho$  is a sequence of states and action profiles. We slightly abuse notation and also write  $\rho \in \text{Obj}_i$ , meaning that the sequence of states in the play  $\rho$  (that

is,  $\rho \upharpoonright_{\text{St}}$  is in  $\text{Obj}_i$ . We define the *payoff* function that associates with each play  $\rho$  a vector  $\text{Payoff}(\rho) \in \{0, 1\}^{n+1}$  defined by

$$\forall i \in \text{Agt}, \text{Payoff}_i(\rho) = 1 \iff \rho \in \text{Obj}_i .$$

We borrow game theoretic vocabulary and say that Agent  $i$  *wins* whenever her payoff is 1. We sometimes abuse this notation and write  $\text{Payoff}_i(\bar{\sigma})$ , which is the payoff of Agent  $i$  associated with the *unique* play induced by  $\bar{\sigma}$ .

In this paper we are interested in winning objectives such as Safety, Reachability, Büchi, coBüchi, and Muller that are defined as follows. Let  $\rho$  be a play in a concurrent game structure  $\mathcal{G}$ . We use the following notations:

$$\text{occ}(\rho) = \{s \in \text{St} \mid \exists j \geq 0 \text{ s.t. } \rho[j] = s\}$$

to denote the set of states that appear along  $\rho$  and

$$\text{inf}(\rho) = \{s \in \text{St} \mid \forall j \geq 0, \exists k \geq j \text{ s.t. } \rho[k] = s\}$$

to denote the set of states appearing infinitely often along  $\rho$ . Then,

- *Reachability*: For some  $T \subseteq \text{St}$ ,  $\text{REACH}(T) = \{\rho \in \text{St}^\omega \mid \text{occ}(\rho) \cap T \neq \emptyset\}$ ;
- *Safety*: For some  $T \subseteq \text{St}$ ,  $\text{SAFE}(T) = \{\rho \in \text{St}^\omega \mid \text{occ}(\rho) \subseteq T\}$ ;
- *Büchi*: For some  $T \subseteq \text{St}$ ,  $\text{BÜCHI}(T) = \{\rho \in \text{St}^\omega \mid \text{inf}(\rho) \cap T \neq \emptyset\}$ ;
- *coBüchi*: For some  $T \subseteq \text{St}$ ,  $\text{COBÜCHI}(T) = \{\rho \in \text{St}^\omega \mid \text{inf}(\rho) \cap T = \emptyset\}$ ;
- *Parity*: For some priority function  $p : \text{St} \rightarrow \mathbb{N}$ ,  $\text{PARITY}(p) = \{\rho \in \text{St}^\omega \mid \min\{p(s) \mid s \in \text{inf}(\rho)\} \text{ is even}\}$ ;
- *Muller*: For some boolean formula  $\mu$  over  $\text{St}$ ,  $\text{MULLER}(\mu) = \{\rho \in \text{St}^\omega \mid \text{inf}(\rho) \models \mu\}$ .

A Nash equilibrium is the formalisation of a situation where no agent can improve her payoff by unilaterally changing her behaviour. Formally:

► **Definition 4.** (Nash equilibrium) A strategy profile  $\bar{\sigma}$  is a Nash equilibrium (NE) if for every agent  $i$  and every strategy  $\sigma'$  of  $i$  the following holds true:

$$\text{Payoff}_i(\bar{\sigma}) \geq \text{Payoff}_i(\langle \bar{\sigma}_{-i}, \sigma' \rangle) .$$

Throughout this paper, we will assume that Agent 0 is the agent for whom we wish to synthesize the strategy, therefore, we use the concept of 0-fixed Nash equilibria.

► **Definition 5** (0-fixed Nash equilibrium). A profile  $\langle \sigma_0, \bar{\sigma}_{-0} \rangle$  is a 0-fixed NE (0-NE), if for every strategy  $\sigma'$  for agent  $i$  in  $\text{Agt} \setminus \{0\}$  the following holds true:

$$\text{Payoff}_i(\langle \sigma_0, \bar{\sigma}_{-0} \rangle) \geq \text{Payoff}_i(\langle \sigma_0, (\bar{\sigma}_{-0})_{-i}, \sigma' \rangle) .$$

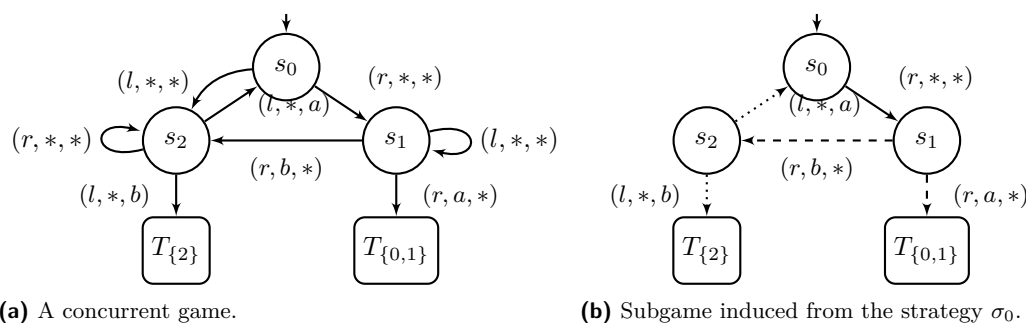
That is, fixing  $\sigma_0$  for Agent 0, the other agents cannot improve their payoff by unilaterally changing their strategy.

### 2.3 Rational synthesis

The rational synthesis can be defined in a optimistic or pessimistic setting. The former one is the so-called Cooperative Rational Synthesis (CRSP) Formally defined as

► **Problem 6.** *Is there a 0-NE  $\bar{\sigma}$  such that  $\text{Payoff}_0(\bar{\sigma}) = 1$ ?*

The latter is the so-called Non Cooperative Rational Synthesis Problem (NCRSP) and is formally defined as



■ **Figure 1**

► **Problem 7.** *Is there a strategy  $\sigma_0$  for Agent 0 such that for every 0-NE  $\bar{\sigma} = \langle \sigma_0, \bar{\sigma}_{-0} \rangle$ , we have  $\text{Payoff}_0(\bar{\sigma}) = 1$ ?*

In this paper we study computational complexity for the rational synthesis problem in both cooperative and non-cooperative settings.

For the CRSP, the complexity results are corollaries of existing work. In particular, for Safety, Reachability, Büchi, co-Büchi, Rabin and Muller objectives, we can apply algorithms from [2] to obtain the same complexities for CRSP as for the turn-based models when the number of agents is not fixed. More precisely, in [2] the problem of finding NE in concurrent games is tackled. In this problem one asks for the existence of NE whose payoff is between two thresholds. Then, by choosing the lower thresholds to be such that only Agent 0 satisfies her objective and the upper thresholds such that all agents win, we reduce to the cooperative rational synthesis problem. Brenguier et al. [2] showed that the existence of constrained NE in concurrent games can be solved in PTIME for Büchi objectives, NP for Safety, Reachability and coBüchi objectives, and PSPACE for Muller objectives. All hardness results are inferred directly from the hardness results in the turn-based setting. This is a consequence of the fact that every turn-based game can be encoded as a concurrent game by allowing at each state at most one agent to have non-vacuous choices. For Streett objectives, by reducing to [2] we only obtain PSPACE-easiness and the NP-hardness comes from the turn-based setting [3].

In the case of non-cooperative rational synthesis, we cannot directly apply the existing results. However, we define an algorithm inspired from the *suspect games* [2]. The suspect game was introduced to decide the existence of pure NE in concurrent games with  $\omega$ -regular objectives. We inspire ourselves from that approach and design a zero-sum game that combines the behaviors of Agent 0 and an extra entity whose goal is to prove, when needed, that the current play is not the outcome of a 0-NE. We also extend the idea in [3] that consists roughly in keeping track of deviations. Recall that the non-cooperative rational synthesis problem consists in designing a strategy  $\sigma_0$  for the protagonist (Agent 0 in our case) such that her objective  $\text{Obj}_0$  is satisfied by all the plays that are outcomes of 0-NE compatible with  $\sigma_0$ . This is equivalent to finding a strategy  $\sigma_0$  for Agent 0 such that for any play  $\rho$  compatible with it, either  $\rho$  satisfies  $\text{Obj}_0$ , or there is no strategy profile  $\bar{\sigma} = \langle \sigma_0, \bar{\sigma}_{-0} \rangle$  that is a 0-NE whose outcome is  $\rho$ .

► **Example 8.** Consider the concurrent game with reachability objectives depicted in Figure 1a. The game starts in the state  $s_0$ . There are three agents, the controller Agent 0, Agent 1, and Agent 2. Agent 0 has two actions  $r$  for right and  $l$  for left. Agents 1 and 2 have two actions, denoted  $a$  and  $b$ . For any subset  $C$  of  $\{0, 1, 2\}$ , the states  $T_C$  indicate that the agents in  $C$  have reached their objectives (These states are sinks). In addition, there are three states  $s_0$ ,

$s_1$ , and  $s_2$ . The edges represent the transitions table. The labels indicate the action profiles e.g. the vector  $(r, a, b)$  means that Agent 0 took action  $r$ , Agent 1 took action  $a$ , and Agent 2 took action  $b$ . Finally action  $*$  stands for the indifferent choice that is any action for a given agent. We can see that at  $s_0$ , Agent 0 is the only agent with non-vacuous choices. He can choose to go to  $s_1$  by playing action  $r$ , or to go to  $s_2$  by playing action  $l$ .

Now consider the strategy  $\sigma_0$  for Agent 0 defined as follows:  $\sigma_0(s_0) = r, \sigma_0(s_1) = r, \sigma_0(s_2) = l$ . We argue that this strategy is a solution to the NCRSP. Indeed, by applying this strategy, we obtain the subgame of Figure 1b. In this game, all the plays falsifying the objective of Agent 0 are the ones where Agent 1 plays  $b$ . Notice now that these plays are not outcomes of a 0-NE since Agent 1 can deviate by playing action  $a$ .

### 3 Solution for Problem 6

We will now describe a general algorithm that solves the NCRSP. As a first step in our procedure, we construct a two-player turn based game.

#### 3.1 Construction of a two-player game

Given a concurrent game  $\mathcal{G} = (\text{St}, s_0, \text{Agt}, (\text{Act}_i)_{i \in \text{Agt}}, \text{Tab})$  we construct a turn-based 2-player zero-sum game  $\mathcal{H} = (Q, q_0, \text{Act}_E, \text{Act}_A, \text{Tab}', \text{Obj})$ .

The game  $\mathcal{H}$  is obtained as follows:

- $q_0 = (s_0, \emptyset, \emptyset)$
- The set  $\text{Act}_E$  is  $\text{Act}_E^a \cup \text{Act}_E^c$  where:
  - $\text{Act}_E^a = \text{Act}_0 \times \prod_{i=1}^n (\text{Act}_i \cup \{-\})$
  - $\text{Act}_E^c = \prod_{i=1}^n (\text{Act}_i \cup \{-\})$ .
- The set  $\text{Act}_A$  is  $\prod_{i=1}^n \text{Act}_i$ .
- The set  $Q$  of states is  $Q_A \cup Q_B \cup Q_C \cup Q_D$  where

$$\begin{aligned} Q_A &= \text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}} \\ Q_B &= \text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}} \times \text{Act}_E^a \\ Q_C &= \text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}} \times \text{Act}_E^a \times \text{Act}_A \\ Q_D &= \text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}} \times \text{Act}_E^a \times \text{Act}_A \times \text{Act}_E^c . \end{aligned}$$

- Player Eve plays in the states in  $Q_A$  and  $Q_C$ , while Player Adam plays in the states in  $Q_B$  and  $Q_D$ . The legal moves are given as follows:
  - From a state  $(s, W, D) \in Q_A$ , Eve plays an action

$$\bar{a} \in \text{Act}_0 \times \prod_{i=1}^n (\text{Act}_i \cup \{-\}) \text{ s.t. } \forall 1 \leq i \leq n, \bar{a}[i] = - \iff i \notin W .$$

- From a state  $(s, W, D, \bar{a}) \in Q_B$ , Adam plays an action  $\bar{b} \in \text{Act}_A$ .
- From a state  $(s, W, D, \bar{a}, \bar{b}) \in Q_C$ , Eve plays an action

$$\bar{c} \in \prod_{i=1}^n (\text{Act}_i \cup \{-\}) \text{ s.t. } i \in W \cup D \implies \bar{c}[i] = - .$$

- From a state  $(s, W, D, \bar{a}, \bar{b}, \bar{c}) \in Q_D$ , Adam plays an action  $\bar{d} \in \text{Act}_A$ .

The transition  $\text{Tab}'$  and the objective  $\text{Obj}$  of the game  $\mathcal{H}$  are described next.

### 3.2 Transition function

The game  $\mathcal{H}$  is best understood as a dialogue between Eve and Adam. In each state  $(s, W, D)$  Eve proposes an action for Agent 0 together with the actions corresponding to the winning strategies of the agents in the set  $W$ . Then, Adam responds with an action profile played by all agents in the environment. In the next step, Eve knows the entire action profile played by the agents and proposes some new deviations for the agents that do not have a deviation yet (they are neither in  $W$  nor in  $D$ ). The last move is performed by Adam, it is his role to “check” that the proposed deviations and winning strategies are correct. Therefore, Adam can choose any continuation for the game and the sets  $W$  and  $D$  are updated according to the previous choices to some new values  $W'$  and  $D'$ . Each dialogue “round” is decomposed into four moves.

The transitions are given by the (partial) function  $\text{Tab}' : Q \times (\text{Act}_E \cup \text{Act}_A) \rightarrow Q$ :

- When  $(s, W, D) \in Q_A$ ,  $\text{Tab}'((s, W, D), \bar{a}) = (s, W, D, \bar{a})$ .
- When  $(s, W, D, \bar{a}) \in Q_B$ ,  $\text{Tab}'((s, W, D, \bar{a}), \bar{b}) = (s, W, D, \bar{a}, \bar{b})$ .
- When  $(s, W, D, \bar{a}, \bar{b}) \in Q_C$ ,  $\text{Tab}'((s, W, D, \bar{a}, \bar{b}), \bar{c}) = (s, W, D, \bar{a}, \bar{b}, \bar{c})$ .
- When  $(s, W, D, \bar{a}, \bar{b}, \bar{c}) \in Q_D$ ,  $\text{Tab}'((s, W, D, \bar{a}, \bar{b}, \bar{c}), \bar{d}) = (s', W', D')$ , such that:
  - $s' = \text{Tab}(s, (\bar{a}[0], \bar{d}))$ .
  - $W' = W \cup \{i \notin W \cup D \mid (\bar{d}[i] = \bar{c}[i]) \text{ and } (\forall j \in \text{Agt} \setminus \{0, i\}, \bar{d}[j] = \bar{b}[j])\} \setminus \{i \in W \mid \bar{d}[i] \neq \bar{a}[i]\}$ . That is, Agent  $i$  is added to the set  $W'$  on the continuations where Agent  $i$  plays the new action proposed by Eve in  $\bar{a}$  (supposedly compatible with a winning strategy) and the other agents do not change their actions with respect to  $\bar{d}$ . Also, any agent for whom Eve proposes an action in  $\bar{c}$  is a hint to Adam that this agent can deviate from that point. It is up to Adam to agree or not. If Adam agrees, we say that he has agreed with the recommendation of Eve. In this case, Eve has to prove that she made the right choice, this will be checked by the winning condition of the game.
  - $D' = D \cup \{i \in W \mid \bar{d}[i] \neq \bar{a}[i]\} \cup \{i \notin W \cup D \mid (\bar{c}[i] \neq -) \text{ and } (\bar{d}[i] \neq \bar{c}[i]) \text{ and } (\forall j \in \text{Agt} \setminus \{0, i\}, \bar{d}[j] = \bar{b}[j])\}$ . This is the opposite case where Adam stood by his choices, in this case the winning condition has to check that this was a wrong decision.

### 3.3 Winning condition

We equip  $Q$  with the canonical projection  $\pi_i$  that is the projection over the  $i$ -th component. In particular, for every  $(s, W, D) \in Q_A$ , we have  $\pi_1((s, W, D)) = s$ ,  $\pi_2((s, W, D)) = W$ , and  $\pi_3((s, W, D)) = D$ . We also extend  $\pi_i$  over  $Q^+$  and  $Q^\omega$  as expected. Histories for Eve are finite words in  $q_0(\text{Act}_E Q \text{Act}_A Q)^*$ . Histories for Adam are finite words in  $q_0(\text{Act}_E Q)^*$ . Plays are infinite sequence in  $q_0(\text{Act}_E Q \text{Act}_A Q)^\omega$ . Let  $r$  be a play, we denote  $r \upharpoonright_{Q_A}$  the restriction of  $r$  over the states in  $Q_A$  which is an infinite sequence in  $Q_A^\omega$ . The set  $\lim \pi_2(r \upharpoonright_{Q_A})$  (resp.  $\lim \pi_3(r \upharpoonright_{Q_A})$ ) is the set of agents in the limit of  $W$ 's (resp.  $D$ 's). The limit  $\lim \pi_3(r \upharpoonright_{Q_A})$  exists because the sets  $D$  occurring in the states  $Q$  along a play are non-decreasing subsets of  $\text{Agt}$ , and  $\text{Agt}$  is finite. The limit  $\lim \pi_2(r \upharpoonright_{Q_A})$  exists because (1) an agent is added into  $W$  only if it is not in  $D$ , and (2) when an agent leaves  $W$ , it gets into  $D$  indefinitely. This means that when an agent leaves from  $W$ , it never goes back.

We define the following sets:

$$S_0 = \{r \in Q^\omega \mid \pi_1(r \upharpoonright_{Q_A}) \in \text{Obj}_0\} \quad , \quad (1)$$

$$S_W = \{r \in Q^\omega \mid \forall i \in \lim \pi_2(r \upharpoonright_{Q_A}), \pi_1(r \upharpoonright_{Q_A}) \in \text{Obj}_i\} \quad , \quad (2)$$

$$S_D = \{r \in Q^\omega \mid \exists i \in \lim \pi_3(r \upharpoonright_{Q_A}), \pi_1(r \upharpoonright_{Q_A}) \notin \text{Obj}_i\} \quad . \quad (3)$$

$$\text{Obj} = (S_0 \cup S_D) \cap S_W \quad .$$

### 3.4 Transformations

#### Lifting of histories

We define a transformation over histories in  $\mathcal{G}$  to create histories in  $\mathcal{H}$ . For every strategy  $\sigma$  for Eve in  $\mathcal{H}$ , we define the transformation  $\mathcal{G}2\mathcal{H}_\sigma$ .

Let  $h$  be a history in  $\mathcal{G}$  and assume that  $h = s_0\bar{m}_0s_1\bar{m}_1\dots s_k\bar{m}_ks_{k+1}$ . The lifting of  $h$  is a history  $\tilde{h}$  in  $\mathcal{H}$  obtained by the mapping  $\mathcal{G}2\mathcal{H}_\sigma$  inductively defined as follows:

$$\mathcal{G}2\mathcal{H}_\sigma(s_0) = (s_0, \emptyset, \emptyset) \quad ,$$

and

$$\mathcal{G}2\mathcal{H}_\sigma(h) = \underbrace{\mathcal{G}2\mathcal{H}_\sigma(s_0\bar{m}_0s_1\bar{m}_1\dots s_k)}_{\tilde{h}'} \bar{a}q_b\bar{b}q_c\bar{c}q_d\bar{d}q_a \quad ,$$

where

$$\begin{aligned} \bar{a} &= \sigma(\tilde{h}') \quad , & q_b &= \text{Tab}'(\text{Last}(\tilde{h}'), \bar{a}) \quad , \\ \bar{b} &= \bar{m}_{k-0} \quad , & q_c &= \text{Tab}'(q_b, \bar{b}) \quad , \\ \bar{c} &= \sigma(\tilde{h}'\bar{a}q_b\bar{b}q_c) \quad , & q_d &= \text{Tab}'(q_c, \bar{c}) \quad , \\ \bar{d} &= \bar{b} = \bar{m}_{k-0} \quad , & q_a &= \text{Tab}'(q_d, \bar{d}) \quad . \end{aligned}$$

Observe that every history  $\mathcal{G}2\mathcal{H}_\sigma(h)$  ends in a state in  $Q_A$ , where Eve plays an action from  $\text{Act}_E^a$ , that always specifies an action for Agent 0. The function  $\mathcal{G}2\mathcal{H}_\sigma$  is thus instrumental in obtaining a strategy  $\sigma_0$  for Agent 0 in  $\mathcal{G}$  from a strategy of Player Eve in  $\mathcal{H}$ . For every history  $h$  in  $\mathcal{G}$ , we define:

$$\sigma_0(h) = \sigma(\mathcal{G}2\mathcal{H}_\sigma(h))[0] \quad . \quad (4)$$

For every strategy  $\sigma$  of Eve, we call 0-strategy the strategy obtained by Equation 4. The following claim is consequence of the same equation.

► **Claim 9.** *Let  $\sigma$  be a strategy for Eve, and let  $\sigma_0$  be the 0-strategy. If a history  $h$  in  $\mathcal{G}$  is compatible with  $\sigma_0$  then the history  $\tilde{h} = \mathcal{G}2\mathcal{H}_\sigma(h)$  in  $\mathcal{H}$  is compatible with  $\sigma$ .*

The function  $\mathcal{G}2\mathcal{H}_\sigma$  maps every history in  $\mathcal{G}$  into a history in  $\mathcal{H}$ . We define  $\mathcal{G}2\mathcal{H}_\sigma^\bullet$  as the natural extension of  $\mathcal{G}2\mathcal{H}_\sigma$  over the domain of plays in  $\mathcal{G}$ . We extend the previous claim as expected.

► **Claim 10.** *Let  $\sigma$  be a strategy for Eve, and let  $\sigma_0$  be the 0-strategy. If a run  $\rho$  in  $\mathcal{G}$  is compatible with  $\sigma_0$  then the run  $r = \mathcal{G}2\mathcal{H}_\sigma^\bullet(\rho)$  in  $\mathcal{H}$  is compatible with  $\sigma$ .*

► **Lemma 11.** *Let  $\sigma$  be a strategy for Eve, let  $\rho$  be a run in  $\mathcal{G}$  compatible with the 0-strategy  $\sigma_0$ . Let  $h$  be a history in  $\mathcal{G}$ , assume  $h$  to be a prefix of  $\rho$ . If  $\tilde{h} = \mathcal{G}2\mathcal{H}_\sigma(h)$  then  $\pi_1(\tilde{h} \upharpoonright_{Q_A}) = h \upharpoonright_{\text{St}}$ .*



**Proof.** By induction on the size of  $h$ . The base case is  $h = s_0$ , in which case  $\mathcal{G}2\mathcal{H}_\sigma(h) = \tilde{h} = (s_0, \emptyset, \emptyset)$ . We have  $\pi_1(\tilde{h} \upharpoonright_{Q_A}) = s_0 = h \upharpoonright_{St}$ . Now assume for induction that  $\pi_1(\tilde{h} \upharpoonright_{Q_A}) = h \upharpoonright_{St}$  for every history  $h = s_0\bar{m}_0s_1\bar{m}_1\dots s_k$  of size  $1 + 2k$  and let  $\mathcal{G}2\mathcal{H}_\sigma(h) = \tilde{h}$ .

Now consider the history  $hm_k s_{k+1}$  by definition  $\mathcal{G}2\mathcal{H}_\sigma(hm_k s_{k+1}) = \tilde{h}\bar{a}q_b\bar{b}q_c\bar{c}q_d\bar{d}q_a$  where  $\bar{a}, \bar{b}, \bar{c}, \bar{d}$  are obtained thanks to  $\mathcal{G}2\mathcal{H}_\sigma$ , by I.H.  $\pi_1(\tilde{h}) = s_0s_1\dots s_k$ , it thus suffices to show that  $\pi_1(q_a) = s_{k+1}$ . For this, one needs to remark that  $m_k = (\sigma(\tilde{h})[0], \bar{d})$ , and that

$$s_{k+1} = \text{Tab}(s_k, m_k) = \pi_1(\text{Tab}'((s, W, D, \bar{a}, \bar{b}, \bar{c}, \bar{d})) = \pi_1(q_a)$$

where the second equality is by definition of the construction.  $\blacktriangleleft$

Since the previous lemma is true for any histories that are respectively prefixes of  $r$  and  $\rho$  we obtain the following claim:

► **Claim 12.** *Let  $\sigma$  be a strategy for Eve, let  $\rho$  be a run in  $\mathcal{G}$  compatible with the 0-strategy  $\sigma_0$ . If  $r = \mathcal{G}2\mathcal{H}_\sigma^\bullet(\rho)$  then  $\pi_1(r \upharpoonright_{Q_A}) = \rho \upharpoonright_{St}$ .*

### Projection of histories

We now define in some sense the reverse operation. Let us define the transformation  $\mathcal{H}2\mathcal{G}$ .

Let  $\tilde{h}$  be a history in  $\mathcal{H}$  ending in a state in  $Q_A$ .

$$\mathcal{H}2\mathcal{G}(q_0) = s_0$$

$$\mathcal{H}2\mathcal{G}(\tilde{h}\bar{a}q_b\bar{b}q_c\bar{c}q_d\bar{d}q_a) = \underbrace{\mathcal{H}2\mathcal{G}(\tilde{h})}_{\text{induction}} \underbrace{(\bar{a}[0], \bar{d}_{-0})}_{\text{action}} q_a$$

► **Lemma 13.** *Let  $\tilde{h}$  be a run in  $\mathcal{H}$ ,  $h$  be a history in  $\mathcal{G}$ . If  $h = \mathcal{H}2\mathcal{G}(\tilde{h})$ , then  $\pi_1(\tilde{h} \upharpoonright_{Q_A}) = h \upharpoonright_{St}$*

**Proof.** By induction over the length of  $\tilde{h}$ . For  $\tilde{h} = (s_0, \emptyset, \emptyset)$  the result trivially true. Assume the result holds for any history  $\tilde{h}$  and let us show that it holds for  $\tilde{h}\bar{a}q_b\bar{b}q_c\bar{c}q_d\bar{d}q_a$ . By induction we have  $\pi_1(\tilde{h} \upharpoonright_{Q_A}) = h \upharpoonright_{St}$ , to conclude notice that

$$\pi_1(q_a) = \text{Tab}(\text{Last}(\mathcal{H}2\mathcal{G}(\tilde{h})), (\bar{a}[0], \bar{d}_{-0})) \quad \blacktriangleleft$$

The function  $\mathcal{H}2\mathcal{G}$  maps every history in  $\mathcal{H}$  ending in a state in  $Q_A$  into a history in  $\mathcal{G}$ . We define  $\mathcal{H}2\mathcal{G}^\bullet$  as the natural extension of  $\mathcal{H}2\mathcal{G}$  over the domain of runs in  $\mathcal{H}$ .

The following claim follows

► **Claim 14.** *Let  $r$  be a run in  $\mathcal{H}$ ,  $\rho$  be a run in  $\mathcal{G}$ . If  $\rho = \mathcal{H}2\mathcal{G}^\bullet(r)$ , then  $\pi_1(r \upharpoonright_{Q_A}) = \rho \upharpoonright_{St}$*

## 4 Main Theorem

► **Theorem 15.** *There exists a solution for the NCRSP iff Eve wins.*

We denote  $\sigma_i^h$  the strategy that mimics the strategy  $\sigma_i$  when the current history is  $h$  i.e.

$$\sigma_i^h(h') = \begin{cases} \sigma_i(h') & \text{if } h' \text{ is a prefix of } h \\ \sigma_i(h \cdot h') & \text{if } h \text{ is a prefix of } h' \\ \perp & \text{otherwise} \end{cases}$$

► **Definition 16.** Let  $\rho$  be a play and let  $h = s_0\bar{a}^0s_1\bar{a}^1\dots s_k\bar{a}^ks_{k+1}$  be a prefix of  $\rho$ . We say that  $h$  is a *good deviation point* for Agent  $i \in \text{Agt} \setminus \{0\}$  if:

## 38:10 The Complexity of Rational Synthesis for Concurrent Games

- $\rho \upharpoonright_{\text{St}} \notin \text{Obj}_i$  and,
- there exists a strategy  $\sigma'_i$  of Agent  $i$  from  $[h]$  such that for all  $(\sigma_j)_{j \in \text{Agt} \setminus \{0,i\}}$  we have:

$$[h] \cdot \text{Out} \left( \sigma_0^{[h]}, \dots, \sigma'_i, \dots, \sigma_n^{[h]} \right) \in \text{Obj}_i \quad , \quad \text{where}$$

$$[h] = \rho[0..k] \cdot \langle \bar{a}_{-i}^k, \sigma'_i(\rho[0..k]) \rangle \cdot \text{Tab} \left( s_k, \langle \bar{a}_{-i}^k, \sigma'_i(\rho[0..k]) \rangle \right) \quad .$$

We say that  $\rho$  has a *good deviation* if some prefix  $h$  of  $\rho$  is a good deviation point.

We use the notion of deviation point in the following lemma. This lemma states that a strategy  $\sigma_0$  is a solution for the NCRSP if any play  $\rho$  compatible with it, either is winning for Agent 0 or some Agent  $i$  would unilaterally deviate and win against any strategy profile of the other agents.

► **Lemma 17.** *A strategy  $\sigma_0$  is a solution for NCRSP iff every play  $\rho$  compatible with  $\sigma_0$  either  $\rho \upharpoonright_{\text{St}} \in \text{Obj}_0$  or,  $\rho$  has a good deviation.*

**Proof.** We start by establishing the if direction, let  $\sigma_0$  be a solution for the NCRSP. If any outcome  $\rho \in \text{Plays}(\sigma_0)$  is such that  $\rho \upharpoonright_{\text{St}} \in \text{Obj}_0$  then there is nothing to prove. Let  $\rho$  be a play in  $\text{Plays}(\sigma_0)$  such that  $\rho$  is not in  $\text{Obj}_0$ . Assume toward a contradiction that  $\rho$  does not contain a good deviation point. Then by Definition 16 we know that for any prefix  $h$  of  $\rho$ , any agent  $i \neq 0$  such that  $\text{Payoff}_i(\rho) = 0$ , and any strategy  $\tau_i$  of  $i$  there exists  $\sigma_1, \dots, \sigma_n$  strategies for agents 1 to  $n$  such that the following holds:

$$[h] \cdot \text{Out} \left( \sigma_0^h, \sigma_1^h, \dots, \tau_i^h, \dots, \sigma_n^h \right) \notin \text{Obj}_i \quad .$$

The above equation implies that Agent  $i$  does not have a profitable deviation under the strategy  $\sigma_0$ , hence the profile  $\langle \sigma_0, \dots, \sigma_n \rangle$  is a 0-fixed NE contradicting the fact that  $\sigma_0$  is a solution for the NCRSP.

For the only if direction, let  $\sigma_0$  be a strategy for agent 0, assume that every  $\rho$  in  $\text{Plays}(\sigma_0)$  satisfies

1.  $\rho \upharpoonright_{\text{St}} \in \text{Obj}_0$  or,
2.  $\rho$  has a good deviation.

If every play  $\rho$  in  $\text{Plays}(\sigma_0)$  is in  $\text{Obj}_0$  then  $\sigma_0$  is a solution for NCRSP. Let  $\rho$  be a play in  $\text{Plays}(\sigma_0)$  such that it is not in  $\text{Obj}_0$ . By assumption,  $\rho$  has a good deviation point i.e. there exists an Agent  $i \neq 0$  and a strategy  $\tau_i$  for the same agent such that: *i)*  $\rho \upharpoonright_{\text{St}} \notin \text{Obj}_i$  and *ii)* after a finite prefix  $h$  of  $\rho$  for any tuple of strategies  $(\sigma_j)_{j \in \text{Agt} \setminus \{0,i\}}$  the following holds:

$$[h] \cdot \text{Out} \left( \sigma_0^h, \sigma_1^h, \dots, \tau_i^h, \dots, \sigma_n^h \right) \in \text{Obj}_i \quad .$$

Hence,  $\rho$  is not the outcome of a 0-fixed NE and therefore  $\sigma_0$  is a solution for the NCRSP. ◀

### 4.1 Correctness

► **Definition 18.** Eve wins if she has a strategy that ensures Obj against any strategy of Adam.

► **Proposition 19.** *If Eve wins then there exists a solution for the NCRSP.*

**Proof.** Let  $\sigma_E$  be a winning strategy for Eve in  $\mathcal{H}$ , and let  $\sigma_0$  be the strategy for Agent 0 in  $\mathcal{G}$  obtained by the construction in Sec. 3.4 Equation (4), that is, for every history  $h$  in  $\mathcal{G}$ ,  $\sigma_0(h) = \sigma_E(\mathcal{G}2\mathcal{H}_{\sigma_E}(h))[0]$ . We show that  $\sigma_0$  is solution to the NCRSP.

Let  $\rho$  be an arbitrary run in  $\mathcal{G}$  compatible with  $\sigma_0$ .

According to Lemma 17 it is sufficient to show that  $\rho$  is in  $\text{Obj}_0$  or  $\rho$  has a good deviation point. Consider the run  $r = \mathcal{G}2\mathcal{H}_{\sigma_E}^\bullet(\rho)$  in  $\mathcal{H}$ . As a consequence of Claim 10, we have that  $r$  is compatible with  $\sigma_E$ . Since  $\sigma_E$  is winning, we also have  $r \in \text{Obj}$ , i.e.,

$$r \in (S_0 \cup S_D) \cap S_W = (S_0 \cap S_W) \cup (S_D \cap S_W) .$$

As a first case, assume that  $r \in S_0 \cap S_W$  implying  $\pi_1(r \upharpoonright_{Q_A}) \in \text{Obj}_0$ . By Claim 12 we can write  $\pi_1(r \upharpoonright_{Q_A}) = \rho \upharpoonright_{\text{St}}$ , and thus  $\rho \upharpoonright_{\text{St}} \in \text{Obj}_0$ .

As a second case, assume  $r \in S_D \cap S_W$ . It implies that there exists a state  $q_a$  in  $Q_A$  along  $r$  such that  $q_a = (s, W, D)$  and there exists an agent  $i$  in  $D$  such that  $i$  in  $\lim \pi_3(r \upharpoonright_{Q_A})$  and  $\pi_1(r \upharpoonright_{Q_A}) \notin \text{Obj}_i$ .

We argue that Agent  $i$  has a profitable deviation from a prefix of  $\rho$  entailing that  $\rho$  contains a good deviation point.

Assume w.l.o.g. that  $q_a$  is the first state along  $r$  for which there exists an Agent  $i$  in  $D$  such that  $i$  in  $\lim \pi_3(r \upharpoonright_{Q_A})$  and  $\pi_1(r \upharpoonright_{Q_A}) \notin \text{Obj}_i$ . The run  $r$  is of the form:

$$r = \tilde{h}\bar{a}p_b\bar{b}p_c\bar{c}p_d\bar{d}q_a\tilde{t} \tag{5}$$

where  $\tilde{h}$  is a finite prefix of  $r$  ending in a state in  $Q_A$ , and  $\tilde{t}$  is an infinite suffix. Remember also that  $r = \mathcal{G}2\mathcal{H}_{\sigma_E}^\bullet(\rho)$ , hence there exists a history  $h$  which is a prefix of  $\rho$  such that  $\tilde{h} = \mathcal{G}2\mathcal{H}(h)$ . We claim that  $h$  is a good deviation point (c.f. Definition 16) for Agent  $i$ . Indeed, we use notation  $\tau_i$  for the strategy defined only after  $h$  has occurred as follows:  $\tau_i(h) = \bar{c}[i]$  where  $\bar{c}$  is the action available for agent  $i$  in Equation (5), and for any history  $hh'$  in  $\mathcal{G}$ :  $\tau_i(hh') = \sigma_E(\mathcal{G}2\mathcal{H}(hh'))[i]$ . (Observe that by construction  $\mathcal{G}2\mathcal{H}(hh')$  always ends in a state in  $Q_A$ , controlled by Eve.)

We define the set  $T$  as the set of all the plays in  $\mathcal{G}$  that start with  $h$  and are compatible with  $\tau_i$ . Let  $\rho'$  be a play in  $T$ , and let  $r' = \mathcal{G}2\mathcal{H}(\rho')$  be a play in  $\mathcal{H}$ . The play  $r'$  enjoys two properties, first  $i \in \lim \pi_2(r' \upharpoonright_{Q_A})$  and second it is compatible with  $\sigma_E$ . Hence  $\pi_1(r' \upharpoonright_{Q_A}) \in \text{Obj}_i$ . This shows that  $\rho$  has a good deviation point after history  $h$ . By Lemma 17 we conclude that  $\sigma_0$  is solution to the problem NCRSP.  $\blacktriangleleft$

## 4.2 Completeness

► **Proposition 20.** *There exists a solution for the NCRSP then Eve wins.*

We first introduce some technical tools.

Deviator :  $\text{Hist}(\sigma_0) \times \text{Agt} \rightarrow \text{Act} \cup \{-\}$

$$(h, i) \mapsto \begin{cases} a & \text{if } h \text{ is a good deviation point for Agent } i \text{ using action } a, \\ - & \text{if not.} \end{cases}$$

Root :  $\text{Hist} \times \text{Agt} \rightarrow \text{Hist} \cup \{\perp\}$

$$(h, i) \mapsto \begin{cases} h' & \text{where } h' \text{ is the shortest prefix of } h \text{ s.t. } \text{Deviator}(h', i) \in \text{Act} \\ \perp & \text{if no such a prefix exists} \end{cases}$$

► **Claim 21.** *Let  $h$  be a history and  $i$  an agent s.t.  $\text{Deviator}(h, i) \in \text{Act}$ , then there exists a winning strategy  $\tau_i$  from  $\text{Root}(h, i)$  for agent  $i$ .*

Indeed, assuming that  $\text{Deviator}(h, i) \in \text{Act}$  and that there is no winning strategy from  $\text{Root}(h, i)$ , would entail that  $\text{Root}(h, i)$  is not a good deviation point.

**Proof of Proposition 20.** Let  $\sigma_0$  be a solution for the NCRSP. Given a history  $\tilde{h}$  in  $\mathcal{H}$  s.t.  $\text{Last}(\tilde{h})$  is in  $Q_A$ , we let  $h = \mathcal{H}2\mathcal{G}(\tilde{h})$ . We construct a strategy  $\sigma_E$  for Eve as follows:  $\sigma_E(\tilde{h}) = \bar{a}$  such that  $\bar{a}[0] = \sigma_0(h)$  and for every  $i$  in  $W$ ,  $\bar{a}[i] = \tau_i(h)$  where  $\tau_i$  is the strategy described by Claim 21. Notice that this strategy is only defined for histories  $h$  that satisfy  $\text{Root}(h, i) \neq \perp$ . This is ensured because  $i$  is in  $W$ , meaning that there exists a prefix  $h'$  of  $h$  such that  $h'$  is a good deviation point for Agent  $i$ .

We also need to define  $\sigma_E$  for histories ending in a  $Q_C$ . Consider any history of the form  $\tilde{h}\bar{a}q_b\bar{b}q_c$ , the strategy  $\sigma_E$  is defined as follows:  $\sigma_E(\tilde{h}\bar{a}q_b\bar{b}q_c) = \bar{c}$  such that for every  $i$  not in  $W \cup D$ ,  $\bar{c}[i] = \text{Deviator}(h, i)$ . Let us show that  $\sigma_E$  is winning for Eve. Let  $r$  be a run compatible with  $\sigma_E$ . We must show that  $r \in (S_0 \cup S_D) \cap S_W$ . Denote  $\rho = \mathcal{H}2\mathcal{G}^\bullet(r)$ . By Claim 14 we have  $\pi_1(r \upharpoonright_{Q_A}) = \rho \upharpoonright_{\text{St}}$ .

If  $\rho \upharpoonright_{\text{St}} \in \text{Obj}_0$  then  $r \in S_0$ . If  $\rho \upharpoonright_{\text{St}} \notin \text{Obj}_0$ , since  $\sigma_0$  is a solution, it follows that along  $\rho$  some player has a good deviation point and is loosing. This entails that at some point  $i$  will be in  $D$  along  $r$  i.e.  $\text{Deviator}(\tilde{h}, i) \in \text{Act}$  for some  $\tilde{h}$  a prefix of  $r$ . Thus  $r \in S_D$ .

It remains to show that  $r \in S_W$  this follows from the facts that 1) any player in  $W$  is due to the mapping  $\text{Deviator}$  that correctly guesses the correct deviations and 2) Claim 21.  $\blacktriangleleft$

## 5 Computational Complexity

In this section, we take advantage of the construction presented in the previous section to give complexity bounds for a variety of winning conditions. In fact, we can adapt the technique used in [3] in order to establish the upper bound complexity for NCRSP.

In the case of Reachability, Safety, Büchi and coBüchi conditions, we reduce the game  $\mathcal{H}$  to a finite duration game. We actually transform the winning condition into a finite horizon condition in a finite duration game. In order to obtain this finite duration game, we simply rewrite the winning objective of  $\mathcal{H}$  and obtain a new game  $\mathcal{H}'$  with Parity objective. The plays in the finite duration game  $\mathcal{H}^f$  are obtained by stopping the plays in the game  $\mathcal{H}'$  after the first loop.

In the remainder of this section, for technical convenience, we assume that the histories in  $\mathcal{H}$  are defined over the set  $Q^*$  and that the plays are defined over  $Q^\omega$ . This does not affect the validity of the results since  $\text{Tab}'$  is deterministic and the actions are encoded in the states.

When the game  $\mathcal{H}'$  is equipped with a parity condition, we construct a finite duration game that stops after the first loop. In particular, if  $\text{Pr} : \text{St}' \rightarrow \mathbb{N}$  is the priority function in  $\mathcal{H}'$ . Then, the finite duration game  $\mathcal{H}^f$  is defined over the same game structure as  $\mathcal{H}'$ , but each play stops after the first loop. We will consider such a play winning if the least parity in the loop is even i.e a play  $r = xy_1y_2y_3 \dots y_l y_1$  where  $x \in q_0 Q^*$  and  $y_1, y_2, \dots, y_l \in Q$  is winning for Eve if  $\min\{\text{Pr}(y_k) \mid 0 \leq k \leq l\}$  is even.

The following lemma establishes the relation between  $\mathcal{H}'$  and  $\mathcal{H}^f$ . It is actually a consequence of a result that appeared in [1].

► **Lemma 22.** *Eve has a winning strategy in the game  $\mathcal{H}'$  with the parity condition  $\text{Pr}$  if and only if she has a winning strategy in the game  $\mathcal{H}^f$ .*

The following lemma establishes the fact that inside a cycle in the game  $\mathcal{H}$ , the values of the sets  $W$  and  $D$  do not change.

► **Lemma 23.** *Let  $r$  be a play in  $\mathcal{H}$ , and consider a loop along  $r$ . Let also  $q$  and  $q'$  be two states on this loop. We have  $\pi_2(q) = \pi_2(q')$  and  $\pi_3(q) = \pi_3(q')$ .*

**Proof.** Let  $r = xqyqz$  be an infinite play in  $\mathcal{H}$ . From the definition of the transition relation in  $\mathcal{H}$ ,  $r' = x(qy)^\omega$  is also a valid play in  $\mathcal{H}$ . Then, assume towards contradiction that there are two states in  $qy$  having different values on states  $W$  or  $D$ . It means that there is at least one player  $i$  that is added or removed to/from  $W$  or  $D$ . Therefore, along  $r'$  we would have an infinite number of additions or removals to/from  $W$  or  $D$ . But, according to the transition relation, this is not possible because once a player is removed from  $W$ , it is added to the set  $D$  and never added to  $W$  again along  $r'$ . Also, once a player added in the set  $D$ , he is never removed. Therefore, along each path, along each loop, the values of  $W$  and  $D$  do not change.  $\blacktriangleleft$

In order to check the condition of reachability, we keep along plays in the game  $\mathcal{H}$  a set  $P$  of players in the environment that already have visited their target states. Then, the resulting game  $\mathcal{H}'$  has states in  $Q \times 2^{\text{Agt}}$  where  $Q$  is the set of states in the game  $\mathcal{H}$ . The set  $P$  is initially equal to the set of players for which the initial state is in their target set. Let  $R_i \subseteq \text{St}$  be the target set of Player  $i$ . Then,  $P_0 = \{i \mid s_0 \in R_i\}$  and the initial state in the resulting game  $\mathcal{H}'$  is  $(s_0, \emptyset, \emptyset, P_0)$ .

The set  $P$  is updated as follows:

- if  $(q, q') \in \text{Tab}'$  in  $\mathcal{H}$  and  $q' = (s, W, D) \in \text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}}$ , then  $((q, P), (q', P \cup \{i \mid s \in R_i\}))$  is the corresponding transition in  $\mathcal{H}'$ .
- if  $(q, q') \in \text{Tab}'$  in  $\mathcal{H}$  and  $q' \notin \text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}}$ , then  $((q, P), (q', P))$  is the corresponding transition in  $\mathcal{H}'$ .

Note that the set  $P$  also eventually stabilizes since it only increases and there is a finite number of players in  $\mathcal{G}$ . Let  $\lim \pi_4(r \upharpoonright_{Q_\lambda})$  be the limit along the play  $r$ .

The objective of Eve in the game  $\mathcal{H}$  is written in  $\mathcal{H}'$  as the Büchi condition  $\text{BÜCHI}(F^R)$  where:

$$F^R = \{(s, W, D, P) \mid (0 \in P \text{ or } D \setminus P \neq \emptyset) \text{ and } (W \subseteq P)\} .$$

Now, the Büchi objective  $\text{BÜCHI}(F^R)$  can be expressed as the parity objective  $\text{PARITY}(\text{Pr})$  with  $\text{Pr}(v) = 0$  if  $v = (s, W, D, P) \in F^R$  and  $\text{Pr}(v) = 1$  otherwise.

We now define the finite duration game  $\mathcal{H}^f$  over the same game arena as  $\mathcal{H}$ , but each play stops when the first state in  $\text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}} \times 2^{\text{Agt}}$  is repeated. Then, each play is of the form  $r = xy_1y_2y_3 \dots y_l y_1$  where  $x \in q_0(Q')^*$  and  $y_1, y_2, \dots, y_l \in Q'$  with  $Q' = \text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}} \times 2^{\text{Agt}}$ . Eve wins in the game  $\mathcal{H}^f$  if  $y_1 = (s, W, D, P)$  is such that  $(0 \in P \text{ or } D \setminus P \neq \emptyset)$  and  $(W \subseteq P)$ . Equivalently, thanks to Lemma 23 and because the value of  $P$  does not change for the same argument that show  $W$  and  $D$  eventually stabilize. Finally Eve wins if  $\min\{\text{Pr}(y_k) \mid 0 \leq j < l\}$  is even.

► **Lemma 24.** *All plays in the game  $\mathcal{H}^f$  have polynomial length in the size of the initial game.*

**Proof.** Since  $D$  and  $P$  are monotone, there are at most  $|\text{Agt}| + 1$  different values that they can take on a path of  $\mathcal{H}$ . Also, in the set  $W$  we can have at most one addition and one removal for each player  $i \in \text{Agt}$  and hence  $2|\text{Agt}| + 1$  different values for  $W$ . Therefore, along a play  $\pi$  there are at most  $r = 1 + (2|\text{Agt}| + 1) \cdot (|\text{Agt}| + 1)^2 \cdot |\text{St}|$  different states in  $\text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}} \times 2^{\text{Agt}}$ . Then, between two states in  $\text{St} \times 2^{\text{Agt}} \times 2^{\text{Agt}} \times 2^{\text{Agt}}$ , there are three intermediate states. Therefore, since all the plays in  $\mathcal{H}^f$  stop after the first cycle, the length of each play is of at most  $4r + 1$  states since there is only one state that appears twice. Therefore, all plays in  $\mathcal{H}^f$  have polynomial length in  $\text{Agt}$  and  $\text{St}$  of the initial play  $\mathcal{G}$ .  $\blacktriangleleft$

► **Proposition 25.** *Deciding if there is a solution for the non-cooperative rational synthesis in concurrent games with Reachability objectives is in PSPACE.*

**Proof.** Using Lemmas 22 and 24, solving the non-cooperative rational synthesis problem, reduces to solving the finite duration game  $\mathcal{H}^f$  which has polynomial length plays. This can be done in PSPACE using an alternating Turing machine running in PTIME. ◀

In the case of Safety, Büchi and coBüchi conditions, we essentially use similar constructions; c.f. long version for full details on the constructions. Roughly speaking, in the case of safety it sufficient to “dualize” the winning condition. In the cases of Büchi and coBüchi objectives, the idea is to transform the game  $\mathcal{H}$  by possibly adding some counters such that Eve’s objective can be written as a parity objective. Note that these constructions are similar to the ones in [3]. In the case of Muller conditions, we have to use Least Appearance Record (LAR) construction to get the parity game  $\mathcal{H}'$  and then the finite duration game would have plays with exponential length in the size of the initial game. This approach would give EXPSpace complexity. Fortunately, the parity condition in the game  $\mathcal{H}'$  that we obtain after applying the LAR construction has an exponential number of states but only a polynomial number of priorities. Then, by using the result from [5, 8], we obtain EXPTIME complexity.

► **Theorem 26.** *Deciding if there is a solution for the non-cooperative rational synthesis problem in concurrent games is in PSPACE for Safety, Reachability, Büchi and co-Büchi objectives and EXPTIME for Muller objectives.*

In the case of a fixed number of agents, the game  $\mathcal{H}$  that we build has polynomial size in the size of the initial game  $\mathcal{G}$  (when considering that the transitions are given explicitly in the table Tab since we build nodes in  $\mathcal{H}$  for each possible action profile). This lowers the complexities that we obtain for the rational synthesis problem. The theorem below holds because the game  $\mathcal{H}$  has polynomial size and Eve’s objective is fixed.

► **Theorem 27.** *Deciding if there is a solution for the non-cooperative rational synthesis in concurrent games with a fixed number of agents and Safety, Reachability, Büchi or co-Büchi objectives is in PTIME.*

► **Theorem 28.** *Deciding if there is a solution for the non-cooperative rational synthesis in concurrent games with a fixed number of agents and Muller objectives is in PSPACE.*

## 6 Conclusions

In some circumstances, the Non Cooperative Rational Synthesis Problem (NCRSP) introduced in [6] and defined here as Problem 6 might arguably accept undesired solutions. It asks whether there is a strategy  $\sigma_0$  for Agent 0 such that *for every* 0-NE, if  $\bar{\sigma} = \langle \sigma_0, \bar{\sigma}_{-0} \rangle$ , then  $\text{Payoff}_0(\bar{\sigma}) = 1$ . One sees the objective of Agent 0 as a critical property satisfied by all rational evolutions of the system. A possibly unwanted consequence is that a strategy  $\sigma_0$  which does not allow any rational evolution of the system, thus forcing anarchy, would be a solution. The original definition of NCRSP can be strengthened so as to ask for a strategy  $\sigma_0$  for Agent 0 such that *there is* at least one 0-NE. Another amendment can also restrict the class of game structures. For instance, one can consider pseudo turn-based games, where 0-NE are certain to exist. It suffices to add in Definition 1 the constraint that in every state, only Agent 0 and at most one other agent have non-vacuous choices. The games are still concurrent. Agent 0 can still effectively control every state, but once her strategy  $\sigma_0$  is fixed, the sub-game induced by  $\sigma_0$  has all the characteristics of a turn-based game, where there is always a 0-NE.

---

**References**

---

- 1 Benjamin Aminof and Sasha Rubin. First-cycle games. *Inf. Comput.*, 254:195–216, 2017. doi:10.1016/j.ic.2016.10.008.
- 2 Patricia Bouyer, Romain Brenguier, Nicolas Markey, and Michael Ummels. Pure nash equilibria in concurrent deterministic games. *Logical Methods in Computer Science*, 11(2), 2015. doi:10.2168/LMCS-11(2:9)2015.
- 3 Rodica Condurache, Emmanuel Filiot, Raffaella Gentilini, and Jean-François Raskin. The complexity of rational synthesis. In *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPICs*, pages 121:1–121:15. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016. doi:10.4230/LIPICs.ICALP.2016.121.
- 4 Dana Fisman, Orna Kupferman, and Yoad Lustig. Rational synthesis. In *Tools and Algorithms for the Construction and Analysis of Systems, 16th International Conference, TACAS 2010, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2010, Paphos, Cyprus, March 20-28, 2010. Proceedings*, volume 6015 of *Lecture Notes in Computer Science*, pages 190–204. Springer, 2010. doi:10.1007/978-3-642-12002-2\_16.
- 5 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>.
- 6 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.
- 7 Orna Kupferman, Giuseppe Perelli, and Moshe Y. Vardi. Synthesis with rational environments. *Ann. Math. Artif. Intell.*, 78(1):3–20, 2016. doi:10.1007/s10472-016-9508-8.
- 8 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.