Synthesis of Controllable Nash Equilibria in Quantitative Objective Games

Shaull Almagor[1], **Orna Kupferman**[2], **Giuseppe Perelli**[1*],
[1] University of Oxford
[2] Hebrew University

Abstract

In *Rational Synthesis*, we consider a multi-agent system in which some of the agents are controllable and some are not. All agents have objectives, and the goal is to synthesize strategies for the controllable agents so that their objectives are satisfied, assuming rationality of the uncontrollable agents. Previous work on rational synthesis considers objectives in LTL, namely ones that describe on-going behaviors, and in Objective-LTL, which allows ranking of LTL formulas. In this paper, we extend rational synthesis to $\text{LTL}[\mathcal{F}]$– an extension of LTL by quality operators. The satisfaction value of an $\text{LTL}[\mathcal{F}]$ formula is a real value in $[0, 1]$, where the higher the value is, the higher is the quality in which the computation satisfies the specification. The extension significantly strengthens the framework of rational synthesis and enables a study its game- and social-choice theoretic aspects. In particular, we study the price of stability and price of anarchy of the rational-synthesis game and use them to explain the cooperative and non-cooperative settings of rational synthesis. Our algorithms make use of strategy logic and decision procedures for it. Thus, we are able to handle the richer quantitative setting using existing tools. In particular, we show that the cooperative and non-cooperative versions of quantitative rational synthesis are 2EXPTIME-complete and in 3EXPTIME, respectively – not harder than the complexity known for their Boolean analogues.

1 Introduction

The *synthesis problem* for LTL (linear temporal logic) gets as input a specification in LTL and outputs a reactive system that satisfies it — if such exists [Pnueli and Rosner, 1989]. The specification is over input signals, controlled by the environment, and output signals, controlled by the system. The system should satisfy the specification in all environments. The environment with which the system interacts is often composed of other systems. For example, the clients interacting with a server are by themselves distinct entities (which we call *agents*). In the traditional approach to synthesis, the agents can be seen as if their only objective is to conspire to fail the system. Hence the term "hostile environment" that is traditionally used in the context of synthesis. In real life, however, many times agents have objectives of their own, other than to fail the system. The approach taken in the field of algorithmic game theory [Nisan *et al.*, 2007] is to assume that agents interacting with a computational system are *rational*; i.e., agents act to achieve their own objectives.

In [Fisman *et al.*, 2010], Fisman et al. introduced *rational synthesis*. The input to the rational-synthesis problem consists of LTL formulas specifying the objectives of the system and the agents that constitute the environment. The signals over which the objectives are defined are partitioned among the system and the agents, so that each of them controls a subset of the signals. There are two approaches to rational synthesis. In *cooperative* rational synthesis, the desired output is a strategy profile such that the objective of the system is satisfied in the computation that is the outcome of the profile, and the agents that constitute the environment have no incentive to deviate from the strategies suggested to them; that is, the profile is a *Nash equilibrium* (NE) [Nash, 1950]. Thus, in the cooperative setting, we assume that once we suggest to the agents strategies that constitute an equilibrium, they follow them. Then, in *non-cooperative* rational synthesis, studied in [Kupferman *et al.*, 2016], the desired output is a strategy for the system such that its objective is satisfied in all NE profiles in which the system follows this strategy. Thus, in the non-cooperative setting, the agents are rational, but need not follow a suggested profile. The rational-synthesis problem for LTL in the cooperative setting is 2EXPTIME-complete [Fisman *et al.*, 2010], as is traditional LTL synthesis. In the non-cooperative setting, the best known complexity is 3EXPTIME [Kupferman *et al.*, 2016].[1]

Traditional games in game theory are finite and their outcome depends on the final position of the game [Nisan and Ronen, 1999; Nisan *et al.*, 2007]. In contrast, the systems

[*]The author is thankful for the financial support of the ERC Advanced Investigator grant 291528 ("RACE") at Oxford.

[1]The complexity specified in [Kupferman *et al.*, 2016] is actually 2EXPTIME-complete, yet the complexity analysis there misses one alternation between strategy quantifiers in the strategy-logic formula to which the problem is reduced. Taking this additional alternation into account, the complexity goes up to 3EXPTIME.

we reason about maintain an *on-going interaction* with their environment [Harel and Pnueli, 1985], and reasoning about their behavior refers not to their final state (in fact, we considers non-terminating systems, with no final state) but rather to the *language* of computations that they generate. While LTL specifications enable the description of rich on-going behaviors, the semantics of LTL is Boolean: a computation may satisfy a specification or it may not. As argued in [Almagor *et al.*, 2016], the Boolean nature of LTL is a real obstacle in synthesis. Indeed, while many systems may satisfy a specification, they may do so at different levels of quality. Consequently, designers would be willing to give up manual design only after being convinced that the automatic procedure that replaces it generates systems of comparable quality. As argued in [Kupferman *et al.*, 2016], the extension of the synthesis problem to the rational setting makes the quantitative setting even more appealing. Indeed, objectives in typical game-theory applications are quantitative, and interesting properties of games often refer to their quantitative aspects. In [Kupferman *et al.*, 2016], the authors add a quantitative layer to LTL and studied rational synthesis for *Objective* LTL (OLTL, for short). There, each specification is a set Ψ of specifications, and a reward function maps each subset of Ψ to the reward gained when this subset of formulas is satisfied. In the rational synthesis problem for OLTL, the input consists of OLTL specifications for the system and the other agents, and the objective of the system is to maximize its reward with respect to environments that are in an equilibrium.

We study the rational-synthesis problem for a much stronger quantitative formalism, namely $\text{LTL}[\mathcal{F}]$. The logic $\text{LTL}[\mathcal{F}]$ is a multi-valued logic that augments LTL with quality operators [Almagor *et al.*, 2016]. The satisfaction value of an $\text{LTL}[\mathcal{F}]$ formula is a real value in $[0, 1]$, where the higher the value is, the higher is the quality in which the computation satisfies the specification. The quality operators in $\mathcal{F}$ can prioritize different scenarios or reduce the satisfaction value of computations in which delays occur. For example, as in earlier work on multi-valued extensions of LTL (c.f., [Faella *et al.*, 2008]), the set $\mathcal{F}$ may contain the $\min\{x, y\}$, $\max\{x, y\}$, and $1-x$ functions, which are the standard quantitative analogues of the $\wedge$, $\vee$, and $\neg$ operators. The novelty of $\text{LTL}[\mathcal{F}]$ is the ability to manipulate values by arbitrary functions. For example, $\mathcal{F}$ may contain the weighted-average function $\oplus_\lambda$. The satisfaction value of the formula $\varphi \oplus_\lambda \psi$ is the weighted (according to λ) average between the satisfaction values of φ and ψ. This enables the specification of the quality of the system to interpolate different aspects of it. As an example, consider the $\text{LTL}[\mathcal{F}]$ formula $\mathsf{G}(req \to (grant \oplus_{\frac{2}{3}} \mathsf{X} grant))$. The formula states that we want requests to be granted immediately and the grant to hold for two transactions. When this always holds, the satisfaction value is $\frac{2}{3} + \frac{1}{3} = 1$. We are quite okay with grants that are given immediately and last for only one transaction, in which case the satisfaction value is $\frac{2}{3}$, and less content when grants arrive with a delay, in which case the satisfaction value is $\frac{1}{3}$.

The extension to $\text{LTL}[\mathcal{F}]$ significantly strengthens the framework of rational synthesis. In addition, we study the stability of rational synthesis and additional game- and social-choice theoretic aspects of it. We generalize the setting to an arbitrary partition of the set of agents to controllable and uncontrollable ones. In particular, the case there are no controllable agents corresponds to interactions with no authority. We refine the stability-inefficiency measures of *price of stability* (PoS) [Anshelevich *et al.*, 2008] and *price of anarchy* (PoA) [Koutsoupias and Papadimitriou, 2009; Papadimitriou, 2001] to a setting where some of the agents are controllable. Essentially, these notions measure how much we lose from the absence of a central authority by comparing the utility of a social-optimum profile (that is, a profile that maximizes the profits of all agents together) with that of NE profiles. Our refinement enables a distinction between cases where the behavior of the controllable agents is fixed and cases it is not.

Studying the stability of rational synthesis, we prove that a rational-synthesis game need not have an NE, and that for some utility functions, the PoS and PoA may not be bounded. We relate the cooperative and non-cooperative settings with the two stability-inefficiency measures. In the cooperative setting, we may suggest to the agents a best NE, thus the cooperative setting corresponds to the PoS measure. On the other hand, in the non-cooperative setting, the agents may follow the worst NE, which corresponds to the PoA measure. This settles a discussion in the community about the necessity of both settings, and also implies that the profit to the controllable components in the non-cooperative setting may be unboundedly smaller than the profit in the cooperative setting.

We solve decision problems for rational synthesis with $\text{LTL}[\mathcal{F}]$ objectives. Our algorithms make use of *strategy logic* and decision procedures for it [Chatterjee *et al.*, 2007; Mogavero *et al.*, 2010; 2012; 2014]. Thus, we are able to handle the richer quantitative setting using existing tools. In particular, we show that the cooperative and non-cooperative versions of $\text{LTL}[\mathcal{F}]$ rational synthesis are 2EXPTIME-complete and in 3EXPTIME, respectively, and that so are the problems of calculating the various stability-inefficiency measures, and other measures that quantify the game and its outcomes. Thus, the complexity of rational synthesis in the quantitative setting is not harder than the best known complexity in the Boolean setting. Due to the lack of space, some of the proofs are omitted and can be found in the full version, in the authors' URLs.

Related Work In [Gutierrez *et al.*, 2015; Wooldridge *et al.*, 2016], the authors introduce the problems of E-Nash and A-Nash for different classes of iterated games, which correspond to the special cases of cooperative and non-cooperative rational synthesis, respectively, in which there is no controllable player, showing that the problems are 2EXPTIME-complete. These problems have been analyzed also in different settings, e.g., imperfect information [Gutierrez *et al.*, 2016; 2018], finite traces goals [Gutierrez *et al.*, 2017b], and lexicographic objectives [Gutierrez *et al.*, 2017a]. In [Condurache *et al.*, 2016], the authors analyze the rational synthesis problem for qualitative goals whose complexity ranges from simple reachability to the full power of ω-regular expressions. In [Almagor *et al.*, 2015], the authors study repair of specifications in multi-player games, where the goal is to reach specifications in which the uncontrollable players are in an NE, in both a cooperative and non-cooperative setting.

2 Preliminaries

The Temporal Logic LTL[$\mathcal{F}$] The linear temporal logic LTL[$\mathcal{F}$], introduced in [Almagor *et al.*, 2016], generalizes LTL by replacing the Boolean operators of LTL with arbitrary functions over $[0,1]$. The logic is actually a family of logics, each parameterized by a set $\mathcal{F}$ of functions.

Syntax. Let AP be a set of Boolean atomic propositions, and let $\mathcal{F} \subseteq \{g : [0,1]^m \to [0,1] \mid m \in \mathbb{N}\}$ be a set of functions over $[0,1]$. Note that the functions in $\mathcal{F}$ may have different arities. An LTL[$\mathcal{F}$] formula is one of the following:

- `True`, `False`, or p, for $p \in AP$.
- $g(\varphi_1, ..., \varphi_m)$, $\mathsf{X}\varphi_1$, or $\varphi_1 \mathsf{U} \varphi_2$, for LTL[$\mathcal{F}$] formulas $\varphi_1, \ldots, \varphi_m$ and a function $g \in \mathcal{F}$.

Semantics. The semantics of LTL[$\mathcal{F}$] formulas is defined with respect to infinite computations over AP. A *computation* is a word $\rho = \rho_0, \rho_1, \ldots \in (2^{AP})^\omega$. We use ρ^i to denote the suffix $\rho_i, \rho_{i+1}, \ldots$. The semantics maps a computation ρ and an LTL[$\mathcal{F}$] formula φ to the *satisfaction value* of φ in ρ, denoted $[\![\rho, \varphi]\!]$. The satisfaction value is defined inductively as described in Table 1 below.[2]

Formula	Satisfaction value
$[\![\rho, \texttt{True}]\!]$	1
$[\![\rho, \texttt{False}]\!]$	0
$[\![\rho, p]\!]$	1 if $p \in \rho_0$ 0 if $p \notin \rho_0$
$[\![\rho, g(\varphi_1, ..., \varphi_m)]\!]$	$g([\![\rho, \varphi_1]\!], ..., [\![\rho, \varphi_m]\!])$
$[\![\rho, \mathsf{X}\varphi_1]\!]$	$[\![\rho^1, \varphi_1]\!]$
$[\![\rho, \varphi_1 \mathsf{U} \varphi_2]\!]$	$\max_{i \geq 0}\{\min\{[\![\rho^i, \varphi_2]\!], \min_{0 \leq j < i} [\![\rho^j, \varphi_1]\!]\}\}$

Table 1: The semantics of LTL[$\mathcal{F}$].

It is not hard to prove, by induction on the structure of the formula, that for every computation ρ and formula φ, it holds that $[\![\rho, \varphi]\!] \in [0,1]$.

The logic LTL coincides with the logic LTL[$\mathcal{F}$] for $\mathcal{F}$ that corresponds to the usual Boolean operators. For simplicity, we use the common such functions as abbreviation, as described below. In addition, we introduce notations for some useful functions. Let $x, y, \lambda \in [0,1]$. Then,

- $x \vee y = \max\{x, y\}$
- $\neg x = 1 - x$
- $\nabla_\lambda x = \lambda x$
- $x \wedge y = \min\{x, y\}$
- $x \oplus_\lambda y = \lambda x + (1-\lambda) y$

To see that LTL indeed coincides with LTL[$\mathcal{F}$] for $\mathcal{F} = \{\neg, \vee, \wedge\}$, note that for this $\mathcal{F}$, all formulas are mapped to $\{0,1\}$ in a way that agrees with the semantics of LTL.

[2] The observant reader may be concerned by our use of max and min where sup and inf are in order. As proved in [Almagor *et al.*, 2016], there are only finitely many satisfaction values for a formula φ, thus the semantics is well defined.

Lemma 1 ([Almagor *et al.*, 2016]). *For every* LTL[$\mathcal{F}$] *formula φ there exists a finite set $V(\varphi) \subseteq [0,1]$ of possible satisfaction values, such that for every computation ρ, it holds that $[\![\rho, \varphi]\!] \in V(\varphi)$ and $|V(\varphi)| = 2^{O(\varphi)}$.*

The Rational-Synthesis Game Consider sets C and U of controllable and uncontrollable agents, respectively. Let $A = C \cup U$. The *rational-synthesis game* (RS-game, for short) is played among the agents in A. For $i \in A$, agent i assigns values to a set X_i of Boolean atomic propositions. For all $i \neq j \in A$, we have that $X_i \cap X_j = \emptyset$. Let $X = \bigcup_{i \in A} X_i$. Each agent $i \in A$ has an objective – an LTL[$\mathcal{F}$] formula φ_i over X.

A *strategy* for agent i is a function $\pi_i : (2^X)^* \to 2^{X_i}$, mapping the history of the computation so far to an assignment to the atomic propositions of agent i. Let Π_i be the set of possible strategies for agent i. A *profile* is a vector of strategies, one for each agent. Let $\Pi = \times_{i \in A} \Pi_i$ denote the set of all possible profiles. We assume that all agents move together. That is, given a profile $P \in \Pi$, the computation generated when all the agents follow their strategies in P is $\rho_P = x_1, x_2, x_3, \ldots \in (2^X)^\omega$, where for all $j \geq 0$, we have $x_j = \bigcup_{i \in A} \pi_i(x_1, x_2, \ldots, x_{j-1})$. We refer to ρ_P as the *outcome* of P. For $i \in A$, the *profit* of agent i in the profile P, denoted $\mathsf{profit}_i(P)$, is the satisfaction value of φ_i in ρ_P. For a subset of the agents $B \subseteq A$, a *partial profile* is a vector of strategies for the agents in B, and we let $\Pi_B = \times_{i \in B} \Pi_i$. For a profile P we denote by $P|_B \in \Pi_B$ its restriction to the agents in B. A profile $P \in \Pi$ *agrees* with a partial profile $P' \in \Pi_B$ if $P|_B = P'$.

In addition to the profits of the individual agents, we are interested in the welfare of the controllable agents (typically, they model the authority) and of the society as a whole. A *utility function* is a function $\mathsf{utility} : [0,1]^{|A|} \to [0,1]$, which maps the profits of the agents to an overall utility of the society. For convenience, we sometimes refer to the utility function as $\mathsf{utility} : \Pi \to [0,1]$, namely as one that operates on profiles rather than on the vector of profits these profiles induce.

Remark 1. The restriction of the range of utility to $[0,1]$ is only to conform with the semantics of LTL[$\mathcal{F}$]. Indeed, by Lemma 1, the domain of utility is finite, hence any range can be normalized to $[0,1]$. We can thus view utility as an LTL[$\mathcal{F}$] formula $\varphi_{\mathsf{utility}}$, defined as $\varphi_{\mathsf{utility}} = \mathsf{utility}(\varphi_1, \ldots, \varphi_k)$ where $A = \{1, \ldots, k\}$.

Example 1. The richness of LTL[$\mathcal{F}$] allows us to capture well-studied utility functions. We demonstrate this on several wellness objectives [Nisan *et al.*, 2007]. Consider a subset $B \subseteq A = \{1, \ldots, k\}$.

- In the *B-utilitarian function*, the utility is the sum of the profits of all the agents in B. By normalizing, this can be captured in LTL[$\mathcal{F}$] by introducing the function $\mathsf{utility}(v_1, \ldots, v_k) = \frac{\sum_{i \in B} v_i}{|B|}$.
- In the *B-egalitarian function*, the utility is the minimum among the profits of the agents in B. We capture this by $\mathsf{utility}(v_1, \ldots, v_k) = \min_{i \in B}\{v_i\}$.

- The *anti-B-utilitarian social welfare function* concerns minimizing the social welfare of a subset of the hostile agents, and is captured by the function $\mathsf{utility}(v_1, \ldots, v_k) = 1 - \frac{\sum_{i \in B} v_i}{|B|}$.
- The *anti-B-egalitarian social welfare function* again concerns hostile agents, this time minimizing their lowest utility. We capture this by $\mathsf{utility}(v_1, \ldots, v_k) = 1 - \min_{i \in B}\{v_i\}$.

For a profile P, an uncontrollable agent $i \in U$, and a strategy $\pi'_i \in \Pi_i$, let $P[i \leftarrow \pi'_i]$ denote the profile obtained from P by replacing the strategy for agent i by π'_i. A profile $P \in \Pi$ is a *controllable Nash equilibrium* (CNE) if no uncontrollable agent can benefit from unilaterally deviating from his strategy in P to another strategy; i.e., for every agent $i \in U$ and every strategy $\pi'_i \in \Pi_i$, it holds that $\mathsf{profit}_i(P[i \leftarrow \pi'_i]) \leq \mathsf{profit}_i(P)$.

Definition 1 (Rational Synthesis). *Consider a game G with agents $A = C \cup U$, objectives φ_i for every agent $i \in A$, and a utility function* $\mathsf{utility} : \Pi \to [0,1]$. *We are given a utility threshold $t \in [0,1]$, and for every agent $i \in C$ we are also given a profit threshold $t_i \in [0,1]$. The* weak (cooperative) rational synthesis *problem is to synthesize a partial profile $P^C \in \Pi_C$ for the controllable agents, such that there exists a CNE P that agrees with P^C,* $\mathsf{utility}(P) \geq t$ *,and for every $i \in C$, it holds that* $\mathsf{profit}_i(P) \geq t_i$. *The* strong (non-cooperative) rational synthesis *problem is to synthesize $P^C \in \Pi_C$ such that for all CNEs P that agree with P^C, we have that* $\mathsf{utility}(P) \geq t$ *and for every $i \in C$, it holds that* $\mathsf{profit}_i(P) \geq t_i$.

Consider a utility function $\mathsf{utility} : \Pi \to [0,1]$, a *social optimum* (SO, for short) is a profile that maximizes the utility. We denote its utility by OPT. Thus, $OPT = \max_{P \in \Pi} \mathsf{utility}(P)$. It is well known that decentralized decision-making may lead to sub-optimal solutions from the point of view of society as a whole. We quantify the inefficiency incurred due to self-interested behavior according to the *price of stability* (PoS) and *price of anarchy* (PoA) [Nisan *et al.*, 2007, Chapter 17] measures. The PoS measures the best-case inefficiency of a Nash equilibrium, while the PoA is the worst-case inefficiency of a Nash equilibrium. Traditionally, PoS (resp. PoA) is the ratio between OPT and the maximal (resp. minimal) NE in the game. In a setting where some of the agents are controllable, however, things become more intricate. In the following we consider two definitions of PoS and PoA for our setting.

Consider a partial profile $P^C \in \Pi_C$. A *P^C-restricted social optimum* (SO(P^C), for short) is a profile that agrees with P^C and maximizes the utility. We denote its utility by $OPT(P^C)$. Thus, $OPT(P^C) = \max\{\mathsf{utility}(P) : P|_C = P^C\}$. We denote by $\Upsilon_G(P^C)$ the set of CNEs in G that agree with P^C. Thus, $P \in \Upsilon_G(P^C)$ iff P is a CNE and $P|_C = P^C$. Let Υ_G denote the set of all CNEs in G.

Definition 2 (Price of Stability and Price of Anarchy). *Let $\mathcal{G}$ be a family of games among controllable and uncontrollable agents, and let G be a game in $\mathcal{G}$ with sets C and U of agents.*

- *The* uncontrollable price of stability *of G is the ratio between the utility of the SO and the maximal utility of a CNE of G. That is,* $\mathrm{UPoS}(G) = \frac{OPT}{\max_{P \in \Upsilon_G} \mathsf{utility}(P)}$.
- *The* uncontrollable price of anarchy *of G is the ratio between the utility of the SO and the minimal utility of a CNE of G. That is,* $\mathrm{UPoA}(G) = \frac{OPT}{\min_{P \in \Upsilon_G} \mathsf{utility}(P)}$.
- *The* controllable price of stability *of G is the minimal ratio between the utility of the SO and the maximal utility of a CNE in a fixed profile for the agents in C. That is,* $\mathrm{CPoS}(G) = \min_{P^C \in \Pi_C} \left(\frac{OPT(P^C)}{\max_{P \in \Upsilon_G(P^C)} \mathsf{utility}(P)} \right)$.
- *The* controllable price of anarchy *of G is the minimal ratio between the utility of the SO and the minimal utility of a CNE in a fixed profile for the agents in C. That is,* $\mathrm{CPoA}(G) = \min_{P^C \in \Pi_C} \left(\frac{OPT(P^C)}{\min_{P \in \Upsilon_G(P^C)} \mathsf{utility}(P)} \right)$.

Intuitively, while in the controlled definitions the controlled agents fix their strategies in the SO and the CNE, in the uncontrollable ones they may use different strategies in each of the profiles.

Remark 2. In the definitions above, if the set of CNEs in the denominator is empty, or the denominator equals 0, we treat the value as ∞.

Lemma 2. *For every game G, it holds that* $\mathrm{CPoS}(G) \leq \mathrm{UPoS}(G)$ *and* $\mathrm{CPoA}(G) \leq \mathrm{UPoA}(G)$.

Proof. Consider a profile $P^C \in \Pi_C$ for which $\max_{P \in \Upsilon_G(P^C)} \mathsf{utility}(P) = \max_{P \in \Upsilon_G} \mathsf{utility}(P)$. Then, clearly $OPT(P^C) \leq OPT$, implying that

$$\frac{OPT(P^C)}{\max_{P \in \Upsilon_G(P^C)} \mathsf{utility}(P)} \leq \frac{OPT}{\max_{P \in \Upsilon_G} \mathsf{utility}(P)},$$

from which we conclude that $\mathrm{CPoS}(G) \leq \mathrm{UPoS}(G)$.

The proof that $\mathrm{CPoA}(G) \leq \mathrm{UPoA}(G)$ is analogous. □

3 On the Stability of the RS Game

Theorem 1. *There is an RS game with no CNE.*

Proof. The theorem holds already for Boolean games [Harrenstein *et al.*, 2001], no controllable agents, and the average utility function. Let $X_1 = \{p\}$, $X_2 = \{q\}$, $\varphi_1 = p \text{ XOR } q$, and $\varphi_2 = \neg(p \text{ XOR } q)$. There is no CNE, as both agents 1 and 2 can always deviate to a strategy that results in a computation that satisfies their objective. Formally, for every profile P, if $\pi_P \models \varphi_1$, then agent 2 can deviate to a strategy in which the value of q is flipped, resulting in a profile P' for which $\pi_{P'} \models \varphi_2$, and similarly for the case $\pi_P \models \varphi_2$, where agent 1 has a beneficial deviation. □

We now turn to consider stability inefficiency, and show that the prices of stability and anarchy are in general unbounded. The result holds already for games with no controllable agents. Note that there, the UPoS and CPoS measures coincide, and we denote them by PoS, and similarly for PoA.

Theorem 2. *The PoA and PoS in the RS game are unbounded: For every $k \geq 2$ and $\epsilon > 0$, there exists a k-agent RS game G_A such that $PoA(G_A) \geq \frac{1}{\epsilon}$, and a k-agent RS game G_S such that $PoS(G_S) \geq \frac{1-\epsilon}{\epsilon}$.*

Proof. In both G_A and G_S, agent i, for $1 \leq i \leq k$, controls an atomic proposition x_i. In both, the utility function gives a profit of ϵ to agent k for satisfying her objective, a profit of $\frac{1-\epsilon}{k-1}$ to agent i, for $1 \leq i \leq k-1$, for satisfying her objective, and sums the profits of all agents.

Consider first G_A. There, the objective of agent i, for $1 \leq i \leq k-1$, is $x_i \wedge \neg x_k$, and the objective of agent k is $x_k \vee (\bigwedge_{i=1}^{k-1} x_i)$. The profile in which $x_k = false$ and $x_i = true$ for all $1 \leq i \leq k-1$ is an SO in which all objectives are satisfied, thus $OPT(G_A) = 1$. The profile in which $x_k = true$ and $x_i = false$ for every $1 \leq i \leq k-1$ is a CNE in which only agent k satisfies her objective. Thus, the utility in this profile is ϵ. Hence, $PoA(G_A) \geq \frac{1}{\epsilon}$.

Consider now G_S. There, the objective of agent i, for $1 \leq i \leq k-1$, is $\bigwedge_{i=1}^{k} x_i$, and the objective of agent k is $\bigwedge_{i=1}^{k-1} x_i \wedge \neg x_k$. In the profile in which $x_i = true$ for all $1 \leq i \leq k$, agents $1, \ldots, k-1$ satisfy their objectives and agent k does not. Thus, $OPT(G_S) \geq 1 - \epsilon$. On the other hand, the best CNE profile is the one in which all $x_1, \ldots, x_{k-1}$ are $true$ and $x_k = false$. There, only agent k satisfies her objective, thus the utility of the only CNE is ϵ. Thus, $PoS(G_S) \geq \frac{1-\epsilon}{\epsilon}$. $\square$

Note that while the objectives used in the games in the proof of Theorem 2 are LTL formulas, the utility function is not uniform. Alternatively, we could have defined a uniform utility function and use LTL[$\mathcal{F}$] quality operators in order to weight the agents differently.

Finally, note that the SO profile in the game G_A described in the proof of Theorem 2 is a CNE. Since we can add a controllable agent whose objective is to maximize the number of satisfied uncontrollable agents, we can conclude with the following.

Theorem 3. *The ratio between the PoS and PoA is unbounded, and so is the ratio between the profit of the controllable agents in the weak and strong settings of rational synthesis.*

4 Decision Procedures for RS

In this section we solve several decision problems related to rational synthesis. Our main tools are decision procedures for Strategy Logic, and the translation of LTL[$\mathcal{F}$] to automata.

Strategy Logic [Chatterjee *et al.*, 2007; Mogavero *et al.*, 2014] (SL, for short) is a logic that allows to quantify over strategies in games as explicit first-order objects. Intuitively, such quantification, together with a syntactic operator called *binding*, enables the formula to quantify restricted classes of strategy profiles, inducing a subset of paths in which a temporal specification needs to be satisfied.

From a syntactic point of view, SL is an extension of LTL with disjoint sets of strategy variables $V_0, \ldots, V_k$, where V_i is a set of strategy variables for agent i, existential ($\langle\langle x_i \rangle\rangle$) and universal ($[[x_i]]$) strategy quantifiers, and a binding operator of the form (i, x_i), which couples an agent i with one of its variables $x_i \in V_i$.

We first introduce some technical notation. For a tuple $t = (t_0, \ldots, t_k)$, we denote by $t[i \leftarrow d]$ the tuple obtained from t by replacing the i-th component with d. We use $\vec{x}$ as an abbreviation for the tuple $(x_0, \ldots, x_k) \in V_0 \times \ldots \times V_k$. By $\langle\langle \vec{x} \rangle\rangle = \langle\langle x_0 \rangle\rangle \ldots \langle\langle x_k \rangle\rangle$, $[[\vec{x}]] = [[x_0]] \ldots [[x_k]]$, and $\flat(\vec{x}) = (0, x_0) \ldots (k, x_k)$ we denote the existential and universal quantification, and the binding of all the agents to the strategy profile variable $\vec{x}$, respectively. Finally, by $\flat(\vec{x}_{-i}, y_i) = (0, x_0) \ldots (i, y_i) \ldots (k, x_k)$ we denote the changing of binding for agent i from the strategy variable x_i to the strategy variable y_i in the global binding $\flat(\vec{x})$.

Here we define and use a slight variant of the *Boolean-Goal* fragment of SL, namely SL[BG], introduced in [Mogavero *et al.*, 2014]. Formulas in SL[BG] are defined with respect to the set AP of atomic proposition, the set A of agents, and sets V_i of strategy variables for agent $i \in A$. The set of SL[BG] formulas is defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathsf{X}\varphi \mid \varphi\, \mathsf{U} \varphi \\ \langle\langle x_i \rangle\rangle\varphi \mid [[x_i]]\varphi \mid \flat(\vec{x})\varphi,$$

where, $p \in AP$ is an atomic proposition and $x_i \in V_i$ is a variable for agent i.

The LTL part has the classical meaning. The formula $\langle\langle x_i \rangle\rangle\varphi$ states that there exists a strategy for agent i such that the formula φ holds. The formula $[[x_i]]\varphi$ states that, for all possible strategies for agent i, the formula φ holds. Finally, the formula $\flat(\vec{x})\varphi$ states that the formula φ holds under the assumption that the agents in A adhere to the strategy evaluation of the variable x_i coupled in $\flat(\vec{x})$.

For a more detailed definition of the semantics, the reader is referred to [Mogavero *et al.*, 2014]

The *alternation depth* of an SL formula is the maximum number of quantifier switches $\langle\langle x_i \rangle\rangle[[x_j]]$ or $[[x_i]]\langle\langle x_j \rangle\rangle$ in the formula. As in first-order logic, the alternation depth plays an important role in the complexity:

Theorem 4. [Mogavero *et al.*, 2014] *The model-checking problem for* SL[BG] *can be solved in $(d+1)$EXPTIME, with d being the alternation depth of the specification.*

For LTL[$\mathcal{F}$], we use the following:

Theorem 5. [Almagor *et al.*, 2016] *Let φ be an* LTL[$\mathcal{F}$] *formula and $V \subseteq [0,1]$ be a predicate.*

1. *There exists a nondeterministic generalized-Büchi automaton $\mathcal{A}_{\varphi,V}$ such that for every computation $\rho \in (2^{AP})^\omega$, it holds that $[[\rho, \varphi]] \in V$ iff $\mathcal{A}_{\varphi,V}$ accepts ρ. Furthermore, $\mathcal{A}_{\varphi,V}$ has at most $2^{(|\varphi|^2)}$ states and its index is at most $|\varphi|$.*
2. *There exists an* LTL *formula $Bool(\varphi, V)$, of length at most exponential in φ, such that for every computation $\rho \in (2^{AP})^\omega$, it holds that $[[\rho, \varphi]] \in V$ iff $\rho \models Bool(\varphi, V)$.*

A particularly useful case of Theorems 5 is when $V = [t, 1]$ (resp. $V = [t, t]$) for some threshold t. In this case, we denote $Bool(\varphi, V)$ by $\varphi^{\geq t}$ (resp. $\varphi^{=t}$).

Solving Rational Synthesis We now turn to show how to solve the RS problem. Our solution relies on a combination of techniques for $\text{LTL}[\mathcal{F}]$ and SL. We first reduce the RS problem into a simpler form.

An RS game is *simple* if $A = \{\alpha_0\} \cup U$, where α_0 is a single controllable agent and its objective is a (Boolean) LTL formula φ_0 with threshold $t_0 = 1$. In addition, the game does not have a utility function; i.e., it is a constant function that can be ignored when solving the problem. It is not hard to transform a given RS game G into a simple one:

Lemma 3. *Given an RS game G with* $\text{LTL}[\mathcal{F}]$ *objectives, we can construct a simple game G' with* $\text{LTL}[\mathcal{F}]$ *objectives such that G has a solution for weak/strong RS iff G' has a solution for weak/strong RS. Moreover, a solution G can be extracted from a solution in G'.*

We now solve the RS problems for simple games.

Theorem 6. *Solving a simple weak or strong RS game with* $\text{LTL}[\mathcal{F}]$ *objectives can be reduced to model-checking an* $\text{SL}[\text{BG}]$ *formula of alternation depth* 1 *or* 2, *respectively.*

Proof. For an agent $i \in U$ and its $\text{LTL}[\mathcal{F}]$ objective φ_i, consider the $\text{SL}[\text{BG}]$ formula with free variables $\vec{y}$ and $\vec{y'}$ defined as $\Phi_i(\vec{y}, \vec{y'}) = \bigwedge_{t \in V(\varphi_i)} (\flat(\vec{y})\varphi_i^{=t} \rightarrow \flat(\vec{y'})\varphi_i^{\leq t})$, and stating that, for every threshold value t, whenever the value of φ_i on the run generated by the profile $\vec{y}$ is t, then the value of φ_i on the run generated by the profile $\vec{y'}$ is less or equal than t. Observe that, by means of this formula, we can express the fact that $\vec{y}$ is an NE over the uncontrollable agents as $\varphi_{NE}(\vec{y}) = [\![\vec{x}]\!] \bigwedge_{i \in U} \Phi_i(\vec{y}, (\vec{y}_{-i}, x_i))$.

Now, we can specify solutions to the weak and strong RS problems by the following formulas: [3]

- $\Phi^{wRS} = \langle\!\langle \vec{y}_{\alpha_0} \rangle\!\rangle \langle\!\langle \vec{y}_U \rangle\!\rangle (\varphi_{NE}(\vec{y}) \wedge \flat(y)\varphi_0^{\geq 1})$, and
- $\Phi^{sRS} = \langle\!\langle \vec{y}_{\alpha_0} \rangle\!\rangle [\![\vec{y}_U]\!] (\neg\varphi_{NE}(\vec{y}) \vee \flat(y)\varphi_0^{\geq 1})$.

Note that that Φ^{wRS} and Φ^{sRS} are $\text{SL}[\text{BG}]$ formulas of alternation depth at most 1 and 2, respectively. Indeed, the formula $\varphi_{NE}(\vec{y})$ contains only a sequence of universal quantifications that, combined with the existential quantifications on top of the formula Φ^{wRS}, gives an alternation depth 1. For the case of strong RS, the formula $\neg\varphi_{NE}$ contains a sequence of existential quantifications that, combined with the quantifications on top of Φ^{sRS}, produces an alternation dept of 2.

Thus, weak and strong RS is reduced to model checking the SL formulas Φ^{wRS} and Φ^{sRS}, respectively.

We should, however, take care when analyzing the complexity of the procedure, for two reasons: first, the formulas $\Phi_i(\vec{y}, \vec{y'})$, which occur in Φ^{wRS}, involve a conjunction over the set of satisfaction values of every φ_i, and second, the formulas $\varphi_i^{=t}$ and $\varphi_i^{\leq t}$, as well as $\varphi_0^{\geq 1}$, may themselves be of exponential length, as per Theorem 5(2).

To overcome these additional exponential blow-ups, we proceed as follows. First, we notice that the model-checking algorithm in [Mogavero *et al.*, 2014, Lemma 5.6] translates LTL into automata. Theorem 5(1) allows us to perform a similar translation from $\text{LTL}[\mathcal{F}]$ with only a single exponential blowup. To address the exponential conjunction in $\Phi_i(\vec{y}, \vec{y'})$, we use universal automata, rather than nondeterministic ones. Thus, we take the intersection of an exponential number of universal automata of size exponential. The result is exponential, and we can then proceed with the algorithm of [Mogavero *et al.*, 2014].

Hence, the overall complexity is that of exponentially many iterations of a doubly or triply exponential procedure, which remains doubly or triply exponential, respectively. □

Combining Lemma 3, Theorem 6, Theorem 4, and the blow-ups and complexities in their proofs, we get a 2EXPTIME upper bound to the weak and a 3EXPTIME upper bound to the strong RS problems with $\text{LTL}[\mathcal{F}]$ objectives. A matching lower bound for the weak RS follows from hardness in 2EXPTIME to the problem for LTL [Kupferman *et al.*, 2016].

Corollary 1. *The weak and strong RS problems with* $\text{LTL}[\mathcal{F}]$ *objectives are 2EXPTIME-complete and in 3EXPTIME, respectively.*

Computing PoS and PoA In this section we consider the problem of computing the stability measures UPoS, UPoA, CPoS and CPoA for an RS game G. Since the range of the utility function is finite, the possible values of the stability measures are also finite. We therefore focus on the decision version of the problem, namely deciding whether e.g., $\text{UPoS}(G) \leq t$ for a given threshold t. We show that while computing UPoS and UPoA can be reduced to the RS problem, computing CPoS and CPoA is more involved, as we have to go over all possible profiles of the controllable agents.

We start with the uncontrollable measures (i.e., UPoS and UPoA), and compute their value by separately computing OPT and $\max_{P \in \Upsilon_G} \mathsf{utility}(P)$ (for UPoS) or $\min_{P \in \Upsilon_G} \mathsf{utility}(P)$ (for UPoA), as described below.

Theorem 7. *Given an RS game G with* $\text{LTL}[\mathcal{F}]$ *objectives and a threshold t, deciding whether* $\text{UPoS}(G) \leq t$ *is 2EXPTIME-complete, while deciding whether* $\text{UPoA}(G) \leq t$ *can be solved in 3EXPTIME.*

Proof. We consider the case of $\text{UPoS}(G)$, handling $\text{UPoA}(G)$ is analogous. Thus, we want to decide whether $\frac{OPT}{\max_{P \in \Upsilon_G} \mathsf{utility}(P)} \leq t$. We start by computing OPT, which amounts to computing the maximal satisfaction value of the $\text{LTL}[\mathcal{F}]$ formula $\mathsf{utility}(\varphi_0, \ldots, \varphi_n)$, where $\varphi_0, \ldots, \varphi_n$ are the objectives for the players. By [Almagor *et al.*, 2016], this can be done in PSPACE. Once OPT is computed, it remains to decide whether $\max_{P \in \Upsilon_G} \mathsf{utility}(P) \geq \frac{OPT}{t}$. This amount to solving the weak RS problem for G with threshold $\frac{OPT}{t}$ for the utility, and no thresholds (i.e., threshold 0) for the players. □

The controllable setting poses a bigger challenge, as we have to fix the profile P^C with which we compute $OPT(P^C)$ and e.g., $\max_{P \in \Upsilon_{G(P^C)}} \mathsf{utility}(P)$. We address this by formalizing the problem in SL.

[3] For simplicity, we assume that $\vec{y_{\alpha_0}}$ and $\vec{y_U}$ are the vectors of the controllable and uncontrollable agents, respectively.

Theorem 8. *Given an RS game G with* $\mathrm{LTL}[\mathcal{F}]$ *objectives and a threshold t, deciding whether* $\mathrm{CPoS}(G) \leq t$ *is 2EXPTIME-complete, while deciding whether* $\mathrm{CPoA}(G) \leq t$ *can be solved in 3EXPTIME.*

Proof. We consider here the case of $\mathrm{CPoS}(G)$, handling $\mathrm{CPoA}(G)$ is similar. Thus, we want to decide whether there is a partial profile $P^C \in \Pi_C$ such that $\frac{OPT(P^C)}{\max_{P \in \Upsilon_G(P^C)} \mathsf{utility}(P)} \leq t$. Consider the set $Range(\mathsf{utility}) = \{u_1, \ldots, u_m\}$ of possible values of utility. By Lemma 1, we have that m is single exponential in the description of G. Let $T = \{(t_1, t_2) : t_1, t_2 \in Range(\mathsf{utility}) \wedge t_1/t_2 \leq t\}$. Then, $\mathrm{CPoS}(G) \leq t$ iff there exist $(t_1, t_2) \in T$ and a partial profile $P^C \in \Pi_C$ such that $OPT(P^C) \leq t_1$ and $\max_{P \in \Upsilon_G(P^C)} \mathsf{utility}(P) \geq t_2$. Equivalently, the latter condition means that there exists a CNE $P \in \Pi$ that agrees with P^C and for which $\mathsf{utility}(P) \geq t_2$.

Accordingly, given $(t_1, t_2) \in T$, we can formulate the above in SL as follows: $\Phi^{\mathrm{CPoS}}(t_1, t_2) = \langle\langle \vec{x}_C \rangle\rangle ([\![\vec{y}_U]\!] \flat(x_C, y_U) \varphi_0^{<t_1} \wedge \langle\langle \vec{z}_U \rangle\rangle \flat(x_C, z_U)(\varphi_{NE}(z_U) \wedge \varphi_0^{>t_2}))$. Finally, we can decide whether $\mathrm{CPoS}(G) \leq t$ by model checking the formula $\bigvee_{(t_1,t_2) \in T} \Phi^{\mathrm{CPoS}}(t_1, t_2)$. Note that this can be done in 2EXPTIME using similar arguments as those made in Section 4. □

Remark 3. Recall the measures of B-utilitarian and B-egalitarian (and their anti- variants) [Nisan *et al.*, 2007] discussed in Example 1. As demonstrated there, for every game with $\mathrm{LTL}[\mathcal{F}]$ objectives, and every measure ν, we can describe the utility function that corresponds to ν by an $\mathrm{LTL}[\mathcal{F}]$ formula of linear size. Hence, calculation of the measures can be reduced to solving an RS game and is between 2EXPTIME and 3EXPTIME, according to which kind of RS is required to be used.

Acknowledgments

We are grateful to Bastien Maubert for the careful reading and for detecting an error in the complexity analysis of the strong rational-synthesis problem in an earlier version of this paper.

References

[Almagor *et al.*, 2015] S. Almagor, G. Avni, and O. Kupferman. Repairing Multi-Player Games. In *CONCUR'15*, pages 325–339, 2015.

[Almagor *et al.*, 2016] S. Almagor, U. Boker, and O. Kupferman. Formalizing and reasoning about quality. *J. ACM*, 63(3), 2016.

[Anshelevich *et al.*, 2008] E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. The price of stability for network design with fair cost allocation. *SIAM J. Comput.*, 38(4):1602–1623, 2008.

[Condurache *et al.*, 2016] R. Condurache, E. Filiot, R. Gentilini, and J. Raskin. The complexity of rational synthesis. In *Proc. ICALP'16*, pages 121:1–121:15, 2016.

[Chatterjee *et al.*, 2007] K. Chatterjee, T. Henzinger, and N. Piterman. Strategy Logic. In *Proc. CONCUR'07*, pages 59–73, 2007.

[Faella *et al.*, 2008] M. Faella, A. Legay, and M. Stoelinga. Model checking quantitative linear time logic. *Electr. Notes Theor. Comput. Sci.*, 220(3):61–77, 2008.

[Fisman *et al.*, 2010] D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In Proc. 16th TACAS, LNCS 6015, pages 190–204, 2010.

[Gutierrez *et al.*, 2015] J. Gutierrez, P. Harrenstein, and M. Wooldridge. Iterated Boolean Games. *Information and Computation*, 242:53–79, 2015.

[Gutierrez *et al.*, 2016] J. Gutierrez, G. Perelli, and M. Wooldridge. Imperfect Information in Reactive Modules Games. In *Proc. KR'16*, pages 390–400, 2016.

[Gutierrez *et al.*, 2017a] J. Gutierrez, A. Murano, G. Perelli, S. Rubin, and M. Wooldridge. Nash equilibria in concurrent games with lexicographic preferences. In *Proc. IJCAI'17*, pages 1067–1073, 2017.

[Gutierrez *et al.*, 2017b] J. Gutierrez, G. Perelli, and M. Wooldridge. Iterated games with LDL goals over finite traces. In *Proc. AAMAS'17*, pages 696–704, 2017.

[Gutierrez *et al.*, 2018] J. Gutierrez, G. Perelli, and M. Wooldridge. Imperfect information in reactive modules games. *Information and Computation*, 2018. Article in Press.

[Harel and Pnueli, 1985] D. Harel and A. Pnueli. On the development of reactive systems. In *Logics and Models of Concurrent Systems*, pages 477–498. Springer, 1985.

[Harrenstein *et al.*, 2001] P. Harrenstein, W. van der Hoek, J. Meyer, and C. Witteveen. Boolean games. In *Proc. 8th TARK*, pages 287–298, 2001.

[Koutsoupias and Papadimitriou, 2009] E. Koutsoupias and C. Papadimitriou. Worst-case equilibria. *Computer Science Review*, 3(2):65–69, 2009.

[Kupferman *et al.*, 2016] O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. *Ann. Math. Artif. Intell.*, 78(1):3–20, 2016.

[Mogavero *et al.*, 2010] F. Mogavero, A. Murano, and M.Y. Vardi. Reasoning About Strategies. In *Proc. 30th FSTTCS*, LIPIcs 8, pages 133–144, 2010.

[Mogavero *et al.*, 2012] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. What Makes ATL* Decidable? A Decidable Fragment of Strategy Logic. In *Proc. 23rd CONCUR*, LNCS 7454, pages 193–208, 2012.

[Mogavero *et al.*, 2014] F. Mogavero, A. Murano, G. Perelli, and M.Y. Vardi. Reasoning About Strategies: On the Model-Checking Problem. *ACM Transactions on Computational Logic*, 15(4):34:1–47, 2014.

[Mogavero *et al.*, 2017] F. Mogavero, A. Murano, G. Perelli, and M. Y. Vardi. Reasoning about strategies: on the satisfiability problem. *Logical Methods in Computer Science*, 13(1), 2017.

[Nash, 1950] J.F. Nash. Equilibrium points in n-person games. In *Proceedings of the National Academy of Sciences of the United States of America*, 1950.

[Nisan and Ronen, 1999] N. Nisan and A. Ronen. Algorithmic mechanism design. In *Proc. 31st STOC*, pages 129–140, 1999.

[Nisan *et al.*, 2007] N. Nisan, T. Roughgarden, E. Tardos, and V.V. Vazirani. *Algorithmic Game Theory*. Cambridge University Press, 2007.

[Papadimitriou, 2001] C. H. Papadimitriou. Algorithms, games, and the internet. In *Proc. 33rd STOC*, pages 749–753, 2001.

[Pnueli and Rosner, 1989] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th POPL*, pages 179–190, 1989.

[Wooldridge *et al.*, 2016] M. Wooldridge, J. Gutierrez, P. Harrenstein, E. Marchioni, G. Perelli, and A. Toumi. Rational verification: From model checking to equilibrium checking. In *Proc. AAAI'16*, pages 4184–4191. AAAI Press, 2016.