Ann Math Artif Intell (2016) 78:3–20
DOI 10.1007/s10472-016-9508-8

Synthesis with rational environments

Orna Kupferman[1] · Giuseppe Perelli[2] · Moshe Y. Vardi[3]

Published online: 21 June 2016

Abstract *Synthesis* is the automated construction of a system from its specification. The system has to satisfy its specification in all possible environments. The environment often consists of agents that have objectives of their own. Thus, it makes sense to soften the universal quantification on the behavior of the environment and take the objectives of its underlying agents into an account. Fisman et al. introduced *rational synthesis*: the problem of synthesis in the context of rational agents. The input to the problem consists of temporal logic formulas specifying the objectives of the system and the agents that constitute the environment, and a solution concept (e.g., Nash equilibrium). The output is a profile of strategies, for the system and the agents, such that the objective of the system is satisfied in the computation that is the outcome of the strategies, and the profile is stable according to the solution concept; that is, the agents that constitute the environment have no incentive to deviate from the strategies suggested to them. In this paper we continue to study rational synthesis. First, we suggest an alternative definition to rational synthesis, in which the

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC grant agreement 278410, by the Israel Science Foundation (grant 1229/10), and by the US-Israel Binational Science Foundation (grant 2010431).

Work partially done when the author was a Ph.D. student at the Università degli Studi di Napoli Federico II and while visiting Rice University. Research partially supported by the ERC Advanced Grant RACE (291528) at Oxford.

NSF Expeditions in Computing project "ExCAPE: Expeditions in Computer Augmented Program Engineering

✉ Giuseppe Perelli
perelli.gi@gmail.com

[1] The Hebrew University, Jerusalem, Israel

[2] University of Oxford, Oxford, UK

[3] Rice University, Houston, TX, USA

agents are rational but not cooperative. We call such problem *strong rational synthesis*. In the strong rational synthesis setting, one cannot assume that the agents that constitute the environment take into account the strategies suggested to them. Accordingly, the output is a strategy for the system only, and the objective of the system has to be satisfied in all the compositions that are the outcome of a stable profile in which the system follows this strategy. We show that strong rational synthesis is 2EXPTIME-COMPLETE, thus it is not more complex than traditional synthesis or rational synthesis. Second, we study a richer specification formalism, where the objectives of the system and the agents are not Boolean but quantitative. In this setting, the objective of the system and the agents is to maximize their outcome. The quantitative setting significantly extends the scope of rational synthesis, making the game-theoretic approach much more relevant. Finally, we enrich the setting to one that allows *coalitions* of agents that constitute the system or the environment.

Keywords Synthesis · Multi-agent systems · Formal methods · Logics for games

Mathematics Subject Classification (2010) 68

1 Introduction

Synthesis is the automated construction of a system from its specification. The basic idea is simple and appealing: instead of developing a system and verifying that it adheres to its specification, we would like to have an automated procedure that, given a specification, constructs a system that is correct by construction. The first formulation of synthesis goes back to Church [12]; the modern approach to synthesis was initiated by Pnueli and Rosner, who introduced LTL (linear temporal logic) synthesis [26]. The *LTL synthesis problem* receives as input a specification in LTL and outputs a reactive system modeled by a finite-state transducer satisfying the given specification — if such exists. It is important to distinguish between input signals, assigned by the environment, and output signals, assigned by the system. A system should be able to cope with all values of the input signals, while setting the output signals to desired values [26]. Therefore, the quantification structure on input and output signals is different. Input signals are universally quantified while output signals are existentially quantified.

Modern systems often interact with other systems. For example, the clients interacting with a server are by themselves distinct entities (which we call agents) and are many times implemented by systems. In the traditional approach to synthesis, the way in which the environment is composed of its underlying agents is abstracted. In particular, 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, often agents have objectives of their own, other than to fail the system. The approach taken in the field of algorithmic game theory [24] is to assume that agents interacting with a computational system are *rational*, i.e., agents act to achieve their own objectives. Assuming agents' rationality is a restriction on the agents behavior and is therefore equivalent to softening the universal quantification on the environment.[1] Thus, the following question arises: can

[1] Early work on synthesis has realized that the universal quantification on the behaviors of the environment is often too restrictive. The way to address this point, however, has been by adding assumptions on the environment, which can be part of the specification (cf., [9]).

system synthesizers capitalize on the rationality and objectives of agents interacting with the system?

In [14], Fisman et al. positively answered this question by introducing and studying *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, and a solution concept, e.g., dominant strategies, Nash Equilibria, and the like. The atomic propositions over which the objectives are defined are partitioned among the system and the agents, so that each of them controls a subset of the propositions. 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. Fisman et al. showed that there are specifications that cannot be realized in a hostile environment but are realizable in a rational environment. Moreover, the rational synthesis problem for LTL and common solution concepts used in game theory can be solved in 2ExpTime, thus its complexity coincides with that of usual synthesis.

In [17], we continued the study of rational synthesis and in this paper we extend those results. We present the following contributions. First, we suggest a strong definition of rational synthesis, in which the agents are rational but not cooperative. Second, we study a richer specification formalism, where the objectives of the system and the agents are not Boolean but quantitative. Third, we enrich the setting to one that allows *coalitions* of agents that constitute the system or the environment. Finally, we show that all these variants of the rational synthesis problems can be reduced to model checking in fragments of *Strategy Logic* [20, 22]. Before we describe our contributions in more detail, let us highlight a different way to consider rational synthesis and our contribution here. *Mechanism design* is a field in game theory and economics studying the design of games whose outcome (assuming agents rationality) achieves some goal [23, 24]. The outcome of traditional games depends on the final position of the game. In contrast, the systems we reason about maintain an *on-going interaction* with their environment, and we reason about their behavior by referring not to their final state (in fact, we consider non-terminating systems, with no final state) but rather to the *language* of computations that they generate. Rational synthesis can be viewed as a variant of mechanism design in which the game is induced by the objective of the system, and the objectives of both the system and the agents refer to their on-going interaction and are specified by temporal logic formulas. Our contributions here correspond to the classic setting assumed in mechanism design: the agents need not be cooperative, and the outcome is not Boolean.

We argue that the definition of rational synthesis in [14] is *cooperative*, in the sense that the agents that constitute the environment are assumed to follow the strategy profile suggested to them (as long as it is in an equilibrium). Here, we consider also a *strong* setting, in which the agents that constitute the environment may follow any strategy profile that is in an equilibrium, and not necessarily the one suggested to them by the synthesis algorithm. In many scenarios, the rational synthesis setting is indeed too optimistic, as the system cannot assume that the environment, even if it is rational, would follow a suggested strategy, rather than a strategy that is as good for it. Moreover, sometimes there is no way to communicate with the environment and suggest a strategy for it. From a technical point of view, we show that the strong rational synthesis setting requires reasoning about *all* possible equilibria, yet, despite this more sophisticated reasoning, it stays 2ExpTime-complete. We achieve the upper bound by reducing rational synthesis to the model-checking problem for Strategy Logic (Sl, for short). Sl is a specification formalism that allows to explicitly quantify over strategies in games as first-order objects [10]. While the model-checking problem for

strategy logic is in general non-elementary, we show that it is possible to express rational synthesis in the restricted *Nested-Goal* fragment of SL, introduced in [20], which leads to the desired complexity. It is important to observe the following difference between the non-strong and the strong settings. In the non-strong one, we synthesize strategies for all agents, with the assumption that the agent that corresponds to the system always follows his suggested strategy and the agents that constitute the environment decide in a rational manner whether to follow their strategies. On the other hand, in the strong setting, we synthesize a strategy only for the agent that corresponds to the system, and we assume that the agents that constitute the environment are rational, thus the suggested strategy has to win against all rational behaviors of the environment.

Recall that we study rational synthesis for different solution concepts in game theory. The most popular concept is Nash equilibrium, where a profile is in an equilibrium if no agent has an incentive to deviate from his strategy provided that the other agents stay with their strategies. A weakness of Nash equilibrium is that it is not stable: if one of the other agents deviates from his strategy, nothing is guaranteed. We address this weakness by formalizing and studying rational synthesis in the presence of coalitions. There, we allow settings in which the system and the environment are both composed of several agents that cooperate against the coalition of the other agents. We show that even this richer setting can be reduced to the Nested-Goal fragment of SL.

We then turn to address a weakness of the classical synthesis problem, a weakness that is more apparent in the rational setting. In classical synthesis, the specification is Boolean and describes the expected behavior of the system. In many applications, systems can satisfy their specifications at different levels of quality. Thus, synthesis with respect to Boolean specifications does not address designers's needs. This latter problem is a real obstacle, as 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. In the last years we see a lot of effort on developing formalisms that would enable the specification of such quality measures [2, 6].

Classical applications of game theory consider games with quantitative payoffs. In the Boolean setting, we assumed that the payoff of an agent is 1, if its objective is satisfied, and is 0 otherwise. In particular, this means that agents whose objectives are not satisfied have no incentive to follow any strategy, even if the profile satisfies the solution concept. In real-life, rational objectives are rarely Boolean. Thus, even beyond our goal of synthesizing systems of high quality, the extension of the synthesis problem to the rational setting calls also for an extension to a quantitative setting. Unfortunately, the full quantitative setting is undecidable already in the context of model checking [16]. In [14], Fisman et al. extended rational synthesis to objectives in the multi-valued logic LLTL, where specifications take truth values from a finite lattice.

We introduce here a new quantitative specification formalism, termed *Objective* LTL, (OLTL, for short). We first define the logic, and then study its rational synthesis. Essentially, an OLTL specification is a pair $\theta = \langle \Psi, \mathsf{f} \rangle$, where $\Psi = \langle \psi_1, \psi_2, \ldots, \psi_m \rangle$ is a tuple of LTL formulas and $\mathsf{f} : \{0,1\}^m \to \mathbb{Z}$ is a *reward function*, mapping Boolean vectors of length m to an integer. A computation η then maps θ to a reward in the expected way, according to the subset of formulas that are satisfied in η. 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. Again, we distinguish between a non-strong and a strong setting. Note that the notion of an equilibrium in the quantitative setting is much more interesting, as it means that all agents in the environment cannot expect to increase their payoffs. We show that

the quantitative setting is not more complex than the non-quantitative one, thus quantitative rational synthesis is complete for EXPTIME in both the non-strong and strong settings.

2 Preliminaries

2.1 Games

A *concurrent game structure* (CGS, for short) [4] is a tuple $\mathcal{G} \triangleq \langle \Phi, \Omega, (A_i)_{i\in\Omega}, S$, where Φ and $\Omega = \{\alpha_0, \ldots, \alpha_k\}$ are finite sets of *atomic propositions* and *agents*, α_i are disjoint sets of *actions*, one for each agent α_i, S is a finite set of *states*, $s_0 \in S$ is a designated *initial state*, and $\lambda : S \to 2^{\Phi}$ is a *labeling function* that maps each state to the set of atomic propositions true in that state. By $A \triangleq \bigcup_{i\in\Omega} \alpha_i$ we denote the union of all possible actions for all the agents. Let $D \triangleq A_0 \times \ldots \times A_k$ be the set of *decisions*, i.e., $(k+1)$-tuples of actions representing the choices of an action for each agent. Then, $\tau : S \times D \to S$ is a deterministic *transition function* mapping a pair of a state and a decision to a state.

A *path* in a CGS $\mathcal{G}$ is an infinite sequence of states $\eta = \eta 0 \cdot \eta 1 \cdot \ldots \in S^{\omega}$ that agrees with the transition function, i.e., such that, for all $i \in \mathbb{N}$, there exists a decision $\mathsf{d} \in D$ such that $\eta i + 1 = \tau(\eta i, \mathsf{d})$. A *track* in a CGS $\mathcal{G}$ is a prefix ρ of a path η, also denoted by $\eta{\leq} n$, for a suitable $n \in \mathbb{N}$. A track ρ is *non-trivial* if $|\rho| > 0$,[2] i.e., $\rho \neq \varepsilon$. We use $Pth \subseteq S^{\omega}$ and $Trk \subseteq S^{+}$ to denote the set of all paths and non-trivial tracks, respectively. Also, for a given $s \in S$, we use Pth^s and Trk^s to denote the subsets of paths and tracks starting from $s \in S$. Intuitively, the game starts in the state s_0 and, at each step, each agent selects an action in its set. The game then deterministically proceeds to the next state according to the corresponding decision. Thus, the outcome of a CGS is a path, regulated by individual actions of the agents.

A *strategy* for Agent α_i is a tool used to decide which decision to take at each phase of the game. Formally, it is a function $\pi_i : Trk \to \alpha_i$ that maps each non-trivial track to a possible action of Agent α_i. By Π_i we denote the set of all possible strategies for agent α_i. A *strategy profile* is a $(k+1)$-tuple $\mathsf{P} = \langle \pi_0, \ldots, \pi_k \rangle \in \Pi_0 \times \ldots \times \Pi_k$ that assigns a strategy to each agent. We denote by $\mathcal{P} \triangleq \Pi_0 \times \ldots \times \Pi_k$ the set of all possible strategy profiles. Moreover, given a strategy profile P and a set of strategies P_A, one per each agent α in A, by $\mathsf{P}[A \leftarrow \mathsf{P}[A]]$ we denote the strategy profile obtained from P by replacing for agents in A, the strategies given in $\mathsf{P}[A]$.

For a strategy profile P and a state s, we use $\eta = \mathsf{play}(\mathsf{P}, s)$ to denote the path that is the outcome of a game that starts in s and agrees with P, i.e., for all $i \in \mathbb{N}$, it holds that $\eta[i+1] = \tau(\eta[i], \mathsf{d}[i])$, where $\mathsf{d}[i] = (\pi_0(\eta[\leq i]), \ldots, \pi_k(\eta[\leq i]))$. By $\mathsf{play}(\mathsf{P}) = \mathsf{play}(\mathsf{P}, s_0)$ we denote the unique path starting from s_0 obtained from P.

We model reactive systems by deterministic transducers. A *transducer* is a tuple $\mathcal{T} = \langle \mathrm{I}, \mathrm{O}, \mathrm{S}, s_0, \delta, \mathsf{L} \rangle$, where I is a set of input signals assigned by the environment, O is a set of output signals, assigned by the system, S is a set of states, s_0 is an initial state, $\delta : \mathrm{S} \times 2^{\mathrm{I}} \to \mathrm{S}$ is a transition function, and $\mathsf{L} : \mathrm{S} \to 2^{\mathrm{O}}$ is a labeling function. When the system is in state $s \in \mathrm{S}$ and it reads an input assignment $\sigma \in 2^{\mathrm{I}}$, it changes its state to $s' = \delta(s, \sigma)$ where it outputs the assignment $\mathsf{L}(s')$. Given a sequence $\varrho = \sigma_1, \sigma_2, \sigma_3, \ldots \in (2^{\mathrm{I}})^{\omega}$ of inputs, the *execution* of $\mathcal{T}$ on ϱ is the sequence of states $s_0, s_1, s_2, \ldots$ such that for all

[2] By $|\rho|$ we denote the length of ρ, classically defined.

$j \geq 0$, we have $s_{j+1} = \delta(s_j, \sigma_j)$. The *computation* $\eta \in (2^{\mathrm{I}} \times 2^{\mathrm{O}})^\omega$ of S on ϱ is then $\langle \mathsf{L}(s_0), \sigma_1 \rangle, \langle \mathsf{L}(s_1), \sigma_2 \rangle, \langle \mathsf{L}(s_2), \sigma_3 \rangle, \ldots$.

2.2 Strategy logic

Strategy Logic [21, 22] (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*, allows us to restrict attention to restricted classes of strategy profiles, determining a subset of paths, in which a temporal specification is desired to be satisfied. Since nesting of quantifications and bindings is possible, such temporal specifications can be recursively formulated by an SL subsentence. From a syntactic point of view, SL is an extension of LTL with disjoint sets of strategy variables $\mathrm{Var}_0, \ldots, \mathrm{Var}_k$, where Var_i is a set of strategy variables for Agent α_i, existential ($\langle\langle x_i \rangle\rangle$) and universal ($[x_i]$) strategy quantifiers, and a binding operator of the form (α_i, x_i) that couples an agent α_i with one of its variables $x_i \in \mathrm{Var}_i$.

We first introduce some technical notation. For a tuple $t = (t_0, \ldots, t_k)$, by $t[i \leftarrow d]$ we denote the tuple obtained from t by replacing the i-th component with d. We use $\vec{x}$ as an abbreviation for the tuple $(x_0, \ldots, x_k) \in \mathrm{Var}_0 \times \ldots \times \mathrm{Var}_k$. 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 $\flat(\vec{x}) = (\alpha_0, x_0) \ldots (\alpha_k, x_k)$ we denote the existential and universal quantification, and the binding of all the agents to the strategy profile variable $\vec{x}$, respectively. Finally, by $\flat(\vec{x}_{-i}, y_i) = (\alpha_0, x_0) \ldots (\alpha_i, y_i) \ldots (\alpha_k, x_k)$ we denote the changing of binding for Agent α_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 *Nested-Goal* fragment of SL, namely SL[NG], introduced in [20]. Formulas in SL[NG] are defined with respect to a set Φ of atomic proposition, a set Ω of agents, and sets Var_i of strategy variables for Agent $\alpha_i \in \Omega$. The set of SL[NG] formulas is defined by the following grammar:

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

where, $p \in \Phi$ is an atomic proposition, $x_i \in \mathrm{Var}_i$ is a variable, and $\vec{x} \in \mathrm{Var}_0 \times \ldots \times \mathrm{Var}_k$ is a tuple of variables, one for each agent.

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

As an example, $\langle\langle x_0 \rangle\rangle [[x_1]] [[x_2]] \flat(\vec{x})(p \mathsf{U}\, q) \vee [[y_0]] \langle\langle y_1 \rangle\rangle \langle\langle y_2 \rangle\rangle \flat(\vec{y})(\mathsf{G}\,\mathsf{F}\, p \wedge \mathsf{G}\, \neg q)$ is an SL[NG] formula stating that either the system α_0 has a strategy x_0 to enforce $p \mathsf{U}\, q$, no matters what the environment agents $\alpha[1]$ and $\alpha[2]$ do, or, for all possible behaviors y_0 of the system, the environment agents $\alpha[1]$ and $\alpha[2]$ have strategies y_1 and y_2 to enforce both $\mathsf{G}\,\mathsf{F}\, p$ and $\mathsf{G}\, \neg q$.

We denote by $\mathsf{free}(\varphi)$ the set of strategy variables occurring in φ but not in a scope of a quantifier. A formula φ is *closed* if $\mathsf{free}(\varphi) = \emptyset$.

Similarly to the case of first order logic, an important concept that characterizes the syntax of SL is the one of *alternation depth* of quantifiers, i.e., the maximum number of quantifier switches $\langle\langle x_i \rangle\rangle [[x_j]]$, $[[x_i]] \langle\langle x_j \rangle\rangle$, in the formula. A precise formalization of the concepts of alternation depth can be found in [21, 22].

Now, in order to define the semantics of SL, we use the auxiliary concept of assignment. Let $\mathrm{Var} = \bigcup_{i=0}^{k} \mathrm{Var}_i$ be a set of variables for the agents in Ω, an *assignment* is a function $\chi : \mathrm{Var} \cup \Omega \rightarrow \Pi$ mapping variables and agents to a relevant strategy, i.e., for all $\alpha_i \in \Omega$

and $x_i \in \mathrm{Var}_i$, we have that $\chi(\alpha_i), \chi(x_i) \in \Pi_i$. Let $\mathrm{Asg} \stackrel{\triangle}{=} \Pi^{\mathrm{Var}\cup\Omega}$ denote the set of all assignments. For an assignment χ and elements $l \in \mathrm{Var} \cup \Omega$, we use $\chi[l \mapsto \pi] \in \mathrm{Asg}$ to denote the new assignment that returns π on l and the value of χ on the other ones, i.e., $\chi[l \mapsto \pi](l) \stackrel{\triangle}{=} \pi$ and $\chi[l \mapsto \pi](l') \stackrel{\triangle}{=} \chi(l')$, for all $l' \in (\mathrm{Var} \cup \Omega) \backslash \{l\}$. By $\mathsf{play}(\chi, s)$ we denote the path $\mathsf{play}(\mathsf{P}, s)$, for the strategy profile P that is compatible with χ.

We now describe when a given game $\mathcal{G}$ and a given assignment χ satisfy an SL formula φ, where $\mathsf{dom}(\chi)$[3] $= \mathsf{free}(\varphi) \cup \Omega$. We use $\mathcal{G}, \chi, s \models \varphi$ to indicate that the path $\mathsf{play}(\chi, s)$ satisfies φ over the CGS $\mathcal{G}$. For φ in LTL, the semantics is as usual [19]. For the other operators, the semantics is as follows.

1. $\mathcal{G}, \chi, s \models \langle\langle x_i \rangle\rangle \varphi$ if there exists a strategy π_i for α_i such that $\mathcal{G}, \chi[x_i \mapsto \pi_i], s \models \varphi$;
2. $\mathcal{G}, \chi, s \models [[x_i]]\varphi$ if, for all strategies π_i for α_i, it holds that $\mathcal{G}, \chi[x_i \mapsto \pi_i], s \models \varphi$;
3. $\mathcal{G}, \chi, s \models \flat(\vec{x})\varphi$ if it holds that $\mathcal{G}, \chi[\alpha_0 \mapsto x_0] \ldots [\alpha_k \mapsto x_k], s \models \varphi$.

Finally, we say that $\mathcal{G}$ satisfies φ, and write $\mathcal{G} \models \varphi$, if there exists an assignment χ such that $\mathcal{G}, \chi, s_0 \models \varphi$.

Intuitively, at Items 1 and 2, we evaluate the existential and universal quantifiers over a variable x_i by associating with it a suitable strategy. At Item 3 we commit the agents to use the strategy contained in the tuple variable $\vec{x}$.

Theorem 1 [20] *The model-checking problem for* SL[NG] *is of* $(d+1)$EXPTIME *complexity, with* d *being the alternation depth of the specification, and of* PTIME*complexity with respect to the size of the model.*

3 Rational synthesis

We define two variants of rational synthesis. The first, *rational synthesis*, was introduced in [14]. The second, *strong rational synthesis*, is new.

We work with the following model: the world consists of a *system* and an environment composed of k agents: $\alpha[1], \ldots, \alpha_k$. For uniformity, we refer to the system as Agent α_0. We assume that Agent α_i controls a set X_i of propositions, and the different sets are pairwise disjoint. At each point in time, each agent sets his propositions to certain values. Let $\mathrm{X} = \bigcup_{0 \le i \le k} \mathrm{X}_i$, and $\mathrm{X}_{-i} = \mathrm{X} \setminus \mathrm{X}_i$. Each agent α_i (including the system) has an objective φ_i, specified as an LTL formula over X.

This setting induces the CGS $\mathcal{G}[Syn] = \langle \Phi, \Omega, (A_i)_{i \varepsilon \Omega}, S$ defined as follows. The set of agents $\Omega = \{\alpha_0, \alpha[1], \ldots, \alpha_k\}$ consists of the system and the agents that constitute the environment. The actions of Agent α_i are the possible assignments to its variables. Thus, $\alpha_i = 2^{\mathrm{X}_i}$. We use A and A_{-i} to denote the sets 2^{X} and $2^{\mathrm{X}_{-i}}$, respectively. The states of the game record the current assignment to the variables. Hence, $S = A$, and for all $s \in S$ and $\langle \sigma_0, \ldots, \sigma_k \rangle \in A[0] \times A[1] \times \cdots \times A[k]$, we have $\delta(s, \sigma_0, \ldots, \sigma_k) = \langle \sigma_0, \cdots, \sigma_k \rangle$.

A strategy for the system is a function $\pi_0 : Trk \rightarrow A[0]$. In the standard synthesis problem, we say that π_0 realizes φ_0 if, no matter which strategies the agents composing the environment follow, all the paths in which the system follows π_0 satisfy φ_0. In rational synthesis, on the contrary, we assume that the agents that constitute the environment are rational, which softens the universal quantification on the behavior of the environment.

[3] By $\mathsf{dom}(\mathsf{f})$ we denote the domain of the function f.

Recall that the rational synthesis problem gets a solution concept as a parameter. As discussed in Section 1, the fact that a strategy profile is a solution with respect to the concept guarantees that it is not worthwhile for the agents constituting the environment to deviate from the strategies assigned to them. Several solution concepts are studied and motivated in game theory. Here, we focus on the concepts of *dominant strategy* and *Nash equilibrium*, defined below.

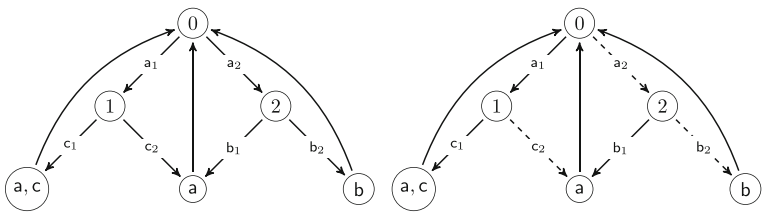
The common setting in game theory is that the objective for each agent is to maximize his *payoff* – a real number that is a function of the outcome of the game. We use $\mathsf{payoff}_i : Pth \to \mathbb{R}$ to denote the payoff function of Agent α_i. That is, payoff_i assigns to each possible path η a real number $\mathsf{payoff}_i(\eta)$ expressing the payoff of α_i on η. For a strategy profile P, we use $\mathsf{payoff}_i(\mathsf{P})$ to abbreviate $\mathsf{payoff}_i(\mathsf{play}(\mathsf{P}, s_0))$. In the case of an LTL objective ψ_i, we define $\mathsf{payoff}_i(\eta) = 1$ if $\eta \models \psi_i$ and $\mathsf{payoff}_i(\eta) = 0$, otherwise.

3.1 Solution concepts

The simplest and most appealing solution concept is dominant-strategies solution [25]. A *dominant strategy* is a strategy that maximizes agent's payoff, regardless of the strategies of the other agents. Therefore, if there is a profile of strategies $\mathsf{P} = \langle \pi_0, \ldots, \pi_k \rangle$ in which all strategies π_i are dominant, then no agent has an incentive to deviate from the strategy assigned to him in P. Formally, P is a *dominant strategy profile* if for every $1 \leq i \leq k$ and for every (other) profile P', we have that $\mathsf{payoff}_i(\mathsf{P}') \leq \mathsf{payoff}_i(\mathsf{P}'[i \leftarrow \pi_i])$.

As an example, consider the game in Fig. 1a, played by three agents, Alice, Bob, and Charlie, whose actions are $\{\mathsf{a}_1, \mathsf{a}_2\}$, $\{\mathsf{b}_1, \mathsf{b}_2\}$, and $\{\mathsf{c}_1, \mathsf{c}_2\}$, respectively. The arrows are labeled with the possible actions of the agents. Each agent wants to visit a state marked with his initial letter, infinitely often. In this game, the strategy for Alice of always choosing a_1 on node 0 is dominant. Indeed, no matter what is the choice for the other players, the execution hits infinitely often a state labeled with an a, thus satisfying Alice's objective. On the other hand, Bob and Charlie have no dominant strategies, since their objectives depend in a decisive way on the strategies adopted by Alice. In several games, it can happen that agents have no dominant strategy. For this reason, one would consider also other kinds of solution concepts.

Another well known solution concept is Nash equilibrium [25]. A strategy profile is a *Nash equilibrium* if no agent has an incentive to deviate from his strategy in P provided that the other agents adhere to the strategies assigned to them in P. Formally, P is a *Nash equilibrium profile* if for every $1 \leq i \leq k$ and for every (other) strategy π_i' for agent α_i, we have that $\mathsf{payoff}_i(\mathsf{P}[i \leftarrow \pi_i']) \leq \mathsf{payoff}_i(\mathsf{P})$. An important advantage of Nash equilibrium



(a) The game (b) Nash equilibrium

Fig. 1 A game and a Nash Equilibrium on it

is that it is more likely to exist than an equilibrium of dominant strategies [25].[4] As an example, consider again the game in Fig. 1a and the strategy profile illustrated by the non-dotted arrows as in Fig. 1b, i.e., players a, b and c pick the actions a_1, b_1 and c_1, respectively. It is easy to see that it is a Nash Equilibrium. Indeed, players a and c reach a winning point, while player b, assuming the other players adhere to their strategies, can never improve his payoff, as his winning point is never reachable.

For the case of repeated-turn games like infinite games, a suitable refinement of Nash Equilibria is the *Subgame perfect Nash-equilibrium* [27] (SPE, for short). A strategy profile $\mathsf{P} = \langle \pi_0, \ldots, \pi_k \rangle$ is an SPE if for every possible history of the game, no agent α_i has an incentive to deviate from her strategy π_i, assuming that the other agents follow their strategies in P. Intuitively, an SPE requires the existence of a Nash Equilibrium for each subgame starting from any possible finite path of the original one. In [14], the authors have studied rational synthesis also for the solution concept of SPE. To do this, the synthesis algorithm in [14] was extended to consider all possible histories of the game. In SL such a path property can be expressed combining strategy quantifiers with temporal operators. Indeed, the formula $\varphi = [\![\vec{x}]\!](\vec{x})\mathsf{G}\,\psi(\vec{y})$, with $\mathsf{free}(\varphi) = \vec{y}$, states that, for all possible strategy profile the agents can follow, the game is always in a position in which the formula $\psi(\vec{y})$ holds. Thus, for all possible paths that can be generated by agents, the property holds. By replacing $\psi(\vec{y})$ with the above formula, we then obtain a formula that represents SPEs. Hence, the non-strong and strong synthesis problems can be asserted in SL also for SPE, and our results hold also for this solution concept.

A weakness of Nash equilibrium is that it is not nearly as stable as a dominant-strategy solution: if one of the other agents deviates from his assigned strategy, nothing is guaranteed. Indeed, consider again the game in Fig. 1 and the Nash Equilibrium described above. If players a and b collaborate, they can improve the payoff of b by taking the action a_2 and b_2, respectively. So, NE is not expressive enough to handle *coalitions* and to identify deviations due to collaborations among agents. To handle also this case, the concept of h-resilient has been introduced [5, 15]. A strategy profile P is a h-resilient equilibrium if it tolerates deviations of up to h players in the game. Formally, for all sets $\mathrm{A} \subseteq \Omega$ with $|\mathrm{A}| \leq h$, and for all sets of strategies $\mathsf{P}[\mathrm{A}]$, it holds that $\mathsf{payoff}_i(\mathsf{P}[\mathrm{A} \leftarrow \mathsf{P}[\mathrm{A}]]) \leq \mathsf{payoff}_i(\mathsf{P})$, for all $i \in \mathrm{A}$. In the case that deviation from a given strategy profile is unavoidable, it may happen that such deviation affects the non deviating agents payoff. Indeed, in the case the agents a and b deviate from the NE described in Fig. 1b, player c loses. To investigate this problem, the concept of t-immune equilibrium has been introduced [1, 15]. A strategy profile P is a t-immune equilibrium if, in the case up to t agents deviate from it, the remaining ones do not decrease their own payoff. Formally, for all sets $\mathrm{A} \subseteq \Omega$ with $|\mathrm{A}| \leq t$, and for all sets of strategies $\mathsf{P}[\mathrm{A}]$, it holds that $\mathsf{payoff}_i(\mathsf{P}) \leq \mathsf{payoff}_i(\mathsf{P}[\mathrm{A} \leftarrow \mathsf{P}[\mathrm{A}]])$, for all $i \in \Omega \setminus \mathrm{A}$.

Clearly, we can combine h-resilient and t-immune equilibria. We say that a strategy profile P is (h, t)-robust if it is both h-resilient and t-immune.

According to their definitions, it is not hard to see that, for games with k players, dominant strategies equilibria corresponds exactly to $(0, k-1)$-robust equilibria, while NE corresponds exactly to $(1, 0)$-robust equilibria.

In rational synthesis, we control the strategy of the system and assume that the agents that constitute the environment are rational. Consider a strategy profile $\mathsf{P} = \langle \pi_0, \ldots, \pi_k \rangle$

[4] In particular, all k-agent turn-based games with ω-regular objectives have Nash equilibrium [11].

and a solution concept γ (that is, dominant, NE, SPE, k-resilient, or t-immune). We say that P is *correct* if $\mathsf{play}(\mathsf{P})$ satisfies φ_0. We say that P is in a π_0-fixed γ-equilibrium if the agents composing the environment have no incentive to deviate from their strategies according to the solution concept γ, assuming that the system continues to follow π_0. More generally, we say that P is in a $\mathsf{P}[\mathrm{A}]$-fixed γ-equilibrium if the agents in $\Omega \setminus \mathrm{A}$ have no incentive to deviate from their strategies according to the solution concept γ, assuming the agents in A continue to follow the strategies provided in P.

3.2 Formal definition of rational synthesis

In the context of objectives in LTL, we assume the following simple payoffs. If the objective φ_i of Agent α_i holds, then his payoff is 1, and if φ_i does not hold, then the payoff of Agent i is 0. Accordingly, $\mathsf{P} = \langle \pi_0, \ldots, \pi_k \rangle$ is in a π_0-fixed dominant-strategy equilibrium if for every $1 \leq i \leq k$ and profile $\mathsf{P}' = \langle \pi'_0, \ldots, \pi'_k \rangle$ with $\pi'_0 = \pi_0$, if $\mathsf{play}(\mathsf{P}') \models \varphi_i$, then $\mathsf{play}(\mathsf{P}'[i \leftarrow \pi_i]) \models \varphi_i$. Also, P is in a Nash-equilibrium if for every $1 \leq i \leq k$ and strategy π'_i, if $\mathsf{play}(\mathsf{P}[i \leftarrow \pi'_i]) \models \varphi_i$, then $\mathsf{play}(\mathsf{P}) \models \varphi_i$.

Definition 1 (Rational synthesis) The input to the rational strategy problem is a set X of atomic propositions, partitioned into $\mathrm{X}_0, \ldots, \mathrm{X}_k$, LTL formulas $\varphi_0, \ldots, \varphi_k$, describing the objectives of the system and the agents composing the environment, and a solution concept γ. We distinguish between two variants of the problem:

1. In *Rational synthesis* [14], the desired output is a strategy profile P such that $\mathsf{play}(\mathsf{P})$ satisfies φ_0 and P is a π_0-fixed γ-equilibrium.
2. In *Strong rational synthesis*, the desired output is a strategy π_0 for the system such that for every strategy profile P that includes π_0 and is a π_0-fixed γ-equilibrium, we have that $\mathsf{play}(\mathsf{P})$ satisfies φ_0.

Thus, in the variant of [14], we assume that once we suggest to the agents in the environment strategies that are in a γ-equilibrium, they will adhere to the suggested strategies. In the strong variant we introduce here, the agents may follow any strategy profile that is in a γ-equilibrium, and thus we require the outcome of all these profiles to satisfy φ_0. It is shown in [14] that the rational synthesis problem is 2ExpTime-complete.

As a more general case, we can assume that also the system is composed by a set of agents, rather than a single one. For this case, it makes sense to consider the whole set of agents divided into two *coalitions*, one for the system agents, and the other for the environment ones, both of them having assigned a specific solution concept to deal with. Formally, we have the following.

Definition 2 (Coalition vs Coalition Rational Synthesis) The input of the Coalition vs Coalition rational synthesis problem is given by two sets of agents $\mathrm{S} = \{\alpha_1, \ldots, \alpha_n\}$ and $\mathrm{E} = \{\beta_1, \ldots, \beta_m\}$, representing the system and the environment, respectively, a set X of atomic propositions, partitioned into $\mathrm{X}_{\alpha_1}, \ldots, \mathrm{X}_{\alpha_n}, \mathrm{X}_{\beta_1}, \ldots \mathrm{X}_{\beta_m}$, LTL formulas $\phi_1, \ldots, \psi_n, \phi_1, \ldots, \phi_m$ describing the objectives of the system and environment coalitions, and two solution concepts γ_{S} and γ_{E}. We distinguish between two variants of the problem:

1. In *Rational synthesis*, the desired output is a strategy profile P that is in a $\mathsf{P}[\mathrm{E}]$-fixed γ_{S}-equilibrium and in a $\mathsf{P}[\mathrm{S}]$-fixed γ_{E}-equilibrium.

2. In *Strong rational synthesis*, the desired output is a partial strategy profile $\mathsf{P}[S]$ for the system coalition such that for every strategy profile P that extends $\mathsf{P}[S]$ and is a $\mathsf{P}[S]$-fixed γ_E-equilibrium, we have that it is also a $\mathsf{P}[E]$-fixed γ_S-equilibrium.

Remark 1 The h-resilient equilibria solution concepts are used to measure the "degree of stability" of a given strategy profile. Instead of what Definition 2 prescribes, they do not describe any behavior of a specific coalition, but only that there is no group of size smaller than a fixed parameter having incentive to cooperatively deviate from the suggested strategy. We can then combine the coalition rational synthesis with such a kind of equilibria, in order to obtain a solution concept specification, saying that a given coalition is stable up to h deviators inside it.

Note that the input to the rational synthesis problem may not have a solution, so when we solve the rational synthesis problem, we first solve the *rational realizability* problem, which asks if a solution exists. As with classical synthesis, the fact that SL model-checking algorithms can be easily modified to return a regular witness for the involved strategies in case an existentially quantified strategy exists, makes the realizability and synthesis problems strongly related.

In the following example, we show that Rational Synthesis and Strong Rational Synthesis problems can have different answers on the same instances. We start with the solution concept of dominant strategies (Example 1), and then continue to Nash Equilibrium (Examples 2 and 3).

Example 1 Consider the three-player game in which the agents α_0 (system), $\alpha[1]$ and $\alpha[2]$ (environment) control $\{\mathsf{t}\}$, $\{\mathsf{p}, \mathsf{q}\}$, and $\{\mathsf{r}, \mathsf{s}\}$, respectively. Let the players' objective be $\varphi_0 = (\mathsf{F}\,\mathsf{p}) \wedge (\mathsf{F}\,\mathsf{r}) \wedge (\mathsf{F}\,\mathsf{t})$, $\varphi_1 = (\mathsf{F}\,\mathsf{p}) \vee (\mathsf{F}\,\mathsf{q})$, and $\varphi_2 = (\mathsf{F}\,\mathsf{r}) \vee (\mathsf{F}\,\mathsf{s})$, respectively. It is easy to see that the strategies "eventually play t", "eventually play p, and "eventually play r" for α_0, $\alpha[1]$, and $\alpha[2]$, respectively, is a solution for the Rational Synthesis problem with dominant strategies as solution concept. However, also the strategies "eventually play q and never play p" and "eventually play s and never play r" for $\alpha[1]$ and $\alpha[2]$ is a pair of dominant strategies, no matters what α_0 plays, and φ_0 cannot be satisfied. Hence, the Strong Rational Synthesis has a negative answer.

This difference between Rational Synthesis and Strong Rational Synthesis applies also in the case the solution concept is the Nash Equilibrium, as the following example shows.

Example 2 Consider a file-sharing network with the system and an environment consisting of two agents. The system controls the signals d_1 and d_2 (Agent $\alpha[1]$ and $\alpha[2]$ can download, respectively) and it makes sure that an agent can download only when the other agent has uploaded in the previous step. The system's objective is that both agents will upload infinitely often. Agent $\alpha[1]$ controls the signal u_1 (Agent $\alpha[1]$ uploads), and similarly for Agent $\alpha[2]$ and u_2. The objective of both agents is to download infinitely often.

Formally, the set of atomic propositions is $X = \{\mathsf{d}_1, \mathsf{d}_2, \mathsf{u}_1, \mathsf{u}_2\}$, partitioned into $X_0 = \{\mathsf{d}_1, \mathsf{d}_2\}$, $X_1 = \{\mathsf{u}_1\}$, and $X_2 = \{\mathsf{u}_2\}$. The objectives of the system and the environment are as follows.

– $\varphi_0 = \mathsf{G}\,(\neg \mathsf{u}_1 \rightarrow \mathsf{X}\,\neg \mathsf{d}_2) \wedge \mathsf{G}\,(\neg \mathsf{u}_2 \rightarrow \mathsf{X}\,\neg \mathsf{d}_1) \wedge \mathsf{G}\,\mathsf{F}\,\mathsf{u}_1 \wedge \mathsf{G}\,\mathsf{F}\,\mathsf{u}_2$,

- $\varphi_1 = \mathsf{G}\,\mathsf{F}\,\mathsf{d}_1$,
- $\varphi_2 = \mathsf{G}\,\mathsf{F}\,\mathsf{d}_2$.

First, note that in standard synthesis, φ_0 is not realizable, as a hostile environment does not need to upload. In the rational synthesis setting, the system can suggest to both agents the following TIT FOR TAT strategy: upload at the first time step, and from that point onward upload iff the other agent uploads. The system itself follows a strategy π_0 according to which it enables downloads whenever possible (that is, d_2 is true whenever Agent $\alpha[1]$ has uploaded in the previous step, and d_1 is true whenever Agent $\alpha[2]$ has uploaded in the previous step). It is not hard to see that the tit for tat strategy profile is a π_0-fixed Nash Equilibrium. Indeed, φ_1 and φ_2 are satisfied and so agents $\alpha[1]$ and $\alpha[2]$ do not have any incentive to deviate. Moreover, also φ_0 is satisfied. Thus the rational synthesis problem with Nash Equilibrium as solution concept has a positive answer. What about the strong rational synthesis? Consider the above strategy π_0 of the system, and consider strategies for the agents that never upload. The tuple of the three strategies is in a π_0-fixed Nash equilibrium. Indeed, assuming the system to hold on π_0, neither agent $\alpha[1]$ nor $\alpha[2]$ can deviate and get its objective achieved. Moreover, note that φ_0 is not satisfied. This means that the strong rational synthesis problem has a negative answer.

Example 3 Consider an ATM system composed by two agents α_1 and α_2, controlling the sets of variables $\{\mathsf{w}_1, \mathsf{w}_2\}$ and $\{\mathsf{fail}\}$, respectively, and two environment agents β_1 and β_2, controlling the variables r_1 and r_2, respectively. The agent α_1 is in charge of performing a withdrawal w_i for the agent β_i whenever it requests by using r_i, unless a failure state fail appears in the execution. Moreover, to make the ATM fair and secure, it cannot allow withdrawing in case no request has been performed and, after a single occurrence of a fail signal, has to immediately stop allowing withdrawals forever. We can represent these properties by $\varphi_{allow} = \bigwedge_{i=1}^{2}(\mathsf{G}\,(\mathsf{r}_i \rightarrow \mathsf{F}\,\mathsf{w}_i))$, $\varphi_{hold} = \bigwedge_{i=1}^{2}\mathsf{G}\,(\mathsf{w}_i \rightarrow \mathsf{X}\,(\neg\mathsf{w}_i\,\mathsf{U}\,\mathsf{r}_i))$, and $\varphi_{fail} = \mathsf{F}\,(\mathsf{fail} \wedge \mathsf{X}\,\mathsf{G}\,(\neg\mathsf{w}_1 \wedge \neg\mathsf{w}_2))$. The agent α_2 is in charge of checking whether the request flow has not been tricked by a bad behavior of the environment. This is done by issuing a failure whenever a request of an agent overlaps the withdrawal execution for the other agent. We can represent this property by means of the formula $\varphi_{failcheck} = \bigvee_{i=1}^{2}(\mathsf{F}\,(\mathsf{r}_i \wedge \neg\mathsf{w}_i\,\mathsf{U}\,\mathsf{r}_{1-i}) \rightarrow \mathsf{F}\,\mathsf{fail})$ We can specify the setting by the following objectives.

- $\varphi_{\alpha_1} = \varphi_{hold} \wedge ((\mathsf{G}\,\neg\mathsf{fail} \wedge \varphi_{allow}) \vee \varphi_{fail})$,
- $\varphi_{\alpha_2} = \mathsf{G}\,\neg\mathsf{fail} \vee \varphi_{failcheck}$,
- $\varphi_{\beta_1} = \mathsf{G}\,\mathsf{F}\,\mathsf{w}_1$,
- $\varphi_{\beta_2} = \mathsf{G}\,\mathsf{F}\,\mathsf{w}_2$.

It is easy to see that the objectives are realizable in the model of coalition-vs-coalition with the solution concept of dominant strategies for the system and a Nash Equilibrium for the environment. Indeed, agent α_2 has a strategy consisting of always setting fail to false, unless one of the two environment agents performs a requests during the withdrawal execution of the other. Furthermore, as far as the variable fail is always false agent α_1 can perform a requested withdrawal for an environment agent. In case fail occurs to be true in some point, then agent α_1 can always perform no withdrawals to get her objective achieved. On the other hand, assuming that the system agents are using these strategies, the environment agents can behave in a way that they never overlap their request, so that their objective is achieved. Note that an environment strategy for agent β_1 cannot be dominant, as player β_2 can deviate in order to make an overlapping request, and the same applies symmetrically

for player β_2. On the other hand, every Nash Equilibrium for the agents β_1 and β_2 makes the objectives φ_{α_1} and φ_{α_2} satisfied. This means that also the strong variant of the problem has a positive answer.

4 Qualitative rational synthesis

In this section we study rational synthesis and strong rational synthesis and show that they can be reduced to the model-checking problem for SL[NG]. Both the rational synthesis problems for several solution concepts can be stated in SL[NG].

We first show how to state that a given strategy profile $\vec{y} = (y_0, \ldots, y_k)$ is in a y_0-fixed γ-equilibrium. For $\alpha_i \in \Omega$, let φ_i be the objective of Agent α_i. For a solution concept γ and a strategy profile $\vec{y} = (y_0, \ldots, y_k)$, the formula $\varphi^{\gamma}(\vec{y})$, expressing that the profile $\vec{y}$ is a y_0-fixed γ-equilibrium, is defined as follows.

- For the solution concept of dominant strategies, we define:
  $\varphi^{\gamma}(\vec{y}) := [\![\vec{z}]\!] \bigwedge_{i=1}^{k} (\flat(\vec{z})\varphi_i \rightarrow \flat(\vec{z}_{-i}, y_i)\varphi_i)$.
- For the solution concept of Nash equilibrium, we define:
  $\varphi^{\gamma}(\vec{y}) := [\![\vec{z}]\!] \bigwedge_{i=1}^{k} (\flat(\vec{y}_{-i}, z_i)\varphi_i \rightarrow \flat(\vec{y})\varphi_i)$.
- For the solution concept of Subgame Perfect Equilibrium, we define:
  $\varphi^{\gamma}(\vec{y}) := [\![\vec{x}]\!]\flat(\vec{x}_{-0}, y_0)\mathsf{G} \bigwedge_{i=1}^{k} [\![z_i]\!]\flat(\vec{y}_{-i}, z_i)\varphi_i \rightarrow \flat(\vec{y})\varphi_i$.
- For the solution concept of h-resilient, we define:
  $\varphi^{\gamma}(\vec{y}) := [\![\vec{x}]\!] \bigwedge_{\mathrm{A}\subseteq\Omega;|\mathrm{A}|\leq h}(\bigwedge_{i\in\mathrm{A}}(\flat(\vec{y}_{-\mathrm{A}}, \vec{x}_{\mathrm{A}})\psi_i) \rightarrow \psi_i)$.
- For the solution concept of t-immune, we define:
  $\varphi^{\gamma}(\vec{y}) := [\![\vec{x}]\!] \bigwedge_{\mathrm{A}\subseteq\Omega;|\mathrm{A}|\leq t}(\bigwedge_{i\in\Omega\setminus\mathrm{A}}(\psi_i) \rightarrow \flat(\vec{y}_{-\mathrm{A}}, \vec{x}_{\mathrm{A}})\psi_i)$.

We can now state the existence of a solution to the non-strong and strong rational synthesis problem, respectively, with input $\varphi_0, \ldots, \varphi_k$ by the closed formulas:

1. $\varphi^{\gamma}_{\subseteq RS} := \langle\!\langle y_0 \rangle\!\rangle \langle\!\langle y_1 \rangle\!\rangle \ldots \langle\!\langle y_k \rangle\!\rangle \flat(\vec{y})(\varphi^{\gamma}(\vec{y}) \wedge \varphi_0)$;
2. $\varphi^{\gamma}_{noncRS} := \langle\!\langle y_0 \rangle\!\rangle [\![y_1]\!] \ldots [\![y_k]\!] \flat(\vec{y})(\varphi^{\gamma}(\vec{y}) \rightarrow \varphi_0)$.

Indeed, the formula 1 specifies the existence of a strategy profile $\mathsf{P} = \langle \pi_0, \ldots, \pi_k \rangle$ that is π_0-fixed γ-equilibrium and such that the outcome satisfies φ_0. On the other hand, the formula 2 specifies the existence of a strategy π_0 for the system such that the outcome of all profiles that are in a π_0-fixed γ-equilibrium satisfy φ_0.

Analogously to the previous case, we can state the existence of a solution to the non-strong and strong coalition vs coalition rational synthesis, with input $\phi_1, \ldots, \phi_n, \psi_1, \ldots, \psi_m$ and two solution concepts γ_{S} and γ_{E}.

1. $\varphi^{\gamma_{\mathrm{S}},\gamma_{\mathrm{E}}}_{tvtc RS} := \langle\!\langle y_1 \rangle\!\rangle \ldots \langle\!\langle y_n \rangle\!\rangle \langle\!\langle x_1 \rangle\!\rangle \ldots \langle\!\langle x_m \rangle\!\rangle \flat(\vec{y}, \vec{x})(\varphi^{\gamma_{\mathrm{E}}}(\vec{y}) \wedge (\varphi^{\gamma_{\mathrm{S}}}(\vec{x}))$;
2. $\varphi^{\gamma_{\mathrm{S}},\gamma_{\mathrm{E}}}_{tvtnoncRS} := \langle\!\langle y_1 \rangle\!\rangle \ldots \langle\!\langle y_n \rangle\!\rangle [\![x_1]\!] \ldots [\![x_m]\!] \flat(\vec{y}, \vec{x})(\varphi^{\gamma_{\mathrm{E}}}(\vec{y}) \rightarrow (\varphi^{\gamma_{\mathrm{S}}}(\vec{x}))$.

As shown above, all the solution concepts we are taking into account can be specified in SL[NG] with formulas whose length is polynomial in the number of the agents and in which the alternation depth of the quantification is 1. Hence we can apply the known complexity results for SL[NG]:

Theorem 2 (Rational synthesis and Strong rational synthesis complexity) *The non-strong and strong rational synthesis and coalition-vs-coalition rational synthesis problems in the qualitative setting are 2*ExpTime*-complete.*

Proof Consider an input $\varphi_0, \ldots, \varphi_k$, X, and γ to the rational synthesis or strong rational synthesis problem. As explained in Section 3, the input induces the game $\mathcal{G}[Syn]$ with nodes in 2^X and the SL[NG] formulas φ^{γ}_{cRS} and $\varphi^{\gamma}_{noncRS}$ for the rational synthesis and strong rational synthesis problems, respectively, in a way that a solution exists if and only if $\mathcal{G}[Syn] \models \varphi^{\gamma}_{cRS}$, for the non-strong case, or $\mathcal{G}[Syn] \models \varphi^{\gamma}_{noncRS}$, for the strong one. Observe that, $\mathcal{G}[Syn]$ is of size exponential w.r.t. the size of the input. For the case of dominant strategies, Nash Equilibria, and Subgame Perfect Nash Equilibria, the upper bound then follows from the fact that the model checking problem for SL[NG] formulas of alternation depth 1 is in EXPTIME in the size of the formula and PTIME in the size of the model (Cf., Theorem 1), thus providing an overall complexity of 2EXPTIME.

For the t-immune (resp., h-resilient) case, observe that the related formula is of length exponential in the size of the input, as it is given by a conjunction ranging over all the possible sets of agents of size at most t (resp., h). Despite this, the procedure is still of 2EXPTIME complexity, as the corresponding automaton used to model-check the formula is built in a way that it universally guesses which of the conjuncts is not satisfied in the structure, which is yet polynomial in the size of the input. Now, since every conjunction of the formula is of polynomial size, we get that the whole formula is checked in $\text{CONPTIME}^{\text{2EXPTIME}}$, which is equivalent to 2EXPTIME.

Moreover, the model-checking algorithm provided in [20] returns finite-state transducers that model strategies that are existentially quantified.

For the lower bound, it is easy to see that the classical LTL synthesis problem is a special case of the non-strong and strong rational synthesis problem. Indeed, $\varphi(I, O)$ is realizable against a hostile environment iff the solution to the strong rational synthesis problem for a system that has an objective φ and controls I and an environment that consists of a single agent that controls O and has an objective *True*, is positive. □

5 Quantitative rational synthesis

As discussed in Section 1, a weakness of classical synthesis algorithms is the fact that specifications are Boolean and miss a reference to the quality of the satisfaction. Applications of game theory consider games with quantitative payoffs. Thus, even more than the classical setting, the rational setting calls for an extension of the synthesis problem to a quantitative setting. In this section we introduce *Objective LTL*, a quantitative extension of LTL, and study an extension of the rational synthesis problem for specifications in Objective LTL. As opposed to other multi-valued specification formalisms used in the context of synthesis of high-quality systems [2, 6], Objective LTL uses the syntax and semantics of LTL and only augments the specification with a reward function that enables a succinct and convenient prioritization of sub-specifications.

5.1 The quantitative setting

Objective LTL (OLTL, for short) is an extension of LTL in which specifications consist of sets of LTL formulas weighted by functions. Formally, an *OLTL specification* over a set X of atomic propositions is a pair $\theta = \langle \Psi, \mathsf{f} \rangle$, where $\Psi = \langle \psi_1, \psi_2, \ldots, \psi_m \rangle$ is a tuple of LTL formulas over X and $\mathsf{f} : \{0, 1\}^m \to \mathbb{Z}$ is a *reward function*, mapping Boolean vectors of length m to integers. We assume that f is given by a polynomial. We use $|\Psi|$ to denote $\sum_{i=1}^{m} |\psi_i|$. For a computation $\eta \in (2^X)^\omega$, the *signature of* Ψ *in* η, denoted $\mathsf{sig}(\Psi, \eta)$, is a vector in $\{0, 1\}^m$ indicating which formulas in Ψ are satisfied in η. Thus, $\mathsf{sig}(\Psi, \eta) =$

$\langle v_1, v_2, \ldots, v_m \rangle$ is such that for all $1 \leq i \leq m$, we have that $v_i = 1$ if $\eta \models \psi_i$ and $v_i = 0$ if $\eta \not\models \psi_i$. The *value of* θ *in* η, denoted $\mathsf{val}(\theta, \eta)$ is then $\mathsf{f}(\mathsf{sig}(\Psi, \eta))$. Thus, the interpretation of an OLTL specification is quantitative. Intuitively, $\mathsf{val}(\theta, \eta)$ indicates the level of satisfaction of the LTL formulas in Ψ in η, as determined by the priorities induced by f. We note that the input of weighted LTL formulas studied in [13] is a special case of Objective LTL.

It is worth to observe that, as the domain of any reward function $\mathsf{f} : \{0, 1\}^m \to \mathbb{Z}$ is a finite set of size 2^m. Then, for a given OLTL specification $\theta = (\Psi, \mathsf{f})$, the set of possible outcomes is of size at most 2^m, thus finite.

Example 4 Consider a system with m buffers, of capacities $\mathsf{c}_1, \ldots, \mathsf{c}_m$. Let full_i, for $1 \leq i \leq m$, indicate that buffer i is full. The OLTL specification $\theta = \langle \Psi, \mathsf{f} \rangle$, with $\Psi = \langle \mathsf{F}\,\mathsf{full}_1, \mathsf{F}\,\mathsf{full}_2, \ldots, \mathsf{F}\,\mathsf{full}_m \rangle$ and $\mathsf{f}(\mathsf{v}) = \mathsf{c}_1 \cdot \mathsf{v}_1 + \cdots + \mathsf{c}_m \cdot \mathsf{v}_m$ enables us to give different satisfaction values to the objective of filling a buffer. Note that $\mathsf{val}(\langle \Psi, \mathsf{f} \rangle, \eta)$ in a computation η is the sum of capacities of all filled buffers.

Example 5 Consider a system with n agents, each of them having their own OLTL objectives $\theta_i = \langle \Psi_i, \mathsf{f}_i \rangle$, with $\Psi_i = \langle \psi_1^i, \ldots, \psi_{m_i}^i \rangle$ and $\mathsf{f}_i : \{0, 1\}^{m_i} \to \mathbb{Z}$. Suppose the system agent α_0 wants to maximize the overall satisfaction. Then, the OLTL objective for the system agent is given by $\langle \Psi_0, \mathsf{f}_0 \rangle$, with $\Psi_0 = \Psi_1, \ldots, \Psi_n$[5] and the function $\mathsf{f}_0 : \{0, 1\}^{m_1 + \ldots + m_n} \to \mathbb{Z}$ is given by $\mathsf{f}_0(\vec{\mathsf{v}}_1, \ldots, \vec{\mathsf{v}}_n) = \sum_{i=1}^{n} \mathsf{f}_i(\vec{\mathsf{v}}_i)$. Suppose, instead, the system wants to ensure that the satisfaction level among the agents is fairly equal. In this case, it needs to minimize the differences between the satisfaction levels. This can be done by means of the formula $\mathsf{f}_0(\vec{\mathsf{v}}_1, \ldots, \vec{\mathsf{v}}_n) = -\sum_{i=1}^{n} \left(\sum_{j=1}^{n} (\mathsf{f}_i(\vec{\mathsf{v}}_i)) - n \cdot \mathsf{f}_j(\vec{\mathsf{v}}_j) \right)^2$.

In the quantitative setting, the objective of Agent α_i is given by means of an OLTL specification $\theta_i = \langle \Psi_i, \mathsf{f}_i \rangle$, specifications describe the payoffs to the agents, and the objective of each agent (including the system) is to maximize his payoff. For a strategy profile P, the payoff for Agent α_i in P is simply $\mathsf{val}(\theta_i, \mathsf{play}(\mathsf{P}))$.

In the quantitative setting, rational synthesis is an optimization problem. Here, in order to solve it, we provide a decision version by making use of a threshold. It is clear that the optimization version can be solved by searching for the best threshold in the decision one. Indeed, since the possible outcomes are of finite number, we need to perform a check for the best threshold a finite number of times.

Definition 3 (Quantitative rational synthesis) The input to the quantitative rational strategy problem is a set X of atomic propositions, partitioned into $\mathrm{X}_0, \ldots, \mathrm{X}_k$, OLTL specifications $\theta_0, \theta_1, \ldots, \theta_k$, with $\theta_i = \langle \Psi_i, \mathsf{f}_i \rangle$, and a solution concept γ. We distinguish between two variants of the problem:

1. In *quantitative rational synthesis*, the desired output for a given threshold $z \in \mathbb{Z}$ is a strategy profile P such that $\mathsf{payoff}_0(\mathsf{P}) \geq z$ and P is in a π_0-fixed γ-equilibrium.
2. In *strong quantitative rational synthesis*, the desired output for a given threshold $z \in \mathbb{Z}$ is a strategy π_0 for the system such that, for each strategy profile P including π_0 and being in a π_0-fixed γ-equilibrium, we have that $\mathsf{payoff}_0(\mathsf{P}) \geq z$.

[5] We are denoting the concatenation product of the Ψ_i's.

Now, we introduce some auxiliary formula that helps us to formulate also the quantitative rational synthesis problem in SL[NG].

For a tuple $\Psi = \langle \psi_1, \ldots, \psi_m \rangle$ of LTL formulas and a signature $v = \{v_1, \ldots, v_m\} \in \{0,1\}^m$, let $\mathsf{mask}(\Psi, v) = (\bigwedge_{i:v_i=0} \neg\psi_i) \wedge (\bigwedge_{i:v_i=1} \psi_i)$ be the LTL formula that characterizes the computations η for which $\mathsf{sig}(\Psi, \eta) = v$.

By means of $\mathsf{mask}(\Psi, v)$, we can adjust the SL formulas $\Phi^{\gamma}(\vec{y})$ described in Section 4 to the quantitative setting. Recall that $\Phi^{\gamma}(\vec{y})$ holds iff the strategy profile assigned to $\vec{y}$ is in a π_0-fixed γ-equilibrium. There, the formula is a conjunction over all agents in $\{\alpha[1], \ldots, \alpha_k\}$, stating that Agent α_i does not have an incentive to change his strategy. In our quantitative setting, this means that the payoff of Agent α_i in an alternative profile is not bigger than his payoff in $\vec{y}$. For two strategy profiles, assigned to $\vec{y}$ and $\vec{y}'$, an SL formula that states that Agent α_i has no incentive that the profile would change from $\vec{y}$ to $\vec{y}'$ can state that the signature of Ψ in $\mathsf{play}(\vec{y}')$ results in a payoff to Agent α_i that is smaller than his current payoff. Formally, we have that:

$\Phi_i^{eq}(\vec{y}, \vec{y}') = \bigvee_{v \in \{0,1\}^{m_i} : \mathsf{f}_i(v) \leq \mathsf{payoff}_i(\vec{y})} \flat(\vec{y}')\mathsf{mask}(\Psi_i, v)$.

We can now adjust $\Phi^{\gamma}(\vec{y})$ for all the cases of solution concepts we are taking into account.

- For the solution concept of dominant strategies, we define:
  $\Phi^{\gamma}(\vec{y}) := \bigwedge_{i \in \{1,\ldots,n\}} [\![\vec{z}]\!] \Phi_i^{eq}(\vec{y}, (\vec{z}_{-0}, y_0))$;
- For the solution concept of Nash equilibrium, we define:
  $\Phi^{\gamma}(\vec{y}) := \bigwedge_{i \in \{1,\ldots,n\}} [\![\vec{z}]\!] \Phi_i^{eq}(\vec{y}, (\vec{y}_{-i}, z_i))$;
- For the solution concept of Subgame Perfect Equilibrium, we define: $\Phi^{\gamma}(\vec{y}) := [\![\vec{x}]\!] \flat(\vec{x}_{-0}, y_0) \mathsf{F} \bigwedge_{i \in \{1,\ldots,n\}} [\![\vec{z}]\!] \Phi_i^{eq}(\vec{y}, (\vec{z}_{-0}, y_0))$;
- For the solution concept of k-resilient, we define:
  $\Phi^{\gamma}(\vec{y}) := [\![\vec{x}]\!] \bigwedge_{\mathrm{A} \subseteq \Omega; |\mathrm{A}| \leq k} (\bigwedge_{i \in \mathrm{A}} \Phi_i^{eq}(\vec{y}, (\vec{y}_{-\mathrm{A}}, z_{\mathrm{A}})))$;
- For the solution concept of t-immune, we define:
  $\Phi^{\gamma}(\vec{y}) := [\![\vec{x}]\!] \bigwedge_{\mathrm{A} \subseteq \Omega; |\mathrm{A}| \leq t} (\bigwedge_{i \in \Omega \setminus \mathrm{A}} \Phi_i^{eq}(\vec{y}, (\vec{y}_{-(\Omega \setminus \mathrm{A})}, z_{\Omega \setminus \mathrm{A}})))$.

Once we adjust $\Phi^{\gamma}(\vec{y})$ to the quantitative setting, we can use the same SL formula used in the non-quantitative setting to state the existence of a solution to the rational synthesis problem.

Theorem 3 *The non-strong and strong quantitative rational synthesis problems are* 2EXPTIME-COMPLETE.

Proof We can reduce the problems to the model-checking problem of the SL formulas Φ_{RS}^{γ} and Φ_{nonRS}^{γ}, respectively. We should, however, take care when analyzing the complexity of the procedure, as the formulas $\Phi_i^{eq}(\vec{y}, \vec{y}')$, which participate in Φ_{RS}^{γ} and Φ_{nonRS}^{γ} involve a disjunction over vectors in $\{0,1\}^{m_i}$, resulting in Φ_{nonRS}^{γ} of an exponential length.

While the above prevents us from using the doubly exponential known bound on SL[NG] model checking for formulas of alternation depth 1 as is, it is not difficult to observe that the run time of the model-checking algorithm in [20], when applied to Φ_{nonRS}^{γ}, is only doubly exponential.

Indeed, the extra exponential blow-up in the construction of the automaton $\mathcal{A}_{\varphi}^{\mathcal{G}}$ in Lemma 5.6 [20] is given by an exponential number of applications of the disjunction rule. From that point the automaton requires a doubly exponential check for each disjunction, due to the innermost construction of the automaton for LTL formulas inside. Hence, the overall complexity is given by an exponential execution of a doubly exponential procedure, which is still doubly exponential, and it can return the witnessing strategies.

Hardness in EXPTIME follows easily from hardness in the non-quantitative setting. □

6 Discussion

The understanding that synthesis corresponds to a game in which the objective of each player is to satisfy his specification calls for a mutual adoption of ideas between formal methods and game theory. In *rational synthesis*, introduced in [14], synthesis is defined in a way that takes into account the rationality of the agents that constitute the environment and involves an assumption that an agent cooperates with a strategy to a strategy profile in which his objective is satisfied. Here and in [17], we extend the idea and consider also *strong* rational synthesis, in which agents need not cooperate with suggested strategies and may prefer different strategies that are at least as beneficial for them.

Many variants of the classical synthesis problem have been studied [7, 8]. It is interesting to examine the combination of the rational setting with the different variants. To begin with, the non-strong and strong settings can be combined into a framework in which one coalition of agents is competing with another coalition of agents, where each coalition is internally cooperative, but the two coalitions are non-cooperative. Furthermore, we plan to study rational synthesis with incomplete information. In particular, we plan to study rational synthesis with *incomplete information* [18], where agents can view only a subset of the signals that other agents output, and rational *stochastic* synthesis [11], which models the unpredictability of nature and involves stochastic agents that assign values to their output signals by means of a distribution function. Beyond a formulation of the richer settings, one needs a corresponding extension of strategy logic and its decision problems.

As discussed in Section 1, classical applications of game theory consider games with quantitative payoffs. We added a quantitative layer to LTL by introducing Objective- LTL and studying its rational synthesis. In recent years, researchers have developed more refined quantitative temporal logics, which enable a formal reasoning about the quality of systems. In particular, we plan to study rational synthesis for the multi-valued logics LTL[F] [2], which enables a prioritization of different satisfaction possibilities, and LTL[D] [3], in which discounting is used in order to reduce the satisfaction value of specifications whose eventualities are delayed. The rational synthesis problem for these logics induce a game with much richer profiles, making the search for a stable solution much more challenging.

References

1. Abraham, I., Dolev, D., Gonen, R., Halpern, J.Y.: Distributed Computing Meets Game Theory: Robust Mechanisms for Rational Secret Sharing and Multiparty Computation. In: Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing, PODC 2006, Denver, CO, USA, July 23-26, 2006, pp. 53–62 (2006). doi:10.1145/1146381.1146393
2. Almagor, S., Boker, U., Kupferman, O.: Formalizing and reasoning about quality. In: ICALP'13, LNCS, vol. 7966, pp. 15–27 (2013)
3. Almagor, S., Boker, U., Kupferman, O.: Discounting in LTL. In: TACAS'14, LNCS, vol. 8413, pp. 424–439. Springer (2014)
4. Alur, R., Henzinger, T., Kupferman, O.: Alternating-Time Temporal logic. JACM **49**(5), 672–713 (2002)

5. Aumann, R.J.: Acceptable points in general cooperative n-person games. Contributions to the Theory of Games **4**, 287–324 (1959)
6. Bloem, R., Chatterjee, K., Henzinger, T., Jobstmann, B.: Better quality in synthesis through quantitative objectives. In: CAV'09, LNCS, vol. 5643, pp. 140–156. Springer (2009)
7. Bouyer, P., Brenguier, R., Markey, N., Ummels, M.: Pure Nash equilibria in concurrent games. In: Logical Methods in Computer Science (2015). To appear
8. Brenguier, R., Raskin, J.F., Sankur, O.: Assume-admissible synthesis. In: CONCUR '15, pp. 100–113 (2015)
9. Chatterjee, K., Henzinger, T., Jobstmann, B.: Environment Assumptions for Synthesis. In: CONCUR'08, LNCS, vol. 5201, pp. 147–161. Springer (2008)
10. Chatterjee, K., Henzinger, T., Piterman, N.: Strategy Logic. In: CONCUR'07, LNCS 4703, pp. 59–73. Springer (2007)
11. Chatterjee, K., Majumdar, R., Jurdzinski, M.: On Nash Equilibria in Stochastic Games. In: CSL, LNCS, vol. 3210, pp. 26–40. Springer (2004)
12. Church, A.: Logic, arithmetics, and automata. In: Proceedings of the International Congress of Mathematicians, pp. 23–35. Institut Mittag-Leffler (1963)
13. Courcoubetis, C., Yannakakis, M.: Markov decision processes and regular events. IEEE Trans. Autom. Control **43**(10), 1399–1418 (1998)
14. Fisman, D., Kupferman, O., Lustig, Y.: Rational Synthesis. In: TACAS'10, LNCS 6015, pp. 190–204. Springer (2010)
15. Halpern, J.Y.: Beyond Nash equilibrium: solution concepts for the 21st Century. In: Gamesec, pp. 1–3 (2011)
16. Henzinger, T.: From boolean to quantitative notions of correctness. In: POPL'10, pp. 157–158. ACM (2010)
17. Kupferman, O., Perelli, G., Vardi, M.Y.: Synthesis with rational environments. In: EUMAS'14, pp. 219–235 (2014). doi:10.1007/978-3-319-17130-2_15
18. Kupferman, O., Vardi, M.Y.: Church's problem revisited. Bull. Symb. Log. **5**(2), 245–263 (1999)
19. Manna, Z., Pnueli, A.: The Temporal Logic of Reactive and Concurrent Systems - Specification. Springer (1992)
20. Mogavero, F., Murano, A., Perelli, G., Vardi, M.: Reasoning about strategies: on the Model-Checking problem. TOCL **15**(4) (2014). doi:10.1145/2631917
21. Mogavero, F., Murano, A., Perelli, G., Vardi, M.Y.: What Makes ATL* Decidable? a Decidable Fragment of Strategy Logic. In: CONCUR'12, LNCS, vol. 7454, pp. 193–208 (2012)
22. Mogavero, F., Murano, A., Vardi, M.: Reasoning about Strategies. In: FSTTCS'10, LIPIcs 8, pp. 133–144 (2010)
23. Nisan, N., Ronen, A.: Algorithmic mechanism design. Games and Economic Behavior **35**(1-2), 166–196 (2001)
24. Nisan, N., Roughgarden, T., Tardos, E., Vazirani, V.: Algorithmic Game Theory. Cambridge University Press (2007)
25. Osborne, M., Rubinstein, A.: A Course in Game Theory. MIT Press (1994)
26. Pnueli, A., Rosner, R.: On the Synthesis of a Reactive Module. In: POPL'89, pp. 179–190 (1989)
27. Selten, R.: Reexamination of the perfectness concept for equilibrium points in extensive games. Int. J. Game Theory **4**(1), 25–55 (1975)