# Perspective Multi-Player Games

Orna Kupferman and Noam Shenwald
School of Computer Science and Engineering
The Hebrew University, Israel

**Abstract**

Perspective games model multi-agent systems in which agents can view only the parts of the system that they own. Unlike the observation-based model of partial visibility, where uncertainty is longitudinal – agents partially observe the full history, uncertainty in perspective games is transverse – agents fully observe parts of the history. So far, researchers studied zero-sum two-player perspective games. There, the objective of one agent (the system) is to satisfy a given specification, and the objective of the second agent (the environment) is to fail the specification.

We study richer and more realistic settings of perspective games. We consider games with *more than two players*, and distinguish between zero-sum games, where the objectives of the players form a partition of all possible behaviors, zero-sum games among *coalitions*, where agents in a coalition share their objectives but do not share their visibility, and *non-zero-sum games*, where each agent has her own objectives and is assumed to be rational rather than hostile. In the non-zero-sum setting, we are interested in stable outcomes of the game; in particular, Nash equilibria.

We show that, as is the case with longitudinal uncertainty, transverse uncertainty leads to undecidability in settings with three or more players that include coalitions or non-zero-sum objectives. We then focus on two-player non-zero-sum perspective games. There, finding and reasoning about stable outcomes is decidable, and in fact, unlike the case with longitudinal uncertainty, can be done in the same complexity as in games with full visibility. In particular, we study *rational synthesis* in the perspective setting, where the goal is to generate systems that satisfy their specification when interacting with rational environments. Our study includes Boolean objectives given by automata or LTL formulas, as well as a *multi-valued setting*, where the objectives are LTL$[\mathcal{F}]$ formulas with satisfaction values in $[0, 1]$, and the agents aim to maximize the satisfaction value of their objectives.

## 1 Introduction

Design and control of *multi-agent systems* correspond to the synthesis of winning strategies in a game that models the interaction between the agents. Different settings induce different classes of games. In all classes, the game is played on a graph and the players generate a *play*, namely a path in the graph, by jointly moving a token along the graph. Classes then differ in the way the players proceed, the type of objectives, and the relation among the player's objectives.

We focus on settings in which each of the players has control in different parts of the system. Thus, the game is *turn-based*, with each player owning a subset of the vertices and deciding the successor vertex when the play reaches a vertex she owns. A *strategy* for a player directs her how to continue a play that reaches her vertices. Each vertex in the game graph is labeled by an assignment to a set $AP$ of atomic propositions, and the objectives of the players are *behavioral* – each objective is a language of infinite words in $(2^{AP})^{\omega}$. In games with *full visibility*, strategies may depend on the full history of the play. In games with *partial visibility*, strategies depend only on visible components of the history. We distinguish between two approaches to partial visibility. The first is the traditional *longitudinal uncertainty* (a.k.a. *observation-based* games), where in all vertices, the players observe the assignment only to an observable subset of the atomic propositions [5, 13, 14, 26, 38]. The second is the recently introduced *transverse uncertainty* (a.k.a. *perspective games*), where players observe the assignment to all the atomic propositions, but only in the vertices they own [25, 23].

Both types of partial visibility correspond to realistic settings. In games with longitudinal uncertainty, we model systems in which each of the underlying components can only view and control a subset of the system's variables. For example, a program that interacts with a user with private variables. In games with transverse uncertainty, we model systems in which control is switched among the underlying components, which can observe only these parts of the interaction that they control. For example, a communication network in which a company that owns part of the routers has to make routing decisions based only on information about visits to its routers [1], a component in a reactive system that is not aware of the activity in other components, for example in software and web services systems [29], and switched systems where components are activated by a scheduler and are not aware of the evolution of the system while being switched off [20, 28, 30, 37].

So far, transverse uncertainty was studied only in *two-player zero-sum* perspective games [25]. There, the objective of one player (the system) is to satisfy a given specification, whereas the second player (the environment) is assumed to be hostile and her objective is to fail the specification. The conceptual difference between longitudinal and transverse uncertainty is reflected in the properties of the game and the algorithms developed for deciding them. In particular, while the complexity of deciding whether the system wins an observation-based game is exponential in the size of the game [5, 38], it is only polynomial in perspective games. Intuitively, this follows from the fact that when a player moves the token in an observation-based game, her partial visibility leads to uncertainty about the location of the token, which requires algorithms to maintain subsets of vertices. On the other hand, in a perspective game, players know the vertex where the token is, and uncertainty is about the path the token has traversed so far.

In this work we study transverse uncertainty in richer and more realistic settings. We first stay in the zero-sum terrain, yet consider perspective games with *more than two players*. Unsurprisingly, if the objectives of the players form a partition of $(2^{AP})^\omega$, then reasoning can be reduced to the setting of two-players zero-sum perspective games. When, however, we allow the players to form *coalitions* [12], where players in a coalition share their objectives but do not share their visibility, then transverse uncertainty can capture the "information fork" that causes the longitudinal setting to be undecidable [36]. Specifically, we show that perspective games with two coalitions can model the undecidable setting of three-player observation-based games, in fact even concurrent ones [17]. Essentially, by carefully ordering the turns in which the players in our perspective game proceed, we can mimic both the concurrency and the observation-based behavior of the players in [17].

In the traditional approach to synthesis, the system has to satisfy its specification in all environments. Thus, the components that compose the environment 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 [35]. In real life, however, the components that compose the environment are often entities that have objectives of their own. The approach taken in the field of algorithmic game theory [33] is to assume that agents interacting with a computational system are *rational*, and thus act to achieve their own objectives. Assuming agents' rationality softens the universal quantification on the environment, and motivates the study of *non-zero-sum games*, which model settings where the players are rational rather than hostile.

Technically, in non-zero-sum games, the objective of each player is still a language of infinite words in $(2^{AP})^\omega$, but now these languages may overlap, and need not cover all behaviors in $(2^{AP})^\omega$. A *profile* in the game is a vector of strategies, one for each player. The interesting questions for non-zero-sum games concern *stable* profiles. In particular, a profile is a *Nash equilibrium* (NE, for short) if no player has an incentive to deviate from her assigned strategy, provided that the other players adhere to the strategies assigned to them [32]. As it turns out, an NE need not exist, even in non-zero-sum games with no uncertainty. There, deciding the existence of an NE can be solved in time polynomial in the size of the graph, for a wide range of structural and behavioral objectives. Increasing the number of players increases the complexity, but it is still decidable [8, 9, 10]. In games with no uncertainty, researchers have studied also settings with *coalitions* [6, 12], *probability* [7, 15], as well as the problem of *repairing* unstable games [2]. Finally, back to synthesis, researchers have studied the problem of *rational synthesis*, where we seek a strategy for the system player that would guarantee the satisfaction of her specification in all rational environments [21, 24]. In settings with no uncertainty, the complexity of rational synthesis is again polynomial in the graph [16].

Non-zero-sum games with longitudinal uncertainty are studied in [22, 19]. In [22], the authors study the NE-existence decision problem, showed that it is undecidable for three or more players, and decidable for two players. For objectives in LTL, the complexity is again 2EXPTIME-complete. The game graph in [22] is implicit, and induced by the objectives of the players. In [19], the authors study rational synthesis in games with longitudinal uncertainty, and the complexities are similar, with an exponential dependency in the graph.

Here, we study *non-zero-sum perspective games*. We show that while undecidability for three or more players in the longitudinal setting is carried over to the transverse one, for games with two players we are able to show decidability, and the problem is only polynomial in the game graph. Our algorithm is based on dividing the general question of NE-existence into four questions, namely existence of NEs that are characterized by the set of players that win. For example, a *WL-NE* is an NE in which only Player 1 wins, and we show that we can reduce the question of existence of a WL-NE to questions about zero-sum perspective games. In more details, by manipulating tree automata used for solving zero-sum perspective games with related objectives, we construct a tree automaton that accepts strategies for Player 1 if they participate in a WL-NE. Similar techniques are used for the other types of NEs, and together enable us to solve the NE-existence problem in a complexity that is only polynomial in the graph game.

As for the complexity in terms of the objectives, we distinguish between objectives given by different types of automata as well as by LTL formulas, and give a complete picture of all formalisms. The bottom line of our results is that unless the specification formalism is a deterministic automaton, the complexities of the problems we study coincide with their complexity in a setting with no uncertainty. Intuitively, this follows from the fact that the transverse setting adds uncertainty only about the state of an automaton that follows the play traversed by the token, and such uncertainty anyway exists in nondeterministic or alternating automata. The details, however, require careful acrobatics that synchronize uncertainty due to partial visibility with uncertainty due to branches in the automaton.

We continue and study rational synthesis for settings with transverse uncertainty. Recall that the basic challenge is to find a strategy for the system player with which her objective is guaranteed to be satisfied, assuming rationality of the other players. The above can be formalized in two different ways [21, 24]. The first is *cooperative rational synthesis* (CRS), where the desired output is an NE in which the objective of the system is satisfied. Thus, the assumption is that we can suggest strategies to the other players, and they would follow our suggestion unless they have a beneficial deviation. This assumption is removed in *non-cooperative rational synthesis* (NRS), where the desired output is a strategy for the system player such that her objective is satisfied in every NE where she follows this strategy, and at least one such NE exists.

While automata and LTL formulas enable the description of rich on-going behaviors, their semantics is Boolean: a play may satisfy an objective or it may not. As argued in [3], 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 [24], 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. We extend rational synthesis with transverse uncertainty to a *multi-valued setting*, where the objectives are LTL[$\mathcal{F}$] formulas – an extension of LTL by quality operators. The satisfaction value of an 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 [3]. The goal of the players is then to maximize the satisfaction value of their objectives. In particular, a profile is an NE if no player can deviate to a strategy with which the satisfaction value of her objective is increased. Rational synthesis for games with LTL[$\mathcal{F}$] objectives and no uncertainty was studied in [4]. Here, we study rational synthesis in settings with transverse uncertainty, for both Boolean and multi-valued objectives. We show that while the two variants are undecidable for three or more players, the case of two players can be solved in the same complexity as the NE-existence problem. Thus, it is not more difficult than rational synthesis in settings with no uncertainty. In fact (see Remark 7.2), our results improve the known upper bound also for rational synthesis in settings with no uncertainty.

# 2 Preliminaries

## 2.1 Perspective Non-Zero-Sum Games

For $k \geq 1$, let $[k] = \{1, \ldots, k\}$. A *k-player game graph* is a tuple $G = \langle AP, \{V_i\}_{i \in [k]}, v_0, E, \tau \rangle$, where $AP$ is a finite set of atomic propositions, $\{V_i\}_{i \in [k]}$ are disjoint sets of vertices, each owned by a different player, and we let $V = \bigcup_{i \in [k]} V_i$. Then, $v_0 \in V_1$ is an initial vertex, which we assume to be owned by Player 1, and $E \subseteq V \times V$ is a total edge relation, thus for every $v \in V$ there is $u \in V$ such that $\langle v, u \rangle \in E$. The function $\tau : V \to 2^{AP}$ maps each vertex to a set of atomic propositions that hold in it. The size $|G|$ of $G$ is $|E|$, namely the number of edges in it.

In a beginning of a play in the game, a token is placed on $v_0$. Then, in each turn, the player that owns the vertex that hosts the token chooses a successor vertex and moves the token to it. Together, the players generate a *play* $\rho = v_0, v_1, \ldots$ in $G$, namely an infinite path that starts in $v_0$ and respects $E$: for all $i \geq 0$, we have that $\langle v_i, v_{i+1} \rangle \in E$. The play $\rho$ induces a *computation* $\tau(\rho) = \tau(v_0), \tau(v_1), ... \in (2^{AP})^{\omega}$.

A *game* is a tuple $\mathcal{G} = \langle G, \{L_i\}_{i \in [k]} \rangle$, where $G$ is a $k$-player game graph, and for every $i \in [k]$, the language $L_i \subseteq (2^{AP})^{\omega}$ is a *behavioral objective*, namely an $\omega$-regular language over the atomic propositions, given by an LTL formula or an automaton. Intuitively, for every $i \in [k]$, Player $i$ aims for a play whose computation is in $L_i$. For a language $L \subseteq (2^{AP})^{\omega}$, let $\overline{L}$ denote the complement of $L$, thus $\overline{L} = (2^{AP})^{\omega} \setminus L$.

Let $\mathsf{Prefs}(G)$ be the set of nonempty prefixes of plays in $G$. For a sequence $\rho = v_0, \ldots, v_n$ of vertices, let $\mathsf{Last}(\rho) = v_n$. For $i \in [k]$, let $\mathsf{Prefs}_i(G) = \{\rho \in$

$\mathsf{Prefs}(G) : \mathsf{Last}(\rho) \in V_i\}$. In games with full visibility, the players have a full view of the generated play. Accordingly, a strategy for Player $i$ maps $\mathsf{Prefs}_i(G)$ to vertices in $V$ in a way that respects $E$. In *perspective games*, Player $i$ can view only visits to $V_i$. Accordingly, strategies are defined as follows. For a prefix $\rho = v_0, \ldots, v_j \in \mathsf{Prefs}(G)$, and $i \in [k]$, *the perspective of player $i$ on $\rho$*, denoted $\mathsf{Persp}_i(\rho)$, is the restriction of $\rho$ to vertices in $V_i$. We denote the perspectives of Player $i$ on prefixes in $\mathsf{Prefs}_i(G)$ by $\mathsf{PPrefs}_i(G)$, namely $\mathsf{PPrefs}_i(G) = \{\mathsf{Persp}_i(\rho) : \rho \in \mathsf{Prefs}_i(G)\}$. Note that $\mathsf{PPrefs}_i(G) \subseteq {V_i}^*$. A *perspective strategy* for Player $i$ (*P-strategy*, for short), is then a function $f_i : \mathsf{PPrefs}_i(G) \to V$ such that for all $\rho \in \mathsf{PPrefs}_i(G)$, we have that $\langle \mathsf{Last}(\rho), f_i(\rho)\rangle \in E$. That is, a P-strategy for Player $i$ maps her perspective of prefixes of plays that end in a vertex $v \in V_i$ to a successor of $v$.

A *profile* is a tuple $\pi = \langle f_1, ..., f_k\rangle$ of P-strategies, one for each player. The *outcome* of a profile $\pi = \langle f_1, ..., f_k\rangle$ is the play obtained when the players follow their P-strategies. Formally, $\mathsf{Outcome}(\pi) = v_0, v_1, ...$ is such that for all $j \geq 0$ and $i \in [k]$, if $v_j \in V_i$, then $v_{j+1} = f_i(\mathsf{Persp}_i(v_0, \ldots, v_j))$.

The set of *winners* in $\pi$, denoted by $\mathsf{Win}(\pi) \subseteq [k]$, is the set of players whose objective is satisfied in $\mathsf{Outcome}(\pi)$. The set of *losers* in $\pi$, denoted $\mathsf{Lose}(\pi)$, is then $[k] \setminus \mathsf{Win}(\pi)$, namely the set of players whose objective is not satisfied in $\mathsf{Outcome}(\pi)$.

The game $\mathcal{G} = \langle G, \{L_i\}_{i\in[k]}\rangle$ is *zero sum* if the objectives of the players form a partition of $(2^{AP})^\omega$. Thus, for $i \neq j$, we have that $L_i \cap L_j = \emptyset$, and $(2^{AP})^\omega = \bigcup_{i\in[k]} L_i$. Then, for every profile $\pi$, we have that $|\mathsf{Win}(\pi)| = 1$ and $|\mathsf{Lose}(\pi)| = k - 1$. For $i \in [k]$, we say that a P-strategy $f_i$ for Player $i$ is a *winning* strategy if for every profile $\pi$ of strategies in which Player $i$ follows $f_i$, we have that $\mathsf{Win}(\pi) = \{i\}$.

When $\mathcal{G}$ is a non-zero-sum game, the objectives of the players may overlap, and we are interested in *stable* profiles in the game. In particular, a profile $\pi = \langle f_1, ..., f_k\rangle$ is a *Nash Equilibrium* (NE, for short) [32] if, intuitively, no (single) player can benefit from unilaterally changing her strategy. In our setting, "benefiting" amounts to moving from the set of losers to the set of winners. Formally, for $i \in [k]$ and some perspective strategy $f'_i$ for Player $i$, let $\pi[i \leftarrow f'_i] = \langle f_1, ..., f_{i-1}, f'_i, f_{i+1}, ..., f_k\rangle$ be the profile in which Player $i$ *deviates* to the strategy $f'_i$. We say that $\pi$ is an NE if for every $i \in [k]$, if $i \in \mathsf{Lose}(\pi)$, then for every perspective strategy $f'_i$ we have that $i \in \mathsf{Lose}(\pi[i \leftarrow f'_i])$.

## 2.2 Linear Temporal Logic

The logic LTL is used for specifying on-going behaviors of reactive systems [34]. Formulas of LTL are constructed from a set $AP$ of atomic propositions using the usual Boolean operators and the temporal operators $X$ ("next time") and $U$ ("until"). Formally, an LTL formula over $AP$ is defined as follows:

- `True`, `False`, or $p$, for $p \in AP$.
- $\neg\psi_1$, $\psi_1 \wedge \psi_2$, $X\psi_1$, or $\psi_1 U \psi_2$, where $\psi_1$ and $\psi_2$ are LTL formulas.

The semantics of LTL is defined with respect to infinite computations in $(2^{AP})^\omega$. Consider a computation $\rho = \sigma_1, \sigma_2, \sigma_3, \ldots$. We denote the suffix $\sigma_j, \sigma_{j+1}, \ldots$ of $\rho$ by $\rho^j$. We use $\rho \models \psi$ to indicate that an LTL formula $\psi$ holds in the computation $\rho$. The relation $\models$ is inductively defined as follows:

- For all $\rho$, we have that $\rho \models \texttt{True}$ and $\rho \not\models \texttt{False}$.
- For an atomic proposition $p \in AP$, we have that $\rho \models p$ iff $p \in \sigma_1$.
- $\rho \models \neg\psi_1$ iff $\rho \not\models \psi_1$.
- $\rho \models \psi_1 \wedge \psi_2$ iff $\rho \models \psi_1$ and $\rho \models \psi_2$.
- $\rho \models X\psi_1$ iff $\rho^2 \models \psi_1$.
- $\rho \models \psi_1 U \psi_2$ iff there exists $k \geq 1$ such that $\rho^k \models \psi_2$ and $\rho^i \models \psi_1$ for all $1 \leq i < k$.

Writing LTL formulas, it is convenient to use the abbreviations $G$ ("always"), $F$ ("eventually"), and $R$ ("release"). Formally, the abbreviations follow the following semantics.

- $F\psi_1 = \mathbf{true} U \psi_1$. That is, $\rho \models F\psi_1$ iff there exists $k \geq 1$ such that $\rho^k \models \psi_1$.
- $G\psi_1 = \neg F \neg \psi_1$. That is, $\rho \models G\psi_1$ iff for all $k \geq 1$ we have that $\rho^k \models \psi_1$.

## 2.3 Trees and Automata

Given a set $D$ of directions, a *$D$-tree* is a set $T \subseteq D^*$ such that if $x \cdot c \in T$, where $x \in D^*$ and $c \in D$, then also $x \in T$. The elements of $T$ are called *nodes*, and the empty word $\varepsilon$ is the *root* of $T$. For every $x \in T$, the nodes $x \cdot c$, for $c \in D$, are the *successors* of $x$. A *path* $\pi$ of a tree $T$ is a set $\pi \subseteq T$ such that $\varepsilon \in \pi$ and for every $x \in \pi$, either $x$ is a leaf or there exists a unique $c \in D$ such that $x \cdot c \in \pi$. Given an alphabet $\Sigma$, a *$\Sigma$-labeled $D$-tree* is a pair $\langle T, \tau \rangle$ where $T$ is a tree and $\tau : T \to \Sigma$ maps each node of $T$ to a letter in $\Sigma$.

Our algorithms use *automata on infinite words and trees*, defined below. For a set $X$, let $\mathcal{B}^+(X)$ be the set of positive Boolean formulas over $X$ (i.e., Boolean formulas built from elements in $X$ using $\wedge$ and $\vee$), where we also allow the formulas **true** and **false**. For a set $Y \subseteq X$ and a formula $\theta \in \mathcal{B}^+(X)$, we say that $Y$ *satisfies* $\theta$ iff assigning **true** to elements in $Y$ and assigning **false** to elements in $X \setminus Y$ makes $\theta$ true. An *alternating tree automaton* is $\mathcal{A} = \langle \Sigma, D, Q, q_{in}, \delta, \alpha \rangle$, where $\Sigma$ is the input alphabet, $D$ is a set of directions, $Q$ is a finite set of states, $\delta : Q \times \Sigma \to \mathcal{B}^+(D \times Q)$ is a transition function, $q_{in} \in Q$ is an initial state, and $\alpha \subseteq Q$ specifies a Büchi or a co-Büchi acceptance condition. For a state $q \in Q$, we use $\mathcal{A}^q$ to denote the automaton obtained from $\mathcal{A}$ by setting the initial state to be $q$. The *size* of $\mathcal{A}$, denoted $|\mathcal{A}|$, is the sum of lengths of formulas that appear in $\delta$.

The alternating automaton $\mathcal{A}$ runs on $\Sigma$-labeled $D$-trees. A *run* of $\mathcal{A}$ over a $\Sigma$-labeled $D$-tree $\langle T, \tau \rangle$ is a $(T \times Q)$-labeled $\mathbb{N}$-tree $\langle T_r, r \rangle$. Each node of $T_r$ corresponds to a node of $T$. A node in $T_r$, labeled by $(x, q)$, describes a copy of the automaton that reads the node $x$ of $T$ and visits the state $q$. Note that many nodes of $T_r$ can correspond to the same node of $T$. The labels of a node and its successors have to satisfy the transition function. Formally, $\langle T_r, r \rangle$ satisfies the following:

1. (1) $\varepsilon \in T_r$ and $r(\varepsilon) = \langle \varepsilon, q_{in} \rangle$.
2. (2) Let $y \in T_r$ with $r(y) = \langle x, q \rangle$ and $\delta(q, \tau(x)) = \theta$. Then there is a (possibly empty) set $S = \{(c_0, q_0), (c_1, q_1), \ldots, (c_{n-1}, q_{n-1})\} \subseteq D \times Q$, such that $S$ satisfies $\theta$, and for all $0 \leq i \leq n-1$, we have $y \cdot i \in T_r$ and $r(y \cdot i) = \langle x \cdot c_i, q_i \rangle$.

For example, if $\langle T, \tau \rangle$ is a $\{0,1\}$-tree with $\tau(\varepsilon) = a$ and $\delta(q_{in}, a) = ((0, q_1) \vee (0, q_2)) \wedge ((0, q_3) \vee (1, q_2))$, then, at level 1, the run $\langle T_r, r \rangle$ includes a node labeled $(0, q_1)$ or a node labeled $(0, q_2)$, and includes a node labeled $(0, q_3)$ or a node labeled $(1, q_2)$. Note that if, for some $y$, the transition function $\delta$ has the value **true**, then $y$ need not have successors. Also, $\delta$ can never have the value **false** in a run.

A run $\langle T_r, r \rangle$ is accepting if all its infinite paths satisfy the acceptance condition. Given a run $\langle T_r, r \rangle$ and an infinite path $\pi \subseteq T_r$, let $inf(\pi) \subseteq Q$ be such that $q \in inf(\pi)$ if and only if there are infinitely many $y \in \pi$ for which $r(y) \in T \times \{q\}$. That is, $inf(\pi)$ contains exactly all the states that appear infinitely often in $\pi$. A path $\pi$ satisfies a *Büchi* acceptance condition $\alpha$ iff $inf(\pi) \cap \alpha \neq \emptyset$, and satisfies a *co-Büchi* acceptance condition $\alpha$ iff $inf(\pi) \cap \alpha = \emptyset$. We also consider the *parity* acceptance condition, where $\alpha : Q \to \{0, 1, \ldots, k\}$ maps each state to a color in $\{0, 1, \ldots, k\}$, and a path $\pi$ satisfies $\alpha$ if the minimal color visited infinitely often is even, thus $\min\{i : inf(\pi) \cap \alpha^{-1}(i) \neq \emptyset\}$ is even. An automaton accepts a tree iff there exists a run that accepts it. We denote by $L(\mathcal{A})$ the set of all $\Sigma$-labeled trees that $\mathcal{A}$ accepts.

The alternating automaton $\mathcal{A}$ is *nondeterministic* if for all the formulas that appear in $\delta$, if $(c_1, q_1)$ and $(c_2, q_2)$ are conjunctively related, then $c_1 \neq c_2$. (i.e., if the transition is rewritten in disjunctive normal form, there is at most one element of $\{c\} \times Q$, for each $c \in D$, in each disjunct). The automaton $\mathcal{A}$ is *universal* if all the formulas that appear in $\delta$ are conjunctions of atoms in $D \times Q$, and $\mathcal{A}$ is *deterministic* if it is both nondeterministic and universal. The automaton $\mathcal{A}$ is a *word* automaton if $|D| = 1$. Then, we can omit $D$ from the specification of the automaton and denote the transition function of $\mathcal{A}$ as $\delta : Q \times \Sigma \to \mathcal{B}^+(Q)$. If the word automaton is nondeterministic or universal, then $\delta : Q \times \Sigma \to 2^Q$.

We denote different types of automata by three-letter acronyms in $\{D, N, U, A\} \times \{F, B, C, P\} \times \{W, T\}$, where the first letter describes the branching mode of the automaton (deterministic, nondeterministic, universal, or alternating), the second letter describes the acceptance condition (finite, Büchi, co-Büchi, or parity), and the third letter describes the object over which the automaton runs (words or trees). For example, UCT stands for a universal co-Büchi tree automaton.

### 2.4 Transducers

A $\Sigma$-labeled $D$-tree $\langle T, \tau \rangle$ is *regular* if for every letter $\sigma \in \Sigma$, we have that $\tau^{-1}(\sigma)$, namely the language of words in $D^*$ that $\tau$ maps to $\sigma$, is a regular language. We describe a regular tree $\langle T, \tau \rangle$ by a *transducer*, which is a deterministic automaton over the alphabet $D$ in which each state is labeled by a letter in $\Sigma$. Then, $\tau(x)$, for $x \in D^*$, is the label of the state that the transducer reaches after reading $x$. Recall that a perspective strategy for Player $i$ is a function $f_i : \mathsf{PPrefs}_i(G) \to V$. Also, as $\mathsf{PPrefs}_i(G) \subseteq V_i^*$, then every perspective strategy is a $V$-labeled $V_i$-tree $\langle V_i^*, f_i \rangle$. When we seek winning strategies, we seek ones that are generated by finite-state transducers.

## 3 Multi-Player Zero-Sum Perspective Games

In [25], the authors study two-player zero-sum perspective games. Thus, games with $k = 2$ in which $L_2 = \overline{L_1}$. When considering such games, we refer to Player 1 and Player 2 as Player OR and Player AND, respectively, denote $V_1$ and $V_2$ by $V_{\rm or}$ and $V_{\rm and}$, respectively, and denote a game by a tuple $\langle G, L \rangle$, namely only with the objective of Player OR.

In this section we stay in the zero-sum setting, but increase the number of players. As is the case in similar settings [5], a reduction to the two-player case is easy, and we describe it for completeness:

**Lemma 3.1** *Consider a zero-sum perspective game $\mathcal{G} = \langle G, \{L_i\}_{i \in [k]} \rangle$, with $G = \langle AP, \{V_i\}_{i \in [k]}, v_0, E, \tau \rangle$. For every $j \in [k]$, Player $j$ has a winning strategy in $\mathcal{G}$ iff Player OR has a winning strategy in the two-player zero-sum game $\mathcal{G}' = \langle G', L_j \rangle$, where $G' = \langle AP, V_{\rm or}, V_{\rm and}, v_0, E, \tau \rangle$ is such that $V_{\rm or} = V_j$ and $V_{\rm and} = \bigcup_{i \in [k] \setminus \{j\}} V_i$.*

**Proof:** If Player OR wins $\mathcal{G}'$, then Player $j$ can win $\mathcal{G}$ by following the winning P-strategy of Player OR. For the other direction, assume Player OR cannot win $\mathcal{G}'$. Then, for every strategy $f$ for Player OR in $\mathcal{G}'$, there is a strategy $f'$ for Player AND such that $\mathsf{Outcome}(f, f') \notin L_j$. Then, $\mathsf{Outcome}(f_1, ..., f_k) = \mathsf{Outcome}(f, f') \notin L_j$, where $f_j = f$, and $f_i$ is the projection of $f'$ on $V_i^*$, for every $i \in [k] \setminus \{j\}$. Namely, for every $\rho \in \mathsf{Prefs}_i(G)$, we define $f_i(\mathsf{Persp}_i(\rho))$ to be $f'(\rho)$. So, for every strategy $f_j$ for Player $j$ in $\mathcal{G}$, there is a set of strategies $\{f_i\}_{i \in [k] \setminus \{j\}}$ such that $\mathsf{Outcome}(f_1, ..., f_k) \notin L_j$. Thus, Player $j$ cannot win $\mathcal{G}$. $\square$

Lemma 3.1, together with the complexity of deciding two-player perspective games [25], implies the following.

**Theorem 3.2** *Consider a zero-sum game $\mathcal{G} = \langle G, \{L_i\}_{i\in[k]}\rangle$, and $j \in [k]$. The problem of deciding whether Player $j$ has a winning P-strategy is EXPTIME-complete (2EXPTIME-complete) when the objectives $L_i$ are given by UPWs (respectively, LTL formulas). The problem can be solved in time polynomial in $|G|$ and exponential in the UPW (respectively, doubly-exponential in the LTL formula) for $L_j$.*

# 4 Multi-Player Zero-Sum Perspective Games with Coalitions

In $k$-player zero-sum games between $m$ coalitions, the $k \geq 2$ players are partitioned into $m$ coalitions, $C_1, C_2, \ldots, C_m \subseteq [k]$, where the players in each coalition $C_j$ all have the same objective $L_j$. The game being zero-sum means that the objectives $L_j$ form a partition of $(2^{AP})^\omega$.

We describe $k$-player zero-sum games between $m$ coalitions by a tuple $\langle G, \{C_j\}_{j\in[m]}, \{L_j\}_{j\in[m]}\rangle$. Visibility is perspective, in the sense that each player views only visits to vertices she owns. It is not hard to see that if players in a coalition share also their views, the game can be easily reduced to an $m$-player zero-sum perspective game in which Player $j$ owns the vertices of all the players in $C_j$. We show that once visibility is perspective, the problem of deciding whether a coalition of players has a strategy to win is undecidable, even for $k = 3$, $m = 2$, $C_1 = \{1, 2\}$ and $C_2 = \{3\}$. Note that reasoning about games between two coalitions, we have that $L_2 = \overline{L_1}$, and we denote the game by $\mathcal{G} = \langle G, C_1, C_2, L\rangle$.

**Theorem 4.1** *The problem of deciding a zero-sum perspective game between two coalitions is undecidable.*

**Proof:** In [17], the authors describe a reduction from the halting problem for deterministic Turing machines to the problem of deciding whether a coalition of Player 1 and Player 2 can win a 3-player *concurrent observation-based* game between two coalitions. The game being concurrent means that all players participate in the transition of the token in all vertices. The game being observation-based means that the players observe only a subset of the atomic propositions, in all vertices. Formally, a *$k$-player concurrent observation-based game graph* is a tuple $G = \langle AP, Q, \tau, \{\sim_i\}_{i\in[k]}, Act, d, \delta\rangle$, where $AP$ is a finite set of atomic propositions, $Q$ is a finite set of states, $\tau : Q \to 2^{AP}$ is a labelling function, $\sim_i \subseteq Q \times Q$ is an equivalence relation, $Act$ is a finite set of actions, $d : Q \times [k] \to 2^{Act} \setminus \emptyset$ describes the set of actions available to the players at each state, satisfying $d(q, k) = d(q', k)$ for $q \sim_k q'$, and $\delta : Q \times Act^k \to Q$ is a transition function. Thus, in each state, each of the players chooses an action, and the play proceeds to a successor state that depends on all actions.

A *$k$-player concurrent observation-based game* is a tuple $\mathcal{G} = \langle G, \{L_i\}_{i\in[k]}\rangle$, where $G$ is a game graph, and for all $i \in [k]$, the language $L_i \subseteq (2^{AP})^\omega$ is an objective for Player $i$. A *strategy* for Player $i$ in $\mathcal{G}$ is then a function $f_i : Q^+ \to Act$ that is compatible with $d$ and $\sim_i$, Thus,

- $f_i(\rho \cdot q) \in d(q, i)$, for every $\rho \in Q^*$ and $q \in Q$;
- $f_i(\rho) = f_i(\rho')$, for every $\rho, \rho' \in Q^+$ such that $\rho \sim_i \rho'$.

The *outcome* of a profile $\pi = \langle f_1, ..., f_k \rangle$ of strategies is the play obtained when the players follow their strategies. Formally, $\mathsf{Outcome}(\pi) = v_0, v_1, ...$ is such that for all $j \geq 0$, we have that $v_{j+1} = \delta(v_i, f_1(v_0, ..., v_i), \ldots, f_k(v_0, ..., v_i))$. As in perspective games, each profile induces a set of winners and losers.

The reduction in [17] constructs, given a Turing machine $M$, a 3-player concurrent observation-based zero-sum game between two coalitions. Thus, $\mathcal{G}_{\mathcal{M}} = \langle G_M, C_1, C_2, L \rangle$, where $C_1 = \{1, 2\}$ and $C_2 = \{3\}$ are a partition of the players into coalitions, and $L$ is a joint objective of Player 1 and Player 2, who compose the coalition $C_1$. The objective of Player 3 is then $\overline{L}$.

The key features of the game $\mathcal{G}_{\mathcal{M}} = \langle G_M, \{1, 2\}, \{3\}, L \rangle$ constructed in [17] are as follows. First, the game graph $G_M = \langle AP, Q, \tau, \{\sim_i\}_{i \in \{1,2,3\}}, Act, d, \delta \rangle$ satisfies the following conditions.

- $AP = \{ok, p_1, p_2\}$.
- There are three states $q_1, q_2$, and $q_{err}$, such that $q_1$ is the only state in which $p_1$ holds, $q_2$ is the only state in which $p_2$ holds, and $q_{err}$ is the only state in which $ok$ does not hold.
- The relation $\sim_3$ is the identity. Thus, Player 3 can distinguish between all the states in $G_M$.
- For $i \in \{1, 2\}$, the equivalence relation $\sim_i$ is defined by $q \sim_i q'$ iff $(p_i \in \tau(q) \Leftrightarrow p_i \in \tau(q'))$. That is, $q$ and $q'$ are $\sim_i$-equivalent if Player $i$ observes $p_i$ either in both $q$ and $q'$, or in none of them. Thus, Player 1 cannot distinguish between states in $Q \setminus \{q_1\}$, and similarly for Player 2 and $Q \setminus \{q_2\}$.
- There are two sets $Act_1, Act_2 \subseteq Act$ such that Player 1 and Player 2 are allowed to take all actions in $Act_1$ and $Act_2$ in all states, respectively. Thus, for all states $q \in Q$ and $i \in \{1, 2\}$, we have that $d(q, i) = Act_i$. Also, for convenience purposes, we assume that $Act_1, Act_2$ and $Act_3$ are disjoint sets.

The objective of the coalition $C_1$ is $L = Gok$, and the coalition $C_1$ wins iff $M$ does not halt on the empty tape.

We show that there is a zero-sum perspective game $\mathcal{G} = \langle G, C_1, C_2, L' \rangle$, again between the coalitions $C_1 = \{1, 2\}$ and $C_2 = \{3\}$, such that $C_1$ wins $\mathcal{G}$ iff $C_1$ wins $\mathcal{G}_{\mathcal{M}}$. Constructing $\mathcal{G}$, we have to address the fact that $\mathcal{G}$ is perspective (rather than with observation-based) and turn-based (rather than concurrent). The idea of our reduction is to use the perspective view in order to mimic the concurrent choices of the players in $\mathcal{G}_{\mathcal{M}}$: We first let Player 3 choose her action. Then, Player 1 chooses her action without knowing the choice of Player 3, and then Player 2 chooses her action without knowing the choices of Player 3 and Player 1. This corresponds to the three choices being made concurrently. Finally, Player 3, the only player that sees all choices (but who made her choice before Player 1 and Player 2) moves the token to the successor vertex. While Player 3 need not follow $\delta$, we define the objectives of the players in a way so that Player 3 has no incentive to do so. Specifically, a computation fulfils the objective of the coalition of Player 1 and Player 2 if it satisfies $Gok$ or includes a violation of $\delta$. Accordingly, it fulfils the objective of Player 3 if it respects $\delta$ and eventually reaches the vertex $q_{err}$, where $ok$ does not hold.

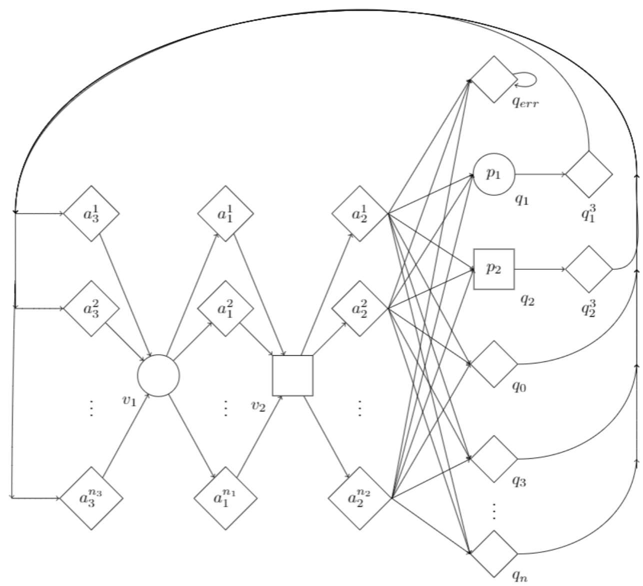


Figure 1: The game graph $G$. The circles, squares, and diamonds are vertices controlled by Player 1, Player 2, and Player 3, respectively.

Formally, $\mathcal{G} = \langle G, \{1,2\}, \{3\}, L' \rangle$ is defined as follows (see Figure 1). The game graph is $G = \langle AP', \{V_i\}_{i\in\{1,2,3\}}, q_0, E, \tau' \rangle$, where

1. $AP' = AP \cup Act \cup Q$.

2. For $i \in \{1,2\}$, we have that $V_i = \{q_i, v_i\}$. Recall that $q_i$ is the only state labeled by $p_i$. The vertex $v_i$ is where Player $i$ chooses her action.

3. $V_3 = Q \setminus \{q_1, q_2\} \cup \{q_1^3, q_2^3\} \cup Act$. For $i \in \{1,2,3\}$, the vertices in $Act_i$ correspond to the different actions that Player $i$ can take. The vertices $q_1^3$ and $q_2^3$ allow Player 3 to choose her action after a visit in $q_1$ and $q_2$, respectively. Note that Player 3 choosing her action before Player 1 and Player 2 guarantees that it is independent of their choices, and thus the sequence of transitions between states in $Q$ in $G$ corresponds to a concurrent transition in $G_M$.

4. The set $E$ contains the following edges:

- $\langle q, a\rangle$, for every $q \in Q \setminus \{q_1, q_2, q_{err}\}$ and $a \in d(q, 3)$.
- $\langle q_i, q_i^3\rangle$, for every $i \in \{1, 2\}$.
- $\langle q_i^3, a\rangle$, for every $i \in \{1, 2\}$ and $a \in d(q_i, 3)$.
- $\langle v_i, a\rangle$, for every $i \in \{1, 2\}$ and $a \in Act_i$.
- $\langle a, v_1\rangle$, for every $a \in Act_3$.
- $\langle a, v_2\rangle$, for every $a \in Act_1$.
- $\langle a, q\rangle$, for every $a \in Act_2$ and $q \in Q$.
- $\langle q_{err}, q_{err}\rangle$.

5. The labelling function $\tau'$ is defined as follows.
   - For every $a \in Act$, we have that $\tau'(a) = \{a, ok\}$.
   - For every $i \in \{1, 2\}$, we have that $\tau'(q_i) = \{p_i, ok\}$,
   - For every $q \in Q \setminus \{q_{err}, q_1, q_2\}$, we have that $\tau'(q) = \{q, ok\}$.
   - For every $i \in \{1, 2\}$, we have $\tau'(q_i^3) = \{q_i, ok\}$.
   - $\tau'(q_{err}) = \{q_{err}\}$.

Now, the objective of Player 1 and Player 2 is

$$L' = Gok \vee \bigvee_{\langle q,a_1,a_2,a_3\rangle \in Q\times Act_1 \times Act_2 \times Act_3} F(q \wedge X(a_3 \wedge XX(a_1 \wedge XX(a_2) \wedge XX\neg\delta(q,(a_1,a_2,a_3))))).$$

Thus[1], either the generated computation satisfies $Gok$, or Player 3 does not respect $\delta$: eventually the computation reaches a state $q$, the players choose actions $a_1, a_2$, and $a_3$, and Player 3 moves the token to a vertex that is not $\delta(q, a_1, a_2, a_3)$.

Recall that the strategies of Player 1 and Player 2 in $\mathcal{G}_{\mathcal{M}}$ observe only visits in $q_1$ and $q_2$, respectively, as well as the number of vertices visited since $q_1$ and $q_2$ have been visited. This is exactly what the perspective view of Player 1 and Player 2 in $\mathcal{G}$ includes. In addition, Player 3 can observe choices made by Player 1 and Player 2 and can always respect $\delta$. Accordingly, the coalition of Player 1 and Player 2 wins in $\mathcal{G}$ iff it wins in $\mathcal{G}_{\mathcal{M}}$, and we are done. $\square$

# 5 Multi-Player Non-Zero-Sum Perspective Games

We proceed to non-zero-sum perspective games. Here, the basic question is whether an NE profile exists.

**Theorem 5.1** *Deciding the existence of an NE in $k$-player non-zero-sum perspective games is undecidable for $k \geq 3$.*

[1] For clarity, our definition of $L'$ ignores the special treatment for $q \in \{q_1, q_2\}$, where the first $X$ should be $XX$, corresponding to the computation passing through $q_1^3$ and $q_2^3$.

**Proof:** We describe a reduction from the problem of deciding a zero-sum perspective game between two coalitions, proved to be undecidable in Theorem 4.1. There, undecidability is shown for $\mathcal{G} = \langle G, C_1, C_2, L \rangle$, with $C_1 = \{1, 2\}$ and $C_2 = \{3\}$. We construct a 3-player non-zero-sum perspective game $\mathcal{G}'$ such that there is an NE in $\mathcal{G}'$ iff $C_1$ wins $\mathcal{G}$.

The idea behind $\mathcal{G}'$ is to add to $\mathcal{G}$ a matching-pennies game, which does not have an NE, between Player 1 and Player 2, and let Player 1 choose whether to play $\mathcal{G}$ or play the matching-pennies game [11]. Now, if $C_1$ wins $\mathcal{G}$, then Player 1 would chose to play $\mathcal{G}$, resulting in a profile in which $C_1$ wins and no deviation of Player 3 can make her winning, and thus that profile is an NE. On the other hand, if $C_1$ does not win $\mathcal{G}$, then no profile in which Player 1 plays $\mathcal{G}$ is an NE. Indeed, Player 1 would deviate to a strategy in which she chooses the matching-pennies game and wins there. Also, no profile in which Player 1 plays the matching-pennies game is an NE. Indeed, the player that looses the matching-pennies game can deviate to a strategy in which she wins the game.

Formally, let $G = \langle AP, \{V_1, V_2, V_3\}, E, v_0, \tau \rangle$. We define $\mathcal{G}' = \langle G', L_1, L_2, L_3 \rangle$ as follows (see Figure 2). The game graph is $G' = \langle AP', \{V'_i\}_{i \in [3]}, v_0^1, E', \tau' \rangle$, where

1. $AP' = AP \cup \{p\}$, for some $p \notin AP$.
2. The vertices owned by Player 1 are $V'_1 = V_1 \cup \{v_0^1, v_p^1, v_{\neg p}^1\}$.
3. The vertices owned by Player 2 are $V'_2 = V_2 \cup \{v^2, v_p^2, v_{\neg p}^2\}$.
4. The vertices owned by Player 3 are $V'_3 = V_3$.
5. The set of edges $E'$ contains $E$ and the following edges.
   - $\langle v_0^1, v \rangle$, for every $v \in \{v_0, v_p^1, v_{\neg p}^1\}$.
   - $\langle v, v^2 \rangle$, for every $v \in \{v_p^1, v_{\neg p}^1\}$.
   - $\langle v^2, v \rangle$ and $\langle v, v \rangle$, for every $v \in \{v_p^2, v_{\neg p}^2\}$.
6. The function $\tau' : V' \to 2^{AP'}$ is defined as follows.
   - $\tau'(v) = \tau(v)$, for every $v \in V$.
   - $\tau'(v) = p$, for every $v \in \{v_p^1, v^2, v_p^2\}$.
   - $\tau'(v) = \emptyset$, for every $v \in \{v_0^1, v_{\neg p}^1, v_{\neg p}^2\}$.

As for the objectives, Player 1 wins the matching-pennies sub-game if she and Player 2 choose vertices that agree on $p$, and Player 2 wins the sub-game if she and Player 1 choose vertices that do not agree on $p$. Recall that the objective of $C_1$ in $\mathcal{G}$ is $L$. Accordingly, $L_1 = (\neg p) \cdot (L + p \cdot p \cdot p^\omega + \neg p \cdot p \cdot \neg p^\omega)$ and $L_2 = (\neg p) \cdot (L + p \cdot p \cdot \neg p^\omega + \neg p \cdot p \cdot p^\omega)$. Then, $L_3 = (\neg p) \cdot \overline{L}$. Thus, Player 1 wins if either she wins $\mathcal{G}$ (with Player 2) or she wins (without Player 2) the matching-pennies game, and similarly for Player 2. Then, Player 3 wins if the game proceeds to $\mathcal{G}$ and her objective there is satisfied. $\square$

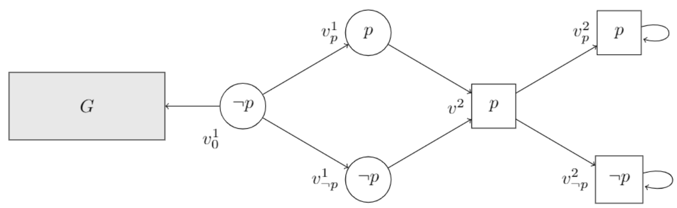


Figure 2: The game graph $G'$. The circles and squares are vertices controlled by Player 1 and Player 2, respectively. At the beginning, Player 1 can choose to proceed to $G$, or move to the "matching-pennies" sub-game on the right.

# 6 Two-Player Non-Zero-Sum Perspective Games

Consider a perspective two-player non-zero-sum game $\mathcal{G} = \langle G, \{L_1, L_2\}\rangle$. We distinguish between four types of NE profiles in $\mathcal{G}$, characterized by the partition of the players to winners and losers. We term the four types LL-NE (both players lose), LW-NE (Player 1 loses and Player 2 wins), WL-NE (Player 1 wins and Player 2 loses), and WW-NE (both players win). We show that we can decide existence for each of the four types, which implies decidability of some NE.

During the section, we fix a game $\mathcal{G} = \langle G, \{L_1, L_2\}\rangle$, where $G = \langle AP, V_1, V_2, E, v_0, \tau\rangle$. Analyzing the complexity of the problem, we consider various possible representations of the objectives $L_1$ and $L_2$. We start with NBWs $\mathcal{N}_1$ and $\mathcal{N}_2$, recognizing $L_1$ and $L_2$, respectively. In Section 6.4, we analyze the complexity also for the case $L_1$ and $L_2$ are given by LTL formulas or other types of automata.

## 6.1 Deciding the existence of an LL-NE

**Lemma 6.1** *A profile $\langle f_1, f_2\rangle$ is an LL-NE in $\mathcal{G}$ iff $f_1$ is a winning P-strategy for Player* OR *in the zero-sum game $\langle G, \overline{L_2}\rangle$ and $f_2$ is a winning P-strategy for Player* AND *in the zero-sum game $\langle G, L_1\rangle$.*

**Proof:** Consider an LL-NE $\langle f_1, f_2\rangle$. By definition, for every strategy $f_2'$ of Player 2, we have that $2 \in \mathsf{Lose}(\pi[2 \leftarrow f_2'])$. Thus, $\tau(\mathsf{Outcome}(f_1, f_2')) \in \overline{L_2}$, and so $f_1$ is a winning P-strategy for Player OR in the zero-sum game with the objective $\overline{L_2}$. Similarly, as $1 \in \mathsf{Lose}(\pi[1 \leftarrow f_1'])$ for every strategy $f_1'$ for Player 1, we have that $f_2$ is a winning P-strategy for Player AND in the zero-sum game where Player AND's objective is $\overline{L_1}$.

For the other direction, assume that $f_1$ and $f_2$ are winning P-strategies for Player OR and Player AND in the zero-sum games $\langle G, \overline{L_2}\rangle$ and $\langle G, L_1\rangle$, respectively. Then, $\tau(\mathsf{Outcome}(f_1, f_2)) \in \overline{L_1} \cap \overline{L_2}$, and neither Player OR nor Player AND has a beneficial deviation. Hence, the profile $\langle f_1, f_2\rangle$ is an LL-NE, and we are done. $\square$

**Theorem 6.2** *Deciding the existence of an LL-NE in $\mathcal{G}$ can be done in time polynomial in $|G|$ and exponential in $|\mathcal{N}_1|$ and $|\mathcal{N}_2|$.*

**Proof:** By Lemma 6.1, the problem can be reduced to deciding the zero-sum games $\langle G, \overline{L_2}\rangle$ and $\langle G, L_1\rangle$. By [25], the latter can be done in time polynomial in $|G|$ and exponential in $|\overline{\mathcal{U}_2}|$ and $|\overline{\mathcal{U}_1}|$, for UCWs $|\overline{\mathcal{U}_2}|$ and $|\overline{\mathcal{U}_1}|$ that recognize $\overline{L_2}$ and $\overline{L_1}$, respectively, and which we can obtain by dualizing $\mathcal{N}_2$ and $\mathcal{N}_1$. □

## 6.2 Deciding the existence of a WW-NE

**Lemma 6.3** *There is a WW-NE in $\mathcal{G}$ iff there is a computation of $G$ that is a member of $L_1 \cap L_2$.*

**Proof:** It is easy to see that if there is a WW-NE in $\mathcal{G}$, then there is a path in $G$ that satisfies $L_1 \cap L_2$. First, it is easy to see that if $\pi = \langle f_1, f_2\rangle$ is a WW-NE, then $\tau(\mathsf{Outcome}(f_1, f_2))$, which is a computation of $G$, is in $L_1 \cap L_2$.

For the other direction, assume there is a path $\rho$ in $G$ such that $\tau(\rho) \in L_1 \cap L_2$. Every two different prefixes $\rho'$ and $\rho''$ of $\rho$ that are of the form $V^* \cdot V_1$, satisfy $|\mathsf{Persp}_1(\rho')| \neq |\mathsf{Persp}_1(\rho'')|$, so in particular, $\mathsf{Persp}_1(\rho') \neq \mathsf{Persp}_1(\rho'')$. Likewise, if $\rho'$ and $\rho''$ are of the form $V^* \cdot V_2$, then $\mathsf{Persp}_2(\rho') \neq \mathsf{Persp}_2(\rho'')$. Thus, the projection of $\rho$ on $V_1^*$ and $V_2^*$ induces two perspective strategies $f_1$ and $f_2$ for Player 1 and Player 2, respectively, such that $\mathsf{Outcome}(f_1, f_2) = \rho$. Hence, $\langle f_1, f_2\rangle$ is a WW-NE. □

**Theorem 6.4** *Deciding the existence of a WW-NE in $\mathcal{G}$ can be done in NLOGSPACE, and in time polynomial in $|\mathcal{G}|$, $|\mathcal{N}_1|$, and $|\mathcal{N}_2|$.*

**Proof:** By Lemma 6.3, deciding the existence of a WW-NE can be reduced to checking the nonemptiness of the intersection of $\mathcal{G}$, $\mathcal{N}_1$, and $\mathcal{N}_2$, implying the desired complexity. □

## 6.3 Deciding the existence of a WL-NE

We say that a strategy $f_1$ for Player 1 is a WL-strategy if there is a strategy $f_2$ for Player 2 such that $1 \in \mathsf{Win}(\langle f_1, f_2\rangle)$, $2 \in \mathsf{Lose}(\langle f_1, f_2\rangle)$, and for every strategy $f_2'$ for Player 2, we have that $2 \in \mathsf{Lose}(\langle f_1, f_2'\rangle)$. We decide the existence of a WL-NE by deciding the existence of a WL-strategy for Player 1.

**Lemma 6.5** *We can construct an NBT $\mathcal{N}$ that is not empty iff there is a WL-strategy for Player 1 in $\mathcal{G}$. The size of $\mathcal{N}$ is polynomial in $|G|$ and $|\mathcal{N}_1|$, and is exponential in $|\mathcal{N}_2|$ .*

**Proof:** We define $\mathcal{N}$ as the intersection of an NBT $\mathcal{A}_1$ that accepts a $V$-labeled $V_1$-tree $\langle V_1^*, f_1\rangle$ iff there is a strategy $f_2$ for Player 2 such that $1 \in \mathsf{Win}(\langle f_1, f_2\rangle)$, and an NBT $\mathcal{A}_2'$ that is *sufficiently equivalent* to a UCT $\mathcal{A}_2$ that accepts a $V$-labeled $V_1$-tree $\langle V_1^*, f_1\rangle$ iff for all strategies $f_2$, we have that $2 \in \mathsf{Lose}(\langle f_1, f_2\rangle)$. Intuitively, in "sufficiently equivalent" we mean that if the intersection of $\mathcal{A}_1$

and $\mathcal{A}_2$ is not empty, then so is the intersection of $\mathcal{A}_1$ and $\mathcal{A}'_2$. We formalize this intuition when we define $\mathcal{A}'_2$ below. The size of $\mathcal{A}_1$ is polynomial in $|G|$ and $|\mathcal{N}_1|$. The size of $\mathcal{A}'_2$ is polynomial in $|G|$ and $|\mathcal{N}_1|$, and exponential in $|\mathcal{N}_2|$. Consequently, as intersection of NBTs involves two copies of their product, so is the size of $\mathcal{N}$.

We start with the construction of the NBT $\mathcal{A}_1$. Note that $\mathcal{A}_1$ should reject $\langle V_1^*, f_1\rangle$ iff for every strategy $f_2$ for Player 2, we have that $1 \in \mathsf{Lose}(\langle f_1, f_2\rangle)$. Equivalently, if $\langle V_1^*, f_1\rangle$ is a winning strategy for Player OR in the two-player zero-sum game $\langle G, \overline{L_1}\rangle$. By [25], we can construct a UCT $\tilde{\mathcal{A}}_1$ that accepts such winning strategies, of size polynomial in $|G|$ and in a UCW for $\overline{L_1}$, which we have by dualizing $\mathcal{N}_1$. We obtain the NBT $\mathcal{A}_1$ by dualizing $\tilde{\mathcal{A}}_1$. Note that the NBT $\mathcal{A}_1$ searches for a path, in the sense that in each state, it sends a single copy to a single successor.

We continue to the construction of the NBT $\mathcal{A}'_2$, and we start with the UCT $\mathcal{A}_2$. Note that $\mathcal{A}_2$ should accept $\langle V_1^*, f_1\rangle$ iff it is a winning strategy for Player OR in the two-player zero-sum game $\langle G, \overline{L_2}\rangle$. By [25], we can construct a UCT that accepts such winning strategies, of size polynomial in $|G|$ and in a UCW for $\overline{L_2}$, which we have by dualizing $\mathcal{N}_2$. Thus, the size of $\mathcal{A}_2$ is polynomial in $|G|$ and $|\mathcal{N}_2|$. As described in [25], the states of the UCT $\mathcal{A}_2$ are triples in $V \times Q_2 \times \{\top, \bot\}$, where $Q_2$ is the state space of $\mathcal{N}_2$. Also, $\mathcal{A}_2$ is deterministic in its $V$-element: all states sent to the same direction $v$ of the tree agree on their $V$-element, which is $v$.

Now we have to transform the UCT $\mathcal{A}_2$ to an NBT. In [27], the authors describe such a transformation, which preserves nonemptiness. Here, we need to preserve nonemptiness of the intersection of $\mathcal{A}_2$ with $\mathcal{A}_1$, so we need to delve into the details of the construction in [27]. The construction there is parameterized by $m \geq 1$, and transforms a UCT $\mathcal{A}$ to an NBT $\mathcal{A}'$ that accepts only trees in $L(\mathcal{A})$, and accepts all trees in $L(\mathcal{A})$ that are generated by a transducer with $m$ states. Since a UCT with $n$ states is nonempty iff it accepts a tree generated by a transducer with $n^{3n} \cdot n^6$ states, taking $m = n^{3n} \cdot n^6$ guarantees that $\mathcal{A}'$ is nonempty iff $\mathcal{A}$ is nonempty. The bound in [27] is a bit tighter (yet less clean), and follows from a bound on the size of a nondeterministic Rabin tree automaton equivalent to $\mathcal{A}$ [31], and the fact a nonempty nondeterministic Rabin tree automaton with $n$ states accepts a tree that is generated by a transducer with $n$ states [18].

Now, the intersection of $\mathcal{A}_2$ with $\mathcal{A}_1$ is not empty iff it contains a tree generated by a transducer with $m$ states, where $m$ bounds the size of a nondeterministic Rabin tree automaton for this intersection. Since $\mathcal{A}_1$ searches a path and $\mathcal{A}_2$ is deterministic in its $V$-element, the size of such a nondeterministic Rabin tree automaton is polynomial in $|G|$ and $|\mathcal{N}_1|$, and is exponential in $|\mathcal{N}_2|$. Applying the construction in [27] with the parameter $m$, again using the fact that $\mathcal{A}_2$ is deterministic in its $V$-element, then results in an NBT $\mathcal{A}'_2$ of size polynomial in $|G|$ and $|\mathcal{N}_1|$, and exponential in $|\mathcal{N}_2|$, which is also the size of the NBT $\mathcal{N}$ for the intersection of $\mathcal{A}_1$ and $\mathcal{A}'_2$. $\square$

**Theorem 6.6** *Deciding the existence of a WL-NE can be done in time polynomial in $|G|$ and $|\mathcal{N}_1|$ and exponential in $|\mathcal{N}_2|$. For LW-NE, the complexity is polynomial in $|G|$ and $|\mathcal{N}_2|$ and exponential in $|\mathcal{N}_1|$.*

**Proof:** By Lemma 6.5, we can construct an NBT of size polynomial in $|G|$ and $|\mathcal{N}_1|$, and exponential in $|\mathcal{N}_2|$, that is not empty iff there is a WL-NE. The complexity for WL-NE then follows from the fact the nonemptiness problem for NBTs can be solved in quadratic time [39]. For LW-NE, we simply switch Player 1 and Player 2. □

## 6.4 Other formalisms

The results from Theorems 6.2, 6.4, and 6.6 can be adjusted to show that the problem of deciding the existence of an NE with objectives given by DFWs and LTL formulas, can be solved in EXPTIME and 2EXPTIME, respectively. In Theorem 6.7 below we describe the exact complexities and provide matching lower bounds for all classes.[2]

**Theorem 6.7** *Consider a two-player non-zero-sum perspective game $\mathcal{G} = \langle G, L_1, L_2 \rangle$. Deciding the existence of an NE can be done in time polynomial in $|G|$ and*

- *exponential in $L_1$ and $L_2$, when given by NBWs or DFWs, in which case the problem is EXPTIME-complete.*
- *doubly-exponential in $L_1$ and $L_2$, when given by LTL formulas, in which case the problem is 2EXPTIME-complete.*

**Proof:** Since there is an NE in $\mathcal{G}$ iff there is an LL, LW, WL, or WW-NE in $\mathcal{G}$, the upper bound for NBWs followsfrom Theorems 6.2, 6.4, and 6.6. Since DBWs are a special case of NBWs, and all considerations can be applied also to objectives describing finite words, the upper bound for DFWs follows. Finally, an exponentially higher bound for LTL follows from the exponential translation of LTL formulas to NBWs [40].

For the lower bound, we describe a reduction from the problem of deciding whether Player OR wins in a zero-sum perspective game, proved to be 2EXPTIME-hard in case the objective is given by an LTL formula and EXPTIME-hard in case it is given by a DFW [25]. Given a two-player zero-sum perspective game $\mathcal{G} = \langle G, L \rangle$, we construct a non-zero-sum perspective game $\mathcal{G}' = \langle G', L_1, L_2 \rangle$, such that there is an NE in $\mathcal{G}'$ iff Player OR wins $\mathcal{G}$.

The construction of $\mathcal{G}' = \langle G', L_1, L_2 \rangle$ is similar to the one in the proof of Theorem 5.1, and consists of adding to $G$ a matching-pennies game between Player 1 and Player 2. The initial vertex of $G'$ is controlled by Player 1, who

[2]Note that objectives in DFWs refer to finite outcomes of games. A lower bound for them implies a lower bound on the special case of DBWs that correspond to safety and co-safety objectives. Also, an upper bound for DPWs (deterministic parity word automata) follows either by an adjustment of the proofs of Theorems 6.2, 6.4, or by a polynomial translation of the DPWs to NBWs.

chooses to either play $G$ or to play the matching-pennies game. In that game (as in Figure 2), Player 1 wins if Player 1 and Player 2 chose to proceed to vertices that agree on the labelling of a new atomic proposition $p$, and Player 2 wins otherwise. Then, the objective $L_1$ of Player 1 is to satisfy $L$ or to win the matching-pennies game. The objective $L_2$ of Player 2 is to satisfy $\overline{L}$ or win the matching-pennies game.

We prove that there is an NE in $\mathcal{G}'$ iff Player OR wins $\mathcal{G}$. First, if Player OR has a winning strategy $f_1$ in $\mathcal{G}$, then a profile in which Player OR proceeds to the sub-graph $G$ and follows $f_1$ is an NE. Indeed, Player OR satisfies $L_1$ and Player 2 has no beneficial deviation. Also, if Player OR does not win $\mathcal{G}$, every profile in which Player 1 proceeds to $G$ is not an NE: if Player 2 loses, she has a beneficial deviation to ensure that $\overline{L}$ is satisfied, and if Player 1 loses, she has a beneficial deviation to win the matching-pennies game. In addition, every profile in which Player 1 proceeds to the matching-pennies game is not an NE either; the loser always has a beneficial deviation.

Finally, in the three formalisms we consider, the objectives $L_1$ and $L_2$ can be obtained by taking a disjunction of $L$ with an objective that specifies winning outcomes in the matching-pennies game. The latter is of a constant size, and the disjunction of $L$ with it (either by $\wedge$, in case of LTL formulas, or by an intersection, in case of automata) does not include a blow-up. Hence, the reduction is polynomial, and we are done. $\square$

# 7 Rational Synthesis

In rational synthesis, we are given a $k$-player non-zero-sum game $\mathcal{G} = \langle G, \{L_i\}_{i\in[k]}\rangle$ and we seek a strategy for Player 1 with which her objective is guaranteed to be satisfied, assuming rationality of the other players. Intuitively, Player 1 is assumed to be the authority, which we control, and thus we do not have to count on her rationality. Technically, in settings when Player $i$ is controllable, we say that a profile $\pi = \langle f_1, ..., f_k\rangle$ is an *$i$-fixed NE*, if no player in $[k] \setminus \{i\}$ has a beneficial deviation. We formalize the intuition behind rational synthesis in two different ways:

**Definition 7.1 [Rational Synthesis]** *Consider a $k$-player non-zero-sum perspective game $\mathcal{G} = \langle G, \{L_i\}_{i\in[k]}\rangle$. We define the following two variants of rational synthesis (RS):*

- Cooperative rational synthesis *(CRS), where the desired output is a 1-fixed NE $\pi$ such that $1 \in \mathsf{Win}(\pi)$.*
- Non-cooperative rational synthesis *(NRS), where the desired output is a strategy $f_1$ for Player 1 such that there is a 1-fixed NE $\pi = \langle f_1, f_2, \ldots, f_k\rangle$ with $1 \in \mathsf{Win}(\pi)$, and for every 1-fixed NE $\pi = \langle f_1, f_2, \ldots, f_k\rangle$, we have that $1 \in \mathsf{Win}(\pi)$.*

As in traditional synthesis, one can also define the corresponding decision problems, of *rational realizability*, where we only need to decide whether the desired profile (in the cooperative variant) or strategy (in the non-cooperative variant) exists. In order to avoid additional notations, we sometimes refer to CRS and NRS also as decision problems.

**Example 7.1** Consider the two-player game $G$ described in Figure 3. Recall that we denote vertices owned by Player 1 by circles and those owned by Player 2 by squares. We first view $G$ as a zero-sum game, where the objective of Player 1

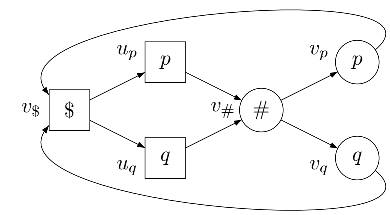


Figure 3: The game graph $G$.

is $L_1 = G((\$ \wedge Xp) \rightarrow XXXp) \wedge ((\$ \wedge Xq) \rightarrow XXXq))$. That is, whenever Player 2 moves the token from $v_\$$ to $u_p$, then Player 1 should move the token from $v_\#$ to $v_p$, and similarly for $u_q$ and $v_q$. Clearly, since Player 1 cannot observe the choice of Player 2, she has no winning strategy. We proceed to the non-zero-sum setting and assume that Player 2 has an objective $L_2 = G(\$ \rightarrow Xp)$. Keeping this in mind, Player 1 has an NRS solution (and then, also a CRS solution). To see this, consider a strategy $f_1$ for Player 1 that always moves the token from $v_\#$ to $v_p$. Then, for every strategy $f_2$ of Player 2, if a profile $\langle f_1, f_2 \rangle$ is a 1-fixed NE, then $\tau(\mathsf{Outcome}(\langle f_1, f_2 \rangle))$ is in $L_1$. Indeed, as Player 2 has a strategy with which her objective is satisfied, namely $f_2'$ that always moves the token from $v_\$$ to $u_p$, and this is the only strategy with which $L_2$ is satisfied, then every profile $\langle f_1, f_2 \rangle$ that is a 1-fixed NE has $f_2 = f_2'$. Since $\tau(\mathsf{Outcome}(\langle f_1, f_2' \rangle))$ is in $L_1$, it follows that $f_1$ is an NRS solution.

Consider now a strategy $f_1'$ of Player 1 that always directs the token from $v_\#$ to $v_q$. It is easy to see that $f_1'$ is not an NRS solution. Indeed, the profile $\langle f_1', f_2' \rangle$ is a 1-fixed NE in which $L_1$ is not satisfied. Note also that $f_1'$ cannot be a part of a CRS solution, as $\langle f_1', f_2' \rangle$ is the only 1-fixed NE containing $f_1'$, and Player 1 loses in it. Finally, note that the strategy $f_1'$ is a solution in a variant of NRS that requires all NEs (rather than all 1-fixed-NEs) to satisfy the objective of Player 1. Indeed, the profile $\langle f_1', f_2' \rangle$ and other profiles in which Player 1 loses are not NEs.

We first show that the two variants of rational synthesis are undecidable in perspective games with three or more players.

**Theorem 7.2** *The cooperative and non-cooperative rational synthesis are undecidable in perspective games with three or more players.*

**Proof:** In the proof of Theorem 5.1, we construct, given a Turing machine $M$, a 3-player non-zero-sum perspective game $\mathcal{G}'$ such that there is an NE $\pi$ in

$\mathcal{G}'$ iff $M$ does not halt on the empty tape. Essentially, $\mathcal{G}'$ is defined by letting Player 1 choose between playing a game on a graph $G$ in which the coalition of Player 1 and Player 2 wins iff $M$ does not halt on the empty tape, or playing a matching-pennies game between Player 1 and Player 2, in which no NE exists. Accordingly, $M$ does not halt on the empty tape iff there is an NE in $\mathcal{G}'$, in which case $\mathsf{Win}(\pi) = \{1, 2\}$ for every NE $\pi$.

We start with CRS. By the characteristics of $\mathcal{G}'$ above, there is a CRS solution for Player 1 iff $M$ does not halt on the empty tape. Indeed, the solution is an NE $\pi$, which by definition is also a 1-fixed NE, with $\mathsf{Win}(\pi) = \{1, 2\}$.

We continue to NRS. Here, we use a variant of $\mathcal{G}'$ where we let Player 1 and Player 3 play the matching-pennies game, and we seek an NRS solution for Player 2. For a strategy $f_2$ of Player 2, we say that a strategy $f_1$ for Player 1 is *good with* $f_2$ if for every strategy $f_3$ for Player 3, we have that $\pi = \langle f_1, f_2, f_3 \rangle$ is an NE with $\mathsf{Win}(\pi) = \{1, 2\}$. Assume first that $M$ does not halt on the empty tape. By the definition of $\mathcal{G}'$, we then have a strategy $f_2$ for Player 2 such that there is a strategy $f_1$ for Player 1 that is good with $f_2$. Indeed, $f_1$ and $f_2$ are the strategies with which the coalition of Player 1 and Player 2 wins when the game proceeds to $G$. We claim that $f_2$ is an NRS solution. First, by the above, there are strategies $f_1$ for Player 1 and $f_3$ for Player 3, such that $\pi = \langle f_1, f_2, f_3 \rangle$ is a 2-fixed NE with $2 \in \mathsf{Win}(\pi)$. Indeed, we can take any strategy $f_1$ for Player 1 that is good with $f_2$ and then any strategy $f_3$ for Player 3. In addition, since there exists a strategy for Player 1 that is good with $f_2$, then for all profiles $\pi = \langle f_1, f_2, f_3 \rangle$, if $\pi$ a 2-fixed NE, and so Player 1 does not deviate, then it must be that $\mathsf{Win}(\pi) = \{1, 2\}$, and we are done. Assume now that $M$ halts on the empty tape. We claim that then, for every strategy $f_2$ of Player 2, the profile $\pi = \langle f_1, f_2, f_3 \rangle$ is not a 2-fixed NE. Indeed, if $f_1$ is such that Player 1 chooses to play in $G$, then Player 3 has a beneficial deviation to win there. Also, if $f_1$ chooses to play the matching-pennies game, then either Player 1 wins there, in which case Player 3 would deviate, or Player 3 wins, in which case Player 1 deviates. Hence, no strategy for Player 2 can be an NRS solution, and we are done. □

Following Theorem 7.2, we continue to study rational synthesis for settings with $k = 2$. We fix $\mathcal{G} = \langle G, L_1, L_2 \rangle$. We assume that $L_1$ and $L_2$ are given by LTL formulas $\psi_1$ and $\psi_2$. This is both because rational synthesis has been traditionally studied for the temporal logic formalism and because our algorithm for the CRS requires complementation of $L_2$, which involves no blow-up for LTL.

**Remark 7.1** We could have defined a variant of NRS where the desired output is a strategy $f_1$ for Player 1 such that for every 1-fixed NE $\pi = \langle f_1, f_2, \ldots, f_k \rangle$, we have that $1 \in \mathsf{Win}(\pi)$. Thus, without requiring a 1-fixed NE to exist. When $k = 2$, the two variants coincide. Indeed, assume $f_1$ is a strategy for Player 1 such that for every 1-fixed NE $\pi = \langle f_1, f_2 \rangle$, we have that $1 \in \mathsf{Win}(\pi)$. Now, if there is $f_2$ such that $2 \in \mathsf{Win}(\langle f_1, f_2 \rangle)$, then $\langle f_1, f_2 \rangle$ is the required 1-fixed NE. Also, if no such $f_2$ exists, then Player 2 has no incentive to deviate from a profile $\langle f_1, f_2 \rangle$, which is therefore a 1-fixed NE. □

## 7.1 Solving cooperative rational synthesis

**Theorem 7.3** *CRS can be solved in time polynomial in $|G|$, exponential in $|\psi_1|$ and doubly-exponential in $|\psi_2|$.*

**Proof:** For every profile $\pi$ with $1 \in \mathsf{Win}(\pi)$, we have that $\pi$ is an NE iff $\pi$ is a 1-fixed NE. Indeed, Player 1 has no incentive to deviate. It is thus not hard to see that a solution $\pi$ to CRS is a WW-NE or a WL-NE. In Theorems 6.4 and 6.6, we solved the corresponding decision problems. Here, we show we can extend them to return the strategies that constitute the NEs. Solving CRS is then done by first searching a WW-NE (both out of courtesy to Player 2 and since it is computationally easier), in time polynomial in $|G|$, and exponential in $|\psi_1|$ and $|\psi_2|$, and then, if no WW-NE exists, searching a WL-NE, which can be done in time polynomial in $|G|$, exponential in $|\psi_1|$ and doubly-exponential in $|\psi_2|$.

In Theorem 6.4, deciding the existence of a WW-NE is reduced checking if there is a path $\rho$ in $G$ such that $\tau(\rho)$ satisfies both $\psi_1$ and $\psi_2$. Once such a path $\rho = v_1, v_2, v_3, ... \in V^\omega$ is found, we define $f_i$, for $i \in \{1, 2\}$ by $f_i(\mathsf{Persp}_i(v_1, ..., v_j)) = v_{j+1}$, for every $j \geq 1$ such that $v_j \in V_i$. Then, we have that $\mathsf{Outcome}(f_1, f_2) = \tau(\rho)$. Since $\rho$ can be found in time polynomial $P$ and exponential in $|\psi_1|$ and $|\psi_2|$, the desired complexity follows.

We continue to a WL-NE. By Theorem 6.6, deciding the existence of a WL-NE, and finding a strategy $f_1$ for Player 1 that is part of a WL-NE, can be done in time polynomial in $|G|$, exponential in $|\psi_1|$, and doubly-exponential in $|\psi_2|$. Note that given a strategy $f_1$ for Player 1, we have that $2 \in \mathsf{Lose}(\langle f_1, f_2\rangle)$ for every strategy $f_2$ for Player 2. Then, it is left to find a path $\rho$ in $G$ that agrees with $f_1$ and satisfies $\psi_1$. The projection of $\rho$ on $V_2^*$ results in a strategy $f_2$ for Player 2 such that $1 \in \mathsf{Win}(\langle f_1, f_2\rangle)$. $\square$

**Remark 7.2 [On the complexity of CRS]** The observant reader may be concerned by the exponential complexity in terms of $|\psi_1|$, as rational synthesis is 2EXPTIME-hard for LTL specifications even in a setting with no uncertainty [21, 24]. The concern is justified: A careful analysis reveals that in a setting with two players, rational synthesis is in fact easier than traditional synthesis. Intuitively, this follows from the fact CRS searches for a single 1-fixed NE, which can be found by reasoning about a nondeterministic automaton for $\psi_1$. Specifically, the reduction from LTL synthesis to LTL rational synthesis (Theorem 2 in [24]) argues that $\psi$ is realizable (in the traditional sense) iff there is a CRS solution to the system player in a game with objectives $\psi$, for the system player, and `True`, for the environment player. This is, however, wrong, as the environment player has no incentive to deviate from any profile. Consequently, every profile in which $\psi$ is satisfied in a 1-fixed NE. $\square$

## 7.2 Solving non-cooperative rational synthesis

Following Remark 7.1, a strategy $f_1$ for Player 1 is an NRS solution if for every 1-fixed NE $\pi = \langle f_1, f_2\rangle$, we have that $1 \in \mathsf{Win}(\pi)$. Searching for strategies that are NRS solutions for Player 1, we distinguish, for every candidate strategy $f_1$ of Player 1, between the case there is a strategy $f_2$ for Player 2 such that $2 \in \mathsf{Win}(\langle f_1, f_2 \rangle)$ (Lemma 7.4), and the case no such strategy exists (Lemma 7.5).

**Lemma 7.4** *Consider a strategy $f_1$ for Player 1. If there is a strategy $f_2$ for Player 2 such that $2 \in \mathsf{Win}(\langle f_1, f_2 \rangle)$, then $f_1$ is an NRS solution iff $f_1$ is a winning strategy for Player* OR *in the zero-sum game $\langle G, L_1 \cup \overline{L_2} \rangle$.*

**Proof:** Assume first that $f_1$ is an NRS solution, and consider a strategy $f_2'$ for Player 2. If the profile $\langle f_1, f_2' \rangle$ is a 1-fixed NE, then as $f_1$ is an NRS solution, we have that $\tau(\mathsf{Outcome}(f_1, f_2')) \in L_1$. If the profile $\langle f_1, f_2' \rangle$ is not a 1-fixed NE, it implies that Player 2 has an inventive to deviate, and so $\tau(\mathsf{Outcome}(f_1, f_2')) \in \overline{L_2}$. It follows that for every strategy $f_2'$ for Player 2, we have that $\tau(\mathsf{Outcome}(f_1, f_2')) \in L_1$ or $\tau(\mathsf{Outcome}(f_1, f_2')) \in \overline{L_2}$. Hence, $f_1$ is a winning strategy for Player OR in the zero-sum game $\langle G, L_1 \cup \overline{L_2} \rangle$.

For the other direction, assume that $f_1$ is a winning strategy for Player OR in the zero-sum game $\langle G, L_1 \cup \overline{L_2} \rangle$. Then, for every strategy $f_2'$, we have that either $\tau(\mathsf{Outcome}(f_1, f_2')) \in L_1$ or $\tau(\mathsf{Outcome}(f_1, f_2')) \in \overline{L_2}$. If $\tau(\mathsf{Outcome}(f_1, f_2')) \in \overline{L_2}$, then the profile $\langle f_1, f_2' \rangle$ is not a 1-fixed NE, as Player 2 can deviate to $f_2$. Hence, every profile $\pi = \langle f_1, f_2' \rangle$ in which $L_1$ is not fulfilled is not an 1-fixed NE. Thus, $f_1$ is an NRS solution. $\square$

**Lemma 7.5** *Consider a strategy $f_1$ for Player 1. If $2 \in \mathsf{Lose}(\langle f_1, f_2 \rangle)$ for every strategy $f_2$ of Player 2, then $f_1$ is an NRS solution iff $f_1$ is a winning strategy for Player* OR *in the zero-sum game $\langle G, L_1 \rangle$.*

**Proof:** Since $2 \in \mathsf{Lose}(\langle f_1, f_2 \rangle)$ for every strategy $f_2$ of Player 2, then every profile $\langle f_1, f_2 \rangle$ is a 1-fixed NE. Hence, $f_1$ is an NRS solution iff $1 \in \mathsf{Win}(\langle f_1, f_2 \rangle)$ for every strategy $f_2$ for Player 2, which holds iff $f_1$ is a winning strategy for Player OR in the zero-sum game $\langle G, L_1 \rangle$. $\square$

**Theorem 7.6** *Finding an NRS solution for Player 1 can be done in time polynomial in $|G|$ and doubly-exponential in $|\psi_1|$ and $|\psi_2|$.*

**Proof:** We start by constructing an NBT $\mathcal{N}$ that is not empty iff there is a strategy $f_1$ such that $2 \in \mathsf{Win}(\langle f_1, f_2 \rangle)$ for some strategy $f_2$ for Player 2, and $f_1$ is a winning strategy for Player OR in the zero-sum game $\langle G, L_1 \cup \overline{L_2} \rangle$.

Next, we construct an NBT $\mathcal{N}'$ that is not empty iff there is a strategy $f_1$ such that $2 \in \mathsf{Lose}(\langle f_1, f_2 \rangle)$ for every strategy $f_2$ for Player 2, and $f_1$ is a winning strategy for Player OR in the zero-sum game $\langle G, L_1 \rangle$.

By Lemmas 7.4 and 7.5, there is an NRS solution for Player 1 iff the NBTs $\mathcal{N}$ or $\mathcal{N}'$ are not empty. We show that $\mathcal{N}$ and $\mathcal{N}'$ are of size polynomial in $|G|$ and doubly-exponential in $|\psi_1|$ and $|\psi_2|$. The complexity then follows from the polynomial nonemptiness check for NBTs [39].

First, we define $\mathcal{N}$ as the intersection of an NBT $\mathcal{A}_1$ that accepts a $V$-labeled $V_1^*$-tree $\langle V_1^*, f_1 \rangle$ iff there is a strategy $f_2$ for Player 2 such that $2 \in \mathsf{Win}(\langle f_1, f_2 \rangle)$, and an NBT $A'_2$ that is sufficiently equivalent to a UCT $\mathcal{A}_2$ that accepts a $V$-labeled $V_1^*$-tree $\langle V_1^*, f_1\rangle$ iff it is a winning strategy for Player OR in the zero-sum game $\langle G, L_1 \cup \overline{L_2}\rangle$. The NBTs $\mathcal{A}_1$ and $A'_2$ are defined in a similar manner to the automata described in the proof of Lemma 6.5, with the appropriate objectives. Accordingly, the size of $\mathcal{A}_1$ is polynomial in $|G|$ and exponential in $|\psi_2|$, and the size of $\mathcal{A}'_2$ is polynomial in $|G|$, exponential in $|\psi_2|$, and doubly-exponential in $|\psi_1 \vee \neg\psi_2|$.

If $\mathcal{N}$ is empty, we proceed to check the nonemptiness of $\mathcal{N}'$. We start by defining a UCT $\mathcal{U}$ as the intersection of the UCT $\tilde{A}_1$ that accepts a $V$-labeled $V_1^*$-tree $\langle V_1^*, f_1\rangle$ iff $2 \in \mathsf{Lose}(\langle f_1, f_2\rangle)$ for every strategy $f_2$ for Player 2, and the UCT $\mathcal{A}_3$ that accepts a $V$-labeled $V_1^*$-tree $\langle V_1^*, f_1\rangle$ iff it is a winning strategy for Player OR in the zero-sum game $\langle G, L_1\rangle$. The size of $\tilde{A}_1$ is polynomial in $|G|$ and exponential in $|\psi_2|$, and the size of $\mathcal{A}_3$ is polynomial in $|G|$ and exponential in $|\psi_1|$. The NBT $\mathcal{N}'$ is then obtained from $\mathcal{U}$ by applying the construction in [25], using the fact that $\tilde{\mathcal{A}}_1$ and $\mathcal{A}_3$ are deterministic in their $V$-component. The size of $\mathcal{N}'$ is then polynomial in $|G|$ and doubly exponential in $|\psi_1|$ and $|\psi_2|$. $\square$

# 8 Multi-valued Objectives

The linear temporal logic LTL[$\mathcal{F}$], introduced in [3], 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.

Let $AP$ be a set of Boolean atomic propositions, and let $\mathcal{F} \subseteq \{g : [0,1]^m \to [0,1] | m \in \mathbb{N}\}$ 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 $g \in \mathcal{F}$.

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

It is not hard to prove, by induction on the structure of the formula, that for every $\varphi$, there exists a finite set $V(\varphi) \subseteq [0,1]$ of possible satisfaction values, such that for every computation $\rho$, it holds that $[\![\rho, \varphi]\!] \in V(\varphi)$ and $|V(\varphi)| = 2^{O(|\varphi|)}$ [3].

The logic LTL coincides with the logic LTL[$\mathcal{F}$] for $\mathcal{F}$ that corresponds to the usual Boolean operators. The novelty of 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 $\lambda$) average between the satisfaction values of $\varphi$ and $\psi$. This enables

| Formula | Satisfaction value |
|---|---|
| $[\![\rho, \mathtt{True}]\!]$ | 1 |
| $[\![\rho, \mathtt{False}]\!]$ | 0 |
| $[\![\rho, p]\!]$ | 1 if $p \in \rho_0$<br>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$[\mathcal{F}]$.

the quality of the system to be an interpolation of different aspects of it. As an example, consider the LTL$[\mathcal{F}]$ formula $G(req \to (grant \oplus_{\frac{2}{3}} X grant))$. The formula specifies the fact that we want requests to be granted immediately and the grant to hold for two transactions. When this always holds, the satisfaction value is $1 \oplus_{\frac{2}{3}} 1 = \frac{2}{3} + \frac{1}{3} = 1$. 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 $\frac{1}{3}$.

**Theorem 8.1** [3] *Let $\varphi$ be an* LTL$[\mathcal{F}]$ *formula and $V \subseteq [0, 1]$ be a predicate. There exists an NBW $\mathcal{A}_{\varphi,V}$ such that for every computation $\rho \in (2^{AP})^\omega$, it holds that $[\![\rho, \varphi]\!] \in V$ iff $\mathcal{A}_{\varphi,V}$ accepts $\rho$. Furthermore, $\mathcal{A}_{\varphi,V}$ has at most $2^{O(|\varphi|^2)}$ states.*

Consider a two-player non-zero-sum game $\mathcal{G} = \langle G, \varphi_1, \varphi_2 \rangle$, for LTL$[\mathcal{F}]$ objectives $\varphi_1$ and $\varphi_2$. For a profile $\pi = \langle f_1, f_2 \rangle$ and an LTL$[\mathcal{F}]$ formula $\varphi$, we use $[\![\pi, \varphi]\!]$ to denote $[\![\tau(\mathsf{Outcome}(\pi)), \varphi]\!]$, namely the satisfaction value of $\varphi$ when the players follow the strategies in $\pi$. A profile $\pi = \langle f_1, f_2 \rangle$ is an NE if for all $i \in \{1, 2\}$ and strategies $f'_i$ for Player $i$, we have that $[\![\pi[i \leftarrow f'_i], \varphi_i]\!] \leq [\![\pi, \varphi_i]\!]$. Thus, no player has a beneficial deviation – one that would increase the satisfaction value of her objective. Then, $\pi$ is a *1-fixed-NE* if Player 2 does not have a beneficial deviation in $\pi$.

**Definition 8.1 [Multi-Valued Rational Synthesis]** *Consider a two-player non-zero-sum perspective game $\mathcal{G} = \langle G, \varphi_1, \varphi_2 \rangle$, for* LTL$[\mathcal{F}]$ *formulas $\varphi_1$ and $\varphi_2$, and a desired satisfaction value $v_1$ for Player 1. We define the following two variants of multi-valued rational synthesis:*

- Cooperative rational synthesis *($v_1$-CRS), where the desired output is an 1-fixed-NE $\pi$ such that $[\![\pi, \varphi_1]\!] \geq v_1$.*
- Non-cooperative rational synthesis *($v_1$-NRS), where the desired output is a strategy $f_1$ for Player 1 such that there is a 1-fixed-NE $\pi = \langle f_1, f_2 \rangle$ with $[\![\pi, \varphi_1]\!] \geq v_1$, and for every 1-fixed-NE $\pi = \langle f_1, f_2 \rangle$, we have that $[\![\pi, \varphi_1]\!] \geq v_1$.*

As has been the case in the Boolean setting (see Remark 7.1), a strategy $f_1$ for Player 1 is a $v_1$-NRS solution iff for every 1-fixed-NE $\pi = \langle f_1, f_2 \rangle$, we have that $[\![\pi, \varphi_1]\!] \geq v_1$.

Note that we could have also defined multi-valued rational synthesis for $k \geq 3$ players. Since, however, LTL is a special case of LTL[$\mathcal{F}$], and, by Theorem 7.2, this setting is undecidable, we restrict attention to two-player games.

Before we solve multi-valued rational synthesis, we show that for zero-sum perspective games, the framework in [25] can be easily extended to LTL[$\mathcal{F}$] objectives. An *atomic objective* is a pair $\langle \varphi, V \rangle$ for an LTL[$\mathcal{F}$] formula $\varphi$ and a predicate $V \subseteq [0, 1]$. A computation $\rho$ satisfies an atomic objective $\langle \varphi, V \rangle$ iff $[\![\rho, \varphi]\!] \in V$. An objective is then a Boolean assertion $\theta$ of atomic objectives, with the expected semantics. For example, a computation $\rho$ satisfies the objective $\langle \varphi_1, [v_1, 1] \rangle \vee \langle \varphi_2, [0, v_2) \rangle$ if $[\![\rho, \varphi_1]\!] \geq v_1$ or $[\![\rho, \varphi_1]\!] < v_2$. We say that a strategy $f_1$ for Player OR is *P-winning* in a game $\mathcal{G} = \langle G, \theta \rangle$ if for every strategy $f_2$ for Player AND, we have that $\tau(\mathsf{Outcome}(f_1, f_2))$ satisfies $\theta$. By dualizing Theorem 8.1, we can construct, given an atomic objective $\langle \varphi, V \rangle$, a UCW $\mathcal{A}_{\varphi,V}$ that accepts exactly all computations $\rho \in (2^{AP})^\omega$ for which $[\![\rho, \varphi]\!] \in V$. For a Boolean assertion $\theta$ of atomic objectives, we can use closure properties of UCWs and construct the desired UCW. Since the algorithm in [25] is based on UCWs for the objectives, we have the following.

**Theorem 8.2** *Consider a perspective zero-sum multi-valued game $\mathcal{G} = \langle G, \theta \rangle$, where $\theta$ is a Boolean assertion of atomic objectives. Deciding whether Player* OR *has a winning P-strategy in $\mathcal{G}$, and finding a winning P-strategy, is 2EXPTIME-complete, and PTIME-complete in the size of the graph.*

## 8.1 Solving cooperative rational synthesis

**Theorem 8.3** *For a value $v_1 \in V(\varphi_1)$, we have that $v_1$-CRS can be solved in time polynomial in $|G|$, exponential in $|\varphi_1|$, and doubly-exponential in $|\varphi_2|$.*

**Proof:** By definition, a solution $\pi = \langle f_1, f_2 \rangle$ to $v_1$-CRS satisfies $[\![\pi, \varphi_1]\!] \geq v_1$ and for every $v_2 \in V(\varphi_2)$, if $[\![\pi, \varphi_2]\!] = v_2$, then for every strategy $f_2'$ for Player 2, we have that $[\![\langle f_1, f_2' \rangle, \varphi_2]\!] \leq v_2$. In other words, there is $v_2 \in V(\varphi_2)$ such that $[\![\pi, \varphi_1]\!] \geq v_1$, $[\![\pi, \varphi_2]\!] = v_2$, and for every strategy $f_2'$ for Player 2, we have that $[\![\langle f_1, f_2' \rangle, \varphi_2]\!] \leq v_2$. Accordingly, we search for a $v_1$-CRS solution by going over all values $v_2 \in V(\varphi_2)$ and intersecting two NBTs: an NBT $\mathcal{A}_1$, which accepts a strategy $f_1$ for Player 1 iff there is a strategy $f_2$ for Player 2 such that $[\![\langle f_1, f_2 \rangle, \varphi_1]\!] \geq v_1$ and $[\![\langle f_1, f_2 \rangle, \varphi_2]\!] = v_2$, and an NBT $\mathcal{A}_2$, which accepts a strategy $f_1$ for Player 1 iff for all strategies $f_2$ for Player 2, we have $[\![\langle f_1, f_2' \rangle, \varphi_2]\!] \leq v_2$. Both NBTs can be constructed as in the proofs of Theorem 6.6, using the Theorem 8.1 for constructing the corresponding UCWs. In particular, the NBT $\mathcal{A}_1$ should reject a strategy $f_1$ if for all strategies $f_2$, their outcome is not in the language of an NBW that is the intersection of $\mathcal{A}_{\varphi_1,[v_1,1]}$ and $\mathcal{A}_{\varphi_2,[v_2,v_2]}$. If there is a value $v_2 \in V(\varphi_2)$ for which the intersection is not empty, we construct from it the desired 1-fixed NE as described in Theorem 7.3. $\square$

## 8.2 Solving non-cooperative rational synthesis

Recall that a strategy $f_1$ for Player 1 is an NRS solution if for every 1-fixed-NE $\pi = \langle f_1, f_2\rangle$, we have that $[\![\pi, \varphi_1]\!] \geq v_1$. Equivalently, for every $v_2 \in V(\varphi_2)$, if $[\![\langle f_1, f_2\rangle, \varphi_2]\!] = v_2$ and every strategy $f_2'$ for Player 2, we have that $[\![\langle f_1, f_2\rangle, \varphi_2]\!] \leq v_2$, then $[\![\langle f_1, f_2\rangle, \varphi_1]\!] \geq v_1$.

**Lemma 8.4** *Consider a strategy $f_1$ for Player 1. Let $v_2$ be the maximal value such that there is a strategy $f_2$ for Player 2 such that $[\![\langle f_1, f_2\rangle, \varphi_2]\!] = v_2$. Then, $f_1$ is a $v_1$-NRS solution iff $f_1$ is a winning strategy for Player* OR *in the zero-sum game $\langle G, \langle \varphi_1, [v_1, 1]\rangle \vee \langle \varphi_2, [0, v_2)\rangle\rangle$.*

**Proof:** Assume first that $f_1$ is a $v_1$-NRS solution, and consider a strategy $f_2'$ for Player 2. If the profile $\langle f_1, f_2'\rangle$ is a 1-fixed NE, then as $f_1$ is a $v_1$-NRS solution, we have that $[\![\langle f_1, f_2\rangle, \varphi_1]\!] \geq v_1$. If the profile $\langle f_1, f_2'\rangle$ is not a 1-fixed NE, it implies that Player 2 has an incentive to deviate, and so $[\![\langle f_1, f_2\rangle, \varphi_2]\!] < v_2$. It follows that for every strategy $f_2'$ for Player 2, we have that $[\![\langle f_1, f_2\rangle, \varphi_1]\!] \geq v_1$ or $[\![\langle f_1, f_2\rangle, \varphi_2]\!] < v_2$. Hence, $f_1$ is a winning strategy for Player OR in the zero-sum game $\langle G, \langle \varphi_1, [v_1, 1]\rangle \vee \langle \varphi_2, [0, v_2)\rangle\rangle$.

Assume now that $f_1$ is a winning strategy for Player OR in the zero-sum game $\langle G, \langle \varphi_1, [v_1, 1]\rangle \vee \langle \varphi_2, [0, v_2)\rangle\rangle$. Then, for every strategy $f_2'$, we have that $[\![\langle f_1, f_2\rangle, \varphi_1]\!] \geq v_1$ or $[\![\langle f_1, f_2\rangle, \varphi_2]\!] < v_2$. If $[\![\langle f_1, f_2\rangle, \varphi_2]\!] < v_2$, then the profile $\langle f_1, f_2'\rangle$ is not a 1-fixed NE, as Player 2 can deviate to $f_2$. Hence, every profile $\pi = \langle f_1, f_2'\rangle$ in which $[\![\langle f_1, f_2\rangle, \varphi_1]\!] < v_1$ is not a 1-fixed NE. Thus, $f_1$ is a $v_1$-NRS solution. $\square$

Note that Lemmas 7.4 and 7.5 can be viewed as a special case of Lemma 8.4, with $\mathcal{F}$ that includes only the Boolean operators. Then, the only possible satisfaction values of $\varphi_1$ and $\varphi_2$ are 0 and 1.

**Theorem 8.5** *For a value $v_1 \in V(\varphi_1)$, we have that $v_1$-NRS for* LTL[$\mathcal{F}$] *objectives can be solved in time polynomial in $|G|$ and doubly-exponential in $|\varphi_1|$ and $|\varphi_2|$.*

**Proof:** We search for a $v_1$-NRS solution by going over all values $v_2 \in V(\varphi_2)$ and intersecting two NBTs. The first is an NBT $\mathcal{A}_1$, identical to the intersection NBT described in the proof of Theorem 8.3, which accepts a strategy $f_1$ for Player 1 iff there is a strategy $f_2$ for Player 2 such that $[\![\langle f_1, f_2\rangle, \varphi_1]\!] \geq v_1$ and $[\![\langle f_1, f_2\rangle, \varphi_2]\!] = v_2$, and for all strategies $f_2$ for Player 2, we have that $[\![\langle f_1, f_2\rangle, \varphi_2]\!] \leq v_2$. The second, is an NBT $\mathcal{A}_2$ that accepts a strategy $f_1$ for Player 1 iff it is a winning strategy for Player OR in the zero-sum game $\langle G, \langle \varphi_1, [v_1, 1]\rangle \vee \langle \varphi_2, [0, v_2)\rangle\rangle$. Equivalently, if for all strategies $f_2$ for Player 2, we have that $[\![\langle f_1, f_2\rangle, \varphi_1]\!] \geq v_1$ or $[\![\langle f_1, f_2\rangle, \varphi_2]\!] < v_2$. By Lemma 8.4, the intersection accepts a strategy $f_1$ for Player 1 iff it is a $v_1$-NRS solution. If there is a value $v_2 \in V(\varphi_2)$ for which the intersection is not empty, we construct from it the desired strategy for Player 1. $\square$

**Acknowledgments** We thanks Gal Vardi for helpful discussions.

## References

[1] S. Agarwal, M. S. Kodialam, and T. V. Lakshman. Traffic engineering in software defined networks. In *Proc. 32nd IEEE International Conference on Computer Communications*, pages 2211–2219, 2013.

[2] S. Almagor, G. Avni, and O. Kupferman. Repairing multi-player games. In *Proc. 26th Int. Conf. on Concurrency Theory*, *LIPIcs* 42, pages 325–339, 2015.

[3] S. Almagor, U. Boker, and O. Kupferman. Formalizing and reasoning about quality. *Journal of the ACM*, 63(3), 2016.

[4] S. Almagor, O. Kupferman, and G. Perelli. Synthesis of controllable Nash equilibria in quantitative objective game. In *Proc. 27th Int. Joint Conf. on Artificial Intelligence*, pages 35–41, 2018.

[5] R. Alur, T.A. Henzinger, and O. Kupferman. Alternating-time temporal logic. *Journal of the ACM*, 49(5):672–713, 2002.

[6] L. Amgoud. An argumentation-based model for reasoning about coalition structures. In *2nd International Workshop on Argumentation in Multi-Agent Systems*, LNCS 4049, pages 217–228. Springer, 2005.

[7] B. Aminof, M. Kwiatkowska, B. Maubert, A. Murano, and S. Rubin. Probabilistic strategy logic. In *Proc. 28th Int. Joint Conf. on Artificial Intelligence*, pages 32–38, 2019.

[8] P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Nash equilibria in concurrent games with Büchi objectives. In *Proc. 31st Conf. on Foundations of Software Technology and Theoretical Computer Science*, *LIPIcs* 13, pages 375–386. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2011.

[9] P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Concurrent games with ordered objectives. In *Proc. 15th Int. Conf. on Foundations of Software Science and Computation Structures*, LNCS 7213, pages 301–315. Springer, 2012.

[10] P. Bouyer, R. Brenguier, N. Markey, and M. Ummels. Pure Nash equilibria in concurrent deterministic games. *Logical Methods in Computer Science*, 11(2), 2015.

[11] P. Bouyer, N. Markey, and S. Vester. Nash equilibria in symmetric graph games with partial observation. *Information and Computation*, 254:238–258, 2017.

[12] N. Bulling and J. Dix. Modelling and verifying coalitions using argumentation and ATL. *Inteligencia Artif.*, 14(46):45–73, 2010.

[13] K. Chatterjee and L. Doyen. The complexity of partial-observation parity games. In *Proc. 16th Int. Conf. on Logic for Programming Artificial Intelligence and Reasoning*, pages 1–14. Springer, 2010.

[14] K. Chatterjee, L. Doyen, T. A. Henzinger, and J-F. Raskin. Algorithms for $\omega$-regular games with imperfect information. In *Proc. 15th Annual Conf. of the European Association for Computer Science Logic*, LNCS 4207, pages 287–302, 2006.

[15] T. Chen and J. Lu. Probabilistic alternating-time temporal logic and model checking algorithm. In *Fourth International Conference on Fuzzy Systems and Knowledge Discovery*, volume 2, pages 35–39, 2007.

[16] R. Condurache, E. Filiot, R. Gentilini, and J.-F. Raskin. The complexity of rational synthesis. In *Proc. 43th Int. Colloq. on Automata, Languages, and Programming*, *LIPIcs* 55, pages 121:1–121:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[17] C. Dima and F.L. Tiplea. Model-checking ATL under imperfect information and perfect recall semantics is undecidable. *CoRR*, abs/1102.4225, 2011.

[18] E.A. Emerson. Automata, tableaux, and temporal logics. In *Proc. Workshop on Logic of Programs*, LNCS 193, pages 79–87. Springer, 1985.

[19] E. Filiot, R. Gentilini, and J.-F. Raskin. Rational synthesis under imperfect information. In *Proc. 33rd IEEE Symp. on Logic in Computer Science*, pages 422–431. ACM, 2018.

[20] D. Fisman and O. Kupferman. Reasoning about finite-state switched systems. In *5th International Haifa Verification Conference*, LNCS 6405, pages 71–86. Springer, 2009.

[21] D. Fisman, O. Kupferman, and Y. Lustig. Rational synthesis. In *Proc. 16th Int. Conf. on Tools and Algorithms for the Construction and Analysis of Systems*, LNCS 6015, pages 190–204. Springer, 2010.

[22] J. Gutierrez, G. Perelli, and M. J. Wooldridge. Imperfect information in reactive modules games. *Inf. Comput.*, 261:650–675, 2018.

[23] O. Kupferman. Games with full, longitudinal, and transverse observability. In *14th International Conference on Reachability Problems*, LNCS 12448, pages 20–34. Springer, 2020.

[24] O. Kupferman, G. Perelli, and M.Y. Vardi. Synthesis with rational environments. *Annals of Mathematics and Artificial Intelligence*, 78(1):3–20, 2016.

[25] O. Kupferman and G. Vardi. Perspective games. In *Proc. 34th IEEE Symp. on Logic in Computer Science*, pages 1 – 13, 2019.

[26] O. Kupferman and M.Y. Vardi. Synthesis with incomplete information. In *Advances in Temporal Logic*, pages 109–127. Kluwer Academic Publishers, 2000.

[27] O. Kupferman and M.Y. Vardi. Safraless decision procedures. In *Proc. 46th IEEE Symp. on Foundations of Computer Science*, pages 531–540, 2005.

[28] D. Liberzon. *Switching in Systems and Control*. Birkhauser, 2003.

[29] Y. Lustig and M.Y. Vardi. Synthesis from component libraries. *Software Tools for Technology Transfer*, 15(5-6):603–618, 2013.

[30] M. Margaliot. Stability analysis of switched systems using variational principles: an introduction. *Automatica*, 42(12):2059–2077, 2006.

[31] D.E. Muller and P.E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of theorems of Rabin, McNaughton and Safra. *Theoretical Computer Science*, 141:69–107, 1995.

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

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

[34] A. Pnueli. The temporal semantics of concurrent programs. *Theoretical Computer Science*, 13:45–60, 1981.

[35] A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th ACM Symp. on Principles of Programming Languages*, pages 179–190, 1989.

[36] A. Pnueli and R. Rosner. On the synthesis of an asynchronous reactive module. In *Proc. 16th Int. Colloq. on Automata, Languages, and Programming*, LNCS 372, pages 652–671. Springer, 1989.

[37] B. Puchala. Asynchronous omega-regular games with partial information. In *35th Int. Symp. on Mathematical Foundations of Computer Science*, pages 592–603. Springer, 2010.

[38] J.H. Reif. The complexity of two-player games of incomplete information. *Journal of Computer and Systems Science*, 29:274–301, 1984.

[39] M.Y. Vardi and P. Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and Systems Science*, 32(2):182–221, 1986.

[40] M.Y. Vardi and P. Wolper. Reasoning about infinite computations. *Information and Computation*, 115(1):1–37, 1994.