Iterated Boolean games

Julian Gutierrez, Paul Harrenstein, Michael Wooldridge *

Department of Computer Science, University of Oxford, United Kingdom

A R T I C L E   I N F O

A B S T R A C T

Iterated games are well-known in the game theory literature. We study *iterated Boolean games*. These are games in which players repeatedly choose truth values for Boolean variables they have control over. Our model of iterated Boolean games assumes that players have goals given by formulae of Linear Temporal Logic (LTL), a formalism for expressing properties of state sequences. In order to represent the strategies of players in such games, we use a finite state machine model. After introducing and formally defining iterated Boolean games, we investigate the computational complexity of their associated game-theoretic decision problems, as well as semantic conditions characterising classes of LTL properties that are preserved by equilibrium points (pure-strategy Nash equilibria) whenever they exist.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Playing a game more than once against the same opponent can have a dramatic effect on which outcomes of the game can be sustained as equilibria [22, pp. 133–161]. To take a classic example in the literature, in the one-shot Prisoner's Dilemma there is a unique pure Nash equilibrium in which both players defect, leading to payoffs that are worse for both players than the payoffs they would have obtained had they cooperated. However, if instead the same players repeatedly meet each other then cooperation can be rationally sustained—leading to equilibria (outcomes) which can be better than those of the one-shot version of the game. Cooperation is rationally sustainable because the players will meet in the future, and will thereby have the opportunity to *punish* each other for non-cooperation.

We study iterated versions of *Boolean games*. The basic idea of a Boolean game [16] is that each player i is associated with a goal, represented as a logical formula γ_i, and player i's main purpose is to ensure that γ_i is satisfied. The strategies and choices for each player i are defined with respect to a set of Boolean variables Φ_i, drawn from an overall set of variables Φ. Player i is assumed to have unique control over the variables in Φ_i, in that it can assign truth values to these variables in any way it chooses. Strategic concerns arise in Boolean games as the satisfaction of player i's goal γ_i can depend on the variables controlled by other players.

In the version of Boolean games that we study, it is assumed that players interact over an *infinite* series of rounds, where at each round each player makes an assignment to the variables under its control. Goals are expressed as formulae of *Linear Temporal Logic* (LTL), a well-known formalism for expressing properties of distributed and concurrent systems [9,20,21]. Formulae of LTL are essentially predicates over infinite sequences of states. Thus, whether a player's goal is or is not satisfied may depend not just on how players act in one round, but how they act in all future rounds.

---

* Corresponding author.
  E-mail address: Michael.Wooldridge@cs.ox.ac.uk (M. Wooldridge).

Players in Boolean games can be understood as non-deterministic computer programs, and the model thus has great relevance to concurrent and multi-agent systems research. As players can model computer programs, building strategies in Boolean games corresponds to synthesising computer systems from their logical specifications, for instance as the modules for control and synchronisation of concurrent processes [18,23,32].

The paper contains the following main contributions:

- Firstly, it formalises iterated Boolean games. Moreover, it provides a finite state machine representation (an *operational* model) along with a temporal logic theory (a *denotational* model) for strategies and players. This dual operational/denotational model for strategies and players in iterated Boolean games allows us to investigate computational properties regarding the games, such as, for instance, whether a collection of strategies forms a pure-strategy Nash equilibrium.
- Secondly, we study the complexity of various decision problems. Model checking for iterated Boolean games is in PSPACE and thus not harder than the model checking problem with respect to LTL specifications [27]. Synthesis is 2EXPTIME-complete, matching the complexity of the synthesis problem for LTL specifications [24]. Moreover, checking whether a given strategy profile forms a pure-strategy Nash equilibrium and whether an iterated Boolean game has at least one Nash equilibrium are PSPACE-complete and 2EXPTIME-complete problems, respectively. We also study the complexity of checking the existence of equilibria for two other solution concepts, namely, dominant strategies and a refinement of subgame perfect Nash equilibrium. In both cases, the complexity of solving these problems is also 2EXPTIME-complete.
- Thirdly, we give *Nash Folk Theorems* for iterated Boolean games.[1] These theorems provide semantic characterisations of LTL properties that are satisfied in equilibrium outcomes of a game. Some of these Folk Theorems completely characterise games for which some questions can be answered more efficiently; in particular, in some cases synthesis can be done in PSPACE, instead of in 2EXPTIME as in the general case.

Note that our main interest is in the use of game theoretic techniques for the analysis of multi-process computer systems, but the present paper may also be of some minor interest to the game theory community concerned with iterated (repeated) games, in the following sense. Iterated games have been widely studied in the game theory literature, and it is by now commonplace to model strategies for players in infinite games using finite state machines (see, *e.g.*, [3, p. 376]). Now, systems consisting of interacting finite state machines have also been widely-studied within computer science, and a substantial literature has arisen developing techniques for the analysis of such systems. One of the key problems considered within this community is that of how to reason about the infinite computational traces generated by such systems. To this end, temporal logic has been widely promoted as a language for reasoning about such paths. In particular, Linear Temporal Logic, the language that we use in the present paper, has theoretical properties that make it very attractive for this purpose [9]. We believe that there may be some benefit by using such languages in the analysis of infinitely repeated games, and that the present paper gives some indication of how they might be so used.

Finally, in order to improve the readability of the paper, longer proofs have been omitted from the main text and held over to Appendix A.

2. Preliminaries

Propositional logic   Let $\Phi = \{p, q, \ldots\}$ be a finite, fixed, and non-empty vocabulary of Boolean variables. We let L_0 denote the set of formulae of classical propositional logic constructed over the set of Boolean variables Φ using the connectives "$\wedge$" (and), "$\vee$" (or), "$\neg$" (not), "$\rightarrow$" (implies), "$\leftrightarrow$" (iff), "$\top$" (truth), and "$\bot$" (falsity). By $vars(\varphi)$ we denote the propositional variables that occur in φ.

A *valuation* is a subset of propositional variables, that is, $v \subseteq \Phi$, where it is understood that v assigns truth to all the variables it contains and falsity to the others. Let $V(\Phi)$ denote the set of valuations for Φ, *i.e.*, $V(\Phi) = 2^{\Phi}$. Whenever Φ is clear from the context, we also write V for $V(\Phi)$. For $\varphi \in L_0$ we use $v \models \varphi$ to indicate that v *satisfies* φ (and say that φ *holds at* v).

For $\Psi \subseteq \Phi$ and $v \in V(\Phi)$, we define the formula χ_v^{Ψ} in $L(\Phi)$ that uniquely characterises the valuation $v \cap \Psi$, that is, the valuation v as restricted to Ψ:

$$\chi_v^{\Psi} = \bigwedge_{p \in \Psi \cap v} p \wedge \bigwedge_{p \in \Psi \setminus v} \neg p.$$

The superscript Ψ is omitted if $\Psi = \Phi$. Thus, for valuations $v, w \in V(\Phi)$ and $\Psi \subseteq \Phi$ we have,

$$v \models \chi_w^{\Psi} \quad \text{if and only if} \quad v \cap \Psi = w \cap \Psi.$$

---

[1] As the name already suggests, Folk Theorems for repeated or iterated games have long been part of the folklore of game theory, without there being a canonical reference for them. Nowadays, Aumann and Shapley [1] are commonly credited as the originators.

Linear Temporal Logic (LTL) We use the well-known framework of LTL to express properties of plays of iterated Boolean games [9,20,21]. LTL extends classical propositional logic with modal tense operators for expressing properties of infinite sequences. Specifically, LTL extends classical propositional logic with the unary modal operators $\mathbf{X}$ ("next"), $\mathbf{F}$ ("eventually"), $\mathbf{G}$ ("always"), and the binary modal operator $\mathbf{U}$ ("until").

We take $\mathbf{X}$ and $\mathbf{U}$ as our basic operators, and define the remaining LTL connectives in terms of these. For our propositional basis, we use the operators "∨" and "¬", and assume the remaining classical operators are defined in terms of these in the standard way.

Formally, the syntax of LTL is defined with respect to a set Φ of Boolean variables as follows:

$$\varphi ::= p \mid \neg \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\,\varphi \mid \varphi\,\mathbf{U}\,\varphi$$

where $p \in \Phi$. The remaining classical and LTL connectives are then defined in the standard way; in particular, we have $\mathbf{F}\varphi = \top\,\mathbf{U}\,\varphi$, and $\mathbf{G}\varphi = \neg\mathbf{F}\neg\varphi$. Given a set of variables Ψ, let $L(\Psi)$ be the set of LTL formulae over Ψ.

A core concept in the LTL semantics is that of a *run*. A run $\rho \colon \mathbb{N} \to V(\Phi)$ is a function that assigns a valuation to every time point $t \in \mathbb{N}$, indicating which variables are true at that time point. Using square brackets around parameters referring to time points, we will let $\rho[t]$ denote the valuation assigned to time point t by ρ. For $\Psi \subseteq \Phi$ also use $\rho|_\Psi$ to refer to the run ρ restricted to subsets of variables in Ψ, that is, for all time points t,

$$\rho|_\Psi[t] = \rho[t] \cap \Psi.$$

We interpret formulae of LTL with respect to pairs (ρ, t) where ρ is a run and $t \in \mathbb{N}$ is a temporal index into ρ. Formally, the semantics of LTL formulae is as follows:

$$
\begin{aligned}
(\rho, t) &\models p & &\text{iff} & &p \in \rho[t] \\
(\rho, t) &\models \neg\varphi & &\text{iff} & &\text{it is not the case that } (\rho, t) \models \varphi \\
(\rho, t) &\models \varphi \vee \psi & &\text{iff} & &(\rho, t) \models \varphi \text{ or } (\rho, t) \models \psi \\
(\rho, t) &\models \mathbf{X}\,\varphi & &\text{iff} & &(\rho, t+1) \models \varphi \\
(\rho, t) &\models \varphi\,\mathbf{U}\,\psi & &\text{iff} & &\text{for some } t' \geq t \colon ((\rho, t') \models \psi \text{ and for all } t \leq t'' < t' \colon (\rho, t'') \models \varphi).
\end{aligned}
$$

If $(\rho, 0) \models \varphi$, we also write $\rho \models \varphi$ and say that ρ *satisfies* φ. We say that φ and ψ are *equivalent* if for all runs ρ we have $\rho \models \varphi$ if and only if $\rho \models \psi$. An LTL formula $\varphi \in L$ is *satisfiable* if there is some run satisfying φ.

Boolean games In a Boolean game each player i controls a subset of propositional variables Φ_i, meaning that i has the unique ability to choose the value (either true or false) of each variable in Φ_i. A *choice* for player i is a subset v_i of the propositional variables under its control, that is, $v_i \subseteq \Phi_i$. A *choice vector* $\vec{v} = (v_1, \ldots, v_n)$ is a collection of choices, one for each player. When all players choose values for the variables they control, a valuation results. Each player i has a propositional formula γ_i as its goal and chooses values for the variables under its control with the aim of satisfying γ_i. The goal γ_i of player i, however, may contain variables controlled by other players j, who will also be choosing values for their variables in Φ_j so as to try to get their goals γ_j satisfied. The satisfaction of these formulae γ_j may in turn be dependent on the choice that player i makes for the variables in Φ_i.

Formally, a *Boolean game* is a tuple

$$(N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n),$$

where N is a set of n players, Φ is a set of propositional variables, and Φ_i is the subset of Φ that player i controls. Finally, γ_i is a formula in $L_0(\Phi)$ which represents player i's goal. We assume that $\Phi_1, \ldots, \Phi_n$ are pairwise disjoint and $\Phi_1 \cup \cdots \cup \Phi_n = \Phi$. A choice for player i is a valuation $v_i \subseteq \Phi_i$ for the variables under its control. We denote the set of player i's choices by V_i.

Notice that each strategy profile $\vec{v} = (v_1, \ldots, v_n)$ trivially induces a unique valuation for the variables Φ, and vice versa. We will frequently exploit this by treating strategy profiles as if they were valuations, and valuations as if they were strategy profiles, *e.g.*, by writing $p \in \vec{v}$ to denote that the truth value of p under valuation v is set to true.

The basic assumption in Boolean games is that players will strictly prefer outcomes that satisfy their goal over those that do not, but are indifferent between outcomes that satisfy their goal, and indifferent between outcomes that do not satisfy their goal. We capture these preferences in relations $\succsim_i \subseteq V \times V$ for each player i. Thus, each preference relation $\succsim_i$ defines dichotomous preferences over the valuations in the following sense: player i strictly prefers those valuations that satisfy γ_i over those valuations that do not, and is indifferent between outcomes otherwise. Thus we write $v \succsim_i w$ to mean that outcome $v \in V$ is (weakly) preferred over valuation $w \in V$. Formally, for each player i and all $v, w \in V$ we have,

$$v \succsim_i w \quad \text{if and only if} \quad w \models \gamma_i \text{ implies } v \models \gamma_i.$$

The irreflexive part of $\succsim_i$ is defined in the usual way: $v \succ_i w$ if and only if both $v \succsim_i w$ and not $w \succsim_i v$. Because valuations and outcomes are interchangeable, we are justified in writing $\vec{v}_1 \succsim_i \vec{v}_2$, meaning that the valuation induced by $\vec{v}_1$ is (weakly) preferred over the valuation induced by $\vec{v}_2$.

An outcome $\vec{v} = (v_1, \ldots, v_i, \ldots, v_n)$ is said to be a *pure strategy Nash equilibrium* if there is no player i and choice $v_i' \in V_i$ such that

$$(v_1, \ldots, v_i', \ldots, v_n) \succ_i (v_1, \ldots, v_i, \ldots, v_n).$$

In the remainder of the paper, we will simply write "Nash equilibrium" as a shorthand for "pure strategy Nash equilibrium."

3. Iterated Boolean games

Iterated Boolean games are an extension of Boolean games where players interact with each other for infinitely many rounds. As in the standard (one-shot or one-round) setting described before, there are n players each of whom uniquely controls a subset of Boolean variables and strives to get a particular goal formula γ_i satisfied. In an iterated Boolean game, however, players' goals γ_i are LTL formulae rather than propositional logic formulae. Such goals γ_i are interpreted on (infinite) runs ρ and the players' strategies are to determine a valuation $v_i[t]$ in V_i for every time point $t \in \mathbb{N}$. Here, the choice $v_i[t]$ may very well depend on both i's choice $v_i[t']$ and the other players' choices $v_j[t']$ at previous time points $t' < t$.

Formally, an *Iterated Boolean Game* (IBG) is a structure

$$G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n),$$

where $N = \{1, \ldots, n\}$ is a set of agents (the players), $\Phi = \{p, q, \ldots\}$ is a finite set of Boolean variables, $\Phi_i \subseteq \Phi$ is the set of Boolean variables under the unique control of player i, and $\gamma_i \in L$ is the LTL goal of player i. As for regular one-round Boolean games, the sets $\Phi_1, \ldots, \Phi_n$ are assumed to form a partition of Φ, *i.e.*, $\Phi_i \cap \Phi_j = \emptyset$ for all $i \neq j \in N$, and $\Phi = \Phi_1 \cup \cdots \cup \Phi_n$.

We now need to define strategies for iterated Boolean games, and based on that explain why strategy profiles induce runs, that is, infinite sequences of valuations.

Strategies   In iterated Boolean games, a strategy for player i is a function that makes a choice in each round of the game on the basis of the history of the game to date. Formally, we can understand such strategies as functions

$$f_i : V^* \to V_i$$

where V^* denotes the set of finite sequences over V. However, there is a difficulty with this representation from a computational perspective, as the domain of such a function is of infinite size. When considering computational problems associated with strategies, we therefore require a *finite* representation for strategies. In this paper, we model strategies as *deterministic finite state machines with output*. Such representations are widely used to study repeated games in the game theory literature [22, pp. 140–143], and are of course very natural from the point of view of computer science.

Formally, a *machine strategy* σ_i for player i in an iterated Boolean game $(N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$ is a structure:

$$\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$$

where Q_i is a finite and non-empty set of *states*, $q_i^0 \in Q_i$ is the *initial* state, $\delta_i : Q_i \times V \to Q_i$ is the *state transition function*, and $\tau_i : Q_i \to V_i$ is the *choice function*. Let Σ_i denote the class of machine strategies for player i. A *strategy profile* $\vec{\sigma}$ is an n-tuple of strategies, one for each player i, that is, $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$.

Runs induced by strategy profiles   Because strategies are deterministic, a strategy profile $\vec{\sigma}$ induces a *unique* run, which we will denote by $\rho(\vec{\sigma})$. To define $\rho(\vec{\sigma})$ formally, we need a little more notation. First, a *state vector* of a strategy profile is a tuple $\vec{q} = (q_1, \ldots, q_n)$ where for every $i \in N$, we have $q_i \in Q_i$. We denote the ith component of $\vec{q}$ by q_i. With each point of time t we associate a state vector denoted by $\vec{q}[t] = (q_1[t], \ldots, q_n[t])$ and a valuation denoted by $\vec{v}[t] = (v_1[t], \ldots, v_n[t])$. The *history*, $h(\vec{\sigma})$ of a strategy profile $\vec{\sigma}$ is an infinite sequence of interleaved state vectors and valuations

$$h(\vec{\sigma}) = \vec{q}[0] \xrightarrow{\vec{v}[0]} \vec{q}[1] \xrightarrow{\vec{v}[1]} \cdots$$

where, for $t = 0$,

$$\vec{q}[0] = (q_1^0, \ldots, q_n^0),$$

$$\vec{v}[0] = (\tau_1(q_1^0), \ldots, \tau_n(q_n^0)),$$

and for every $t \in \mathbb{N}$ with $t > 0$,

$$\vec{q}[t] = (\delta_1(q_1[t-1], \vec{v}[t-1]), \ldots, \delta_n(q_n[t-1], \vec{v}[t-1])), \text{ and}$$

$$\vec{v}[t] = (\tau_1(q_1[t]), \ldots, \tau_n(q_n[t])).$$

Let $\vec{\sigma}$ be a strategy profile and $h(\vec{\sigma})$ the history induced by $\vec{\sigma}$. Then, the run $\rho(\vec{\sigma})$ induced by $\vec{\sigma}$ satisfies $\rho(\vec{\sigma})[t] = \vec{v}[t]$ for every $t \in \mathbb{N}$ whenever $\vec{v}$ is the valuation vector associated with $h(\vec{\sigma})$, that is,

$$\rho(\vec{\sigma}) = \vec{v}[0], \vec{v}[1], \vec{v}[2], \ldots$$

Preferences and Nash equilibrium When every agent has made their choice in an iterated Boolean game, we obtain a strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, and hence a run $\rho(\vec{\sigma})$. The possible outcomes of an iterated Boolean game are thus the runs $\rho \colon \mathbb{N} \to V(\Phi)$. For each run ρ and each player i, it will be the case that either ρ satisfies the goal γ_i of i or not. In this way, we obtain preference relations $\succsim_i$ for each player. Formally, for all runs ρ, ρ',

$$\rho \succsim_i \rho' \quad \text{if and only if} \quad \rho' \models \gamma_i \text{ implies } \rho \models \gamma_i.$$

Given these relations, we can define the concept of Nash equilibrium for iterated Boolean games. We use the notation $(\vec{\sigma}_{-i}, \sigma_i')$ to denote the strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_i, \ldots, \sigma_n)$ where σ_i is replaced with σ_i', that is, $(\vec{\sigma}_{-i}, \sigma_i')$ denotes the strategy profile $(\sigma_1, \ldots, \sigma_i', \ldots, \sigma_n)$. Consequently, we use the notation $\rho(\vec{\sigma}_{-i}, \sigma_i')$ to refer to the run induced by such a strategy profile.

A strategy profile $\vec{\sigma}$ is then said to be a Nash equilibrium if for all players $i \in N$ and for all strategies $\sigma_i' \in \Sigma_i$ we have

$$\rho(\vec{\sigma}) \succsim_i \rho(\vec{\sigma}_{-i}, \sigma_i').$$

Let $NE(G)$ be the set of Nash equilibrium strategy profiles for a game G.

Due to the dichotomous character of the players' preferences, this means that $\vec{\sigma}$ is a Nash equilibrium if every player whose goal is not satisfied by $\rho(\vec{\sigma})$ cannot unilaterally deviate to get its goal achieved.

Observation 1. *A strategy profile $\vec{\sigma}$ in an iterated Boolean game is a Nash equilibrium if and only if for every player $i \in N$,*

$$\text{either } \rho(\vec{\sigma}) \models \gamma_i \text{ or } \rho(\vec{\sigma}_{-i}, \sigma_i') \not\models \gamma_i \text{ for every strategy } \sigma_i'.$$

Partial strategies It is implicit in the definition of machine strategies that δ_i is a total function, and so for any given state $q \in Q_i$ we must have $\delta_i(q, \vec{v})$ defined for all $2^{|\Phi|}$ valuations $\vec{v}$ in V, resulting in strategies that are invariably of size exponential in $|\Phi|$. To obtain a more compact representation, we may allow δ_i to be a *partial* function defined only on the relevant pairs $(q, \vec{v})$. This will relieve us of the need to explicitly define $\delta_i(q, \vec{v})$ for every valuation $\vec{v}$. Thus, a partial strategy σ_i may sometimes be polynomial in the size of Φ and faithfully represent a strategy σ_i' that is of size exponential in $|\Phi|$.

In order to obtain a full or complete machine strategy σ_i' from a partial one σ_i, we will give a construction that uses a *default state* q_i^d to which all pairs $(q, \vec{v})$—i.e., machine transitions—for which $\delta_i(q, \vec{v})$ is not defined are mapped. Note, however, that partial machine strategies may necessarily have to be exponential in $|\Phi|$, as full machine strategies are a particular case of the more general class of partial machine strategies. Henceforth, we write machine strategy to mean partial strategy, and full machine strategy to mean a machine strategy of size exponential in Φ. The construction of a full machine strategy from a partial one is given next.

A machine strategy $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$, where $\delta_i \colon Q_i \times V \to Q_i$ may be a partial function, *represents* the full machine strategy

$$\sigma_i' = (Q_i', q_i^{00}, \delta_i', \tau_i')$$

where, for some default state $q_i^d \notin Q_i$, we have $Q_i' = Q_i \cup \{q_i^d\}$, $q_i^{00} = q_i^0$, and where δ_i' and τ_i' are such that for all $q_i' \in Q_i'$ and $\vec{v} \in V$,

$$\delta_i'(q_i', \vec{v}) = \begin{cases} \delta_i(q_i', \vec{v}) & \text{if } \delta_i(q_i', \vec{v}) = q_i'' \text{ for some } q_i'' \in Q_i, \\ q_i^d & \text{otherwise.} \end{cases}$$

$$\tau_i'(q_i') = \begin{cases} \tau_i(q_i') & \text{if } q_i' \neq q_i^d, \\ \varnothing & \text{otherwise.} \end{cases}$$

We say a machine strategy $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$ is *complete* or *partial* depending on whether δ_i is complete or partial, respectively.

The run $\rho(\vec{\sigma})$ and history $h(\vec{\sigma})$ induced by a profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ of partial strategies are defined, respectively, by the run and history induced by the profile $\vec{\sigma}' = (\sigma_1', \ldots, \sigma_n')$ of complete strategies that $\vec{\sigma}$ represents, that is,

$$\rho(\vec{\sigma}) = \rho(\vec{\sigma}') \quad \text{and} \quad h(\vec{\sigma}) = h(\vec{\sigma}').$$

To illustrate the definitions given so far as well as the kind of strategic reasoning underlying iterated Boolean games, we now provide some examples.
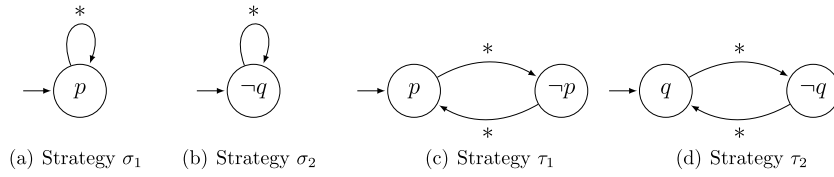
Fig. 1. Four machine strategies used in Example 1. An arc labelled with an asterisk indicates that on reading any input valuation, the machine moves to the indicated state.
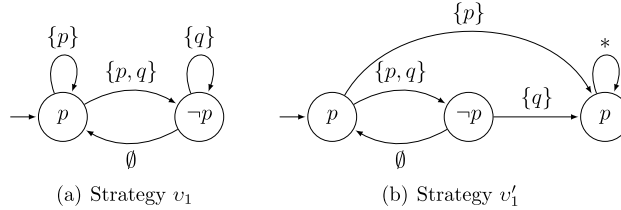

Fig. 2. Strategies υ and υ_1'.

Example 1. Consider the following iterated Boolean game:

$$G = (\{1, 2\}, \{p, q\}, \{p\}, \{q\}, \mathbf{GF}(p \wedge q), \mathbf{GF}(\neg p \wedge \neg q)).$$

In G there are two players, 1 and 2, the former controlling p and the latter q. Player 1, having $\mathbf{GF}(p \wedge q)$ as its goal strives to have both p and q to be true infinitely often, whereas player 2's goal is $\mathbf{GF}(\neg p \wedge \neg q)$ and would like to see both variables false infinitely often. Now consider the strategies σ_1 and σ_2 as depicted in Figs. 1(a) and 1(b), in which player 1 always sets p to true, no matter what player 2 does and player 2 always sets q to false, no matter what player 1 does, respectively. It might also *seem* that against any strategy of the other player, it can never hurt for player 1 to play σ_1 or for player 2 to play σ_2. The strategy profile (σ_1, σ_2) is in fact a Nash equilibrium, be it one in which no player achieves its goal. Yet, as will be seen below, σ_1 and σ_2 are not always optimal responses to every strategy of the other player.

If player 1 were to set p alternately to true and to false, as in strategy τ_1 in Fig. 1(c), and player 2 were to set q alternately to false and true, as in strategy τ_2 in Fig. 1(d), both players would achieve their goal. Accordingly, the strategy profile (τ_1, τ_2) is also an obvious Nash equilibrium in this game. Observe, however, that if player 2 were to play strategy τ_2' which is like τ_2, but now with player 2 starting with setting q to false instead, a run would result in which alternately the valuations $\{p\}$ and $\{q\}$ would be produced, leading to neither player's goal being satisfied. This, however, is not a Nash equilibrium. Player 1 could avert this situation by playing the slightly more sophisticated strategy υ_1, as depicted in Fig. 2(a): set p to true, until $\{p, q\}$ is realised, then set p to false until $\emptyset$ is realised, and so on. Both strategy profiles (υ_1, τ_2) and (υ_1, τ_2') are Nash equilibria and lead to runs that satisfy the goals of both players, namely,

$$\rho(\upsilon_1, \tau_2) = \{p, q\}, \emptyset, \{p, q\}, \emptyset, \ldots$$

and

$$\rho(\upsilon_1, \tau_2') = \{p\}, \{p, q\}, \emptyset, \{p, q\}, \emptyset, \ldots.$$

Finally, strategy υ', depicted in Fig. 2(b), incorporates a mechanism to deter player 2 from playing σ_2. It forces player 2 to play τ_2 (or a strategy displaying the same behaviour against υ_1') by *threatening* to set p to true forever after player 2's first deviation.

Example 2 *(Auctioning an item).* Consider the following auction involving two bidders and one auctioneer, in which one item is sold to the highest bidder. When the auction starts at time 0, the price is £0 and with every time unit the auctioneer raises the price by £1. A bidder bids by raising their hand at time t, indicating that he is prepared to pay £t for the item. The action terminates as soon as one of the bidders no longer places a bid. Then, the item is assigned to the unique bidder who did place a bid at the time the auction terminated. If no such bidder exists or if bidding goes on forever, then the item is left unsold.

We model this situation as an iterated Boolean game with three players: 1, 2, and 3. The former two are the bidders and each control one propositional variable, p and q, respectively, which they can set to true, indicating that they are bidding, or to false, indicating they do not bid, at each time. Player 3 represents the auctioneer and controls two propositional variables, p^* and q^*. The auctioneer sets p^* to true when the auction terminates and bidder 1 wins the auction. Similarly, q^* is set to true at termination of the action if bidder 2 wins the auction.

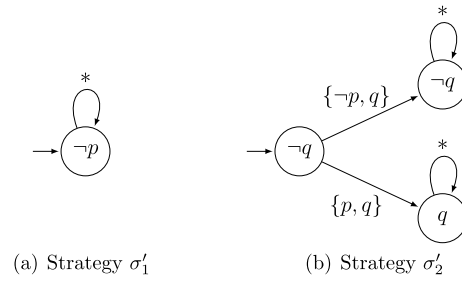(a) Strategy σ_1' (b) Strategy σ_2'

Fig. 3. Strategies for the bidders that are part of a Nash equilibrium in the auction setting described in Example 2.

The rules of the auction can thus be formalised as follows

$$last\_bid_1 = (p \wedge q)\,\mathbf{U}(p \wedge \neg q)$$

$$last\_bid_2 = (p \wedge q)\,\mathbf{U}(q \wedge \neg p)$$

$$no\_last\_bid = \neg last\_bid_1 \wedge \neg last\_bid_2$$

$$assign_1 = \mathbf{G}\neg q^* \wedge (\neg p^*\,\mathbf{U}(p \wedge \neg q \wedge p^*))$$

$$assign_2 = \mathbf{G}\neg p^* \wedge (\neg q^*\,\mathbf{U}(q \wedge \neg p \wedge q^*))$$

$$no\_assign = \mathbf{G}(\neg p^* \wedge \neg q^*).$$

We assume that the auctioneer is only interested in the auction being conducted in good order and, in particular, that the item goes to the highest bidder. Accordingly, the auctioneer's goal γ_3 is given by

$$\gamma_3 = (last\_bid_1 \rightarrow assign_1) \wedge (last\_bid_2 \rightarrow assign_2) \wedge (no\_last\_bid \rightarrow no\_assign).$$

Observe that the auctioneer can always respond to the strategies σ_1, σ_2 adopted by the two bidders by choosing a strategy σ_3' that guarantees γ_3 to be satisfied in $\rho(\sigma_1, \sigma_2, \sigma_3')$. Let σ_3^0 be the strategy for player 3 that outputs $\emptyset$ in every round. If t is the first round such that $\rho(\sigma_1, \sigma_2, \sigma_3^0)[t] \models p \wedge \neg q$ with $\rho(\sigma_1, \sigma_2, \sigma_3^0)[t'] \models p \wedge q$ for all $t' < t$, let σ_3' always output $\emptyset$ except for $\{p^*\}$ at round t. If t is the first round such that $\rho(\sigma_1, \sigma_2, \sigma_3^0)[t] \models \neg p \wedge q$ with $\rho(\sigma_1, \sigma_2, \sigma_3^0)[t'] \models p \wedge q$ for all $t' < t$, let σ_3' always output $\emptyset$ except for $\{q^*\}$ at round t. Otherwise, let $\sigma_3' = \sigma_3^0$. Thus, we may assume that the auctioneer's goal is satisfied in every Nash equilibrium.

Let the bidders have reservation prices (the highest prices they are willing to pay for the item) of £k and £m, respectively. For a bidder with reserve price £x this means that if that bidder submits the highest bid, the other bidder should have stopped bidding at or before the xth round. Let $\mathbf{X}^0\varphi = \varphi$ and $\mathbf{X}^{k+1}\varphi = \mathbf{X}(\mathbf{X}^k\varphi)$. Thus, define:

$$reserve_1^k = last\_bid_1 \rightarrow (\neg q \vee \mathbf{X}\neg q \vee \cdots \vee \mathbf{X}^k\neg q),$$

$$reserve_2^m = last\_bid_2 \rightarrow (\neg p \vee \mathbf{X}\neg p \vee \cdots \vee \mathbf{X}^m\neg p).$$

Now, the players' preferences can be formulated as follows:

$$\gamma_1 = \mathbf{F}\,p^* \wedge reserve_1^k \qquad \gamma_2 = \mathbf{F}\,q^* \wedge reserve_2^m.$$

Thus, each bidder's goal is to obtain the item for his reserve price or less.

There are several Nash equilibria in this game. In one of them, the player with the highest reservation price gets the item. This is, for instance, the case if bidder 1's strategy outputs p k times in a row and henceforth $\neg p$, irrespective of what bidder 2 does, whereas bidder 2 outputs q m times in a row and subsequently only $\neg q$. Assume $m < k$. Then bidder 1 gets the item for £$m + 1$. There is, however, also a Nash equilibrium in which bidder 2, who has the lower reservation price, obtains the item for £0. This is, for instance, the case for the strategies σ_1' and σ_2' as depicted in Fig. 3, where player 2 threatens to bid forever if player 1 dares to bid in the first round.

If, however, we assume the players not to be intimidated so easily and bestow them with a bidder's pride never to concede the auction without having bid their reservation price (but not more than that), better outcomes result in equilibrium. To model this situation, let

$$bidders\_pride_1^k = \neg last\_bid_1 \rightarrow (p \wedge \mathbf{X}\,p \wedge \cdots \wedge \mathbf{X}^k p \wedge \mathbf{X}^{k+1}\neg p)$$

$$bidders\_pride_2^m = \neg last\_bid_2 \rightarrow (q \wedge \mathbf{X}\,p \wedge \cdots \wedge \mathbf{X}^m q \wedge \mathbf{X}^{m+1}\neg q)$$

and reformulate the bidders' preferences as

$$\gamma_1' = reserve_1^k \wedge bidders\_pride_1^k \qquad \gamma_2' = reserve_2^m \wedge bidders\_pride_2^m.$$

Under these assumptions, both bidders can guarantee their goal to be satisfied by bidding exactly as long as their reservation price is high. It is not hard to verify that now in all Nash equilibria the bidder with the highest reservation price obtains the item for the other bidder's reservation price plus £1. If both bidders have the same reservation price, neither of them gets the item. Moreover, in each of these Nash equilibria all players' goals, including that of the auctioneer, are satisfied.

3.1. Properties of machine strategies

Strategies in our framework are deterministic finite state machines with output, also called "sequential transducers" or "Moore machines" in the control or computer science literature. Such strategies are not as powerful as strategies defined as unrestricted functions $f_i : V^* \to V_i$ from (finite) partial plays $\vec{v}[0], \vec{v}[1], \vec{v}[2], \ldots, \vec{v}[k]$ to partial plays $\vec{v}[0], \vec{v}[1], \vec{v}[2], \ldots, v_i[k+1]$. For instance, machine strategies do not have unbounded counting power; consequently, some runs cannot be generated by them. Take, *e.g.*, the run

$$\rho = abaabbaaabbbaaaabbbbaaaaabbbbb \ldots a^k b^k \ldots$$

which alternates finite and increasing sequences of a's and b's. This run cannot be generated by a machine strategy as an unbounded counting power within the strategy model would be needed.

However, because players' goals are LTL formulae, machine strategies are powerful enough in the following sense: if an LTL formula φ is satisfiable, then $\rho(\vec{\sigma}) \models \varphi$, for some strategy profile of machine strategies $\vec{\sigma}$. In other words, if we are interested in modelling players with LTL goals, then, in fact, all we need are machine strategies, loosely speaking, because they are powerful enough to achieve and represent any given LTL goal.

The following lemma formalises this.

Lemma 1. *For every satisfiable LTL formula $\varphi \in L(\Phi)$ there is a strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ such that $\rho(\vec{\sigma}) \models \varphi$.*

We will, in fact, prove a stronger version of this lemma: we will show that all we need for this result are *myopic* strategies, σ_i, which are oblivious to the behaviour of other strategies σ_j. Such strategies simply output a fixed infinite sequence of symbols over Φ_i.

Formally, a myopic strategy σ_i is a machine strategy $(Q_i, q_i^0, \delta_i, \tau_i)$ where

$$\delta_i(q, \vec{v}_1) = \delta_i(q, \vec{v}_2)$$

for all states $q \in Q_i$ and valuations $\vec{v}_1, \vec{v}_2 \in V$.

In order to construct a strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ only in myopic strategies σ_i we use the fact that any satisfiable LTL formula φ is satisfied by an ω-regular word, that is, by an infinite word of the form

$$\alpha \beta^\omega$$

where α is a finite word of size k and $\beta^\omega = \beta\beta\beta \ldots$ is an infinite word built from the concatenation of a finite word β of size p. Thus, if φ is satisfiable, then it is satisfied by some ω-regular word of this form.

The claim and construction is as follows.

Lemma 2. *For every satisfiable LTL formula $\varphi \in L(\Phi)$ there is a strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, where each σ_i is myopic, such that $\rho(\vec{\sigma}) \models \varphi$.*

Proof. As φ is satisfiable, then there is an ω-regular word $\rho = \alpha\beta\beta \ldots$ that satisfies φ. And since ρ is a word over $(2^\Phi)^\omega$, it can be written as the superposition/union of n words over $(2^{\Phi_1} \times \ldots \times 2^{\Phi_n})^\omega$, that is, a word whose projection ρ_i with respect to Φ_i is the ω-regular Φ_i-word

$$\rho_i = \alpha_i[0], \alpha_i[1], \ldots, \alpha_i[k-1], \beta_i[0], \beta_i[1], \ldots, \beta_i[p-1], \beta_i[0], \beta_i[1], \ldots$$

because of the definition of ρ.

We can then define the myopic machine strategy

$$\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$$

where

- $Q_i = \{q_i^s : 0 \le s < k+p\}$,
- $(q_i^s, v, q_i^t) \in \delta_i$ if either
  - $s + 1 = t$ or
  - $s = k + p - 1$ and $t = k$,
- $\tau_i(q_i^s) = \rho_i[s]$.

Note that the correctness of the construction—that in fact σ_i is a myopic machine strategy—lies in the fact that there is a unique q_i^t for each q_i^s. As a consequence, the valuation $\vec{v}$ in the transition function δ_i becomes irrelevant. It, then, immediately follows that $\rho(\vec{\sigma}) \models \varphi$. □

Myopic strategies are machines which simply "write down" an output word, and since any myopic strategy is simply a special type of finite state machine, such a word must be of the form $w = \alpha\beta\beta\beta\ldots$, for two finite words α and β of size k and p, respectively. In other words, any word w generated by a myopic strategy is ω-regular. Note, moreover, that the other direction holds too: every ω-regular word w can be generated by a myopic strategy σ. To see this, and using part of the proof of Lemma 2, simply let w be the run ρ_i and σ be the machine strategy σ_i, thus implying:

Corollary 1. *For every ω-regular word w over 2^Φ there is a strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, where each σ_i is myopic, such that $\rho(\vec{\sigma}) = w$.*

Another property that machine strategies enjoy, whether myopic or not, is extensionality. Machine strategies are *extensional* in the sense that, if two strategies σ_i' and σ_i'' yield the same valuation v_i against a strategy profile σ_{-i} of the other players at every time t, the profiles (σ_{-i}, σ_i') and $(\sigma_{-i}, \sigma_i'')$ produce the same run as well. Formally, this could be stated as

$$\rho(\vec{\sigma}_{-i}, \sigma_i')|_{\Phi_i} = \rho(\vec{\sigma}_{-i}, \sigma_i'')|_{\Phi_i} \quad \text{implies} \quad \rho(\vec{\sigma}_{-i}, \sigma_i') = \rho(\vec{\sigma}_{-i}, \sigma_i'').$$

The following lemma gives a slightly stronger statement (the proof may be found in Appendix A).

Lemma 3. *Let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ and $\vec{\sigma}' = (\sigma_1', \ldots, \sigma_n')$ be strategy profiles and i a player. Then,*

$$\rho(\vec{\sigma})|_{\Phi_i} = \rho(\vec{\sigma}_{-i}, \sigma_i')|_{\Phi_i} \text{ and } \rho(\vec{\sigma})|_{\Phi \setminus \Phi_i} = \rho(\vec{\sigma}'_{-i}, \sigma_i)|_{\Phi \setminus \Phi_i} \text{ imply } \rho(\vec{\sigma}) = \rho(\vec{\sigma}').$$

3.2. Logical characterisation of machine strategies

We now present a lemma which tells us that every strategy profile $\vec{\sigma}$ can be characterised by an LTL formula $TH(\vec{\sigma})$ in the sense that $\rho \models TH(\vec{\sigma})$ if and only if ρ is the unique run induced by the strategy profile $\vec{\sigma}$. A crucial point of the construction is that $TH(\vec{\sigma})$ is guaranteed to be of size at most polynomial in the size of $\vec{\sigma}$. A slight complication is that the formula $TH(\vec{\sigma})$ is in an LTL language $L(\Phi')$ on an extended set of propositional variables, $\Phi' \supseteq \Phi$. We show that for all runs $\rho: \mathbb{N} \to V(\Phi')$ that satisfy $TH(\vec{\sigma})$ it is the case that $\rho|_\Phi = \rho(\vec{\sigma})$.

Let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ be a profile of partial machine strategies $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$. Let, furthermore, $Q_i' = Q_i \cup \{q_i^d\}$, where q_i^d is the default state in the complete strategy that σ_i represents. For each player i and each state $q_i \in Q_i'$ introduce a new ("fresh") Boolean variable, which, as without loss of generality we assume that $\Phi, Q_1', \ldots, Q_n'$ are pairwise disjoint, we will also denote by q_i. Moreover, we use $Q' = Q_1' \cup \cdots \cup Q_n'$ to refer to the entire set of these new variables. For each player i, we define the formula

$$th(\sigma_i) = INIT(\sigma_i) \wedge TRANS(\sigma_i) \wedge INVAR(\sigma_i) \wedge VAL(\sigma_i),$$

where,

$$INIT(\sigma_i) = q_i^0,$$

$$TRANS(\sigma_i) = \mathbf{G}\left(\bigwedge_{\delta_i(q_i, \vec{v}) = q_i'} \left((\chi_{\vec{v}}^\Phi \wedge q_i) \to \mathbf{X} q_i' \right) \wedge \left(\left(\bigwedge_{\delta_i(q_i, \vec{v}) = q_i'} \neg(\chi_{\vec{v}}^\Phi \wedge q_i) \right) \to \mathbf{X} q_i^d \right) \right),$$

$$INVAR(\sigma_i) = \mathbf{G} \bigvee_{q_i \in Q_i \cup \{q_i^d\}} \left(q_i \wedge \bigwedge_{q_i' \neq q_i} \neg q_i' \right),$$

$$VAL(\sigma_i) = \mathbf{G}\left(\bigwedge_{q_i \in Q_i} \left(q_i \to \chi_{\tau(q_i)}^{\Phi_i} \right) \wedge (q_i^d \to \chi_\emptyset^{\Phi_i}) \right).$$

Finally, let

$$TH(\vec{\sigma}) = th(\sigma_1) \wedge \cdots \wedge th(\sigma_n).$$

Intuitively, $th(\sigma_i)$ is the LTL theory of strategy σ_i: the property $INIT(\sigma_i)$ encodes the initial state q_i^0, $TRANS(\sigma_i)$ encodes the transition relation δ_i, $INVAR(\sigma_i)$ ensures that the strategy is in at most one state at any given time, while $VAL(\sigma_i)$ encodes

the choice function τ_i. Note, in particular, that the size of $th(\vec{\sigma}_i)$ is polynomial in the size of $\vec{\sigma}_i$. Now, let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ be a profile of partial strategies and let the history $h(\vec{\sigma})$ be given by

$$\vec{q}[0] \xrightarrow{\vec{v}[0]} \vec{q}[1] \xrightarrow{\vec{v}[1]} \cdots.$$

Define the run $\hat{\rho}(\vec{\sigma}) : \mathbb{N} \to 2^{\Phi \cup Q \cup \{q_1^d, \ldots, q_n^d\}}$ where, for all time points t,

$$\hat{\rho}(\vec{\sigma})[t] = \rho(\vec{\sigma}) \cup \{q_1[t], \ldots, q_n[t]\}.$$

Thus, $\hat{\rho}(\vec{\sigma})$ is exactly like $\rho(\vec{\sigma})$, except that the former also specifies the current state of each of the players' strategies at each time point. Observe that, for all subsets $\Psi \subseteq \Phi$,

$$\hat{\rho}(\vec{\sigma})|_\Psi = \rho(\vec{\sigma})|_\Psi,$$

in particular, $\hat{\rho}(\vec{\sigma})|_\Phi = \rho(\vec{\sigma})$.

With the above logical characterisation of machine strategies in place, we can now show that only the run $\rho(\vec{\sigma})$ induced by a strategy profile $\vec{\sigma}$ satisfies the LTL formula given by its logical characterisation $TH(\vec{\sigma})$.

Lemma 4. *Let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ be a profile of (partial) strategies $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$ for an iterated Boolean game representing complete strategies with default states $q_1^d, \ldots, q_n^d$. Let, furthermore, $\rho : \mathbb{N} \to 2^{\Phi \cup Q \cup \{q_1^d, \ldots, q_n^d\}}$ an ω-regular run and $\vec{\sigma}' = (\sigma_1', \ldots, \sigma_n')$ a strategy profile such that $\rho|_\Phi = \rho(\vec{\sigma}')$. Then, for all players i,*

1. *$\rho \models th(\sigma_i)$ if and only if $\rho|_{\Phi \cup Q_i \cup \{q_i^d\}} = \hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)|_{\Phi \cup Q_i \cup \{q_i^d\}}$,*
2. *$\rho \models TH(\vec{\sigma})$ if and only if $\rho = \hat{\rho}(\vec{\sigma})$.*

The proof of the above lemma can be found in Appendix A.

Note that, while machine strategies provide an *operational* model of the behaviour/interactions of the players in an iterated Boolean game, the formula $th(\sigma_i)$ provides a *denotational* model for σ_i with respect to the semantics of LTL—and hence a denotational model of the behaviour of players.

4. Computational complexity

We now consider the complexity of problems relating to Nash equilibria in iterated Boolean games. We first establish an upper bound on the complexity of a problem that underpins many of others that will be studied later on:

IBG MODEL CHECKING
Given: Game G, strategy profile $\vec{\sigma}$, LTL formula $\varphi \in L(\Phi)$.
Question: Is it the case that $\rho(\vec{\sigma}) \models \varphi$?

This problem is not quite the same as the "standard" LTL model checking problem, which is known to be PSPACE-complete ("truth in an R-structure" [27, p. 741]), and so we need to establish the upper bound. Also, with respect to the set of machine strategies given as input, recall that we allow them to be partial. The reduction in the following proof is to LTL SATISFIABILITY, which is known to be in PSPACE [27].

LTL SATISFIABILITY
Given: LTL formula $\varphi \in L(\Phi)$.
Question: Is there a run $\rho : \mathbb{N} \to 2^\Phi$ such that $\rho \models \varphi$?

We are now in a position to prove the following result.

Proposition 1. IBG MODEL CHECKING *is in PSPACE.*

Proof. We reduce IBG MODEL CHECKING to LTL SATISFIABILITY. Let $\vec{\sigma}$ be a (partial) strategy profile for an iterated Boolean game G and let φ be an LTL formula in $L(\Phi)$. To check whether $\rho(\vec{\sigma}) \models \varphi$, we ask whether $TH(\vec{\sigma}) \wedge \varphi$ is satisfiable. Recall that the size of $TH(\vec{\sigma})$ is polynomial in the size of $\vec{\sigma}$. It now suffices to show that,

$$TH(\vec{\sigma}) \wedge \varphi \text{ is satisfiable} \quad \text{if and only if} \quad \rho(\vec{\sigma}) \models \varphi.$$

For the implication from left to right, assume that $TH(\vec{\sigma}) \wedge \varphi$ is satisfiable. Then, there is some run $\rho : \mathbb{N} \to 2^{\Phi \cup Q \cup \{q_1^d, \ldots, q_n^d\}}$, with $\rho \models TH(\vec{\sigma}) \wedge \varphi$, and some strategy profile $\vec{\sigma}'$ such that $\rho|_\Phi = \rho(\vec{\sigma}')$. By Lemma 4(2) it then follows that $\rho = \hat{\rho}(\vec{\sigma})$, and hence $\hat{\rho}(\vec{\sigma}) \models \varphi$. Since φ is an LTL formula in $L(\Phi)$ and $\rho(\vec{\sigma}) = \hat{\rho}(\vec{\sigma})|_\Phi$, we can conclude that $\rho(\vec{\sigma}) \models \varphi$.

For the implication from right to left, assume that $\rho(\vec{\sigma}) \models \varphi$. Since $\rho(\vec{\sigma}) = \hat{\rho}(\vec{\sigma})|_\Phi$, also $\hat{\rho}(\vec{\sigma}) \models \varphi$. Moreover, observe that, by Lemma 4(2), we also have $\hat{\rho}(\vec{\sigma}) \models TH(\vec{\sigma})$. It follows that $TH(\vec{\sigma}) \wedge \varphi$ is satisfiable. □

```
Input:   Game G and strategy profile σ⃗
Output: "yes" if σ⃗ ∈ NE(G), "no" otherwise.
1  for i := 1 to n do
2      if ρ(σ⃗) ⊭ γᵢ then
3          if γᵢ ∧ ⋀_{j∈N\{i}} th(σⱼ) is satisfiable then
4              return "no"
5          end if
6      end if
7  end for
8  return "yes"
```

Algorithm 1: Algorithm for MEMBERSHIP.

With this result in place, we can consider problems specifically related to equilibria in iterated Boolean games. First, consider the following problem:

MEMBERSHIP
Given: Game G and strategy profile $\vec{\sigma}$.
Question: Is it the case that $\vec{\sigma} \in NE(G)$?

Based on Observation 1, we have the following result.

Proposition 2. MEMBERSHIP *is PSPACE-complete.*

Proof. Observe that Algorithm 1 (see above) solves MEMBERSHIP for iterated Boolean games in PSPACE. By Proposition 1, line 2 can be solved using a PSPACE oracle for IBG MODEL CHECKING. Moreover, line 3 uses a PSPACE oracle for LTL SATISFI-ABILITY. The algorithm checks whether there is some player i such that $\rho(\vec{\sigma}) \not\models \gamma_i$ and $\gamma_i \wedge \bigwedge_{j\in N\setminus\{i\}} th(\sigma_j)$ is satisfiable. It outputs "no" if this is the case and "yes," otherwise.

For soundness, note that, because of Observation 1, it suffices to prove that, for all players i,

$$\gamma_i \wedge \bigwedge_{j\in N\setminus\{i\}} th(\sigma_j) \text{ is satisfiable} \quad \text{if and only if} \quad (\vec{\sigma}_{-i}, \sigma_i') \models \gamma_i \text{ for some } \sigma_i'.$$

Therefore, assume $\gamma_i \wedge \bigwedge_{j\in N\setminus\{i\}} th(\sigma_j)$ is satisfiable, let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, with $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$, and let q_i^d be the default state of the complete strategy that σ_i represents. Then, there is an ω-regular $\rho : \mathbb{N} \to 2^{\Phi \cup Q \cup \{q_1^d, \ldots, q_n^d\}}$ such that

$$\rho \models \gamma_i \wedge \bigwedge_{j\in N\setminus\{i\}} th(\sigma_j).$$

Since ρ is ω-regular, then $\rho|_\Phi$ is also ω-regular and by Corollary 1 there is a strategy profile $\vec{\sigma}'' = (\sigma_1'', \ldots, \sigma_n'')$ where each σ_i'' is myopic such that $\rho|_\Phi = \rho(\vec{\sigma}'')$. Observe that, with $\gamma_i \in L(\Phi)$, also $\rho(\vec{\sigma}'') \models \gamma_i$. Now, for every player j distinct from i, we have $\rho \models th(\sigma_j)$ and, by Lemma 4(1), moreover,

$$\rho(\vec{\sigma}'') = \rho|_\Phi = \hat{\rho}(\vec{\sigma}''_{-j}, \sigma_j)|_\Phi = \rho(\vec{\sigma}''_{-j}, \sigma_j).$$

By an $(n-2)$-fold application of Lemma 3, we obtain $\rho(\vec{\sigma}_{-i}, \sigma_i'') = \rho(\vec{\sigma}'')$. It follows that $(\vec{\sigma}_{-i}, \sigma_i'') \models \gamma_i$.

For the opposite direction assume that $(\vec{\sigma}_{-i}, \sigma_i') \models \gamma_i$ for some strategy σ_i'. Now, observe that Lemma 1 yields

$$\hat{\rho}(\vec{\sigma}_{-i}, \sigma_i') \models \gamma_i \wedge \bigwedge_{j\in N\setminus\{i\}} th(\sigma_j).$$

For hardness, we reduce LTL SATISFIABILITY to MEMBERSHIP. Let φ be an LTL formula in $L(\Phi)$ and let ρ^0 be the run such that $\rho^0[t] = \emptyset$ for all $t \in \mathbb{N}$. Define $G = (\{i\}, \Phi \cup \{p\}, \varphi \wedge p)$ as the one-player iterated Boolean game in which player i has $\varphi \wedge p$ as goal, where p is a fresh propositional variable, that is, $p \notin \Phi$. Let $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$ be the *partial* machine strategy with $Q_i = \{q_i^0\}$ and $\delta_i = \tau_i = \emptyset$. Thus, $\rho(\sigma_i) = \rho^0$ and $\rho(\sigma_i) \not\models \varphi \wedge p$. Clearly, σ_i is polynomial in the size of φ. It can now readily be seen that,

$$\varphi \text{ is satisfiable} \quad \text{if and only if} \quad \sigma_i \notin NE(G).$$

For the left-to-right direction assume that φ is satisfiable. As p does not occur in φ, obviously, $\varphi \wedge p$ is satisfiable. Then, by Lemma 1, there is a strategy profile σ_i' such that $\rho(\sigma_i') \models \varphi \wedge p$. Since $\rho(\sigma_i) \not\models \varphi \wedge p$, by Observation 1, we may conclude that $\sigma_i \notin NE(G)$. Now assume that φ is not satisfiable. Then, neither is $\varphi \wedge p$. By Observation 1, it then follows that $\sigma_i \in NE(G)$. As PSPACE is closed under complement, the result follows. □

Thus, solving Membership for iterated Boolean games is no harder than LTL model checking. However, other apparently closely related problems turn out to be much more complex. Consider the following:

E-Nash
Given: Game G, LTL formula $\varphi \in L$.
Question: Does $\rho(\vec{\sigma}) \models \varphi$ hold for some $\vec{\sigma} \in NE(G)$?

A-Nash
Given: Game G, LTL formula $\varphi \in L$.
Question: Does $\rho(\vec{\sigma}) \models \varphi$ hold for all $\vec{\sigma} \in NE(G)$?

Non-Emptiness
Given: Game G.
Question: Is it the case that $NE(G) \neq \emptyset$?

On the one hand, we first show (i) that E-Nash can be reduced to the *rational synthesis* problem for pure Nash equilibria as first analysed by Fisman et al., known to be 2EXPTIME-complete (Theorem 2 in Fisman et al. [11]). On the other hand, we show (ii) that the *LTL realizability* problem studied by Pnueli and Rosner [25], which is known to be 2EXPTIME-complete too, can be reduced to Non-Emptiness. Since Non-Emptiness can be solved by asking whether $(G, \top)$ is accepted as an instance of E-Nash, then (i) and (ii) imply that both E-Nash and Non-Emptiness are 2EXPTIME-complete. Finally, since 2EXPTIME is a deterministic complexity class, that A-Nash is 2EXPTIME-complete immediately follows as a corollary.

Let us briefly outline the rational synthesis problem (see [11] for a detailed description). In the rational synthesis problem there is a set of players $1, \ldots, n$, called the environment, and a player 0 called the system. Each player has an LTL goal $\gamma_0, \gamma_1, \ldots, \gamma_n$ and as in iterated Boolean games each player controls a set of variables $\Phi_0, \Phi_1, \ldots, \Phi_n$ and the game is played for infinitely many rounds. In a single round of the game the environment players simultaneously give values to the variables they have control over and based on that the system player gives values to the variables it controls, that is, those variables in Φ_0. Such values are either $\top$ or $\bot$ in the Boolean setting. Note, in particular, that in a single round the system gets to see the values that the environment has chosen before giving the values of the variables in Φ_0. The strategies of the environment players $i \in \{1, \ldots, n\}$ are functions

$$f_i : (2^{\Phi_0 \cup \Phi_1 \cup \ldots \cup \Phi_n})^* \to 2^{\Phi_i}$$

and the strategy of the system player is a function

$$f_0 : (2^{\Phi_0 \cup \Phi_1 \cup \ldots \cup \Phi_n})^* (2^{\Phi_1 \cup \ldots \cup \Phi_n}) \to 2^{\Phi_0}$$

showing that the system only plays after its environment. Clearly, each strategy profile $(f_0, f_1, \ldots, f_n)$ induces a unique outcome of the game, namely, an infinite run/word over the set $2^{\Phi_0 \cup \Phi_1 \cup \ldots \cup \Phi_n}$.[2] Now, the rational synthesis problem asks for a strategy profile $(f_0, f_1, \ldots, f_n)$ such that the outcome of the game satisfies γ_0 (the goal of the system) and the strategy profile $(f_1, \ldots, f_n)$ is a pure-strategy Nash equilibrium. The reduction of E-Nash to the rational synthesis problem is now possible.

Proposition 3. *The problems* E-Nash, A-Nash, *and* Non-Emptiness *are 2EXPTIME-complete.*

Proof. To solve E-Nash for a game (G, φ) one can ask for a solution to the *rational synthesis* problem [11] where the system player does not control any variables, *i.e.* $\Phi_0 = \emptyset$, and has φ as goal, that is, $\gamma_0 = \varphi$. Suppose now that this rational synthesis problem has a positive answer, that is, that there is a run, say π, induced by $(f_0, f_1, \ldots, f_n)$ such that both $\pi \models \gamma_0$ and $(f_1, \ldots, f_n)$ is a Nash equilibrium. Since strategy profiles $(f_0, f_1, \ldots, f_n)$ induce unique runs and f_0 is trivial (because $\Phi_0 = \emptyset$)—due to Lemma 1—the unique run over $2^{\Phi_1 \cup \ldots \cup \Phi_n}$, namely π, induced by $(f_0, f_1, \ldots, f_n)$ can be generated by a set of machine strategies $\{\sigma_1, \ldots, \sigma_n\}$. In other words, it follows that $\pi = \rho(\sigma_1, \ldots, \sigma_n)$ and $\rho(\sigma_1, \ldots, \sigma_n) \models \varphi$, for some $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ that forms a Nash equilibrium. The other direction, namely that if the rational synthesis problem has a negative answer then E-Nash has a negative answer too, is trivial. Therefore, E-Nash can be solved in 2EXPTIME.

For hardness, we now show that the *LTL realizability* problem can be reduced to Non-Emptiness which, as mentioned before, can be solved by asking whether $(G, \top)$ is accepted as an instance of E-Nash. We first construct an intermediate game R', which is a variant of the realizability problem $R = (\varphi, I, O)$ as first studied by Pnueli and Rosner [25]. In the standard realizability game $R = (\varphi, I, O)$ the environment always plays first. Then, a play is an infinite sequence of valuations $v_i^0, v_o^0, v_i^1, v_o^1, \ldots$ where each $v_i^k \in 2^I$ and each $v_o^k \in 2^O$, with $k \in \mathbb{N}$. The system controls the variables in O and the environment controls those in I. If φ can be realised, then there is a function $f : (2^{I \cup O})^+ \to 2^O$ that the system can use as a strategy so that every run given by f, that is, every word $\rho = (v_i^0 \cup v_o^0), (v_i^1 \cup v_o^1), \ldots$ satisfies φ.

---

[2] Because the goals are LTL formulae, it is sufficient to assume that strategies are the functions represented by finite state machines with output, *i.e.* by transducers [24].

In the realizability game $R' = (\varphi', I, O)$ we now have both players, the system and the environment, playing simultaneously—*i.e.*, the system no longer plays after the environment. The formula φ' is $\varphi[\mathbf{X}\,p_1, \ldots, \mathbf{X}\,p_m / p_1, \ldots, p_m]$, the LTL formula where every occurrence of a variable $p_j \in O$ is replaced by $\mathbf{X}\,p_j$. We first show that φ is R-realizable if and only if φ' is R'-realizable, and then that R'-realizability can be reduced to Non-Emptiness.

Now, suppose that φ is realizable. It follows that there is a function $f : (2^{I \cup O})^+ \to 2^O$ for which every $\rho = (v_i^0 \cup v_o^0), (v_i^1 \cup v_o^1), \ldots$, with $v_i^k \in 2^I$ and $v_o^k \in 2^O$, satisfies φ. Let $f' : (2^{I \cup O})^+ \to 2^O$ be defined as follows:

$$f'((v_i^0 \cup v_o^0), \ldots, (v_i^k \cup v_o^k)) = \begin{cases} \emptyset & \text{if } k = 0, \\ f((v_i^0 \cup v_o^0), \ldots, (v_i^{k-1} \cup v_o^{k-1})) & \text{otherwise.} \end{cases}$$

Thus, if $\rho = (v_i^0 \cup v_o^0), (v_i^1 \cup v_o^1), \ldots$ is a run in the R-realizability game, then $\rho' = (v_i^0 \cup \emptyset), (v_i^1 \cup v_o^0), (v_i^2 \cup v_o^1) \ldots$ is the induced run in the associated R'-realizability game. Induction on φ' shows that, for every run ρ' in the R'-realizability game constructed from R, we have that $\rho' \models \varphi'$, *i.e.*, that φ' is R'-realizable, because $\rho \models \varphi$ holds in the associated R-realizability game.

For the other direction, we prove the contrapositive: we suppose that φ is not R-realizable and show that, in such a case, φ' is not R'-realizable either. Since φ is not R-realizable, there is a winning strategy for the environment to make φ false whatever strategy the system uses, that is, to generate a run ρ such that $\rho \not\models \varphi$. It then follows that as $\rho = (v_i^0 \cup v_o^0), (v_i^1 \cup v_o^1), \ldots$ does not satisfy φ, then $\rho' = (v_i^0 \cup \emptyset), (v_i^1 \cup v_o^0), (v_i^2 \cup v_o^1) \ldots$ does not satisfy $\rho' = \varphi[\mathbf{X}\,p_1, \ldots, \mathbf{X}\,p_m / p_1, \ldots, p_m]$ either—and hence φ' is not R'-realizable.[3]

Finally, we reduce R'-realizability to Non-Emptiness. Given an R'-realizability game $R' = (\varphi', I, O)$, create a 4-player iterated Boolean game G, as follows. The variables in G are $\Phi = I \cup O \cup \{x_1, x_2\}$, where x_1 and x_2 are new. Player 1 corresponds to the system, controls the variables in O, and has goal $\gamma_1 = \varphi'$. Player 2 corresponds to the environment, controls variables in I, and has goal $\gamma_2 = \neg\varphi'$. Player 3 controls x_1 and has goal $\gamma_3 = \varphi' \vee (x_1 \leftrightarrow x_2)$ while Player 4 controls x_2 and has goal $\gamma_4 = \varphi' \vee \neg(x_1 \leftrightarrow x_2)$. Note that players 3 and 4 ensure that in any Nash equilibrium, φ' must be satisfied: in any strategy profile that does not satisfy φ', one of players 3 or 4 has a beneficial deviation. We claim that φ' is R'-realizable if and only if $NE(G) \neq \emptyset$.

For the left-to-right direction, assume φ' is R'-realizable, and let σ_1 be the machine strategy corresponding to the winning strategy for φ'. Also, let $\sigma_2, \sigma_3, \sigma_4$ be arbitrary strategies for players 2, 3, and 4, respectively. We claim that $(\sigma_1, \sigma_2, \sigma_3, \sigma_4) \in NE(G)$. Observe that players 1, 3, and 4 all have their goal achieved by this strategy profile and can have no beneficial deviation. A beneficial deviation for player 2 would require a strategy σ_2' such that $\rho(\sigma_1, \sigma_2', \sigma_3, \sigma_4) \models \neg\varphi'$. But since σ_1 encodes a winning strategy for φ', no such strategy σ_2' is possible; hence $(\sigma_1, \sigma_2, \sigma_3, \sigma_4) \in NE(G)$.

Finally, for the right-to-left direction, assume $NE(G) \neq \emptyset$, and let the profile $\vec{\sigma} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4)$ be a witness to this fact. Observe that we must have $\rho(\vec{\sigma}) \models \varphi'$, for otherwise, players 3 and 4 would ensure that $\vec{\sigma} \notin NE(G)$. Since $\vec{\sigma}$ is a Nash equilibrium, player 2 cannot deviate through any strategy σ_2' so that $\rho(\vec{\sigma}_{-2}, \sigma_2') \models \neg\varphi'$. It then follows that σ_1 encodes a winning strategy f' for φ', and hence φ' is R'-realizable. □

5. Complexity of other solution concepts

Nash equilibrium is the most important solution concept in game theory, and many relevant solution concepts in the literature are either generalisations or refinements of it. However, there are well-known weaknesses of Nash equilibria; for instance, it is not guaranteed to be unique, it is arguably unstable (nothing is ensured if, *e.g.*, more than one player deviates or irrational moves are made), and it does not account for dynamic behaviour, amongst others. For this reason, we now explore the complexity of other solution concepts, namely, *dominant strategies* and a refinement of *subgame perfect Nash equilibrium*.[4]

The former is a stronger concept when compared with Nash equilibrium. However, it is much more stable: it ensures a player's best response *regardless of the behaviour of the other players*. Thus, unlike Nash equilibria, dominant strategies behave well even with respect to "irrational" moves of other players. The latter solution concept, on the other hand, is a refinement of Nash equilibrium (every subgame perfect Nash equilibrium is a Nash equilibrium) intended to take into account the dynamic behaviour of systems—informally, a Nash equilibrium is subgame perfect if it is a Nash equilibrium in all subgames of the original game. Here, we study a refinement of subgame perfect Nash equilibrium where *reachable*, rather than all, subgames are considered.[5]

---

[3] The preservation of (un)satisfiability for the LTL formulae given by the map from LTL formulae φ to φ'—where output variables are preceded by a next-time operator—is also presented in [7] where strategies as finite state transducers are also considered.

[4] We note that subgame perfect equilibrium can be formulated in different ways for iterated Boolean games; the definition we provide here represents just one of a range of possibilities.

[5] A criticism of subgame perfect equilibrium, as defined by Selten, is that this solution concept requires every strategy in a profile to be a best response, even in situations that the strategy profile precludes from arising, *i.e.*, in unreachable subgames. Selten studied refinements of subgame perfect equilibrium [26] where this defect could be eliminated.

5.1. Dominant strategies

Dominant strategy equilibrium is a very appealing solution concept because of its stability/robustness: it defines a best response for each player, no matter how other players in the game behave. Formally, a strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_i, \ldots, \sigma_n)$ is a *dominant strategy equilibrium* if for every strategy profile $(\sigma'_1, \ldots, \sigma'_n)$ and every player i, we have

$$\rho(\sigma'_1, \ldots, \sigma_i, \ldots, \sigma'_n) \succsim_i \rho(\sigma'_1, \ldots, \sigma'_i, \ldots, \sigma'_n).$$

We now show that computing the set of dominant strategy equilibria of an iterated Boolean game is, as for Nash equilibria, a 2EXPTIME-complete problem. We shall write Non-Emptiness for such a problem. The proof of this result is obtained via a variation of the proof of Proposition 3 (see Appendix A). We also write E-Dominant and A-Dominant for the analogous problems of E-Nash and A-Nash, respectively, now with respect to dominant strategy equilibrium. Formally, we have the following result, whose proof may be found in Appendix A.

Proposition 4. E-Dominant, A-Dominant, *and* Non-Emptiness *for Dominant Strategies are 2EXPTIME-complete.*

5.2. Consistent-subgame perfect Nash equilibrium

A well-known, natural refinement of Nash equilibrium when considering extensive form games [22] is *subgame perfect Nash equilibrium* (SPNE), a solution concept where the dynamic nature of a system is better captured. As iterated Boolean games are played for infinitely many rounds, it is only natural to study this solution concept in our framework.

Specifically, in this section, we show that Non-Emptiness, E-CSPNE, and A-CSPNE (again, the analogous problems of those defined for Nash equilibrium), remain complete for 2EXPTIME when the solution concept is a refinement of SPNE where, informally, one is interested in whether a strategy profile is a Nash equilibrium in *all reachable subgames*, that is, in all subgames that are *consistent* with the strategy profile (the definition of subgame perfect Nash equilibrium requires the strategy profile to be a Nash equilibrium in all subgames, even unreachable ones, that is, those subgames inconsistent with the strategy profile). The proofs of these results are adaptations of the proofs for Nash equilibrium, based—informally—on the following key observation: that if a player has a winning strategy, then it also has one in all *reachable* subgames allowed by (consistent with) such a winning strategy.

We start by defining *subgames* in iterated Boolean games. Subgames are defined with respect to finite runs (finite sequences of valuations), whose concatenation will be denoted by the symbol ";" usually employed to represent the concatenation of two strings/words. Then, given a finite run $\pi = v_0, \ldots, v_k$—with $k \in \mathbb{N}$—and a run $\rho = w_0, w_1, \ldots$ we will write $\pi; \rho$ for the sequence of valuations (*i.e.*, run) $v_0, \ldots, v_k, w_0, w_1, \ldots$.

Now, given an iterated Boolean game

$$G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$$

and a finite run π, the π-*subgame* G_π of G is defined to be the iterated Boolean game G, save that, for all runs ρ, ρ', we have

$$\rho \succsim_i \rho' \quad \text{if and only if} \quad \pi; \rho' \models \gamma_i \text{ implies } \pi; \rho \models \gamma_i,$$

that is, with the satisfaction of the LTL goals $\gamma_1, \ldots, \gamma_n$ subject to the finite run π (an unfinished "partial play") having already happened. Finally, outcomes and strategies in subgames are defined as for iterated Boolean games (possible because any π-subgame G_π of G has the same structure of G).

Observe that once players choose their strategies, some subgames are not reachable. For instance, in Example 1, if player 1 chooses to play with the strategy in Fig. 1(a), then no π-subgame where π contains an occurrence of $\neg p$ is reachable (because such a strategy, namely σ_1, always plays p).

We can then, given a game G, define the subgames of G with respect to the finite runs induced by a particular strategy profile $\vec{\sigma}$. These would be, precisely, the reachable subgames when the strategy profile $\vec{\sigma}$ is being played. Formally, the subgames of G when playing the strategy profile $\vec{\sigma}$, called the $\vec{\sigma}$-*subgames* $G_{\vec{\sigma}}$ of G, are defined to be the set of subgames

$$\{G_{\rho(\vec{\sigma})[0, \ldots, k]} : k \in \mathbb{N}\}.$$

Also, substrategies can be defined for subgames, *i.e.*, given some G and a strategy profile $\vec{\sigma}$, we can define the π-*substrategies* that will be played in every π-subgame. Formally, for every $i \in N$, let σ_i^π be the π-substrategy of σ_i defined to be σ_i save that the new initial state of the strategy is

$$q_i^0 = h(\vec{\sigma})[|\pi|]_i$$

where h is the history of $\vec{\sigma}$ as defined before, and $|\pi|$ denotes the size of π. In other words, $h(\vec{\sigma})[|\pi|]_i$ is the state that the strategy σ_i reaches after reading (being fed with) π. We, thus, write $\vec{\sigma}_\pi$ for the strategy profile $(\sigma_1^\pi, \ldots, \sigma_n^\pi)$.

With the above definitions in place, we can now introduce the best known solution concept (equilibrium) for subgames in iterated Boolean games. Given a game G, we say that $\vec{\sigma}$ is a *subgame perfect Nash equilibrium* of G if $\vec{\sigma}_\pi$ is a Nash
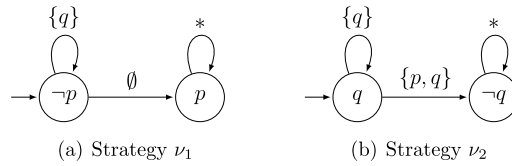
(a) Strategy ν_1 (b) Strategy ν_2

Fig. 4. The strategies ν_1 and ν_2 for player 1 and player 2, respectively.

equilibrium of G_π, for all finite runs π. Moreover, we say it is a *consistent-subgame perfect Nash equilibrium* if it is a Nash equilibrium of G_π, for all runs $\pi \in \{\rho(\vec{\sigma})[0, \ldots, k] : k \in \mathbb{N}\}$, that is, with respect to all reachable subgames of G when playing $\vec{\sigma}$ (*i.e.*, subgames that are consistent with the strategy profile $\vec{\sigma}$). Given the definitions above, we also have the following result, whose proof may be found in Appendix A.

Proposition 5. E-CSPNE*,* A-CSPNE*, and* Non-Emptiness *with respect to Consistent-Subgame Perfect Nash equilibria are 2EXPTIME-complete.*

6. Folk theorems for iterated Boolean games

In game theory, much of the interest in iterated games derives from the *Nash Folk Theorems* [22, p. 143]. These theorems tell us that the range of outcomes that can be sustained as equilibria in iterated games is much wider than one might at first suspect from the component game. Their usual form is that the set of *all* feasible and individually rational payoff vectors are achievable as Nash equilibria in iterated games; the definitions of feasibility and individual rationality depend on the precise setting considered.

To take a famous example, the Nash Folk Theorems tell us that cooperation can be sustained in the iterated Prisoner's Dilemma. The standard device for proving Folk Theorems is a *trigger strategy* [22, p. 143]. A trigger strategy intended to obtain a particular outcome works by punishing any player who deviates from behaviour leading to the intended outcome: no player can benefit from deviation, as this would result in punishment by all other players. The desired outcome is thereby obtained as an equilibrium.

It seems very natural, therefore, to consider Folk Theorems in the context of our iterated Boolean games. Given our interest in using LTL to express properties of equilibria, we can formulate the question of which equilibria can be sustained in an iterated Boolean game in the following way:

Which LTL properties are preserved by the Nash equilibria of an iterated Boolean game?

This question is closely related to the rational synthesis problem in [11], and here formalised in E-Nash. However, the rational synthesis problem, in effect, pertains to *particular* LTL properties being realised in some Nash equilibrium, while our concern is rather with characterising *the set of LTL formulae that can be satisfied in equilibria of iterated Boolean games.*

Let us now consider the following example.

Example 3. Consider again the iterated Boolean game of Example 1 along with the formula $\varphi = \mathbf{G}(\neg p \wedge q)$. Obviously, φ is inconsistent with both player 1's goal $\mathbf{G}\mathbf{F}(p \wedge q)$ and player 2's goal $\mathbf{G}\mathbf{F}(\neg p \wedge \neg q)$. The run that results from player 1 invariably setting p to false and player 2 invariably setting q to true clearly satisfies φ but the strategies involved clearly do not constitute a Nash equilibrium: player 1 would like to deviate to σ_1 and player 1 to σ_2 (see Fig. 1). The same run, however, is produced by the strategies ν_1 and ν_2 as depicted in Fig. 4. Moreover, it can be seen that ν_1 and ν_2 constitute a pure-strategy Nash equilibrium. Note that if player 1 were to set p to true at a certain point in time, it would be answered by player 2 playing $\neg q$ ever after. This would also render player 1's goal false. Similarly, if, at some time point, player 2 were to set q to false, player 1 will invariably play p from the next time point onwards. If so, player 2's goal is rendered unachievable.

In this section, we show that the concepts and techniques used to prove the Nash Folk Theorems can be adapted to our setting. In particular, for games in which players have *safety goals* (of the form $\mathbf{G}\varphi$), we use a punishment strategy construction, similar in spirit to that used to prove the Nash Folk Theorems, to characterise precisely the circumstances under which arbitrary LTL formulae are satisfied in some equilibrium of an iterated game.

To be able to use such a construction, we need to be able to define for our setting some counterpart of the notion of a feasible and individually rational payoff for each player. In game theory, a payoff for one player is individually rational and feasible if it could be enforced by the set of all other players in the game. In our setting, players have goals, rather than payoffs, and so we need to formulate the concept with respect to whether a player's goal can be falsified by the set of all other players.

6.1. Punishable players and goals

A player i, with goal γ_i, is *punishable* if (at any point of time) i's counterparts can jointly find values for the propositional variables under their control that guarantee γ_i will then be false no matter which values i chooses for its variables. Nash Folk Theorems may hold only if all players, at all times, are punishable. This is indeed the case for games with certain kinds of goals, which we define next. We say a goal γ_i is *non-trivial* if γ_i and $\neg\gamma_i$ are satisfiable (*i.e.*, if the goal γ_i is neither a tautology nor a contradiction in LTL). We say γ_i is a *safety goal* when γ_i is equivalent to a formula $\mathbf{G}\varphi_i$ for some LTL formula φ_i. When φ_i is a propositional formula we say that γ_i is a *propositional safety goal*. Folk theorems for iterated Boolean games with propositional safety goals are presented first.

6.2. Propositional safety goals

We now define a predicate *punishable*(i), which formalises when a player i with a propositional safety goal is punishable, that is, if a trigger strategy can be constructed against i. For this we find it useful to introduce a small piece of additional notation. Where i is a player, we let V_{-i} denote the set of valuations $V(\Phi \setminus \Phi_i)$, *i.e.*, the set of valuations for variables controlled by players other than i. Then, for iterated Boolean games with propositional safety goals $\gamma_i = \mathbf{G}\varphi_i$ the predicate *punishable*(i) reduces to the expression

$$\bigvee_{v_{-i} \in V_{-i}} \bigwedge_{v'_i \in V_i} \left(\chi_{(v_{-i}, v'_i)} \to \neg\varphi_i \right)$$

being valid. Say player i is *punishable* when the predicate *punishable*(i) holds (and it is not punishable otherwise).

In the case where a player i is not punishable, we can ensure that its goal will be achieved in any Nash equilibrium, as shown next.

Lemma 5. *Let* $G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$ *be an iterated Boolean game in which player i has a propositional safety goal* $\gamma_i = \mathbf{G}\varphi_i$*. Then, if player i is not punishable, for all* $\vec{\sigma} \in NE(G)$ *we have that* $\rho(\vec{\sigma}) \models \gamma_i$*.*

The proof may be found in Appendix A.

Now, we can prove a Folk Theorem for iterated Boolean games with propositional safety goals. As seen in Lemma 5, non-punishable players invariably achieve their goals in any Nash equilibrium. The next Folk Theorem refers to situations where non-punishable players have to be considered.

Theorem 1. *Let* $G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$ *be an iterated Boolean game in which each player i has a propositional safety goal* $\gamma_i = \mathbf{G}\varphi_i$*. Let, furthermore, ψ be an LTL formula such that* $\psi \wedge \bigwedge\{\gamma_i : i \text{ is not punishable}\}$ *is satisfiable. Then, there is a* $\vec{\sigma} \in NE(G)$ *with* $\rho(\vec{\sigma}) \models \psi$*.*

The proof may be found in Appendix A.

Using Theorem 1, the following statement follows as a corollary.

Theorem 2. *Let* $G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$ *be an iterated Boolean game where each player i has a non-trivial propositional safety goal* $\gamma_i = \mathbf{G}\varphi_i$*. Then, the following two statements are equivalent:*

  (*i*) *all players are punishable, and*
 (*ii*) *for all satisfiable* $\psi \in L$*, there is a* $\vec{\sigma} \in NE(G)$: $\rho(\vec{\sigma}) \models \psi$*.*

Proof. The implication from (*i*) to (*ii*) is immediate by Theorem 1. For the direction from (*ii*) to (*i*), we prove the contrapositive. Assume that some player i is not punishable. As γ_i was assumed to be non-trivial, $\neg\gamma_i$ is satisfiable. Yet, by Lemma 5, we have that $\rho(\vec{\sigma}) \models \gamma_i$ for all equilibria $\vec{\sigma} \in NE(G)$ (take ψ in Lemma 5 to be $\neg\gamma_i$). Hence, there is no $\vec{\sigma} \in NE(G)$ with $\rho(\vec{\sigma}) \models \neg\gamma_i$. $\square$

We can leverage Theorems 1 and 2 to obtain PSPACE-completeness for the E-Nash problem with propositional safety goals.

Proposition 6. *The* E-Nash *and* A-Nash *problems for iterated Boolean games with propositional safety goals are* PSPACE-*complete.*

Proof. For membership of E-Nash in PSPACE, consider an arbitrary game $G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$ with propositional safety goals and an arbitrary $\psi \in L(\Phi)$. Algorithm 2 decides whether ψ is sustained by some pure Nash equilibrium of G.

```
Input:    Game G = (N, Φ, Φ₁, ..., Φₙ, γ₁, ..., γₙ), where γ₁, ..., γₙ are propositional safety goals given by Gφ₁, ..., Gφₙ
Output: "yes" if σ⃗ ∈ NE(G) for some σ⃗ ∈ Σᵢ, "no" otherwise.
 1   N₀ := ∅
 2   for i := 1 to n do
 3       if ∀ Φ \ Φᵢ ∃Φᵢ φᵢ is true then
 4           N₀ := N₀ ∪ {i}
 5       else
 6           N₀ := N₀
 7       end if
 8   end for
 9   if ψ ∧ ⋀ᵢ∈N₀ γᵢ is satisfiable then
10       return "yes"
11   else
12       return "no"
13   end if
```

Algorithm 2: Algorithm for E-Nash with propositional safety goals.

In the for-loop, that is, in lines (1) through (8), this algorithm singles out the players that are not punishable. Checking the quantified Boolean formula $\forall \Phi \setminus \Phi_i \exists \Phi_i \varphi$ for truth is equivalent to checking $\bigwedge_{v_{-i} \in V_{-i}} \bigvee_{v'_i \in V_i} (\chi_{(v_{-i}, v'_i)} \wedge \varphi_i)$ for satisfiability and can be achieved in Σ_2^p. Soundness of this step then follows from Lemma 5. In step (9) satisfiability of $\psi \wedge \bigwedge_{i \in N_0} \gamma_i$ is checked, which can be done in PSPACE. As PSPACE subsumes Σ_2^p, the algorithm runs in PSPACE.

For soundness and completeness, it now suffices to show that the following two statements are equivalent:

(*i*) $\psi \wedge \bigwedge_{i \in N_0} \gamma_i$ is satisfiable,
(*ii*) there is a $\vec{\sigma} \in NE(G)$ with $\rho(\vec{\sigma}) \models \psi$.

For the direction from (*ii*) to (*i*), assume that there is a $\vec{\sigma} \in NE(G)$ with $\rho(\vec{\sigma}) \models \psi$. By Lemma 5, it then follows that $\rho(\vec{\sigma}) \models \gamma_i$ for all $i \in N_0$. Hence, $\psi \wedge \bigwedge_{i \in N_0} \gamma_i$ is satisfiable. The other direction is immediate by Theorem 1.

For PSPACE-hardness, we reduce the LTL satisfiability problem, which is PSPACE-complete. Let ψ be an arbitrary LTL formula in L. We may assume $vars(\psi)$ to contain at least two propositional variables p and q. Now consider the iterated Boolean game $G = (N, \Phi, \Phi_1, ..., \Phi_n, \gamma_1, ..., \gamma_n)$ such that $N = \{1, 2\}$, $\Phi = vars(\psi)$, $p \in \Phi_1$, $q \in \Phi_2$, and $\gamma_1 = \mathbf{G}q$ and $\gamma_2 = \mathbf{G}p$. Clearly, both γ_1 and γ_2 are propositional safety goals. Moreover, it can easily be checked that $\bigvee_{v_{-i} \in V_{-i}} \bigwedge_{v'_i \in V_i} (\chi_{(v_{-i}, v'_i)} \rightarrow \neg \gamma_i)$ is valid for both $i = 1$ and $i = 2$, that is, both 1 and 2 are punishable. It now suffices to show that

$$\psi \text{ is satisfiable} \quad \text{if and only if} \quad \text{there is a } \vec{\sigma} \in NE(G) \text{ with } \rho(\vec{\sigma}) \models \psi.$$

This, however, is an immediate consequence of Theorem 2.

Finally, PSPACE-completeness of A-Nash follows from E-Nash, since PSPACE is a deterministic complexity class. □

6.3. General safety goals

For general safety goals $\gamma_i = \mathbf{G}\varphi_i$ we say that, for each player i, the predicate *punishable*(i) holds if and only if there is some satisfiable LTL formula $\chi \in L(\Phi \setminus \Phi_i)$ such that $\chi \rightarrow \neg \mathbf{G}\varphi_i$ is valid. Based on this definition the next Folk Theorem follows:

Theorem 3. *Let $G = (N, \Phi, \Phi_1, ..., \Phi_n, \gamma_1, ..., \gamma_n)$ be an iterated Boolean game where each player i has a non-trivial safety goal $\gamma_i = \mathbf{G}\varphi_i$. Then, if all players are punishable, for all satisfiable $\psi \in L$, there is a strategy profile $\vec{\sigma} \in NE(G)$ with $\rho(\vec{\sigma}) \models \psi$.*

Proof. Assume that each player is punishable, that is, that for each player i there is some satisfiable formula $\chi_i \in L(\Phi_{-i})$ such that $\chi_i \rightarrow \neg \mathbf{G}\varphi_i$ is valid. Observe that $\rho(\vec{\sigma}) \models \chi_i$ implies that $\rho(\vec{\sigma}_{-i}, \sigma'_i) \models \chi_i$ for all σ'_i in Σ_i. Moreover, let $\psi \in L$ be an arbitrary satisfiable formula.

For arbitrary satisfiable formulae φ, we denote by $\vec{\sigma}[\varphi] = (\sigma_1[\varphi], ..., \sigma_n[\varphi])$ any strategy profile which run satisfies φ, i.e., $\rho(\vec{\sigma}[\varphi]) \models \varphi$. For each player i we write $\sigma_i[\varphi] = (Q_i[\varphi], q_i^0[\varphi], \tau_i[\varphi], \delta_i[\varphi])$. In virtue of Lemma 1 we know such strategies and strategy profiles exist.

Now consider the machine strategy profiles $\vec{\sigma}[\psi], \vec{\sigma}[\chi_1], ..., \vec{\sigma}[\chi_n]$. Without loss of generality we may assume that the sets of states of the machines involved are pairwise disjoint.

For every player i we construct a machine strategy $\sigma_i^* = (Q_i^*, q_i^{00}, \tau_i^*, \delta_i^*)$ that keeps track of what the other players do, chooses in accordance with $\sigma_i[\psi]$ as long as all other players j choose in accordance with $\vec{\sigma}[\psi]$ as well. However, if a player j is the only one to deviate from the behaviour prescribed by $\vec{\sigma}_j[\psi]$, player i reverts to playing $\sigma_i[\chi_j]$ thenceforth. If all players other than j adopt this strategy, j's efforts to achieve his goal $\mathbf{G}\varphi_j$ will be undermined. Each σ_i^* is obtained by

a rather straightforward product construction. We then show

(*i*) $\rho(\sigma_1^*, \ldots, \sigma_n^*) \models \psi$, and
(*ii*) $\vec{\sigma}^* = (\sigma_1^*, \ldots, \sigma_n^*) \in NE(G)$.

Formally, for each player $i \in N$, define the strategy $\sigma_i^* = (Q_i^*, q_i^{00}, \tau_i^*, \delta_i^*)$ as follows:

$$Q_i^* = (Q_1[\psi] \times \cdots \times Q_n[\psi]) \cup Q_i[\chi_1] \cup \cdots \cup Q_i[\chi_n]$$
$$q_i^{00} = (q_1^0[\psi], \ldots, q_n^0[\psi]).$$

We define $\tau_i^*: Q_i^* \to V_i$ such that for all $q_i \in Q_i^*$,

$$\tau_i^*(q_i) = \begin{cases} \tau_i[\psi](q_i[\psi]) & \text{if } q_i = (q_1[\psi], \ldots, q_n[\psi]), \\ \tau_i[\chi_j](q_i) & \text{if } q_i \in Q_i[\chi_j]. \end{cases}$$

And define $\delta_i^*: Q_i^* \times V \to Q_i^*$ such that for every $(q_1, \ldots, q_n)$ in the set $Q_1[\psi] \times \cdots \times Q_n[\psi]$ and every $\vec{v} \in V$,

$$\delta_i^*((q_1, \ldots, q_n), \vec{v}) = \begin{cases} q_i^0[\chi_j] & \text{if } v_j \neq \tau_j(q_j) \text{ and } v_k = \tau_k(q_k) \text{ for all } k \neq j, \\ (\delta_1[\psi](q_1, \vec{v}), \ldots, \delta_n[\psi](q_n, \vec{v})) & \text{otherwise.} \end{cases}$$

Finally, for all $q_i \in Q_i[\chi_j]$ for some $j \in N$,

$$\delta_i^*(q_i, \vec{v}) = \delta_i[\chi_j](q_i, \vec{v}).$$

Let the histories $h(\vec{\sigma})$ and $h(\vec{\sigma}^*)$ be given by, respectively,

$$\vec{q}[0] \xrightarrow{\vec{v}[0]} \vec{q}[1] \xrightarrow{\vec{v}[1]} \cdots, \text{ and}$$
$$\vec{r}[0] \xrightarrow{\vec{w}[0]} \vec{r}[1] \xrightarrow{\vec{w}[1]} \cdots.$$

It can then be shown by a routine induction that for all time points t and each player i both $r_i[t] = \vec{q}[t] = (q_1[\psi][t], \ldots, q_n[\psi][t])$ and $\vec{w}[t] = \vec{v}[t]$.

For (*i*), see that it now follows that $\rho(\vec{\sigma}^*) = \rho(\vec{\sigma})$ and, hence, $\rho(\vec{\sigma}^*) \models \psi$.

For (*ii*), assume for some player j that $\rho(\vec{\sigma}^*) \not\models \mathbf{G}\varphi_j$. In order to get a contradiction, also assume that there is a strategy $\sigma_j' = (Q_j', q_i^{000}, \delta_j', \tau_j')$ such that $\rho(\vec{\sigma}_{-j}^*, \sigma_j') \models \mathbf{G}\varphi_j$. Let the history $h(\vec{\sigma}_{-j}^*, \sigma_j')$, moreover, be given by

$$\vec{p}[0] \xrightarrow{\vec{u}[0]} \vec{p}[1] \xrightarrow{\vec{u}[1]} \cdots.$$

Obviously, $\rho(\vec{\sigma}^*) \neq \rho(\vec{\sigma}_{-j}^*, \sigma_j')$ and let t' be the earliest time point for which $\rho(\vec{\sigma}^*)[t'] \neq \rho(\vec{\sigma}_{-j}^*, \sigma_j')[t']$, that is, $\vec{w}[t'] \neq \vec{u}[t']$. Let $\sigma_j'' = (Q_j', p_j[t'+1], \delta_j', \tau_j')$, that is, the same strategy as σ_j' but with $p_j[t'+1]$ as start state instead of q_j^{000}. Moreover, let the history $h(\vec{\sigma}_{-i}[\chi_j], \sigma_j'')$ be given by

$$\vec{s}[0] \xrightarrow{\vec{x}[0]} \vec{s}[1] \xrightarrow{\vec{x}[1]} \cdots.$$

By induction it can then be shown that, for all time points t,

$$\vec{p}[t + t' + 1] = \vec{s}[t] \quad \text{and} \quad \vec{u}[t + t' + 1] = \vec{x}[t].$$

For the basis, observe that for j obviously, both

$$p_j[0 + t' + 1] = p_j[t' + 1] = s_j[0], \text{ and}$$
$$u_j[0 + t' + 1] = \tau_j'(p_j[0 + t' + 1]) = \tau_j'(s_j[0]) = x_j[0].$$

Now, consider an arbitrary player i distinct from j. It can be seen that then $w_j[t] = v_j[t] = \tau_j(q_j[\psi][t]) \neq u_j[t]$, whereas $w_k[t] = v_k[t] = \tau_k(q_k[\psi][t]) = u_k[t]$ for all k distinct from j. Hence, observing that $q_i^0[\chi_j] \in Q_i[\chi_j]$,

$$p_i[0 + t + 1] = \delta_i^*((q_1[\psi][t], \ldots, q_n[\psi][t]), \vec{u}[t]) = q_i^0[\chi_j] = s_i[0], \text{ and}$$
$$u_i[0 + t' + 1] = \tau_i^*(p_i[0 + t + 1]) = \tau_i^*(q_i^0[\chi_j]) = \tau_i[\chi_j](q_i^0[\chi_j]) = s_i[0].$$

The induction step follows by a routine argument.

It now follows that, for all formulas φ and all time points t,

$$\rho(\vec{\sigma}_{-j}[\chi_j], \sigma_j'')[t] \models \varphi \quad \text{if and only if} \quad \rho(\vec{\sigma}_{-j}^*, \sigma_j')[t + t' + 1] \models \varphi.$$

To conclude, recall that, by a previous observation, $\rho(\vec{\sigma}_{-i}[\chi_j], \sigma_j'') \models \chi_j$ and, hence, $\rho(\vec{\sigma}_{-i}[\chi_j], \sigma_j'') \models \neg\mathbf{G}\varphi_j$. Accordingly, we also have that $\rho(\vec{\sigma}_{-i}^*, \sigma_i')[t + 1] \models \neg\mathbf{G}\varphi_j$. Thus, there is t'' such that $\rho(\vec{\sigma}_{-i}^*, \sigma_i')[t + t' + 1] \not\models \varphi_j$. And hence, $\rho(\vec{\sigma}_{-i}^*, \sigma_i') \not\models \mathbf{G}\varphi_j$, a contradiction as desired. $\square$

7. Related work

Pure-strategy Nash equilibria in Boolean games have been studied previously (*e.g.*, in [8,4,13,5]) but, to the best of our knowledge, apart from a very recent exception (see [15]), not in game models where the players are allowed to interact for an infinite number of rounds, which is the setting our main results pertain to. In particular, in [15], the present framework was extended in order to consider both linear-time and branching-time goals as well as explicit structures (arenas, boards) where the games are played.

However, many games in logic and computer science have been considered where game plays are assumed to have infinite length; see, *e.g.*, [6,12] for surveys on the topic. These games are usually played by two players (in a hostile environment) who have independent and concurrent behaviour. In our setting, we are interested in situations where there are multiple players who do not necessarily have opposing objectives.

Two game models are worth singling out as being particularly relevant to our present work: the *non-zero-sum concurrent n-player games* with qualitative (binary) winning objectives in [6] and the *infinite concurrent multiplayer games* over a Boolean domain in [11]. The former model differs from ours in that (i) there is no explicit notion of control over variables—players simply choose moves in a given set—and (ii) the games are played on graphs. These two features of the games in [6] make them potentially exponentially larger than ours, which can have important implications when dealing with complexity issues. On the other hand, the games in [11] generalise our framework in two ways: first, they use a model of strategies that is more general than machine strategies and, secondly, they allow one player in the game (called the system player) to make choices after all other players have, independently and simultaneously, made their choices. As shown in this paper, the synthesis problem for both games has, however, the same complexity.

Finally, Folk Theorems for other games are mostly found in the game theory literature rather than in logic, computer science, or multi-agent systems research. Yet, there are a few exceptions. In [11] a similar question is asked for a problem—rational synthesis—that is closely related to one of the decision problems for iterated Boolean games. Other results on synthesis can be found too, see, *e.g.*, [7,28,19,18]. In most of this literature the emphasis is on the question whether a strategy can be synthesised that guarantees the satisfaction of an LTL formula. With the Folk Theorems, however, our focus is on semantic representations of complete classes of LTL properties which can be rationally sustained when playing a game.

8. Future work

In the model of iterated Boolean games we studied here the players' goals are represented by LTL formulae. It is also natural to consider players whose behaviour, as motivated by their goals, would be better described by branching-time temporal logics, such as CTL* [10] or the μ-calculus [17]. It seems likely that a framework of this kind would lead to systems (games) with different computational complexity properties.

Even within the linear-time setting, the use of LTL goals presents some limitations: It is known that LTL cannot express all linear-time ω-regular properties. It then seems reasonable to consider linear-time temporal logics with such expressive power, *e.g.*, a natural choice would be the linear-time μ-calculus [2] which elegantly extends LTL with extremal fixpoint operators. The study of a game model that extends iterated Boolean games, and which considers goals given by, *e.g.*, the linear-time or the modal μ-calculus, is presented in [15], where, amongst others, some hardness results for the computational complexity of the synthesis of strategies are provided.

Another important technical component of this work was the formalisation of conditions underpinning the Folk Theorems. Our study covered so-called (propositional) safety goals; other types of goals might be studied as well, for instance, goals with fairness or response properties. This study could be complemented by the definition of a purely semantic characterisation of punishability. Such a definition of punishability could be given in terms of runs and strategies, rather than in terms of the syntactic form of the players' goals. On the one hand, the definition should be strong enough to ensure that in any iterated Boolean game in which all players are punishable, every satisfiable formula is sustainable in a Nash equilibrium. At the same time, the concept should be broad enough so as to capture an as wide as possible range of Folk Theorems. In particular, Theorems 2 and 3 should follow as corollaries of a more general result involving this concept of punishability. It is also an additional challenge to syntactically characterise such a concept of punishability by a class of formulae that render a player punishable if such a player adopts any of them as a goal.[6]

The results in this paper are confined to games where the outcomes of the payoff sets are win-lose (binary) and the strategies are deterministic. Relaxing any of these two design choices may lead to computational problems. Recent work on games has shown that allowing randomization, either in the strategies [31] or in the arenas [29] where the games are played may result in scenarios where computing the Nash equilibria of a game is undecidable, even for reachability objectives which can easily be expressed in LTL. However with respect to binary payoff sets, decidability can be recovered [30].

Acknowledgments

---

[6] We thank one of the anonymous reviewers for suggesting this interesting problem.

Appendix A. Proofs from the main text

Lemma 3. *Let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ and $\vec{\sigma}' = (\sigma_1', \ldots, \sigma_n')$ be strategy profiles and i a player. Then,*

$$\rho(\vec{\sigma})|_{\Phi_i} = \rho(\vec{\sigma}_{-i}, \sigma_i')|_{\Phi_i} \text{ and } \rho(\vec{\sigma})|_{\Phi \backslash \Phi_i} = \rho(\vec{\sigma}_{-i}', \sigma_i)|_{\Phi \backslash \Phi_i} \text{ imply } \rho(\vec{\sigma}) = \rho(\vec{\sigma}').$$

Proof. Let $\sigma_k = (Q_k, q_k^0, \delta_k, \sigma_k')$ and $\sigma_k' = (Q_i', q_i^{00}, \delta_i', \tau_i')$, for each player k. Assume $\rho(\vec{\sigma})|_{\Phi_i} = \rho(\vec{\sigma}_{-i}, \sigma_i')|_{\Phi_i}$ and $\rho(\vec{\sigma})|_{\Phi \backslash \Phi_i} = \rho(\vec{\sigma}_{-i}', \sigma_i)|_{\Phi \backslash \Phi_i}$ Let the histories $h(\vec{\sigma})$, $h(\vec{\sigma}_{-i}, \sigma_i')$, $h(\vec{\sigma}_{-i}', \sigma_i)$, and $h(\vec{\sigma}')$, moreover, be given by, respectively,

$$h(\vec{\sigma}_{-i}, \sigma_i): \quad \vec{q}[0] \xrightarrow{\vec{v}[0]} \vec{q}[1] \xrightarrow{\vec{v}[1]} \cdots$$
$$h(\vec{\sigma}_{-i}, \sigma_i'): \quad \vec{p}[0] \xrightarrow{\vec{u}[0]} \vec{p}[1] \xrightarrow{\vec{u}[1]} \cdots$$
$$h(\vec{\sigma}_{-i}', \sigma_i): \quad \vec{s}[0] \xrightarrow{\vec{x}[0]} \vec{s}[1] \xrightarrow{\vec{x}[1]} \cdots$$
$$h(\vec{\sigma}_{-i}', \sigma_i'): \quad \vec{r}[0] \xrightarrow{\vec{w}[0]} \vec{r}[1] \xrightarrow{\vec{w}[1]} \cdots$$

We prove by (a routine) induction that, for all time points t and all players j distinct from i,

$$q_i[t] = s_i[t], \qquad p_i[t] = r_i[t], \qquad q_j[t] = p_j[t], \qquad s_j[t] = r_j[t],$$

as well as

$$\vec{v}[t] = \vec{u}[t] = \vec{x}[t] = \vec{w}[t].$$

For the basis, let j be an arbitrary player distinct from i. Observe that $q_i[0] = s_i[0]$, $p_i[0] = r_i[0]$, $q_j[0] = p_j[0]$, $s_j[0] = r_j[0]$ follow immediately from the definition of the strategy profiles considered. It now suffices to show that $\vec{v}[0] = \vec{u}[0]$, $\vec{v}[0] = \vec{x}[0]$, and $\vec{v}[0] = \vec{w}[0]$. To see $\vec{v}[0] = \vec{u}[0]$, observe that $v_i[0] = u_i[0]$ by our initial assumption and that,

$$v_j[0] = \tau(q_j[0]) = \tau_j(p_j[0]) = u_j[0].$$

Similarly, to see that $\vec{v}[0] = \vec{x}[0]$, observe that $v_j[0] = x_j[0]$ by the initial assumption and

$$v_i[0] = \tau(q_i[0]) = \tau(s_i[0]) = x_i[0].$$

Finally, for $\vec{v}[0] = \vec{w}[0]$, observe that $v_i[0] = u_i[0]$ and $v_j[0] = x_j[0]$ by the initial assumption. Now the following equations hold:

$$v_i[0] = u_i[0] = \tau_i'(p_i[0]) = \tau_i'(r_i[0]) = w_i[0],$$
$$v_j[0] = x_j[0] = \tau_j'(s_j[0]) = \tau_j'(r_j[0]) = w_j[0].$$

For the induction step, consider player i an arbitrary player j distinct from i. Now the following equations hold.

$$q_i[t+1] = \delta_i(q_i[t], \vec{v}[t]) =_{\text{i.h.}} \delta_i(s_i[t], x_i[t]) = s_i[t+1],$$
$$p_i[t+1] = \delta_i'(p_i[t], \vec{u}[t]) =_{\text{i.h.}} \delta_i(r_i[t], w_i[t]) = r_i[t+1],$$
$$q_j[t+1] = \delta_j(q_j[t], \vec{v}[t]) =_{\text{i.h.}} \delta_j(p_j[t], u_j[t]) = p_j[t+1],$$
$$s_j[t+1] = \delta_j'(q_j[t], \vec{v}[t]) =_{\text{i.h.}} \delta_j(r_j[t], w_j[t]) = w_j[t+1].$$

Finally, we show that $\vec{v}[t+1] = \vec{u}[t+1]$, $\vec{v}[t+1] = \vec{x}[t+1]$, and $\vec{v}[t+1] = \vec{w}[t+1]$. The argument is analogous to the one for the base case. To see $\vec{v}[t+1] = \vec{u}[t+1]$, observe that $v_i[t+1] = u_i[t+1]$ by the initial assumption and that,

$$v_j[t+1] = \tau(q_j[t+1]) = \tau_j(p_j[t+1]) = u_j[t+1].$$

Similarly, to appreciate that $\vec{v}[t+1] = \vec{x}[t+1]$, observe that $v_j[t+1] = x_j[t+1]$ by the initial assumption and

$$v_i[t+1] = \tau(q_i[t+1]) = \tau(s_i[t+1]) = x_i[t+1].$$

Finally, for $\vec{v}[t+1] = \vec{w}[t+1]$, observe that $v_i[t+1] = u_i[t+1]$ and $v_j[t+1] = x_j[t+1]$ by the initial assumption. Now the following equations hold:

$$v_i[t+1] = u_i[t+1] = \tau_i'(p_i[t+1]) = \tau_i'(r_i[t+1]) = w_i[t+1],$$
$$v_j[t+1] = x_j[t+1] = \tau_j'(s_j[t+1]) = \tau_j'(r_j[t+1]) = w_j[t+1].$$

This concludes the proof. $\quad\square$

Lemma 4. *Let $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ be a profile of (partial) strategies $\sigma_i = (Q_i, q_i^0, \delta_i, \tau_i)$ for an iterated Boolean game representing complete strategies with default states $q_1^d, \ldots, q_n^d$. Let, furthermore, $\rho : \mathbb{N} \to 2^{\Phi \cup Q \cup \{q_1^d, \ldots, q_n^d\}}$ an ω-regular run and $\vec{\sigma}' = (\sigma_1', \ldots, \sigma_n')$ a strategy profile such that $\rho|_\Phi = \rho(\vec{\sigma}')$. Then, for all players i,*

1. *$\rho \models th(\sigma_i)$ if and only if $\rho|_{\Phi \cup Q_i \cup \{q_i^d\}} = \hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)|_{\Phi \cup Q_i \cup \{q_i^d\}}$,*
2. *$\rho \models TH(\vec{\sigma})$ if and only if $\rho = \hat{\rho}(\vec{\sigma})$.*

Proof. For item 1 above, consider an arbitrary player i, an arbitrary run $\rho : \mathbb{N} \to 2^{\Phi \cup Q \cup \{q_1^d, \ldots, q_n^d\}}$ and a strategy profile $\vec{\sigma}' = (\sigma_1', \ldots, \sigma_n')$ such that $\rho|_\Phi = \rho(\vec{\sigma}')$. Let $\sigma_j' = (Q_j', q_j^{00}, \delta_j', \tau_j')$ for each player j. Let furthermore the histories $h(\vec{\sigma}'_{-i}, \sigma_i)$ and $h(\vec{\sigma}')$ be given by, respectively,

$$\vec{q}[0] \xrightarrow{\vec{v}[0]} \vec{q}[1] \xrightarrow{\vec{v}[1]} \cdots$$
$$\vec{r}[0] \xrightarrow{\vec{w}[0]} \vec{r}[1] \xrightarrow{\vec{w}[1]} \cdots$$

First assume that $\rho \models th(\sigma_i)$. Recall that

$$\hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)[t] = v[t] \cup \{q_1[t], \ldots, q_n[t]\}$$

for each time point t. It now suffices to show by induction, for all time points t, that

$$\rho[t] \cap (Q_i \cup \{q_i^d\}) = \{q_i[t]\} \quad \text{and} \quad \vec{v}[t] = \vec{w}[t],$$

as well as that $q_j[t] = r_j[t]$ for all players j distinct from i.

For the *basis*, let $t = 0$. Observe that $q_i[0] = q_i^0$. Moreover, $\rho \models INIT(\sigma_i)$ and, hence, $\rho[0] \models q_i^0$. Also, $\rho \models INVAR(\sigma_i)$, from which follows that $\rho[0] \not\models q_i'$ for all $q_i' \in Q_i \cup \{q_i^d\}$ with $q_i \neq q_i^0$. Accordingly,

$$\rho[0] \cap (Q_i \cup \{q_i^d\}) = \{q_i[0]\}.$$

Furthermore, $\rho \models VAL(\sigma_i)$. Hence, in particular, $\rho[0] \models q_i^0 \to \chi_{\tau(q_i^0)}^{\Phi_i}$. With $\rho[0] \models q_i^0$ we then obtain $\rho[0] \models \chi_{\tau_i(q_i^0)}^{\Phi_i}$. With $\rho|_\Phi = \rho(\vec{\sigma}')$, also $\rho(\vec{\sigma}', \sigma_i')[0] \models \chi_{\tau_i(q_i^0)}^{\Phi_i}$ and, thus, $\rho(\vec{\sigma}')[0] \cap \Phi_i = \tau_i(q_i^0)$. Then,

$$v_i[0] = \rho(\vec{\sigma}'_{-i}, \sigma_i)[0] \cap \Phi_i = \tau_i(q_i^0) = \rho(\vec{\sigma}')[0] \cap \Phi_i = w_i[0].$$

It can, moreover, easily be appreciated that for every player j distinct from i we have $q_j[0] = q_j^{00} = r_j[0]$ and

$$v_j[0] = \tau_j'(q_j[0]) = \tau_j'(r_j[0]) = w_j[0].$$

It follows that $\vec{v}[0] = \vec{w}[0]$, as desired.

For the *induction step*, we may assume that

$$\rho[t] \cap (Q_i \cup \{q_i^d\}) = \{q_i[t]\} \quad \text{and} \quad \vec{v}[t] = \vec{w}[t],$$

as well as that $q_j[t] = r_j[t]$ for all players j distinct from i. We show that,

$$\rho[t+1] \cap (Q_i \cup \{q_i^d\}) = \{q_i[t+1]\} \quad \text{and} \quad \vec{v}[t+1] = \vec{w}[t+1],$$

as well as that $q_j[t+1] = r_j[t+1]$ for all players j distinct from i.

First, consider an arbitrary player j distinct from i. By a routine application of the induction hypothesis, it can easily be seen that both $q_j[t+1] = r_j[t+1]$ and $v_j[t+1] = w_j[t+1]$.

Now distinguish the following cases. Either there is some $q_i' \in Q_i$ such that $\delta_i(q_i[t], \vec{v}[t]) = q_i'$ or there is no such $q_i' \in Q_i$.

If the former, $q_i' = \delta_i(q_i[t], \vec{v}[t]) = q_i[t+1]$. By the induction hypothesis we obtain $\rho[t] \models q_i[t]$. Since $\rho \models INVAR(\sigma_i)$, moreover, $\rho[t] \not\models q_i'$ for all states q_i' distinct from $q_i[t]$. By another application of the induction hypothesis, $\rho(\vec{\sigma}')[t] \models \chi_{\vec{v}[t]}^{\Phi_i}$. Recall that $\rho|_\Phi = \rho(\vec{\sigma}')$ and, thus, also $\rho[t] \models q[t]$. As, moreover, $\rho \models TRANS(\sigma_i)$, in particular,

$$\rho[t] \models (\chi_{\vec{v}[t]}^\Phi \wedge q_i[t]) \to \mathbf{X}\, q_i[t+1].$$

Hence, subsequently, $\rho[t] \models \mathbf{X}\, q_i[t+1]$ and $\rho[t+1] \models q_i[t+1]$. Moreover, $\rho \models INVAR(\sigma_i)$, from which follows that $\rho[t+1] \not\models q_i'$ or all $q_i' \in Q \cup \{q_i^d\}$ with $q_i' \neq q_i[t+1]$. Accordingly,

$$\rho[t+1] \cap (Q_i \cup \{q_i^d\}) = \{\vec{q}_i[t+1]\}.$$

Furthermore, since $\rho \models VAL(\sigma_i)$, in particular, $\rho[t+1] \models q_i[t+1] \to \chi^{\Phi}_{\tau(q_i[t+1])}$. Recall that $\rho[t+1] \models q_i[t+1]$ and, therefore, $\rho[t+1] \models \chi^{\Phi}_{\tau_i(q_i[t+1])}$. Recall that $\rho|_\Phi = \rho(\vec{\sigma}')$ and, hence, also $\rho(\vec{\sigma}'_{-i}, \sigma'_i)[t+1] \models \chi^{\Phi}_{\tau_i(q_i[t+1])}$. Therefore, $\rho(\vec{\sigma}'_{-i}, \sigma'_i)[t+1] \cap \Phi_i = \tau_i(q_i[t+1])$ and, thus,

$$v_i[t+1] = \rho(\vec{\sigma}'_{-i}, \sigma_i)[t+1] \cap \Phi_i = \tau_i(q_i^0) = \rho(\vec{\sigma}'_{-i}, \sigma'_i)[t+1] \cap \Phi_i = w_i[t+1].$$

We may conclude that in this case $v[t+1] = w[t+1]$.

If the latter, we have $q_i[t+1] = q_i^d$. By virtue of the induction hypothesis, we have $\rho[t] \models q_i[t]$. As $\rho \models INVAR(\sigma_i)$, moreover, $\rho \not\models q'_i$ for all $q'_i \neq q_i[t]$. By another application of the induction hypothesis, $\rho(\vec{\sigma}') \models \chi^{\Phi}_{\vec{v}[t]}$. As $\rho|_\Phi = \rho(\vec{\sigma})$, we also have $\rho[t] \models \chi^{\Phi}_{\vec{v}[t]}$. Now consider arbitrary $q_i, q'_i \in Q$ and $\vec{v}$ such that $\delta_i(q_i, \vec{v}) = q'_i$. Then, either $q_i \neq q[t]$ or $\vec{v} \neq \vec{v}[t]$. It follows that $\rho[t] \models \neg(\chi^{\Phi}_{\vec{v}} \wedge q_i)$ and, with q, q', and $\vec{v}$ having been chosen arbitrarily, $\rho[t] \models \bigwedge_{\delta_i(q_i, \vec{v})=q'_i} \neg(\chi^{\Phi}_{\vec{v}} \wedge q_i)$. As $\rho \models TRANS(\sigma_i)$, in particular,

$$\rho[t] \models \left(\bigwedge_{\delta_i(q_i, \vec{v})=q'_i} \neg(\chi^{\Phi}_{\vec{v}} \wedge q_i) \right) \to \mathbf{X} q_i^d.$$

Therefore, $\rho[t] \models \mathbf{X} q_i^d$ and, subsequently, $\rho[t+1] \models q_i^d$. Also observe that, as $\rho \models INVAR(\sigma_i)$, also $\rho[t+1] \not\models q_i$ for every q_i distinct from q_i^d. It now follows that

$$\rho(\vec{\sigma}')[t+1] \cap (Q_i \cup \{q_i^d\}) = \{q_i[t+1]\}.$$

To appreciate $v_i[t+1] = w_i[t+1]$, observe that

$$v_i[t+1] = \tau_i(q_i[t+1]) = \tau_i(q_i^d) = \emptyset.$$

Moreover, in virtue of $\rho \models VAL(\sigma_i)$, we have $\rho[t+1] \models q_i^d \to \chi^{\Phi_i}_\emptyset$. Also, $\rho[t+1] \models q_i^d$, and, hence, $\rho[t+1] \models \chi^{\Phi_i}_\emptyset$. Recall that $\rho|_\Phi = \rho(\vec{\sigma}')$, and, hence, $\rho(\vec{\sigma}') \models \chi^{\Phi_i}_\emptyset$. Therefore, $\rho(\vec{\sigma}') \cap \Phi_i = \emptyset$ and, thus,

$$v_i[t+1] = \emptyset = \rho(\vec{\sigma}'_{-i}, \sigma'_i)[t+1] \cap \Phi_i = w_i[t+1].$$

Also in this case, we may conclude that $\vec{v}[t+1] = \vec{w}[t+1]$.

For the opposite direction, assume $\rho|_{\Phi_i \cup Q_i \cup \{q_i^d\}} = \hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)|_{\Phi_i \cup Q_i \cup \{q_i^d\}}$. From the definition of $\hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)$ it immediately follows that for all $t \in \mathbb{N}$,

$$\hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)[t] \models \chi^{\Phi_i}_{w_i[t]}, \qquad \hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)[t] \models \chi^{\Phi}_{\vec{w}[t]}, \qquad \hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)[t] \models r_i[t],$$

and, for all $q'_i \in Q_i$ with $q'_i \neq r_i[t]$, moreover, $\hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)[t] \not\models q'_i$. With our assumption, for all time points t and all $q'_i \in Q_i$ with $q'_i \neq q_i[t]$, we then also have that,

$$\hat{\rho}[t] \models \chi^{\Phi_i}_{v_i[t]}, \qquad \hat{\rho}[t] \models \chi^{\Phi}_{\vec{v}[t]}, \qquad \hat{\rho}[t] \models q_i[t],$$

and, for all $q'_i \in Q_i$ with $q'_i \neq q_i[t]$, moreover, $\hat{\rho}(\vec{\sigma}'_{-i}, \sigma_i)[t] \not\models q'_i$. It can now straightforwardly be shown that,

$$\rho \models INIT(\sigma_i) \wedge TRANS(\sigma_i) \wedge INVAR(\sigma_i) \wedge VAL(\sigma_i).$$

The tedious details we omit.

Finally, for 2, first assume that $\rho = \hat{\rho}(\vec{\sigma})$. Then, in particular, $\rho|_\Phi = \rho(\vec{\sigma})$. Now consider an arbitrary player i. Then, moreover, $\rho|_{\Phi \cup Q_i \cup \{q_i^d\}} = \rho(\vec{\sigma}_{-i}, \sigma_i)|_{\Phi \cup Q_i \cup \{q_i^d\}}$. By 1, it then follows that $\rho \models th(\sigma_i)$. Having chosen i arbitrarily, we also have $\rho \models TH(\vec{\sigma})$. For the opposite direction, assume $\rho \models TH(\vec{\sigma})$. Thus, $\rho \models th(\sigma_j)$ for all players j. Consider an arbitrary player i. Without loss of generality and for ease of notation, we assume that $i = n$. Now consider the sequence of strategy profiles

$$\vec{\sigma}^1 = (\sigma'_1, \sigma'_2, \sigma'_3, \ldots, \sigma'_{n-1}, \sigma'_n),$$
$$\vec{\sigma}^2 = (\sigma_1, \sigma'_2, \sigma'_3, \ldots, \sigma'_{n-1}, \sigma'_n),$$
$$\vec{\sigma}^3 = (\sigma_1, \sigma_2, \sigma'_3, \ldots, \sigma'_{n-1}, \sigma'_n),$$
$$\vdots$$
$$\vec{\sigma}^n = (\sigma_1, \sigma_2, \sigma_3, \ldots, \sigma_{n-1}, \sigma'_n).$$

We show for all j with $1 \leq j \leq n$,

$$\rho|_{\Phi \cup Q_j \cup \{q_j^d\}} = \hat{\rho}(\vec{\sigma}^j_{-j}, \sigma_j)|_{\Phi \cup Q_j \cup \{q_j^d\}}.$$

As $\rho \models th(\sigma_1)$, the basis is immediate by 1. For the induction step, assume that $\rho|_{\Phi \cup Q_j \cup \{q_j^d\}} = \hat{\rho}(\vec{\sigma}_{-j}^j, \sigma_j)|_{\Phi \cup Q_j \cup \{q_j^d\}}$. Then, in particular, $\rho|_\Phi = \rho(\vec{\sigma}_{-j}^j, \sigma_j) = \rho(\vec{\sigma}^{j+1})$. As $\rho \models th(\sigma_{j+1})$, an application of 1 yields

$$\rho|_{\Phi \cup Q_{j+1} \cup \{q_{j+1}^d\}} = \hat{\rho}(\vec{\sigma}_{-j+1}^{j+1}, \sigma_{j+1})|_{\Phi \cup Q_{j+1} \cup \{q_{j+1}^d\}}.$$

For $i = n$, we then find that

$$\rho|_{\Phi \cup Q_n \cup \{q_n^d\}} = \hat{\rho}(\vec{\sigma}_{-n}^n, \sigma_n)|_{\Phi \cup Q_n \cup \{q_n^d\}} = \hat{\rho}(\vec{\sigma})|_{\Phi \cup Q_n \cup \{q_n^d\}}.$$

It follows that $\rho|_{\Phi \cup Q_i \cup \{q_i^d\}} = \hat{\rho}(\vec{\sigma})|_{\Phi \cup Q_i \cup \{q_i^d\}}$ for all players i. We may conclude that $\rho = \hat{\rho}(\vec{\sigma})$. $\quad\square$

Proposition 4. E-DOMINANT, A-DOMINANT, *and* NON-EMPTINESS *for Dominant Strategies are 2EXPTIME-complete.*

Proof. Membership in 2EXPTIME is proven exactly as for Nash equilibrium, using [11]. For hardness, we also first use the reduction from LTL realizability to its variant (namely, the R'-realizability game $R' = (\varphi', I, O)$) where both the system and the environment play simultaneously. Then, we prove the following claim: that the LTL formula φ' is R'-realizable if and only if $DS(G) \neq \emptyset$, where $DS(G)$ is the set of dominant strategy equilibria of the iterated Boolean game G, defined as described below.

Given an R'-realizability game $R' = (\varphi', I, O)$, create a two-player iterated Boolean game G where player 1 controls the variables in I as well as a fresh Boolean variable x_1, player 2 controls the variables in O as well as a fresh Boolean variable x_2, and the goals are $\gamma_1 = \varphi' \vee (x_1 \leftrightarrow x_2)$ and $\gamma_2 = \bot$. Variables x_1 and x_2 are used to show that player 1 has a dominant strategy (winning) that satisfies φ'. Now, for the left-to-right direction, assume that φ' is R'-realizable, and let σ_1 be the machine strategy corresponding to the winning strategy for φ' and σ_2 be any strategy. We claim that $(\sigma_1, \sigma_2) \in DS(G)$. Observe that σ_1 ensures γ_1, thus it is a dominant strategy for player 1. On the other hand, any σ_2 is a dominant strategy for player 2 since γ_2 can never be satisfied. Hence $(\sigma_1, \sigma_2) \in DS(G)$. For the right-to-left direction, assume $DS(G) \neq \emptyset$, and let the profile $\vec{\sigma} = (\sigma_1, \sigma_2) \in DS(G)$. From the definition of dominant strategy equilibrium we know that

$$\rho(\sigma_1, \sigma_2') \succsim_1 \rho(\sigma_1', \sigma_2')$$

for all strategies σ_1' and σ_2' for player 1 and player 2, respectively. In other words, we know that

$$\text{if } \rho(\sigma_1, \sigma_2') \not\models \gamma_1 \text{ then } \rho(\sigma_1', \sigma_2') \not\models \gamma_1.$$

We first show that $\rho(\vec{\sigma}) \models \gamma_1$. Let σ_2' be σ_2. Then, for a contradiction, assume that $\rho(\vec{\sigma}) \not\models \gamma_1$. It follows that $\rho(\sigma_1', \sigma_2') \not\models \gamma_1$, for any σ_1'. But, necessarily, there is σ_1' for which $x_1 = x_2$, thus satisfying γ_1—which is a contradiction. Therefore, we have that $\rho(\vec{\sigma}) \models \gamma_1$.

Now, we show that, in addition, $\rho(\vec{\sigma}) \models \varphi'$. Again, for a contradiction, suppose that $\rho(\vec{\sigma}) \not\models \varphi'$. Because $\rho(\vec{\sigma}) \models \gamma_1$, it must be the case that $\rho(\vec{\sigma}) \models (x_1 \leftrightarrow x_2)$. Let σ_2' be the strategy that plays exactly as σ_2, save that it plays a different value for x_2, i.e., x_2 false/true in σ_2' if true/false in σ_2. Then, $\rho(\sigma_1, \sigma_2') \not\models \gamma_1$. As before, it then follows that $\rho(\sigma_1', \sigma_2') \not\models \gamma_1$, for any σ_1'. But, necessarily, there is σ_1' for which $x_1 = x_2$, thus satisfying the goal γ_1—which is a contradiction. Therefore, we have that $\rho(\vec{\sigma}) \models \varphi'$ too.

Now, we show that σ_1 is, in fact, a winning strategy for φ'. Suppose, for a contradiction, that σ_1 is not. Then, there is σ_2' such that $\rho(\sigma_1, \sigma_2') \not\models \varphi'$. Let, without loss of any generality, σ_2' be such that $\rho(\sigma_1, \sigma_2') \not\models (x_1 \leftrightarrow x_2)$. Then, $\rho(\sigma_1, \sigma_2') \not\models \gamma_1$. It follows that $\rho(\sigma_1', \sigma_2') \not\models \gamma_1$, for any σ_1'. But, there is σ_1' for which $x_1 = x_2$, thus satisfying γ_1—which is a contradiction. Therefore, it follows that $\rho(\sigma_1, \sigma_2') \models \varphi'$, for all σ_2', that is, σ_1 is a strategy that ensures φ'. It thus follows that σ_1 encodes a winning strategy f' for φ' in the R'-realizability game, and hence φ' is R'-realizable.

We can conclude that NON-EMPTINESS for Dominant Strategies in iterated Boolean games is a 2EXPTIME-complete problem. The hardness results for the E-NASH and A-NASH problems immediately follow from that of NON-EMPTINESS, as done in the case of Nash equilibrium. $\quad\square$

Proposition 5. E-CSPNE, A-CSPNE, *and* NON-EMPTINESS *with respect to Consistent-Subgame Perfect Nash equilibria are 2EXPTIME-complete.*

Proof. Membership in 2EXPTIME and hardness, up to the end of the construction of the 4-player game, is as for Nash equilibrium, but we claim, instead, that φ' is R'-realizable iff $CSPNE(G) \neq \emptyset$, where $CSPNE(G)$ is the set of consistent-subgame perfect Nash equilibria of G. We now show that a small variant of the argument for Nash equilibria holds in this case.

For the left-to-right direction, assume φ' is R'-realizable, and let σ_1 be the machine strategy corresponding to the winning strategy for φ'. Also, let $\sigma_2, \sigma_3, \sigma_4$ be any strategies for players 2, 3, and 4. We now claim that $\vec{\sigma} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \in CSPNE(G)$. Take any reachable subgame G_π, that is, a subgame where $\pi \in \{\rho(\vec{\sigma})[0, \ldots, k] : k \in \mathbb{N}\}$. The π-substrategies in this subgame are $(\sigma_i^\pi)_{i \in N}$, each with initial state given by $q_i^0 = h(\vec{\sigma})[|\pi|]_i$. First, observe that since π is consistent with $\vec{\sigma}$, then there is a run ρ' such that $\pi; \rho' \models \varphi'$. Since player 1 is using a winning strategy to satisfy φ', and π is consistent with

such a winning strategy, then, in fact, every ρ', consistent with σ_1^π, also satisfies φ'. That is, $\pi; \rho' \models \gamma_1$, for all such ρ'. Then, the goal of player 1 is also satisfied in the subgame G_π, and hence also the goals of players 3 and 4. And, as for Nash equilibrium, a beneficial deviation for player 2 requires a $\sigma_2'^\pi$ such that $\pi; \rho(\sigma_1^\pi, \sigma_2'^\pi, \sigma_3^\pi, \sigma_4^\pi) \not\models \varphi'$, which is impossible since σ_1^π encodes a winning π-substrategy for φ'; hence $(\sigma_1^\pi, \sigma_2^\pi, \sigma_3^\pi, \sigma_4^\pi) \in NE(G_\pi)$. Since the finite run π was chosen arbitrarily, the result necessarily holds for all π in $\{\rho(\vec{\sigma})[0, \ldots, k] : k \in \mathbb{N}\}$—and hence for all subgames G_π consistent with the strategy profile $\vec{\sigma}$—implying that $\vec{\sigma} \in CSPNE(G)$.

The right-to-left direction is almost immediate. Simply assume that $CSPNE(G) \neq \emptyset$, and let $\vec{\sigma} = (\sigma_1, \sigma_2, \sigma_3, \sigma_4) \in CSPNE(G)$. Observe that $\vec{\sigma}$ is, in particular, also a Nash equilibrium. Therefore, exactly the same argument for Nash equilibrium strategy profiles holds to ensure that σ_1 encodes a winning strategy f' for φ', and hence, to show that φ' is R'-realizable. $\square$

Lemma 5. *Let* $G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$ *be an iterated Boolean game in which player* i *has a propositional safety goal* $\gamma_i = \mathbf{G}\,\varphi_i$. *Then, if player* i *is not punishable, for all* $\vec{\sigma} \in NE(G)$ *we have that* $\rho(\vec{\sigma}) \models \gamma_i$.

Proof. Without loss of generality let $i = n$ and assume that player n is not punishable. Thus, $\bigvee_{v_{-n} \in V_{-n}} \bigwedge_{v_n' \in V_n} \left(\chi_{(v_{-n}, v_n')} \rightarrow \neg\varphi_n \right)$ is not valid, that is, $\bigwedge_{v_{-n} \in V_{-n}} \bigvee_{v_n' \in V_n} \left(\chi_{(v_{-n}, v_n')} \wedge \varphi_n \right)$ is satisfiable. From the latter it can easily be seen that for all $v_{-n} \in V_{-n}$ there is some $v_n' \in V_n$ such that $(v_{-n}, v_n') \models \varphi_n$ and we may assume the existence of a function $f_n : V_{-n} \rightarrow V_n$ such that, for every valuation $(v_1, \ldots, v_{n-1}) \in V_{-n}$,

$$\rho(v_1, \ldots, v_{n-1}, f_n(v_1, \ldots, v_{n-1})) \models \varphi_n.$$

Now consider an arbitrary strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$, where $\sigma_i = (Q_j, q_j^0, \tau_j, \delta_j)$ for each player j, and assume that $\rho(\vec{\sigma}) \not\models \mathbf{G}\,\varphi_n$. To show that $\vec{\sigma} \notin NE(G)$, it suffices to construct a machine strategy σ_n^* for n such that $\rho(\vec{\sigma}_{-n}, \sigma_n^*) \models \mathbf{G}\,\varphi_n$.

Define $\sigma_n^* = (Q_n^*, q_n^{00}, \tau_n^*, \delta_n^*)$ in such a way that

$$Q_n^* = Q_1 \times \cdots \times Q_{n-1} \times V$$
$$q_n^{00} = (q_1^0, \ldots, q_{n-1}^0, (v_1^0, \ldots, v_n^0)),$$

where $(v_1^0, \ldots, v_n^0)$ is such that for all players i,

$$v_j^0 = \begin{cases} f_n(\tau_1(q_1^0), \ldots, \tau_{n-1}(q_{n-1}^0)) & i = n, \\ \tau_j(q_j^0) & \text{otherwise.} \end{cases}$$

Also define $\tau_n^* : Q_n^* \rightarrow V_n$ such that for all $(q_1, \ldots, q_{n-1}, (v_1, \ldots, v_n)) \in Q_n^*$,

$$\tau_n^*(q_1, \ldots, q_{n-1}, (v_1, \ldots, v_n)) = f_n(\tau_1(q_1), \ldots, \tau_{n-1}(q_{n-1})).$$

Moreover, for every state $(q_1, \ldots, q_{n-1}, \vec{v}) \in Q_n^*$ and all valuations $\vec{w} \in V$, set

$$\delta_n^*((q_1, \ldots, q_{n-1}, \vec{v}), \vec{w}) = (\delta_1(q_1, \vec{v}), \ldots, \delta_n(q_{n-1}, \vec{v}), \vec{v}^*),$$

where $\vec{v}^* = (v_1^*, \ldots, v_n^*)$ is such that, for all players $j \in N$,

$$v_j^* = \begin{cases} \tau_n^*(\delta_1(q_1, \vec{v}), \ldots, \delta_{n-1}(q_{n-1}, \vec{v}), \vec{v}) & \text{if } j = n, \\ \tau_j(\delta_j(q_j, \vec{v})) & \text{otherwise.} \end{cases}$$

Observe that δ_n^* does not depend on its second argument $\vec{w}$.

To show that $\rho(\vec{\sigma}_{-n}, \sigma_n^*) \models \mathbf{G}\,\varphi_n$, consider the history $h(\sigma_1, \ldots, \sigma_{n-1}, \sigma_n^*)$ and assume it to be given by,

$$\vec{r}[0] \xrightarrow{\vec{w}[0]} \vec{r}[1] \xrightarrow{\vec{w}[1]} \cdots$$

By a straightforward induction on t, it can be shown that for all $\vec{r}[t] = (r_1[t], \ldots, r_n[t])$ and $\vec{w}[t] = (w_1[t], \ldots, w_n[t])$: we have:

$$r_j[t] = (r_1[t], \ldots, r_{n-1}[t], \vec{w}[t])$$
$$w_j[t] = f_n(w_1[t], \ldots, w_{n-1}[t]).$$

Intuitively, the former says that σ_n^* correctly predicts the behaviour of $\vec{\sigma}_{-n}$ at each time t, the latter that σ_n^* always selects the appropriate and winning answer to this behaviour.

We may conclude that for every time point $t \in \mathbb{N}$,

$$\rho(\vec{\sigma}_{-n}, \sigma_n^*)[t] = (w_1[t], \ldots, w_{n-1}[t], f_n(w_1[t], \ldots, w_{n-1}[t])).$$

Hence, $\rho(\vec{\sigma}_{-n}, \sigma_n^*)[t] \models \varphi_n$ and, accordingly, $\rho(\vec{\sigma}_{-n}, \sigma_n^*) \models \mathbf{G}\,\varphi_n$. Having assumed that $\rho(\vec{\sigma}) \not\models \mathbf{G}\,\varphi_i$, we may conclude that $\vec{\sigma} \notin NE(G)$. $\square$

Theorem 1. *Let $G = (N, \Phi, \Phi_1, \ldots, \Phi_n, \gamma_1, \ldots, \gamma_n)$ be an iterated Boolean game in which each player i has a propositional safety goal $\gamma_i \equiv \mathbf{G}\varphi_i$. Let, furthermore, ψ be an LTL formula such that $\psi \wedge \bigwedge\{\gamma_i : i \text{ is not punishable}\}$ is satisfiable. Then, there is a $\vec{\sigma} \in NE(G)$ with $\rho(\vec{\sigma}) \models \psi$.*

Proof. Having assumed that $\psi \wedge \bigwedge\{\gamma_i : i \text{ is not punishable}\}$ is satisfiable, by Lemma 1, there is some strategy profile $\vec{\sigma} = (\sigma_1, \ldots, \sigma_n)$ such that $\rho(\vec{\sigma}) \models \psi \wedge \bigwedge\{\gamma_i : i \text{ is not punishable}\}$. Moreover, for every punishable player j, we have that

$$\bigvee_{v_{-j} \in V_{-j}} \bigwedge_{v'_j \in V_j} (\chi_{(v_{-j}, v'_j)} \to \neg\varphi_j) \text{ is valid.}$$

Thus, for every punishable player j, there is some $(v_1^j, \ldots, v_n^j) \in V$ such that for all $v'_j \in V_j$,

$$(v_1^j, \ldots, v_{j-1}^j, v'_j, v_{j+1}^j, \ldots, v_n^j) \not\models \varphi_j.$$

For every player i, punishable or not punishable, we construct a machine strategy $\sigma_i^* = (Q_i^*, q_i^0, \tau_i^*, \delta_i^*)$ that keeps track of what the other players do, chooses in accordance with σ_i if all other players j choose according to σ_j, but chooses v_i^j at $t+1$ whenever j is the only punishable player whose choice at t deviates from $\rho(\vec{\sigma})[t] \cap \Phi_j$. We then show that

(a) $\rho(\sigma_1^*, \ldots, \sigma_n^*) \models \psi$, and
(b) $\vec{\sigma}^* = (\sigma_1^*, \ldots, \sigma_n^*) \in NE(G)$.

Formally, for each player i we define $\sigma_i^* = (Q_i^*, q_i^{00}, \tau_i^*, \delta_i^*)$ such that

$$Q_i^* = (Q_1 \times \cdots \times Q_n) \cup \{q_i^1, \ldots, q_i^{i-1}, q_i^{i+1}, \ldots, q_i^n\}, \text{ and}$$
$$q_i^{00} = (q_1^0, \ldots, q_n^0).$$

Moreover, let $\tau_i^* : Q_i^* \to V_i$ be such that for all $q_i^* \in Q_i^*$,

$$\tau_i^*(q_i^*) = \begin{cases} \tau_i(q_i) & \text{if } q_i^* = (q_1, \ldots, q_n), \\ v_i^j & \text{if } q_i^* = q_i^j \text{ for some punishable } j \neq i, \\ \emptyset & \text{otherwise,} \end{cases}$$

where $q_i^1, \ldots, q_i^{i-1}, q_i^{i+1}, \ldots, q_i^n$ are fresh and mutually distinct. Finally, we define $\delta_i^* : (Q_i^* \cup V) \to Q_i^*$ such that for all $\vec{v} = (v_1, \ldots, v_n)$ in V, all $q_i^* = (q_1, \ldots, q_n)$ in $Q_1 \times \cdots \times Q_n$,

$$\delta_i^*(q_i^*, \vec{v}) = \begin{cases} q_i^j & \text{if } v_j \neq \tau_j(q_j) \text{ and } v_k = \tau_k(q_k) \text{ for all } k \neq j, \\ (\delta_1(q_1, \vec{v}), \ldots, \delta_n(q_n, \vec{v})) & \text{otherwise.} \end{cases}$$

Finally, for all $\vec{v} = (v_1, \ldots, v_n)$ in V and all $j \neq i$ we stipulate that,

$$\delta_i^*(q_i^j, \vec{v}) = q_i^j.$$

For (a), take the histories $h(\vec{\sigma})$ and $h(\vec{\sigma}^*)$ and assume they are given by

$$\vec{q}[0] \xrightarrow{\vec{v}[0]} \vec{q}[1] \xrightarrow{\vec{v}[1]} \cdots, \text{ and}$$
$$\vec{r}[0] \xrightarrow{\vec{w}[0]} \vec{r}[1] \xrightarrow{\vec{w}[1]} \cdots,$$

respectively. By a straightforward induction on t it follows that, for all $t \in \mathbb{N}$,

$$\vec{r}[t] = ((q_1[t], \ldots, q_n[t]), \ldots, (q_1[t], \ldots, q_n[t])), \text{ and}$$
$$\vec{w}[t] = \vec{v}[t].$$

Hence, for all $t \in \mathbb{N}$, we have $\rho(\vec{\sigma})[t] = \rho(\vec{\sigma}^*)[t]$; it follows that $\rho(\vec{\sigma}^*) \models \psi$, as desired, and that $\rho(\vec{\sigma}^*) \models \gamma_k$ for each player k who is not punishable.

To show (b), assume for contradiction that $\vec{\sigma}^*$ is not a Nash equilibrium. Then, there is a player j and some $\sigma'_j \in \Sigma_j$ such that $\rho(\vec{\sigma}^*) \not\models \gamma_j$ as well as $\rho(\sigma_{-j}^*, \sigma'_j) \models \gamma_j$. Observe that j has to be a punishable player. Moreover, it is obvious that $\rho(\vec{\sigma}^*) \neq \rho(\sigma_{-j}^*, \sigma'_j)$. Accordingly, let t be the *earliest* time point such that

$$\rho(\vec{\sigma}^*)[t] \neq \rho(\sigma_{-j}^*, \sigma'_j)[t].$$

Let the histories $h(\vec{\sigma}^*)$ and $h(\vec{\sigma}^*_{-j}, \sigma'_j)$ for $\vec{\sigma}$ and $\vec{\sigma}^*$ up to $t+1$ be given by, respectively,

$$\vec{r}[0] \xrightarrow{\vec{w}[0]} \vec{r}[1] \xrightarrow{\vec{w}[1]} \cdots \xrightarrow{\vec{w}[t-1]} \vec{r}[t] \xrightarrow{\vec{w}[t]} \vec{r}[t+1] \xrightarrow{\vec{w}[t+1]},$$

$$\vec{s}[0] \xrightarrow{\vec{u}[0]} \vec{s}[1] \xrightarrow{\vec{u}[1]} \cdots \xrightarrow{\vec{u}[t-1]} \vec{s}[t] \xrightarrow{\vec{u}[t]} \vec{s}[t+1] \xrightarrow{\vec{u}[t+1]}.$$

Then, $\vec{w}[t'] = \vec{u}[t']$ for all $t' < t$, and $\vec{v}[t] \neq \vec{w}[t]$. Now, consider an arbitrary player $i \neq j$. Recall that player i's strategy in $(\vec{\sigma}^*_{-j}, \sigma'_j)$ is σ^*_i. As $\vec{u}[t'] = \vec{w}[t']$ for all $t' < t$, it follows that, for all $t' \leq t$,

$$s_i[t'] = r_i[t'] = (q_1[t'], \ldots, q_n[t']).$$

In particular this holds for $t' = t$ and hence, $s_i[t] = (q_1[t], \ldots, q_n[t])$. Now,

$$u_i[t] = \tau^*(s_i[t]) = \tau^*(q_1[t], \ldots, q_n[t]) = \tau(q_i[t]).$$

Thus, having chosen i arbitrarily, we have,

$$u_k[t] = \tau(q_k[t]) \quad \text{for all } k \neq j.$$

As $u[t] \neq w[t]$, it moreover follows that, $u_j[t] \neq w_j[t]$. Also observe that

$$w_j[t] = v_j[t] = \tau^*_j(q[t]) = \tau^*(q_1[t], \ldots, q_n[t]) = \tau_j(q_j[t]).$$

Hence, $u_j[t] \neq \tau_j(q_j)$. Accordingly, we obtain

$$\vec{u}[t] = (\tau(q_1[t]), \ldots, \tau(q_{j-1}[t]), u_j[t], \tau(q_{j+1}[t]), \ldots, \tau(q_n[t])),$$

where $u_j[t] \neq \tau_j(q_j)$. Now, observe that

$$s_i[t+1] = \delta^*_i(s_i[t], \vec{u}[t]) = \delta^*_i((q_1[t], \ldots, q_n[t]), \vec{u}[t]) = \delta^*_i(q^*_i[t], \vec{u}[t]) = q^j_i.$$

As a consequence we also have

$$u_i[t+1] = \tau^*_i(s_i[t+1]) = \tau^*_i(q^j_i) = v^j_i.$$

With i having been chosen arbitrarily,

$$\vec{u}[t+1] = (v^j_1, \ldots, v^j_{j-1}, u_j[t+2], v^j_{j+1}, \ldots, v^j_n).$$

As we know that $(v^j_1, \ldots, v^j_{j-1}, v'_j, v^j_{j+1}, \ldots, v^j_n) \not\models \varphi_j$ for all $v'_j \in V_j$, then,

$$(v^j_1, \ldots, v^j_{j-1}, u_j[t+1], v^j_{j+1}, \ldots, v^j_n) \not\models \varphi_j.$$

Since

$$(v^j_1, \ldots, v^j_{j-1}, v'_j, v^j_{j+1}, \ldots, v^j_n) = u[t+1] = \rho(\sigma^*_{-j}, \sigma'_j),$$

we obtain $(\rho(\sigma^*_{-j}, \sigma'_j), t+1) \not\models \varphi_j$. It follows that $\rho(\sigma^*_{-j}, \sigma'_j) \not\models \mathbf{G}\varphi_j$ and we can conclude that $\rho(\sigma^*_{-j}, \sigma'_j) \not\models \gamma_j$, a contradiction. □

References

[1] R. Aumann, L. Shapley, Long-term competition: a game-theoretic analysis, in: Essays in Game Theory in Honor of Michael Maschler, Springer, 1994, pp. 1–15.
[2] H. Barringer, R. Kuiper, A. Pnueli, A really abstract concurrent model and its temporal logic, in: POPL, ACM Press, 1986, pp. 173–183.
[3] K. Binmore, Fun and Games, D. C. Heath and Company, 1992.
[4] E. Bonzon, M.C. Lagasquie-Schiex, J. Lang, Dependencies between players in Boolean games, Int. J. Approx. Reason. 50 (6) (2009) 899–914.
[5] J. Bradfield, J. Gutierrez, M. Wooldridge, On the structure of events in Boolean games, in: LOFT, 2014.
[6] K. Chatterjee, T.A. Henzinger, A survey of stochastic ω-regular games, J. Comput. Syst. Sci. 78 (2) (2012) 394–413.
[7] K. Chatterjee, T.A. Henzinger, B. Jobstmann, Environment assumptions for synthesis, in: CONCUR, in: LNCS, vol. 5201, Springer, 2008, pp. 147–161.
[8] P.E. Dunne, W. van der Hoek, S. Kraus, M. Wooldridge, Cooperative Boolean games, in: AAMAS (2), IFAAMAS, 2008, pp. 1015–1022.
[9] E.A. Emerson, Temporal and modal logic, in: Handbook of Theoretical Computer Science, vol. B: Formal Models and Semantics (B), Elsevier, 1990, pp. 995–1072.
[10] E.A. Emerson, J.Y. Halpern, "Sometimes" and "not never" revisited: on branching versus linear time temporal logic, J. ACM 33 (1) (1986) 151–178.
[11] D. Fisman, O. Kupferman, Y. Lustig, Rational synthesis, in: TACAS, in: LNCS, vol. 6015, Springer, 2010, pp. 190–204.
[12] E. Grädel, M. Ummels, Solution concepts and algorithms for infinite multiplayer games, in: New Perspectives on Games and Interaction, in: Texts in Logic and Games, vol. 4, Amsterdam University Press, 2008, pp. 151–178.
[13] J. Grant, S. Kraus, M. Wooldridge, I. Zuckerman, Manipulating Boolean games through communication, in: IJCAI, IJCAI/AAAI, 2011, pp. 210–215.
[14] J. Gutierrez, P. Harrenstein, M. Wooldridge, Iterated Boolean games, in: IJCAI, IJCAI/AAAI, 2013.

[15] J. Gutierrez, P. Harrenstein, M. Wooldridge, Reasoning about equilibria in game-like concurrent systems, in: KR, AAAI Press, 2014.
[16] P. Harrenstein, W. van der Hoek, J.J. Meyer, C. Witteveen, Boolean games, in: TARK VIII, 2001, pp. 287–298.
[17] D. Kozen, Results on the propositional mu-calculus, Theor. Comput. Sci. 27 (1983) 333–354.
[18] O. Kupferman, P. Madhusudan, P.S. Thiagarajan, M.Y. Vardi, Open systems in reactive environments: control and synthesis, in: CONCUR, in: LNCS, vol. 1877, Springer, 2000, pp. 92–107.
[19] Y. Lustig, M.Y. Vardi, Synthesis from component libraries, Int. J. Softw. Tools Technol. Transf. 15 (5–6) (2013) 603–618.
[20] Z. Manna, A. Pnueli, The Temporal Logic of Reactive and Concurrent Systems – Specification, Springer, 1992.
[21] Z. Manna, A. Pnueli, Temporal Verification of Reactive Systems – Safety, Springer, 1995.
[22] M.J. Osborne, A. Rubinstein, A Course in Game Theory, The MIT Press, 1994.
[23] A. Pnueli, R. Rosner, A framework for the synthesis of reactive modules, in: Concurrency, in: LNCS, vol. 335, Springer, 1988, pp. 4–17.
[24] A. Pnueli, R. Rosner, On the synthesis of a reactive module, in: POPL, ACM Press, 1989, pp. 179–190.
[25] A. Pnueli, R. Rosner, On the synthesis of an asynchronous reactive module, in: ICALP, in: LNCS, vol. 372, Springer, 1989, pp. 652–671.
[26] R. Selten, Re-examination of the perfectness concept for equilibrium points in extensive games, Int. J. Game Theory 4 (1975) 25–55.
[27] A.P. Sistla, E.M. Clarke, The complexity of propositional linear temporal logics, J. ACM 32 (3) (1985) 733–749.
[28] M. Ummels, Rational behaviour and strategy construction in infinite multiplayer games, in: FSTTCS, in: LNCS, vol. 4337, Springer, 2006, pp. 212–223.
[29] M. Ummels, D. Wojtczak, The complexity of Nash equilibria in simple stochastic multiplayer games, in: ICALP (2), in: LNCS, vol. 5556, Springer, 2009, pp. 297–308.
[30] M. Ummels, D. Wojtczak, Decision problems for Nash equilibria in stochastic games, in: CSL, in: LNCS, vol. 5771, Springer, 2009, pp. 515–529.
[31] M. Ummels, D. Wojtczak, The complexity of Nash equilibria in limit-average games, in: CONCUR, in: LNCS, vol. 6901, Springer, 2011, pp. 482–496.
[32] M.Y. Vardi, From verification to synthesis, in: VSTTE, in: LNCS, vol. 5295, Springer, 2008, p. 2.