Boolean games revisited

Elise Bonzon[1] and **Marie-Christine Lagasquie-Schiex**[1] and **Jérôme Lang**[1] and **Bruno Zanuttini**[2]

Abstract. Game theory is a widely used formal model for studying strategical interactions between agents. *Boolean games* [7] are two players, zero-sum static games where player's utility functions are binary and described by a single propositional formula, and the strategies available to a player consist of truth assignments to each of a given set of propositional variables (the variables *controlled* by the player.) We generalize the framework to *n*-players games which are not necessarily zero-sum. We give simple characterizations of Nash equilibria and dominated strategies, and investigate the computational complexity of the related problems.

1 Introduction

Game theory is the most successful formal model for the study of strategical interactions between agents. Informally, a game consists of a set of agents (or players), and for each agent, a set of possible strategies and an utility function mapping every possible combination of strategies to a real value. In this paper we consider only *static* games, where agents choose their strategies in parallel, without observing the others' choices. While game theory considers several formats for specifying a game (especially extended form and normal form, which coincide as far as static games are concerned), they usually consider that utility functions are represented explicitly, by listing the values for each combination of strategies.

In many real-world domains, the strategies available to an agent consist in assigning a value to each of a given set of variables. Now, representing utility functions explicitly leads to a description whose size is exponential both in the number of agents ($n \times 2^n$ values for n agents each with two available strategies) and in the number of variables controlled by the agents ($2 \times 2^p \times 2^p$ values for two agents each controlling p variables). Thus, in many cases explicitly specifying utility functions is unreasonable, as well as computing solution concepts (such as pure-strategy Nash equilibria) using a naive algorithm which enumerates combinations of strategies.

Now, specifying utilities, or more generally preferences, in a compact way, is an issue addressed by many AI researchers in the last years. Languages have been studied which allow for a concise representation of preference relations or utility functions on combinatorial domains, exploiting to a large extent the structural properties of preferences. In this paper, without much loss of generality we focus on the case where each agent has control over a set of propositional (binary) variables. Under this assumption, preferences can be represented within logic-based preference representation languages. Using propositional logic allies a rich expressivity to the possibility of using a wide variety of algorithms.

Since the specification of a static game needs the description of the agents' preferences, it seems natural to specify them using such languages for compact preference representation. Here, for the sake of simplicity we focus on the simplest possible way of using propositional logic for representing games (the extent to which it can be extended is discussed at the end of the paper), namely by using a revisited form of *Boolean games*. These games [7, 6, 4] are two-players zero-sum games where the players' utilities are binary and specified by a mere propositional formula φ (the *Boolean form* of the game).

Some background is given in Section 2. In Section 3, we give a (simplified) description of Boolean games and generalize them so as to represent non zero-sum games with an arbitrary number of players (but we keep the assumption that each player's preferences are represented by a unique propositional formula, inducing a binary utility function). In Sections 4 and 5, we show how well-known tools from propositional logic can be used so as to give simple characterizations of two of the most important game-theoretic notions, namely pure-strategy Nash equilibria and dominated strategies, and so as to derive complexity results for their computation. Sections 6 and 7 respectively address related work and further issues.

2 Background

Let $V = \{a, b, \ldots\}$ be a finite set of propositional variables and let L_V be the propositional language built from V and Boolean constants $\top$ (*true*) and $\bot$ (*false*) with the usual connectives. Formulas of L_V are denoted by φ, ψ etc. A *literal* is a variable x of V or the negation of one. A *term* is a consistent conjunction of literals. $Lit(\alpha)$ denotes the set of literals forming the term α. A formula φ is in DNF when it is written as a disjunction of terms.

2^V is the set of the interpretations for V, with the usual convention that for $M \in 2^V$ and $x \in V$, M gives the value *true* to x if $x \in M$ and *false* otherwise. $\models$ denotes the classical logical consequence relation.

Let $X \subseteq V$. 2^X is the set of X*-interpretations*. A *partial interpretation* (for V) is an X-interpretation for some $X \subseteq V$. Partial interpretations are denoted by listing all variables of X, with a $^{-}$ symbol when the variable is set to false: for instance, let $X = \{a, b, d\}$, then the X-interpretation $M = \{a, d\}$ is denoted $\{a, \bar{b}, d\}$. If $\{V_1, \ldots, V_p\}$ is a partition of V and $\{M_1, \ldots, M_p\}$ are partial interpretations, where $M_i \in 2^{V_i}$, $(M_1, \ldots, M_p)$ denotes the interpretation $M_1 \cup \ldots \cup M_p$.

Let ψ be a propositional formula. A term α is an *implicant* of ψ iff $\alpha \models \psi$ holds. α is a *prime implicant* of ψ iff α is an implicant of ψ and for every implicant α' of ψ, if $\alpha \models \alpha'$ holds, then $\alpha' \models \alpha$ holds. $PI(\psi)$ denotes the set of all the prime implicants of ψ. If $X \subseteq V$, an X-prime implicant of ψ is a prime implicant of ψ such that $Lit(\alpha) \subseteq X$. $PI_X(\psi)$ denotes the set of all the X-prime implicants of ψ.

Let $\varphi \in L_V$ and $X \subseteq V$. The *forgetting* of X in φ [12], denoted by $\exists X : \varphi$, is defined inductively by: (i) $\exists \varnothing : \varphi = \varphi$; (ii) $\exists \{x\} : \varphi = \varphi_{x \leftarrow \top} \vee \varphi_{x \leftarrow \bot}$; (iii) $\exists (X \cup \{x\}) : \varphi = \exists X : (\exists \{x\} : \varphi)$. Note that $\exists X : \varphi$ is the logically weakest consequence of φ containing only variables from $V \setminus X$ [9, 11].

[1] IRIT, UPS, F-31062 Toulouse Cedex 9,{bonzon, lagasq, lang}@irit.fr
[2] GREYC, Université de Caen, F-14032 Caen,zanutti@info.unicaen.fr

Finally, we denote the partial instantiation of a formula φ by an X-interpretation M_X by: $(\varphi)_{M_X} = \varphi_{v\in M_X \leftarrow \top, v\in X\setminus M_X \leftarrow \bot}$.

3 Boolean games

Given a set of propositional variables V, a Boolean game on V [7, 6] is a zero-sum game with two players (1 and 2), where the actions available to each player consist in assigning a truth value to each variable in a given subset of V. The utility functions of the two players are represented by a propositional formula φ formed upon the variables in V and called *Boolean form* of the game. φ represents the goal of player 1: her payoff is 1 when φ is satisfied, and 0 otherwise. Since the game is zero-sum[3], the goal of player 2 is $\neg\varphi$. [4]

Example 1
Consider $V = \{a,b,c\}$. Player 1 controls a and c while 2 controls b. Player 1's goal is $\varphi_1 = (a \leftrightarrow b) \vee (\neg a \wedge b \wedge \neg c)$ and therefore, 2's is $\varphi_2 = \neg\varphi_1 \equiv (\neg a \wedge b \wedge c) \vee (a \wedge \neg b)$. The normal form of this game is depicted on the right (in each (x,y), x—resp. y—represents the payoff of player 1—resp. 2):

1 \ 2	$\{b\}$	$\{\overline{b}\}$
$\{a,c\}$	(1, 0)	(0, 1)
$\{a,\overline{c}\}$	(1, 0)	(0, 1)
$\{\overline{a},c\}$	(0, 1)	(1, 0)
$\{\overline{a},\overline{c}\}$	(1, 0)	(1, 0)

We now give a more general definition of a Boolean game, with any number of players and not necessarily zero-sum (we will see further that the original definition [7, 6] is a special case of this more general framework).

Definition 1 *A* Boolean game *is a 4-tuple (A,V,π,Φ), where $A = \{1,2,\dots,n\}$ is a set of players, V is a set of propositional variables (called* decision variables*), $\pi : A \mapsto V$ is a control assignment function, and $\Phi = \{\varphi_1,\dots,\varphi_n\}$ is a collection of formulas of L_V.*

The control assignment function π maps each player to the variables she controls. For the sake of notation, the set of all the variables controlled by i is written π_i instead of $\pi(i)$. We require that each variable be controlled by one and only one agent, i.e., $\{\pi_1,\dots,\pi_n\}$ forms a partition of V.

Definition 2 *Let $G = (A,V,\pi,\Phi)$ be a Boolean game. A* strategy *s_i for a player i in G is a π_i-interpretation. A* strategy profile *S for G is an n-tuple $S = (s_1,s_2,\dots,s_n)$ where for all i, $s_i \in 2^{\pi_i}$.*

In other words, a strategy for i is a truth assignment for all the variables i controls. Note that since $\{\pi_1,\dots,\pi_n\}$ forms a partition of V, a strategy profile S is an interpretation for V, i.e., $S \in 2^V$.
In the rest of the paper we make use of the following notation, which is standard in game theory. Let $G = (A,V,\pi,\Phi)$ be a Boolean game with $A = \{1,\dots,n\}$, and $S = (s_1,\dots,s_n)$, $S' = (s'_1,\dots,s'_n)$ be two strategy profiles. s_{-i} denotes the projection of S on $A\setminus\{i\}$: $s_{-i} = (s_1,s_2,\dots,s_{i-1},s_{i+1},\dots,s_n)$.
Similarly, π_{-i} denotes the set of the variables controlled by all players except i: $\pi_{-i} = V\setminus\pi_i$.

[3] Stricto sensu, the obtained games are not zero-sum, but constant-sum (the sum of utilities being 1) – the difference is irrelevant and we use the terminology “zero-sum” nevertheless.

[4] The original definition [7, 6] is inductive: a Boolean game consists of a finite dynamic game. We use here the equivalent, simpler definition of [4], who showed that this tree-like construction is unnecessary.

Finally, (s_{-i},s'_i) denotes the strategy profile obtained from S by replacing s_i with s'_i without changing the other strategies: $(s_{-i},s'_i) = (s_1,s_2,\dots,s_{i-1},s'_i,s_{i+1},\dots,s_n)$.
Players' utilities in Boolean games are binary: player i is satisfied by a strategy profile (and gets utility 1) if and only if her goal φ_i is satisfied, and she gets utility 0 otherwise. Therefore, the goals $\{\varphi_i, i=1,\dots,n\}$ play the role of the utility functions $u_1,\dots,u_n$.

Definition 3 *For every player i and strategy profile S: $u_i(S) = 0$ if $S \models \neg\varphi_i$ and $u_i(S) = 1$ if $S \models \varphi_i$.*

Example 2 *We consider here a Boolean n-players version of the well-known prisoners' dilemma. n prisoners (denoted by $1,\dots,n$) are kept in separate cells. The same proposal is made to each of them: "Either you denounce your accomplices (denoted by D_i, $i=1,\dots,n$) or you cover them (C_i, $i=1,\dots,n$). Denouncing makes you freed while your partners will be sent to prison (except those who denounced you as well; these ones will also be freed). But if none of you chooses to denounce, everyone will be freed."*
Here is the representation of this game in normal form for $n=3$:

	strategy of 3: C_3		strategy of 3: D_3	
1 \ 2	C_2	D_2	C_2	D_2
C_1	(1,1,1)	(0,1,0)	(0,0,1)	(0,1,1)
D_1	(1,0,0)	(1,1,0)	(1,0,1)	(1,1,1)

So, for n prisoners, we have an n-dimension matrix, therefore 2^n n-tuples must be specified. Now, this game can be expressed much more compactly by the following Boolean game $G = \{A,V,\pi,\Phi\}$: $A = \{1,2,\dots,n\}$, $V = \{C_1,\dots,C_n\}$ (with $\neg C_i = D_i$ for every i), $\forall i \in \{1,\dots,n\}, \pi_i = \{C_i\}$, and $\forall i \in \{1,\dots,n\}, \varphi_i = \{(C_1 \wedge C_2 \wedge \dots \wedge C_n) \vee \neg C_i\}$.
Here, each player i has two possible strategies: $s_i = \{C_i\}$ and $s'_i = \{\overline{C_i}\}$. There are 8 strategy profiles for G, including $S_1 = (C_1,C_2,C_3)$ and $S_2 = (\overline{C_1},C_2,C_3)$. Under S_1, players 1, 2 and 3 have their goal satisfied, while S_2 satisfies only the goal of player 1.

Note that this choice of binary utilities implies a loss of generality, but it is essentially a starting point for the study of Boolean games, which moreover will gives us lower complexity bounds. See Section 7.

Definition 4 *Let $G = (A,V,\pi,\Phi)$ be a Boolean game, with $\Phi = \{\varphi_1,\dots,\varphi_n\}$ and $A = \{1,\dots,n\}$. Strategy s_i is a* winning strategy *for i if $\forall s_{-i} \in 2^{\pi_{-i}}, (s_{-i},s_i) \models \varphi_i$.*

Proposition 1 *Let $G = \{A,V,\pi,\Phi\}$ be a Boolean game. Player $i \in A$ has a winning strategy iff $PI_{\pi_i}(\varphi_i) \neq \emptyset$.*

Clearly enough, deciding the existence of a winning strategy for a given player is an instance of the controllability problem [2, 10] and can be reduced to the resolution of a $\mathsf{QBF}_{2,\exists}$ instance.
It is also easily seen that Boolean games as studied by Harrenstein et al [7, 6] are a special case of our n-players Boolean games, obtained by making the following two assumptions: $n = 2$ (two players) and $\varphi_2 = \neg\varphi_1$ (zero-sum).

4 Nash equilibria

Pure-strategy Nash equilibria (PNE) for n-players Boolean games are defined exactly as usual in game theory (see for instance [13]), having in mind that utility functions are induced from the player's goals $\varphi_1,\dots,\varphi_n$. A PNE is a strategy profile such that each player's strategy is an optimum response to the other players' strategies.

Definition 5 *Let $G = (A,V,\pi,\Phi)$ be a Boolean game with $A = \{1,\dots,n\}$. $S = \{s_1,\dots,s_n\}$ is a* pure-strategy Nash equilibrium *(PNE) if and only if $\forall i \in \{1,\dots,n\}, \forall s'_i \in 2^{\pi_i}, u_i(S) \geq u_i(s_{-i},s'_i)$.*

Example 3 *Let $G = \{A,V,\pi,\Phi\}$ be the Boolean game defined by $V = \{a,b,c\}$, $A = \{1,2,3\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\varphi_1 = \neg a \vee (a \wedge b \wedge \neg c)$, $\varphi_2 = (a \leftrightarrow (b \leftrightarrow c))$ and $\varphi_3 = ((a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge c))$.*
Player 1 has a winning strategy, namely setting a to false. It can be checked that the strategy profile $S = \{\overline{a},\overline{b},c\}$ is the only PNE of G.

In some examples, several PNE appear: in Ex. 1, the PNE are $\{\overline{a}b\overline{c}\}$ and $\{\overline{a}\overline{b}\overline{c}\}$, and in Ex. 2, the PNE are $\{C_1C_2C_3\}$ and $\{\overline{C_1C_2C_3}\}$.
We now give simple characterizations of pure-strategy Nash equilibria in Boolean games, starting with the following one:

Proposition 2 *Let $G = (A,V,\pi,\Phi)$ be a Boolean game and let $S \in 2^V$. S is a pure-strategy Nash equilibrium for G iff for all $i \in A$, either $S \models \varphi_i$ or $s_{-i} \models \neg\varphi_i$ holds.*

Proof: S is a PNE for G iff $\forall i \in A, \forall s'_i \in 2^{\pi_i}, u_i(S) \geq u_i(s_{-i},s'_i)$, i.e., $\forall i \in A, \forall s'_i \in 2^{\pi_i}, u_i(S) = 1$ or $u_i(s_{-i},s'_i) = 0$, i.e., $\forall i \in A$, $u_i(S) = 1$ or $\forall s'_i \in 2^{\pi_i}, u_i(s_{-i},s'_i) = 0$. Finally, $u_i(S) = 1 \Leftrightarrow S \models \varphi_i$, and $\forall s'_i \in 2^{\pi_i}, u_i(s_{-i},s'_i) = 0 \Leftrightarrow \forall s'_i \in 2^{\pi_i}$, $(s_{-i},s'_i) \models \neg\varphi_i$, i.e., $s_{-i} \models \neg\varphi_i$. ■

Since $s_{-i} \models \neg\varphi_i$ means that $\neg\varphi_i$ follows from s_{-i} *whatever* the instantiation of the variables controlled by player i, the previous characterization of PNE can be simplified again, using the forgetting operator.

Proposition 3 *Let $S \in 2^V$. S is a pure-strategy Nash equilibrium for G if and only if $S \models \bigwedge_i(\varphi_i \vee (\neg\exists i : \varphi_i))$.*

Proof: We have the following chain of equivalences: $s_{-i} \models \neg\varphi_i \Leftrightarrow s_{-i} \models \neg\exists i : \varphi_i \Leftrightarrow (s_i,s_{-i}) \models \neg\exists i : \varphi_i$ (because variables controlled by player i have disappeared from $\neg\exists i : \varphi_i) \Leftrightarrow S \models \neg\exists i : \varphi_i$. Putting everything together, we get: $(\forall i \in A, S \models \varphi_i$ or $s_{-i} \models \neg\varphi_i) \Leftrightarrow (\forall i \in A, S \models \varphi_i$ or $S \models \neg\exists i : \varphi_i) \Leftrightarrow \forall i \in A, S \models \varphi_i \vee (\neg\exists i : \varphi_i) \Leftrightarrow S \models \bigwedge_i(\varphi_i \vee (\neg\exists i : \varphi_i))$ ■

In the particular case of two-players zero-sum Boolean games, we recover the well-known fact that pure-strategy Nash equilibria coincide with winning strategies for one of the players.

Proposition 4 *If G is a two-players zero-sum Boolean game, $S = (s_1,s_2)$ is a pure-strategy Nash equilibrium iff s_1 is a winning strategy for 1 or s_2 is a winning strategy for 2.*

Proof: Let $S = (s_1,s_2)$ be a PNE. Assume $u_1(S) = 1$ (the case $u_2(S) = 1$ is symmetric). Since G is zero-sum, we have $u_2(S) = 0$. Now since S is a PNE, $\forall s'_2, u_2(S) \geq u_2(s_1,s'_2)$, which entails $\forall s'_2, u_2(s_1,s'_2) = 0$. It follows $\forall s'_2, (s_1,s'_2) \models \neg\varphi_2$, which entails that $\forall s'_2, (s_1,s'_2) \models \varphi_1$. Thus s_1 is a winning strategy for 1.
Conversely, assume that s_1 is a winning strategy for 1 (the case of 2 is symmetric). Then we have $\forall s_2, u_1(s_1,s_2) = 1$ and $\forall s_2, u_2(s_1,s_2) = 0$. Let $S = (s_1,s_2)$ where $s_2 \in 2^{\pi_2}$. We have $\forall s'_1, u_1(S) \geq u_1(s'_1,s_2)$ and $\forall s'_2, u_2(S) \geq u_2(s_1,s'_2)$. Thus S is a PNE. ■

This fact enables us to easily determine the complexity of deciding whether there is a pure-strategy Nash equilibrium in a given Boolean game. Recall that $\Sigma_2^p = \mathsf{NP}^{\mathsf{NP}}$ is the class of all the languages that can be recognized in polynomial time by a nondeterministic Turing machine equipped with NP oracles [14].

Proposition 5 *Deciding whether there is a pure-strategy Nash equilibrium in a Boolean game is Σ_2^p-complete. Completeness holds even under the restriction to two-players zero-sum games.*

Proof: Membership in Σ_2^p is immediate. Hardness is obtained by a reduction from the problem of deciding the validity of a $\mathtt{QBF}_{2,\exists}$. Given $Q = \exists A, \forall B, \Phi$, we define a two-players zero-sum Boolean game by $\varphi_1 = \Phi \vee (x \leftrightarrow y)$, where x,y are new variables and $\pi_1 = A \cup \{x\}$. Obviously, this game can be built in polynomial time given Q.
Clearly, if Q is valid with $M_A \in 2^A$ as a witness, then both (M_A,x) and $(M_A,\overline{x})$ are winning strategies for 1. Conversely, if Q is not valid, then whatever $M_A \in 2^A$ 1 plays, 2 can play $M_B \in 2^B$ such that $(M_A,M_B) \not\models \Phi$, and 2 can play $\overline{y}$ (resp. y) if 1 plays x (resp. $\overline{x}$), resulting in both cases in 2 winning (so, 1 has no winning strategy). Now it is easily seen that 2 has no winning strategy. Finally, there is a winning strategy for 1 (or 2, vacuously) if and only if Q is valid, and Proposition 4 concludes. ■

The fact that this problem lies at the second level of the polynomial hierarchy can intuitively be explained by the presence of two independent sources of complexity: the search for the "good" strategy profiles, and the test whether this strategy profile is indeed a pure-strategy Nash equilibrium. Once again, this result is related to the complexity of controllability [10]. Actually, since the existence of a Nash equilibrium is a more general problem than controllability, the fact that it has the same complexity is rather good news.
We now briefly investigate syntactic restrictions on the formulas representing the players' goals which make the problem easier. We are especially interested in DNF formulas. Recall that any Boolean function can be represented by such a formula, and thus that this is a syntactic but not a semantic restriction.
As far as 2-players zero-sum games are concerned, since deciding the validity of $\exists A, \forall B, \Phi$, a $\mathtt{QBF}_{2,\exists}$ formula, is Σ_2^P-complete even if Φ is restricted to be in DNF, Proposition 5 holds even if player 1's goal is restricted to be in DNF (and player 2's is implicit). However, when we explicitly represent the goals of each player in DNF, the complexity of the problem goes down to NP, as the next proposition shows.

Proposition 6 *Let G be a Boolean game. If the goal φ_i of every player is in DNF, then deciding whether there is a pure-strategy Nash equilibrium is NP-complete. Completeness holds even if both we restrict the number of players to 2 and one player controls only one variable.*

Proof: If φ_i is in DNF, then $\exists i : \varphi_i$ can be computed in linear time [9, Propositions 17–18]. Thus if every φ_i is in DNF, a formula $\psi \equiv \bigwedge_i(\varphi_i \vee (\neg\exists i : \varphi_i))$ can be computed in linear time. By Proposition 3 it is enough to guess a strategy profile S and check $S \models \psi$, thus the problem is in NP.
As for hardness, we give a reduction from (the complement of) the problem of deciding whether a DNF $\Phi = \bigvee_{i=1}^k T_i$ is tautological, a well-known *coNP*-complete problem. Write X for the set of variables of Φ and let $x,y \notin X$. Define a two-players game G by $\varphi_1 = \bigvee_{i=1}^k (T_i \wedge x \wedge \neg y) \vee (T_i \wedge \neg x \wedge y)$, $\pi_1 = \{y\}$, $\varphi_2 = (x \wedge y) \vee (\neg x \wedge \neg y)$, $\pi_2 = X \cup \{x\}$. Clearly, G can be built in linear time and φ_1, φ_2 are in DNF. Observe $\varphi_1 \equiv \Phi \wedge (x \neq y)$ and $\varphi_2 \equiv (x = y)$. By Proposition 3, there is a PNE in G if and only if $((\Phi \wedge (x \neq y)) \vee \neg\Phi) \wedge (x = y)$ is satisfiable. Indeed: (i) since y does not occur in Φ we have $\neg\exists y : (\Phi \wedge x \neq y) \equiv \neg(\Phi \wedge \exists y : x \neq y) \equiv \neg(\Phi \wedge \top) \equiv \neg\Phi$, and (ii) $\neg\exists X \cup \{x\} : (x = y) \equiv \perp$. Since $\Phi \wedge (x \neq y) \wedge (x = y)$ is unsatisfiable, there is a PNE in G iff $\neg\Phi \wedge (x = y)$ is satisfiable, i.e., iff $\neg\Phi$ is satisfiable since x and y do

not occur in Φ. Finally, there is a PNE in G iff Φ is not tautological. ■

When restricting to two-players games, the complexity of deciding whether a game has a PNE can even be lowered to P. This is the case if goals are represented in (i) Horn-renamable DNF, (ii) affine form, (iii) 2CNF or (iv) monotone CNF. This is ensured by tractability of projection in these cases, and the same proof as for abduction [16, Section 6] can be used. However, as far as we know the practical interest of these restrictions in our context has not been studied.

5 Dominated strategies

Another key concept in game theory is *dominance*. A strategy s_i for player i *strictly dominates* another strategy s'_i if it does strictly better than it against all possible combinations of other players' strategies, and *weakly dominates* it if it does at least as well against all possible combinations of other players' strategies, and strictly better against at least one. The key idea is that dominated strategies are not useful and can be eliminated (iteratively, until a fixpoint is reached). This process relies on the hypothesis that every player behaves in a rational way and knows that the other players are rational.

Definition 6 *Let $s_i \in 2^{\pi_i}$ be a strategy for player i. s_i is* strictly dominated *if $\exists s'_i \in 2^{\pi_i}$ s.t. $\forall s_{-i} \in 2^{\pi_{-i}}$, $u_i(s_i, s_{-i}) < u_i(s'_i, s_{-i})$.*
s_i is weakly dominated *if $\exists s'_i \in 2^{\pi_i}$ s.t. $\forall s_{-i} \in 2^{\pi_{-i}}$, $u_i(s_i, s_{-i}) \leq u_i(s'_i, s_{-i})$ and $\exists s_{-i} \in 2^{\pi_{-i}}$ s.t. $u_i(s_i, s_{-i}) < u_i(s'_i, s_{-i})$.*

The following simple example shows the interest of eliminating dominated strategies.

Example 4 *Let $G = \{A, V, \pi, \Phi\}$ be the Boolean game defined by $V = \{a, b\}$, $A = \{1, 2\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\varphi_1 = \varphi_2 = a \wedge \neg b$.*
This game has two PNE: strategy profiles $S_1 = \{a, \overline{b}\}$ and $S_2 = \{\overline{a}, b\}$. Nevertheless, only one of these equilibria is interesting. Indeed, if 1 and 2 are rational, they will both choose strategy profile S_1, which makes both of them win. This result may be obtained by eliminating dominated strategies: for player 1 (resp. 2), strategy $\{a\}$ (resp. $\{\overline{b}\}$) weakly dominates strategy $\{\overline{a}\}$ (resp. $\{b\}$).

This interest also appears in Ex. 2 (the resulting strategy profile is $\{\overline{C_1 C_2 C_3}\}$), but not in Ex. 1 (the resulting strategy profiles are exactly the PNE). It is a well-known fact from game theory that a strictly dominated strategy is not present in any Nash equilibrium, whereas a weakly dominated strategy can be present in one (see for instance [8].) Moreover, the order of elimination of strictly dominated strategies does not affect the final result, which is no longer true for weakly dominated strategies. Since the latter negative result holds for general games (with no restriction on the players' utility functions), it is worth wondering whether it still holds for Boolean games. It actually does, as shown on the following example.

Example 5 *$G = \{A, V, \pi, \Phi\}$, where $V = \{a, b\}$, $A = \{1, 2\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\varphi_1 = a \wedge b$, $\varphi_2 = a \wedge \neg b$. For player 1 (resp. 2), strategy $\{a\}$ (resp. $\{\overline{b}\}$) weakly dominates strategy $\{\overline{a}\}$ (resp. $\{b\}$). If we first eliminate $\{\overline{a}\}$, then $\{\overline{b}\}$ weakly dominates $\{b\}$ and only one strategy profile remains, namely $\{a\overline{b}\}$. Now, if we first eliminate $\{b\}$, then $\{a\}$ no longer dominates $\{\overline{a}\}$ any more, and two strategy profiles remain, namely $\{a\overline{b}\}$ and $\{\overline{a}\overline{b}\}$.*

We now study properties and characterizations of dominating strategies. A first result, that we just state as a remark, is that in a Boolean game, if strategy s_i strictly dominates strategy s'_i, then s_i is a winning strategy for i. Stated in more formal terms, s_i strictly dominates strategy s'_i if and only if: $s_i \models (\neg\exists - i : \neg\varphi_i)$ and $s'_i \models (\neg\exists - i : \varphi_i)$. This shows that, due to the restriction to binary utilities, the notion of strict dominance degenerates and loses its interest. This is however not the case for weak dominance. We have the following simple characterization of weak dominance:

Proposition 7 *Strategy s_i* weakly dominates *strategy s'_i if and only if $(\varphi_i)_{s'_i} \models (\varphi_i)_{s_i}$ and $(\varphi_i)_{s_i} \not\models (\varphi_i)_{s'_i}$.*

Proof: Strategy s_i weakly dominates s'_i iff (i) $\forall s_{-i} \in 2^{\pi_{-i}}, u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i})$ and (ii) $\exists s_{-i} \in 2^{\pi_{-i}}, u_i(s_i, s_{-i}) > u_i(s'_i, s_{-i})$.
Now (i) $\Leftrightarrow \forall s_{-i} \in 2^{\pi_{-i}}, (u_i(s_i, s_{-i}) = 1$ or $u_i(s'_i, s_{-i}) = 0) \Leftrightarrow \forall s{-i} \in 2^{\pi_{-i}}$, if $(s'_i, s_{-i}) \models \varphi_i$ then $(s_i, s_{-i}) \models \varphi_i \Leftrightarrow \forall s{-i} \in 2^{\pi_{-i}}$, if $s_{-i} \models (\varphi_i)_{s'_i}$ then $s_{-i} \models (\varphi_i)_{s_i} \Leftrightarrow (\varphi_i)_{s'_i} \models (\varphi_i)_{s_i}$.
Finally, (ii) $\Leftrightarrow \neg$(i) if we swap s_i and s'_i; thus (ii) $\Leftrightarrow (\varphi_i)_{s_i} \not\models (\varphi_i)_{s'_i}$. ■

Like for Nash equilibria, this characterization allows us to derive the complexity of deciding weak dominance.

Proposition 8 *Deciding whether a given strategy s'_i is weakly dominated is Σ_2^p-complete. Hardness holds even if φ_i is restricted to be in DNF.*

Proof: Membership in Σ_2^p is immediate. Hardness is obtained again by a reduction from the problem of deciding the validity of a $\mathtt{QBF}_{2,\exists}$. Given $Q = \exists A, \forall B, \Phi$, let a, b be two new variables, and define $\varphi_1 = (a \wedge \Phi) \vee (\neg a \wedge b)$, $\pi_1 = A \cup \{a\}$, $\pi_2 = B \cup \{b\}$ (φ_2 does not matter). Let M'_A be any A-interpretation and s'_1 be $(M'_A, \overline{a})$. We have $(\varphi_1)_{s'_1} \equiv (b)$.
Assume Q is valid with $M_A \in 2^A$ as a witness, and let $s_1 = (M_A, a)$. Then clearly s_1 is a winning strategy for 1 whereas s'_1 is not, thus s_1 weakly dominates s'_1. Conversely, assume Q is not valid, and let $M_A \in 2^A$. Let $s_1 = (M_A, \overline{a})$. Then $(\varphi_1)_{s_1} \equiv (b) \equiv (\varphi_1)_{s'_1}$, thus by Proposition 7, s_1 does not weakly dominate s'_1. Now let $s_1 = (M_A, a)$. Since Q is not valid, there is $M_B \in 2^B$ such that $(M_A, M_B) \not\models \Phi$. Thus $(M_B, b) \models (\varphi_1)_{s'_1}$ but $(M_B, b) \not\models (\varphi_1)_{s_1}$, and by Proposition 7, s_1 does not weakly dominate s'_1. Finally, s'_1 is weakly dominated iff Q is valid. For goals in DNF, just note (i) if Φ is in DNF then $\exists A, \forall B, \Phi$ is still Σ_2^p-complete and (ii) a DNF for φ_1 can be built efficiently. ■

6 Related work

Our work is not the first one that gives a logical account to the study of concepts such as Nash equilibria and dominating strategies. Apart from Boolean games [7, 6, 4], a number of other works considered static games from the point of view of logic and AI.
Two recent lines of work allow for expressing games with *ordinal* preferences within well-developed AI frameworks.
In Foo et al [5], a game in normal form is mapped into a *logic program with ordered disjunction* (LPOD) where each player owns a set of clauses that encode the player's preference over her possible actions given every possible strategy profile of other players. It is then shown that pure-strategy Nash equilibria correspond exactly to the most preferred answer sets. The given translation suffers from a limitation, namely its size: the size of the LPOD is the same as that of the normal form of the game (since each player needs a number of clauses equal to the number of possible other strategy profiles for other players). However, this limitation is due to the way LOPDs are

induced from games and could be overwhelmed by allowing to express the players' preferences by any LPODs (and prioritized goals), which then would allow for a possible much more compact representation.

In De Vos et al [3], a strategic game is represented using a *choice logic program*, where a set of rules express that a player will select a "best response" given the other players' choices. Then, for every strategic game, there exists a choice logic program such that the set of stable models of the program coincides with the set of Nash equilibria of the game. This property provides a systematic method to compute Nash equilibria for finite strategic games.

In Apt et al [1], CP-nets are viewed as games in normal form and vice versa. Each player i corresponds to a variable X_i of the CP-net, whose domain $D(X_i)$ is the set of available actions to the player. Preferences over a player's actions given the other players' strategies are then expressed in a conditional preference table. The CP-net expression of the game can sometimes be more compact than its normal form explicit representation, provided that some players' preferences depend only on the actions of a subset of other players. A first important difference with our framework is that we allow players to control an arbitrary set of variables, and thus we do not view players as variables; the only way of expressing in a CP-net that a player controls several variables would consist in introducing a new variable whose domain would be the set of all combination of values for these variables—and the size of the CP-net would then be exponential in the number of variables. A second important difference, which holds as well for the comparison with Foo et al [5] and De Vos et al [3], is that players can express arbitrary binary preferences, including extreme cases where the satisfaction of a player's goal may depend only of variables controlled by other players. A last (less technical and more foundational) difference with both lines of work, which actually explains the first two above, is that we do not *map* normal form games into anything but we *express* games using a logical language.

Admittedly, on the other hand, the restriction to binary preferences is a strong limitation, but our work can be viewed as a preliminary, but necessary step, and the extension of our notions, and of some of our results, to non-degenerated (ordinal or numerical) preferences do not present any particular problem (see Section 7).

In Section 4 we mentioned a relationship to propositional controllability, as studied by Boutilier [2] and Lang et al [10]. A recent line of work within this alley [15] studies a cooperation logic in which each agent is assumed to control a set of propositional variables. While we focus on preferences and solution concepts, a recent work on controllability, namely the Coalition Logic of Propositional Control [15] focuses on the effective power of agents, that is, they reason about the state of affairs that a group of agents can bring about.

7 Conclusion

In this paper we extended Boolean games to an arbitrary number of players and to arbitrary Boolean goals. Extended Boolean games are a first step towards a more general framework for expressing and reasoning with interacting agents when the set of strategy profiles has a combinatorial structure.

Clearly, the restriction to a single goal for each agent—and therefore to binary utilities—is a strong one. However, as often, proceeding by "building blocks" is valuable, for at least two reasons. First, once this Boolean games framework is defined, extending it so as to allow for more general preferences does not present any particular difficulty: the definition remains unchanged *except* the agents' goals $\varphi_1, \ldots, \varphi_n$, which are replaced by more complex structures, expressed within logical languages for compact representation, such as prioritized goals, weighted goals, conditional goals, or CP-nets with binary variables. These frameworks allow for representing compactly either *numerical* preferences (utility functions on 2^V) or *ordinal* preferences (partial or complete orderings on 2^V). Nash equilibria and dominated strategies are defined in the very same way[5].

Now, binary utilities are a degenerate case of both numerical and ordinal preferences. This implies that the complexity results identified in this paper provide *lower bounds* for complexity results in the aforementioned possible extensions of the framework[6].

Thus, this paper is a first step towards a more general propositional framework for representing and reasoning with static games where the set of strategy profiles has a combinatorial structure. Further work includes not only extensions to more expressive preferences as explained above, but also more sophisticated games such as dynamic games or games with incomplete information.

REFERENCES

[1] K. R. Apt, F. Rossi, and K. B. Venable. CP-nets and Nash equilibria. In Elsevier, editor, *Proc. CIRAS 2005 (Third International Conference on Computational Intelligence, Robotics and Autonomous Systems)*, Singapore, December 13-16 2005.

[2] C. Boutilier. Toward a logic for qualitative decision theory. In *KR-94*, pages 75–86, 1994.

[3] M. De Vos and D. Vermeir. Choice logic programs and Nash equilibria in strategic games. In Jorg Flum and Mario Rodriguez-Artalejo, editors, *Computer Science Logic (CSL'99)*, volume 1683, pages 266–276, 1999.

[4] P.E. Dunne and W. van der Hoek. Representation and complexity in boolean games. In *Proc. of JELIA2004*, volume LNCS 3229, pages 347–359. José Júlio Alferes et João Alexandre Leite (eds), 2004.

[5] N. Foo, T. Meyer, and G. Brewka. LPOD answer sets and Nash equilibria. In M. Maher, editor, *Proceedings of the 9th Asian Computer Science Conference (ASIAN 2004)*, pages 343–351. Chiang Mai, Thailand, Springer LNCS 3321, 2004.

[6] P. Harrenstein. *Logic in Conflict*. PhD thesis, Utrecht University, 2004.

[7] P. Harrenstein, W. van der Hoek, J.J Meyer, and C. Witteveen. Boolean games. In J. van Benthem, editor, *Proceedings of TARK 2001*, pages 287–298. Morgan Kaufmann, 2001.

[8] J. Hillas and E. Kohlberg. Foundations of strategic equilibrium. In R. Aumann and S. Hart, editors, *Handbook of Game Theory*, volume 3, pages 1598–1663. North-Holland, 2002.

[9] J. Lang, P. Liberatore, and P. Marquis. Propositional independence - formula-variable independence and forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.

[10] J. Lang and P. Marquis. Two forms of dependence in propositional logic: controllability and definability. In *Proceedings of AAAI-98*, pages 268–273, 1998.

[11] F. Lin. On the strongest necessary and weakest sufficient conditions. *Artificial Intelligence*, 128:143–159, 2001.

[12] F. Lin and R. Reiter. Forget it. In *In proc. of the AAAI falls symposium on Relevance*, pages 154–159, 1994.

[13] M. Osborne and A. Rubinstein. *A course in game theory*. MIT Press, 1994.

[14] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[15] W. van der Hoek and M. Wooldridge. On the logic of cooperation and propositional control. *Artificial Intelligence*, 164(1-2):81–119, 2005.

[16] B. Zanuttini. New polynomial classes for logic-based abduction. *Journal of Artificial Intelligence Research*, 19:1–10, 2003.

[5] As to the notion of winning strategy, it has to be refined into that of strategy guaranteeing to obtain a given satisfaction threshold.

[6] For Proposition 5, this lower bound will also be an upper bound for most extensions of the framework.