HAL

archives-ouvertes.fr

Compact preference representation and Boolean games

Élise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang, Bruno Zanuttini

▶ To cite this version:

HAL Id: hal-00947826

https://hal.archives-ouvertes.fr/hal-00947826

Submitted on 17 Feb 2014

Compact preference representation and Boolean games[*]

Elise Bonzon[† ‡]   Marie-Christine Lagasquie-Schiex[‡]   Jérôme Lang[‡]

Bruno Zanuttini[§]

Abstract

Game theory is a widely used formal model for studying strategical interactions between agents. *Boolean games* [23, 22] yield a compact representation of 2-player zero-sum static games with binary preferences: an agent's strategy consists of a truth assignment of the propositional variables she controls, and a player's preferences are expressed by a plain propositional formula. These restrictions (2-player, zero-sum, binary preferences) strongly limit the expressivity of the framework. We first generalize the framework to n-player games which are not necessarily zero-sum. We give simple characterizations of Nash equilibria and dominated strategies, and investigate the computational complexity of the associated problems. Then, we relax the last restriction by coupling Boolean games with a representation, namely, CP-nets.

Keywords: game theory, propositional logic, preference representation, CP-nets

1   Introduction

Game theory attempts to formally analyze strategic interactions between agents. Roughly speaking, a non-cooperative game consists of a set of agents (or players), and for each agent, a set of possible strategies and a utility function mapping every possible combination of strategies to a real value. In this paper we consider only *one-shot* games, where agents choose their strategies in parallel, without observing the others' choices.

While there are several different formats for formally specifying a game (most notably extended form and normal form, which coincide as far as static games are concerned), utility functions are usually represented explicitly, by listing the values for each combination of strategies (an exception being the recent framework of graphical games, which we discuss later in the paper). However, the number of utility values which must be specified, that is, the number of possible combinations of strategies, is exponential in the number of players, which renders such an explicit way of representing the preferences of the players unreasonable when the number of players is not very small. This becomes even more problematic when the set of strategies available to an agent consists in assigning a value from a finite domain to each of a given set of variables (which is the case in many real-world domains). In this case, 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 Boolean variables). Thus, in all these cases, specifying players' preferences explicitly is clearly unreasonable, both because it would need exponential space, and because studying these games (for instance by computing solution concepts such as pure-strategy Nash equilibria) would require accessing all of these utility values at least once and hence would take time exponential in the numbers of agents and variables in all cases.

How can this be dealt with? A way out consists in using a language for representing individual preferences (either preference relations or utility functions) on combinatorial (multivariable) domains in a *structured* and *compact* way. These languages have been actively studied recently, especially in the AI community. They exploit to a large extent the structural properties of preferences (such as conditional independencies between variables). The relevance of compact preference representation on multivariable domains to the specification and the study of games looks obvious (in the simple case where each agent has only two possible strategies, then there is exactly one binary variable per agent), and yet this connection has rarely been exploited (up to a few exceptions, which we discuss in Section 6).

In this paper, we focus without much loss of generality on the case in which each agent has control over a set of *Boolean* (binary) variables. Not only does this simplify the presentation, but it allows us to exploit logical languages for preference representation. It is also consistent with previous work on *Boolean games* [23, 22, 14]. In Section 7 we discuss in more details to which extent this assumption induces a loss of generality.

The simplest possible way to represent preferences in Boolean games consists in assuming that each player has a specific propositional formula that she wants to be satisfied. Under this assumption, each player has a *dichotomous* preference (either

she is satisfied or not, with no possible gradation). Furthermore, Boolean games as defined in [23, 14] are two-player zero-sum games where the players' utilities are binary and specified by a single propositional formula φ (the *Boolean form* of the game) expressing the goal of player number 1. These three restrictions (2-player, zero-sum, binary preferences) strongly limit the expressivity of the framework. They were first relaxed in Chapter 8 of [22], which we discuss in detail in Section 4.2.

In Section 3 we stick to the assumption that preferences are dichotomous (but relax the other two restrictions), and revisit Boolean games by defining the agents' preferences as an arbitrary n-tuple of propositional formulas (see [4]). Then, we focus on the third restriction. Whereas a single propositional formula (goal) φ cannot express more than a binary preference relation on interpretations (models of φ are strictly better than models of ¬φ), expressing arbitrary (non-binary) preferences within a propositional framework is possible, making use of a more sophisticated propositional language for compact preference representation. Clearly, many choices for such a language are possible, and for each of them, it is worth giving characterizations of Nash equilibria and other solution concepts as well as addressing complexity issues. After briefly discussing the proposal in Chapter 8 of [22] and some of its possible generalizations, we focus on the language of CP-nets, for several reasons: first, this is one of the most popular and well-developed compact representation language (both because it is easy to elicit preferences represented by CP-nets, and because of its computational properties); and second, because this language has enough *structure* to allow for interesting characterizations of our solution concepts. The good point with ordinal preferences is that they are often easier to elicit, mainly for cognitive reasons. The bad point is that some game-theoretical concepts, such as mixed-strategy Nash equilibria, need numerical preferences. However, some do not, including pure-strategy Nash equilibria and dominated strategies: these notions only require ordinal preferences.

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). Then, 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-theoretical notions, namely pure-strategy Nash equilibria and dominated strategies, and so as to derive complexity results for their computation. In Section 4, Boolean games are coupled with non-dichotomous preferences: we introduce some languages for compact preference representation, and then we briefly discuss Harrenstein's proposal given in Chapter 8 of [22] and some of its generali-

sation. Boolean games are then coupled with propositionalized CP-nets in Section 5. Sections 6 and 7 respectively address related work and further issues. Proofs are given in Appendix A.

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 a variable. A *term* is a consistent conjunction of literals. $Var(\varphi)$ (resp. $Lit(\varphi)$) denotes the set of variables (resp. literals) occuring in the formula φ. A formula φ is in DNF when it is written as a disjunction (possibly empty) 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.

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 $M_1 \cup \ldots \cup M_p$.

If M is an interpretation for V and $Var(\varphi) \subseteq V$, we write $M \models \varphi$ if M satisfies φ. For two formulas φ, ψ, $\varphi \models \psi$ denotes the classical consequence relation, that is, $\varphi \models \psi$ if every model of φ is a model of ψ. If M is a partial interpretation of $Var(\varphi)$, we write $M \models \varphi$ if φ is satisfied by every interpretation for $Var(\varphi)$ which agrees with M; equivalently, $M \models \varphi$ if the conjunction of all literals assigned true by M logically entails φ. Due to this equivalence, we use the same notation for entailment and satisfaction, as is standard in propositional logic.

For $v \in V$, we denote by $v \leftarrow x$ the fact that the variable v is assigned to the value x. The partial instantiation of a formula φ by an X-interpretation M_X is denoted by:

$$(\varphi)_{M_X} = \varphi_{v \in M_X \leftarrow \top, v \in X \setminus M_X \leftarrow \bot}$$

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 there is no implicant α' of ψ with $Lit(\alpha') \subset Lit(\alpha)$. $PI(\psi)$ 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 $Var(\alpha) \subseteq X$. $PI_X(\psi)$ 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 [33], 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) $\exists(X \cup \{x\}) : \varphi = \exists X : (\exists\{x\} : \varphi)$.

$\exists X : \varphi$ is the logically strongest consequence of φ containing only variables from $V \setminus X$ [29, 32]. In other terms, it is the projection of φ on $\overline{X}$. We also need the dual of forgetting defined by $\forall X : \varphi \equiv \neg\exists X : \neg\varphi$.

Lastly, if φ is a propositional formula, then $DepVar(\varphi)$ is the set of propositional variables on which φ depends [29], defined by: for $x \in V$, $x \in DepVar(\varphi)$ if and only if $\varphi_{x:=\top} \not\equiv \varphi_{x:=\bot}$, that is if the two propositional formulas obtained from φ by substituting x by $\top$ and by $\bot$ define distinct propositional functions.

3 n-player Boolean games with binary preferences

Given a set of propositional variables V, a Boolean game on V [23, 22] 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 built from the variables in V and called the *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[1], the goal of player 2 is $\neg\varphi$.[2]

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 goal 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 here (in each (x, y), x—resp. y—represents the payoff of player 1—-resp. 2):

---

[1]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.

[2]The original definition [23] is inductive: a Boolean game consists of a finite dynamic game. We use here the equivalent, simpler definition of [14], who showed that this tree-like construction is unnecessary.

1 \\ 2	b	$\bar{b}$
ac	*(1, 0)*	*(0, 1)*
$a\bar{c}$	*(1, 0)*	*(0, 1)*
$\bar{a}c$	*(0, 1)*	*(1, 0)*
$\bar{a}\bar{c}$	*(1, 0)*	*(1, 0)*

We now give a more general definition of a Boolean game, which may have any number of players and is not necessarily zero-sum.

Definition 1 (Boolean game)
*An n-**player Boolean game** is a 4-tuple (N, V, π, Φ), where $N = \{1, 2, \ldots, n\}$ is a set of players, V is a set of propositional variables (called* decision variables*), $\pi : N \mapsto 2^V$ is a control assignment function, and $\Phi = \langle \varphi_1, \ldots, \varphi_n \rangle$ 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., $\{\pi_1, \ldots, \pi_n\}$ forms a partition of V.

Our notion of Boolean games corresponds to the specialization of distributed evaluation games defined in Chapter 8 of [22] – see Section 4.2 for more details. It is easily seen that Boolean games as studied by [23, 22, 14] are a special case of our *n*-player Boolean games, obtained by making the following two assumptions: $n = 2$ (two players) and $\varphi_2 = \neg \varphi_1$ (zero-sum).

Definition 2 (Strategy, strategy profile)
*Let $G = (N, V, \pi, \Phi)$ 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 a n-tuple $s = (s_1, s_2, \ldots, s_n)$ 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 $\{\pi_1, \ldots, \pi_n\}$ forms a partition of V, a strategy profile s defines an (unambiguous) interpretation for V. Slightly abusing notation and words, we write $s \in 2^V$, to refer to the value assigned by s to some variable, etc. S denotes the set of all strategy profiles for G.

More generally, we could add to Boolean games *constraints* restricting the set of individual strategies: a Boolean game would then contain a collection of propositional formulae $\{\gamma_i, i \in N\}$, where, for each $i \in N$, γ_i is a propositional formula of

$L_{\pi(i)}$. The set S_i of strategies available to player i would then be the set of partial interpretations in 2^{π_i} that satisfy γ_i. For the sake of simplicity, we choose to omit constraints. Most of our results can easily be generalized to Boolean games with constraints (for those that cannot we explicitly mention it).

In the rest of the paper we make use of the following notation, which is standard in game theory. Let $G = (N, V, \pi, \Phi)$ be a Boolean game with $N = \{1, \ldots, n\}$, and $s = (s_1, \ldots, s_n)$, $s' = (s'_1, \ldots, s'_n)$ be two strategy profiles. s_{-i} denotes the projection of s onto $N \setminus \{i\}$: $s_{-i} = (s_1, s_2, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$.

Similarly, π_{-i} denotes the set of the variables controlled by all players except i: $\pi_{-i} = V \setminus \pi_i$.

Finally, (s_{-i}, s'_i) denotes the strategy profile obtained from s by replacing s_i with s'_i without changing the other strategies: $(s_{-i}, s'_i) = (s_1, s_2, \ldots, s_{i-1}, s'_i, s_{i+1}, \ldots, s_n)$.

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 $\{\varphi_i, i = 1, \ldots, n\}$ play the role of the utility functions $u_1, \ldots, u_n$.[3]

Example 2 *We consider here a Boolean n-player (simplified) version of the well-known prisoners' dilemma. n prisoners (denoted by $1, \ldots, n$) are kept in separate cells. The same proposal is made to each of them: "Either you cover for your accomplices (denoted by C_i, $i = 1, \ldots, n$) or you denounce them ($\neg C_i$, $i = 1, \ldots, n$). Denouncing makes you free while your partners will be sent to prison (except those who denounced you as well; these ones will also be free). But if none of you chooses to denounce, everyone will be free[4]".*

Here is the representation of this game in normal form for $n = 3$:

strategy of 3: C_3				strategy of 3: $\overline{C_3}$		
1 \ 2	C_2	$\overline{C_2}$		1 \ 2	C_2	$\overline{C_2}$
C_1	$(1,1,1)$	$(0,1,0)$		C_1	$(0,0,1)$	$(0,1,1)$
$\overline{C_1}$	$(1,0,0)$	$(1,1,0)$		$\overline{C_1}$	$(1,0,1)$	$(1,1,1)$

So, for n prisoners, we have a n-dimension matrix, therefore 2^n n-tuples must be specified.

---

[3] Alternatively, we can define, for every player i, the (binary) utility function induced by her goals φ_i by: for every strategy profile s, $u_i(s) = 0$ if $s \models \neg \varphi_i$ and $u_i(s) = 1$ if $s \models \varphi_i$.

[4] Notice that the limitation to binary preferences make it impossible to express that a player prefers the situation where he denounces and the others cooperate to the situation where everyone cooperates, and the latter to a situation where everyone denounces. In order to do so we need a more sophisticated language – see Section 5.

This game can be expressed much more compactly by the following Boolean game $G = (N, V, \pi, \Phi)$, where $N = \{1, 2, \ldots, n\}$, $V = \{C_1, \ldots, C_n\}$, $\forall i \in \{1, \ldots, n\}, \pi_i = \{C_i\}$, and $\forall i \in \{1, \ldots, n\}, \varphi_i = (C_1 \wedge C_2 \wedge \ldots \wedge C_n) \vee \neg C_i$.

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 $C_1 C_2 C_3$ and $\overline{C_1} C_2 C_3$.
Under $C_1 C_2 C_3$, players 1, 2 and 3 have their goal satisfied, while $\overline{C_1} C_2 C_3$ satisfies only the goal of player 1.

Note that this choice of binary utilities implies a loss of generality, but it provides a good starting point for the study of Boolean games, and it will give us lower complexity bounds. See Section 5 for less restrictive approaches.

Definition 3 (Winning strategy)
*Let $G = (N, V, \pi, \Phi)$ be a Boolean game, with $\Phi = \langle \varphi_1, \ldots, \varphi_n \rangle$ and $N = \{1, \ldots, n\}$. 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 $G = (N, V, \pi, \Phi)$ be a Boolean game. Player $i \in N$ has a winning strategy iff $PI_{\pi_i}(\varphi_i) \neq \varnothing$.

Clearly enough, deciding the existence of a winning strategy for a given player is an instance of the controllability problem [5, 30, 40] and can be reduced to the resolution of a $\mathrm{QBF}_{2, \exists}$ instance.

3.1 Nash equilibria

Pure-strategy Nash equilibria (PNE) for *n*-player Boolean games are defined exactly as usual in game theory (see for instance [35]), keeping in mind that utility functions are induced from players' goals $\varphi_1, \ldots, \varphi_n$. A PNE is a strategy profile in which each player's strategy is an optimum response to the other players' strategies.

Definition 4 (Pure-strategy Nash equilibria)
*Let $G = (N, V, \pi, \Phi)$ be a Boolean game with $N = \{1, \ldots, n\}$. $s = (s_1, \ldots, s_n)$ is a **pure-strategy Nash equilibrium** (PNE) if and only if $\forall i \in \{1, \ldots, n\}, \forall s'_i \in 2^{\pi_i}, u_i(s) \geq u_i(s_{-i}, s'_i)$.*

Example 3 *Let $G = (N, V, \pi, \Phi)$ be the Boolean game defined by $V = \{a, b, c\}$, $N = \{1, 2, 3\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\varphi_1 = \neg a \vee (a \wedge b \wedge \neg c)$, $\varphi_2 = (a \leftrightarrow (b \leftrightarrow c))$ 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 $s = \overline{a} b c$ is the only PNE of G.

In some examples, several PNE exist: in Ex. 1, the PNE are $\bar{a}b\bar{c}$ and $\bar{a}\bar{b}\bar{c}$, 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 $G = (N, V, \pi, \Phi)$ be a Boolean game and let $s \in 2^V$. s is a pure-strategy Nash equilibrium for G iff for all $i \in N$, $s \not\models \varphi_i$ implies $s_{-i} \models \neg\varphi_i$.

Example 3, continued: *The strategy profile $s = \bar{a}\bar{b}c$ is a PNE (notice that it is the only one). Indeed, we have:*

    1. $s = \bar{a}\bar{b}c \models \varphi_1 = \neg a \vee (a \wedge b \wedge \neg c)$

    2. $s = \bar{a}\bar{b}c \models \varphi_2 = (a \leftrightarrow (b \leftrightarrow c))$

    3. $s_{-3} = \bar{a}\bar{b} \models \neg\varphi_3 = ((\neg a \vee b \vee c) \wedge (a \vee \neg b \vee \neg c))$

As $s_{-i} \models \neg\varphi_i$ 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. For the sake of notation, we use the notation $\exists i : \varphi_i$ instead of $\exists \pi_i : \varphi_i$.

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

This result can be seen as the syntactical counterpart of Proposition 9.5.2 and Corollary 9.5.3 in [22] (the latter result being established in the more general setting of distributed evaluation games – see Section 4.2).

Example 3, continued: *Let again $s = \bar{a}\bar{b}c$. $\exists 3 : \varphi_3 = ((a \wedge \neg b \wedge \neg\top) \vee (\neg a \wedge b \wedge \top)) \vee ((a \wedge \neg b \wedge \neg\bot) \vee (\neg a \wedge b \wedge \bot)) = (\neg a \wedge b) \vee (a \wedge \neg b)$. We easily check that $s \models \varphi_1 \wedge \varphi_2 \wedge \neg\exists 3 : \varphi_3$.*
In the particular case of 2-player 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 2-player 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.

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 (see for instance [36]).

Proposition 5

Deciding whether there exists a pure-strategy Nash equilibrium in a Boolean game is Σ_2^p-complete. Completeness holds even under the restriction to 2-player zero-sum games.

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.

This result can be positioned with respect to other results about the complexity of the existence of Nash equilibria in compactly represented games. [20] show that in a graphical game where the utility function of each player depends only on a number of variables bounded by a constant, the existence of a PNE is NP-complete. This result heavily depends on this "locality" assumption that each player's utility depends on a small number of variables controlled by other players. A similar assumption in our framework leads to a similar complexity gap: indeed, for the family of Boolean games such that for some constant K, $|Var(\varphi_i)| \leq K$ for every i, deciding whether there exists a PNE is in NP (we can see from Proposition 3 that the number of variables we have to eliminate is constant, so the computation of $s \models \bigwedge_i (\varphi_i \vee (\neg \exists i : \varphi_i))$ is polynomial).

We now briefly investigate *syntactical* 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 syntactical but not a semantical restriction[5]. A DNF goal intuitively represents an enumeration of "winning situations", where a situation is a partial combination of strategies.

As far as 2-player zero-sum games are concerned, since deciding the validity of $\exists N, \forall B, \Phi$, 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 goal is implicit). However, when we explicitly represent the goal of each player in DNF, the complexity of the problem goes down to NP, as the next proposition shows.

---

[5]However, of course, the equivalent DNF representation of a formula may be exponentially larger than this formula.

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 we restrict the number of players to 2.

When restricting to 2-player 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 with the same polarity for each variable in both players' goals. This is ensured by tractability of projection in these cases, and algorithms similar to those for abduction [41, Section 6] can be used.

The case of goals expressed by Horn-renamable DNFs encompasses that of monotone DNF. That is, the case where each variable is monotone for one player, but may have opposite polarity for both players. This means that two players can express preferred combinations of variables (DNF) with opposite preferences over individual variables (e.g., there is no case where player 1 prefers x to $\neg x$ and where player 2 prefers $\neg x$ to x), and one can decide whether there is a PNE in polynomial time, that is, a tradeoff between their "opposite" preferences.

3.2 Dominated strategies

Another key concept in game theory is *dominance*. A strategy s_i for player i *strictly dominates* another strategy s_i' if s_i does strictly better than s_i' against all possible combinations of other players' strategies, and *weakly dominates* s_i' if s_i 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 5 (Strictly/weakly dominated strategies)
*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 $G = (N, V, \pi, \Phi)$ be the Boolean game defined by $V = \{a, b\}$, $N = \{1, 2\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\varphi_1 = \varphi_2 = a \wedge \neg b$.*

This game has two PNEs: strategy profiles $a\overline{b}$ and $\overline{a}b$. Nevertheless, only one of these equilibria is interesting. Indeed, if 1 and 2 are rational, they will both choose strategy profile $a\overline{b}$, which makes both of them win. This result may be obtained by eliminating weakly dominated strategies: for player 1 (resp. 2), strategy a (resp. $\overline{b}$) weakly dominates strategy $\overline{a}$ (resp. b).

This interest also appears in Example 2 (the resulting strategy profile is $\overline{C_1 C_2 C_3}$), but not in Example 1 (the resulting strategy profiles are exactly the PNE). It is a well-known fact from game theory that there is no strictly dominated strategy in any Nash equilibrium, whereas a weakly dominated strategy can be present in one (see for instance [24]).

Example 5 $G = (N, V, \pi, \Phi)$, where $V = \{a, b\}$, $N = \{1, 2\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\varphi_1 = (a \wedge b) \vee \neg a$, $\varphi_2 = a \wedge b$.
This game has three PNEs: strategy profiles ab, $\overline{a}b$ and $\overline{a}\overline{b}$. However, the strategy a is weakly dominated by the strategy $\overline{a}$ for the player 1, even if it appears in a PNE.

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 by the following example.

Example 6 $G = (N, V, \pi, \Phi)$, where $V = \{a, b\}$, $N = \{1, 2\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\varphi_1 = a \wedge b$, $\varphi_2 = a \wedge \neg b$. *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 dominated 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:

$$s_i \models (\neg \exists - i : \neg \varphi_i) \text{ and } s_i' \models (\neg \exists - i : \varphi_i)$$

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* $(\varphi_i)_{s_i} \not\models (\varphi_i)_{s_i'}$.

For example, we have for the game presented in Example 6: $(\varphi_1)_a = b$ and $(\varphi_1)_{\overline{a}} = \perp$. So, $(\varphi_1)_{\overline{a}} \models (\varphi_1)_a$, and $(\varphi_1)_a \not\models (\varphi_1)_{\overline{a}}$: as seen in Example 6, a weakly dominates $\overline{a}$.

This characterization allows us to derive the following complexity result.

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.

Characterizing strategies that survive iterated elimination of weakly dominated strategies looks much more complicated, as well as finding the complexity of deciding whether a given strategy survives any iterated elimination of weakly dominated strategies. This is left for further research.

4   Coupling Boolean games and compact preference representation languages: principles and first examples

4.1   Towards Boolean games with non-dichotomous preferences

The choice of dichotomous utilities (where agents can only express plain satisfaction or plain dissatisfaction, with no intermediate levels) is an important loss of generality. Fortunately, this restriction can easily be relaxed, at least from the point of view of generalizing definitions. It suffices to replace the preference component of a Boolean game by an input expressed in a (propositional) language for *compact preference representation*. Recall that languages for preference representation may be either *ordinal* (i.e., expressed by weak orders) or *numerical* (i.e., expressed by utility functions). For the sake of the exposition the discussion below supposes that preferences are ordinal (the case for numerical preferences is similar).

A *preference relation* $\succeq$ is a reflexive and transitive binary relation (not necessarily complete) on S. The strict preference $\succ$ associated with $\succeq$ is defined as usual by $s \succ s'$ if and only if $s \succeq s'$ and not $s' \succeq s$.

Let L be a propositional language for compact representation of ordinal preferences, equipped with a function $Induce_L$ that maps any input of L to a preference relation $\succeq$ on 2^V. An *L-Boolean game* is defined to be a 4-uple $G = (N, V, \pi, \Phi)$, where $N = \{1, \ldots, n\}$, V and π are as before and $\Phi = \langle \Phi_1, \ldots, \Phi_n \rangle$, where for each

i, Φ_i is a compact representation, in L, of the preference relation $\succeq_i$ of agent i on S, that is, $Induce(\Phi_i) = \succeq_i$. We let $Pref_G = \langle \succeq_1, \ldots, \succeq_n \rangle$.

Since most propositional languages for compact ordinal preference representation can be used to represent partial preference relations, we start by redefining pure-strategy Nash equilibria and dominated strategies, since the usual definition of these notions assume that players' preferences are complete.

We first have to define *two* notions of PNEs, a weak one and a strong one (these notions correspond respectively to the notions of maximal and maximum equilibria in [22]).

Definition 6 (Weak, strong Nash equilibrium)
Let $G = (N, V, \pi, \Phi)$ and $Pref_G = \langle \succeq_1, \ldots, \succeq_n \rangle$ the collection of preference relations on S induced from Φ. Let $s = (s_1, \ldots, s_n) \in S$.
*s is a **weak PNE** (WPNE) for G iff $\forall i \in \{1, \ldots, n\}$, $\forall s'_i \in 2^{\pi_i}$, $(s'_i, s_{-i}) \not\succ_i (s_i, s_{-i})$.*
*s is a **strong PNE** (SPNE) for G iff $\forall i \in \{1, \ldots, n\}$, $\forall s'_i \in 2^{\pi_i}$, $(s'_i, s_{-i}) \preceq_i (s_i, s_{-i})$.*
$NE_s(G)$ and $NE_w(G)$ denote respectively the set of strong and weak[6] PNEs for G.

Clearly, any SPNE is a WPNE, that is, $NE_s(G) \subseteq NE_w(G)$.

The notions of dominated strategies, initially defined with binary utilities (Definition 5), have to be refined too.

Definition 7 (Strictly/weakly dominated strategies)
Let $s_i \in 2^{\pi_i}$ be a strategy for player i, and $\succeq_i$ her preference relation on S.
*s_i is **strictly dominated** if $\exists s'_i \in 2^{\pi_i}$ s.t. $\forall s_{-i} \in 2^{\pi_{-i}}$, $(s_i, s_{-i}) \prec_i (s'_i, s_{-i})$.*
*s_i is **weakly dominated** if $\exists s'_i \in 2^{\pi_i}$ s.t. $\forall s_{-i} \in 2^{\pi_{-i}}$, $(s_i, s_{-i}) \preceq_i (s'_i, s_{-i})$ and $\exists s_{-i} \in 2^{\pi_{-i}}$ s.t. $(s_i, s_{-i}) \prec_i (s'_i, s_{-i})$.*

The introduction of partial preferences allows us to introduce a new notion of dominated strategies. This new notion is very weak: all strategies can be partially dominated.

Definition 8 (Partially dominated strategies)
Let $s_i \in 2^{\pi_i}$ be a strategy for player i, and $\succeq_i$ her preference relation on S.
*s_i is **partially dominated** if $\exists s'_i \in 2^{\pi_i}$ s.t. $\forall s_{-i} \in 2^{\pi_{-i}}, (s_i, s_{-i}) \not\succ_i (s'_i, s_{-i})$ and $\exists s_{-i} \in 2^{\pi_{-i}}$ s.t. $(s_i, s_{-i}) \prec_i (s'_i, s_{-i})$*

---

[6]There is an interesting equivalent way of defining weak and strong Nash equilibria. A partial preference relation $\succeq_i$ can be identified with the set of all $Ext(\succeq_i)$ of complete preference relations that extend it, and a *n*-uple of partial preference relations $P = \langle \succeq_1, \ldots, \succeq_n \rangle$ is then identified with the set $Ext(P)$ of all *n*-uples of complete preference relations $\langle \sqsupseteq_1, \ldots, \sqsupseteq_n \rangle$ such that for every i, $\sqsupseteq_i$ extends $\succeq_i$. Formally, $Ext(P) = Ext(\succeq_1) \times \ldots \times Ext(\succeq_n)$. Finally, let G be a Boolean game G, then s is a strong (respectively weak) Nash equilibrium for G iff s is a Nash equilibrium for every (respectively, at least one) game whose preferential component is in $Ext(P)$.

If $\succeq_i$ is a total pre-order, then weakly and partially dominated strategies are equivalent.

Moreover, we still have [34]:

- the order of elimination of strictly dominated strategies *does not affect* the final result.

- the order of elimination of weakly and partially dominated strategies *may affect* the final result.

4.2 Distributed evaluation games

A first way of extending Boolean games so as to allow players' preferences to be non-dichotomous has been proposed in Chapter 8 of [22]. In a *distributed evaluation game*, the preferences of a player are expressed by a finite set of propositional formulae expressing elementary goals that she wants to see satisfied. We choose to give a formal presentation of distributed evaluation games in our own terms.

Definition 9 (Distributed evaluation games, reformulated)
An individual goal base *is a set* $\Sigma = \{\varphi_1, \dots, \varphi_q\}$ *of propositional formulas. A* **distributed evaluation game** *is a 4-uple* $G = (N, V, \pi, \Phi)$*, where* $\Phi = (\Sigma_1, \dots, \Sigma_n)$ *is a collection of goals bases.*

The preference of a player is induced from her goal base in the following way: she prefers a state s to a state s' if the set of goals satisfied by s contains the set of goals satisfied by s'. Formally:

Definition 10
Let Σ be a goal base. For any state $s \in S$, let $Sat(s, \Sigma) = \{\varphi \in \Sigma \mid s \models \varphi\}$. The preference relation $\succeq_\Sigma$ on S induced from Σ is defined by

$$s \succeq_\Sigma s' \text{ iff } Sat(s, \Sigma) \supseteq Sat(s', \Sigma)$$

Note that $\succeq_\Sigma$ may not be complete. An alternative to finding a preference relation from a goal base is to use cardinality instead of inclusion: $s \succeq_\Sigma s'$ iff $|Sat(s, \Sigma)| \geq |Sat(s', \Sigma)|$.
Now that we have a preference relation for each player, results of Subsection 4.1 apply.

The following example shows how the prisoner's dilemma can be represented as a distributed evaluation game.

Example 7 *Let $G = (N, V, \pi, \Phi)$, with $N = \{1, 2\}$, $V = \{C_1, C_2\}$, $\pi_1 = \{C_1\}$, $\pi_2 = \{C_2\}$, $\Sigma_1 = \{C_1 \rightarrow C_2, \neg C_1, \neg C_1 \wedge C_2\}$ and $\Sigma_2 = \{C_2 \rightarrow C_1, \neg C_2, \neg C_2 \wedge C_1\}$. The preference relations associated with players* 1 *and* 2 *are the following:*

$$\overline{C}_1 C_2 \succ_1 \overline{C}_1 \overline{C}_2 \succ_1 C_1 C_2 \succ_1 C_1 \overline{C}_2$$

$$C_1 \overline{C}_2 \succ_2 \overline{C}_1 \overline{C}_2 \succ_2 C_1 C_2 \succ_2 \overline{C}_1 C_2$$

We can give a simple characterization of pure strategy Nash equilibrium in this framework, which is a reformulation of Corollary 9.5.3 in [22] in our framework.

Proposition 9 *Let $G = (N, V, \pi, \Phi)$ be a distributed evaluation game and let $s \in S$. If s is a SPNE then for all $i \in N$, $Sat(s, \Sigma_i)$ is maximal s_{-i}-consistent.*[7]

Note that the previous implication can be extended to an equivalence if we consider WPNE instead of SPNE.

Example 7, continued: *$C_1 C_2$ is a SPNE of G: we have $Sat(C_1 C_2, \Sigma_1) = \{C_2 \rightarrow C_1\}$ maximal C_2-consistent (because $Sat(\overline{C}_1 C_2, \Sigma_1) = \emptyset$), and $Sat(C_1 C_2, \Sigma_2) = \{C_2 \rightarrow C_1\}$ maximal C_1-consistent.*
We can check that this is the only SPNE of this game.

Let us say a word about complexity. It is easy to check that the problem of deciding whether there exists a strong or weak PNE is in Σ_2^p. Now, when Σ_i is a singleton, SPNEs coninde with PNEs as defined in Section 3.1. Therefore, deciding whether there exists a strong pure-strategy Nash equilibrium in a Boolean game is Σ_2^p-complete.

4.3 Generalizing distributed evaluation games: Boolean games and prioritized goals

Generalizing a simple goal to a set of goals is a first step to allow to have more expressing preferences. However, expressing preferences using set of goals does not allow to express the relative importance of goals. A practical way to do so is to introduce priority over goals. Thus, a simple generalisation of distributed evaluation games is Boolean games with prioritized goals [3]. The preferences of a single player in this framework are expressed by a set of goals ordered by a priority relation.

---

[7]Recall that a set $W \subseteq \Sigma_i$ is maximal s_i-consistent if it is consistent with s_i, and if there is no $W' \subseteq \Sigma_i$ consistent with s_i such that $W \subset W'$.

A *prioritized goal base* Σ is a collection $\langle \Sigma^1; \dots; \Sigma^p \rangle$ of sets of propositional formulas. Σ^j represents the set of goals of priority j, with the convention that the smaller j, the more prioritary the formulas in Σ^j.

Several criteria can be used in order to generate a preference relation $\succeq$ from Σ, as for example the *discrimin* criteria based on inclusion, or the *leximin* criteria based on cardinality. We choose to skip the formal definition of these criteria, and simply give an example where the two well-known criteria discrimin and leximin coincide.

Example 8 *Prisoner's dilemma can be translated into the following Boolean game with prioritized goals $G = (N, V, \pi, \Phi)$: $N = \{1, 2\}$, $\pi_1 = \{C_1\}$, $\pi_2 = \{C_2\}$, $\Sigma_1 = \langle C_2; \neg C_1 \rangle$, $\Sigma_2 = \langle C_1; \neg C_2 \rangle$.*

The preference relations associated with players 1 and 2 are the following (they are identical with leximin or discrimin) :

$$C_1 \overline{C}_2 \succ_1 \overline{C}_1 \overline{C}_2 \succ_1 C_1 C_2 \succ_1 \overline{C}_1 C_2$$

$$\overline{C}_1 C_2 \succ_2 \overline{C}_1 \overline{C}_2 \succ_2 C_1 C_2 \succ_2 C_1 \overline{C}_2$$

We choose to omit details about Boolean games with prioritized goals because this would require a significant amount of space, while we have no significant results other than simple generalization of results of Section 3. These results can be found on [3].

5 Coupling Boolean games and compact preference representation languages, a case study: CP-nets and Boolean games

5.1 CP-nets

In this section we consider a very popular language for compact preference representation on combinatorial domains, namely CP-nets.

This graphical model exploits conditional preferential independence in order to structure the decision maker's preferences under a *ceteris paribus* assumption. They were introduced in [8] and extensively studied in many subsequent papers, most notably [6, 7].

Although CP-nets generally consider variables with arbitrary finite domains, for the sake of simplicity (and homogeneity with the rest of the paper) here we consider only "propositionalized" CP-nets, that is, CP-nets with binary variables (note that this is not a real loss of generality, as all our definitions and results can be easily lifted to the more general case of non-binary variables).

**Definition 11 (Conditional preferential independence)**
Let V be a set of propositional variables and $\{X, Y, Z\}$ a partition of V. X is
conditionally preferentially independent *of Y given Z if and only if $\forall z \in 2^Z$,*
$\forall x_1, x_2 \in 2^X$ and $\forall y_1, y_2 \in 2^Y$ we have: $x_1 y_1 z \succeq x_2 y_1 z$ iff $x_1 y_2 z \succeq x_2 y_2 z$.

For each variable X, the agent specifies a set of *parent variables $Pa(X)$* that can
affect her preferences over the values of X. Formally, X and $V \setminus (\{X\} \cup Pa(X))$ are
conditionally preferentially independent given $Pa(X)$. This information is used to
create the CP-net:

Definition 12 (CP-net)
*Let V be a set of propositional variables. $\mathcal{N} = \langle \mathcal{G}, \mathcal{T} \rangle$ is a **CP-net on** V, where $\mathcal{G}$*
is a directed graph over V, and $\mathcal{T}$ is a set of conditional preference tables $CPT(X_j)$
for each $X_j \in V$. Each $CPT(X_j)$ associates a linear order $\succ_p^j$ with each instantia-
tion $p \in 2^{Pa(X_j)}$.

Example 9 *Consider the following simple CP-net that expresses my preference*
over dinner configurations. This network consists of two variables S and W, stand-
ing respectively for the soup and wine. I strictly prefer fish soup (S_f) to vegetable
soup (S_v), while my preference between red (W_r) and white (W_w) wine is condi-
tioned on the soup to be served: I prefer red wine if served a vegetable soup, and
white wine if served a fish soup.

$$S_f \succ S_v$$

S_f	$W_w \succ W_r$
S_v	$W_r \succ W_w$
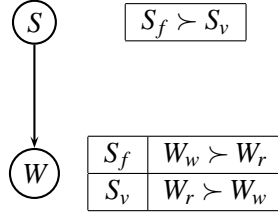
The preference information captured by a CP-net $\mathcal{N}$ can be viewed as a set of
logical assertions about a user's preference ordering over complete assignments to
variables in the network. These statements are generally not complete, that is, they
do not determine a unique preference ordering. Those orderings consistent with
$\mathcal{N}$ can be viewed as possible models of the user's preferences, and any preference
assertion that holds in all such models can be viewed as a consequence of the CP-
net [7].

Definition 13
Let $\mathcal{N}$ be a CP-net over propositional variables V, and $o, o' \in 2^V$ be two interpre-
*tations (also called outcomes). $o \succ o'$ is a **consequence** of $\mathcal{N}$, written $\mathcal{N} \models o \succ o'$,*
iff $o \succ o'$ holds in all preference orderings consistent with the ceteris paribus pref-
erence statements encoded by the CPTs of $\mathcal{N}$.

The set of consequences $o \succ o'$ of a CP-net constitutes a partial ordering over outcomes: o is preferred to o' in this ordering iff $\mathcal{N} \models o \succ o'$. This partial ordering can be represented by a directed graph, referred to as the induced preference graph:

1. The nodes of the induced preference graph correspond to the complete assignments to the variables of the network, and

2. there is an edge from node o to node o' if and only if the assignments at o and o' differ only in the value of a single variable X, and given the values assigned by o and o' to $Pa(X)$, the value assigned by o to X is preferred to the value assigned by o' to X[8].

In turn, the transitive closure of this graph specifies the partial ordering over outcomes induced by the CP-net. More precisely, we have that $\mathcal{N} \models o \succ o'$ if and only if there is a directed path from o to o' in the induced preference graph of $\mathcal{N}$.

Formally, the induced preference relation of $\mathcal{N}$, represented by the induced preference graph, is defined as follows:

Definition 14 (Induced preference relation)
*The **preference relation** over outcomes induced by a CP-net $\mathcal{N}$ is denoted by $\succ_{\mathcal{N}}$, and defined by $\forall o, o' \in 2^V$, $o \succ_{\mathcal{N}} o'$ if and only if $\mathcal{N} \models o \succ o'$.*

Informally, a CP-net $\mathcal{N}$ is satisfied by $\succ$ if $\succ$ satisfies each of the conditional preferences expressed in the CPTs of $\mathcal{N}$ under the *ceteris paribus* interpretation.

Definition 15 (Satisfiability of a CP-net)
*A CP-net $\mathcal{N}$ is **satisfiable** iff there is some ranking $\succ$ such that $\forall o$, $\mathcal{N} \not\models o \succ \ldots \succ o$.*

Example 9, continued: *The following figure shows the preference graph over outcomes induced by this CP-net. The bottom element ($S_v \wedge W_w$) is the worst outcome while the top element ($S_f \wedge W_w$) is the best.*
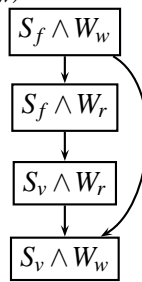
We can totally order the outcomes: $(S_f \wedge W_w) \succ (S_f \wedge W_r) \succ (S_v \wedge W_r) \succ (S_v \wedge W_w)$. This relation $\succ$ is the only ranking that satisfies this CP-net.

---

[8]In [6, 7], the arrows are oriented from less preferred to more preferred, but we choose here the opposite representation in order to stick with the other graphical representations used in this paper.

5.2 CP-Boolean games

Definition 16 (CP-Boolean games)
*A **CP-Boolean game** is a 4-uple $G = (N, V, \pi, \Phi)$, where $N = \{1, \ldots, n\}$ is a set of players, $V = \{x_1, \ldots, x_p\}$ is a set of propositional variables, $\pi : N \mapsto V$ is a control assignment function, and $\Phi = \langle \mathcal{N}_1, \ldots, \mathcal{N}_n \rangle$. Each $\mathcal{N}_i$ is a CP-net on V, and $\forall i \in N$, $\succeq_i = \succeq_{\mathcal{N}_i}$.*

Each CP-net $\mathcal{N}_i$ is a compact representation of the preference relation of player i on S.

Example 10 *$G = (N, V, \pi, \Phi)$ where $N = \{1, 2\}$, $V = \{a, b, c\}$, $\pi_1 = \{a, b\}$, $\pi_2 = \{c\}$, $\mathcal{N}_1$ and $\mathcal{N}_2$, as well as $\succeq_1 = \succeq_{\mathcal{N}_1}$ and $\succeq_2 = \succeq_{\mathcal{N}_2}$, are represented on the following figure.*

Arrows are oriented from more preferred to less preferred strategy profiles; we do not draw edges that are obtained from others by transitivity; and the dotted arrows indicate the links taken into account in order to compute Nash equilibria.

$$\mathcal{N}_1 \qquad \succeq_1 \qquad \mathcal{N}_2 \qquad \succeq_2$$
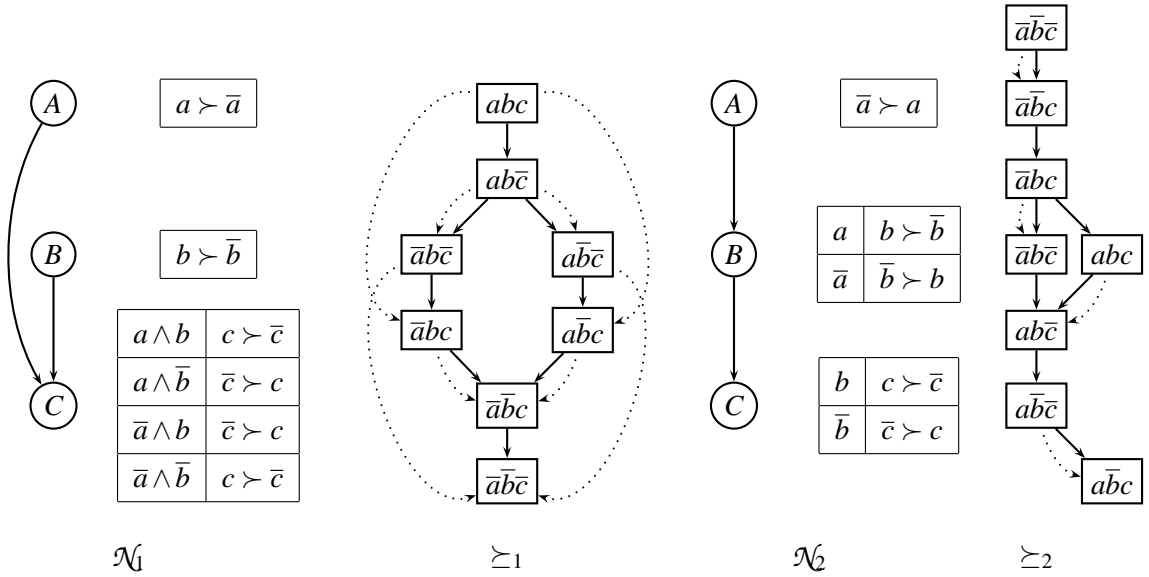
Using these partial pre-orders, Nash equilibria are: $NE_s = NE_w = \{abc\}$.

Consider now the CP-net $\mathcal{N}_2$ of player 2. Only two rankings satisfy this network:

$$\overline{a}\overline{b}\overline{c} \succ_2 \overline{a}\overline{b}c \succ_2 \overline{a}bc \succ_2 \overline{a}b\overline{c} \succ_2 abc \succ_2 ab\overline{c} \succ_2 a\overline{b}\overline{c} \succ_2 a\overline{b}c$$

$$\overline{a}\overline{b}\overline{c} \succ_2 \overline{a}\overline{b}c \succ_2 \overline{a}bc \succ_2 abc \succ_2 \overline{a}b\overline{c} \succ_2 ab\overline{c} \succ_2 a\overline{b}\overline{c} \succ_2 a\overline{b}c$$

Notice that we have $\mathcal{N}_2 \models \overline{a}\overline{b}\overline{c} \succ_2 \overline{a}bc$, but $\mathcal{N}_2 \not\models \overline{a}b\overline{c} \succ_2 abc$.

In [8] it was shown that, given a CP-net $\mathcal{N}$, we can easily determine the best outcome (or one of the nondominated outcomes, if the best outcome is not unique) among those preference rankings that satisfy $\mathcal{N}$. In our framework, a best outcome is a strategy profile which is preferred in a CP-net $\mathcal{N}$ in every preference ordering that satisfies $\mathcal{N}$.

Definition 17 (Best outcome)
Let $\mathcal{N}$ be a CP-net over the set of variables V.
*A strategy profile s is a **best outcome of** $\mathcal{N}$ if and only if $\forall s' \in S$, $\mathcal{N} \models s \succ s'$.*

Determining the best outcome, or one of the nondominated outcomes, among the preference rankings that satisfy $\mathcal{N}$ is called *outcome optimization*. If $\mathcal{N}$ is acyclic, the network does not generally determine a unique ranking but determines a unique best outcome.
This query can be answered using the *forward sweep* procedure, taking time linear in the network size [8, 6]. This procedure consists in instantiating variables following an order compatible with the graph, choosing for each variable its preferred value given the value of the parents.

Lemma 1 [8]
Let $\mathcal{N}$ be an acyclic CP-net over the set of variables V. The forward sweep procedure constructs the best outcome in S.

Since indifferences are not allowed, strictly and weakly dominated strategies are the same.

Proposition 10 *Let G be a CP-Boolean game such that the graph $\mathcal{G}_i$ for the player i is acyclic.*

- *i has a **dominated strategy** if and only if there is a variable controlled by i and independent (for $\succeq_i$) from all other variables, i.e. if and only if there is a variable $v \in \pi_i$ such that $Pa(v)$ could be simplified to $Pa(v) = \varnothing$ without changing preferences.[9]*
  Moreover, in this case, s_i dominates s'_i if and only if $s_i[v] \succ_i s'_i[v]$, where $s_i[v]$ denotes the value of v in the strategy[10] s_i.

- *i has a **dominant strategy**, that is a strategy which dominates all the others, if and only if every variable controlled by i is independent (for $\succeq_i$) from all other variables, i.e. if and only if for all variables $v \in \pi_i$, $Pa(v) \subseteq \pi_i$.*

---

[9]A variable w could appear in the parents of another variable v without having an influence on the preferences on v: we could have $w \in Pa(v)$ without v depending on w. We consider here that v depends on w if and only if $w \in Pa(v)$ (otherwise we remove w in $Pa(v)$).

[10]So, i prefers the value of v in s_i over the value of v in s'_i.

If $\mathcal{G}_i$ is not acyclic, i can have a dominated strategy even if there is no $v \in \pi_i$ such that $Pa(v) = \varnothing$.

Example 11 $G = (N, V, \pi, \Phi)$ *where* $N = \{1, 2\}$, $V = \{a, b, c, d\}$, $\pi_1 = \{a, b\}$, $\pi_2 = \{c, d\}$, $\mathcal{N}_1$ *and* $\mathcal{N}_2$ *are represented on the following figure.*

$\mathcal{N}_1$ $\mathcal{N}_2$
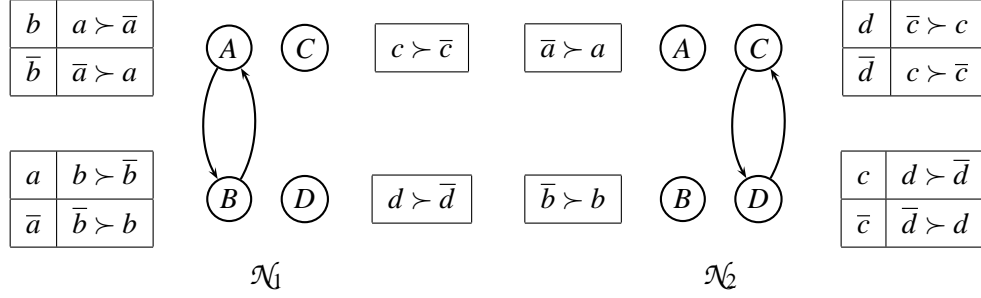
For player 1, *strategy* ab *dominates all other strategies, even if* $Pa(a) = b$ *and* $Pa(b) = a$.

In $\mathcal{N}_1$, *the configuration of a and b, controlled by player* 1, *is the same as the configuration of c and d in* $\mathcal{N}_2$, *which are controlled by* 2. *However,* 2 *has no dominated strategies.*

The previous example shows that cycles in a CP-net $\mathcal{N}$ can lead to an inconsistency in the associated preference relation $\succ_{\mathcal{N}}$. In this case, our results can be applied.[11]

The following lemma introduces the notion of best response, useful for characterizing Nash equilibria in CP-Boolean games.

Lemma 2 *Let* $G = (N, V, \pi, \Phi)$ *be a CP-Boolean game such that the graphs* $\mathcal{G}_i$ *are all acyclic, and* $i \in N$. *For all* $s_{-i} \in 2^{V \setminus \pi_i}$ *there exists a best response for i, that is, a strategy* $s_i^* \in 2^{\pi_i}$ *such that for all* $s_i' \neq s_i^*$, $(s_i^*, s_{-i}) \succ_i (s_i', s_{-i})$.

Notice that s_i^* depends only on s_{-i}, therefore $r_i(s_{-i})$ will denote the best response by i to s_{-i}. Then:

Lemma 3 s *is a SPNE if and only if for each i,* $s_i = r_i(s_{-i})$.

These two lemmas allow us to show the following propositions:

---

[11]If we want to generalize our results to cyclic CP-net, it is possible to take the semantics introduced by [9], which coincides with the standard semantics when $\succ_{\mathcal{N}}$ is consistent. In Brafman and Dimopoulos semantics, the preferences expressed in the CPT of a CP-net are *weak*. They defined $o \succ o'$ as $o \succeq o'$ and $o' \not\succeq o$.
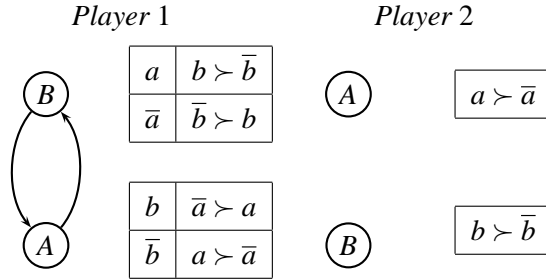
Proposition 11 *Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game such that the graphs $\mathcal{G}_i$ are all acyclic. Deciding whether a given strategy profile s is a SPNE of G is polynomial.*

Proposition 12

In a CP-Boolean game where the graph associated to each player's CP-net is acyclic, SPNE and WPNE coincide.

If the graphs are not all acyclic, that is, if there is at least one player whose graph is cyclic, this property is no longer guaranteed, and we can have a WPNE which is not a SPNE:

Example 12 *Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game, with $N = \{1, 2\}$, $V = \{a, b\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$. Players' preferences are represented on the following figure.*

Player 1 Player 2

a	$b \succ \bar{b}$
$\bar{a}$	$\bar{b} \succ b$

$a \succ \bar{a}$

b	$\bar{a} \succ a$
$\bar{b}$	$a \succ \bar{a}$

$b \succ \bar{b}$

Using these partial pre-orders, Nash equilibria are: $NE_s = \varnothing$, $NE_w = \{ab, \bar{a}b\}$.

Consequently, in the following we will talk of PNEs (instead of SPNEs and WP-NEs) as soon as the graph associated to each player's CP-net is acyclic.
The second property concerns a very interesting case where the existence and the uniqueness of PNE hold:

Proposition 13

Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game such that the graphs $\mathcal{G}_i$ are all identical ($\forall i, j, \mathcal{G}_i = \mathcal{G}_j$) and acyclic. Then G has one and only one PNE.

If the graphs are all acyclic but not identical, that is, if there are two players whose graphs differ over at least one edge, neither existence nor uniqueness are guaranteed:

Example 13 *Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game, with $N = \{1, 2, 3\}$, $V = \{a, b, c\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$. Players' preferences are represented in the following figure.*

$\mathcal{N}_1$ $\mathcal{N}_2$ $\mathcal{N}_3$
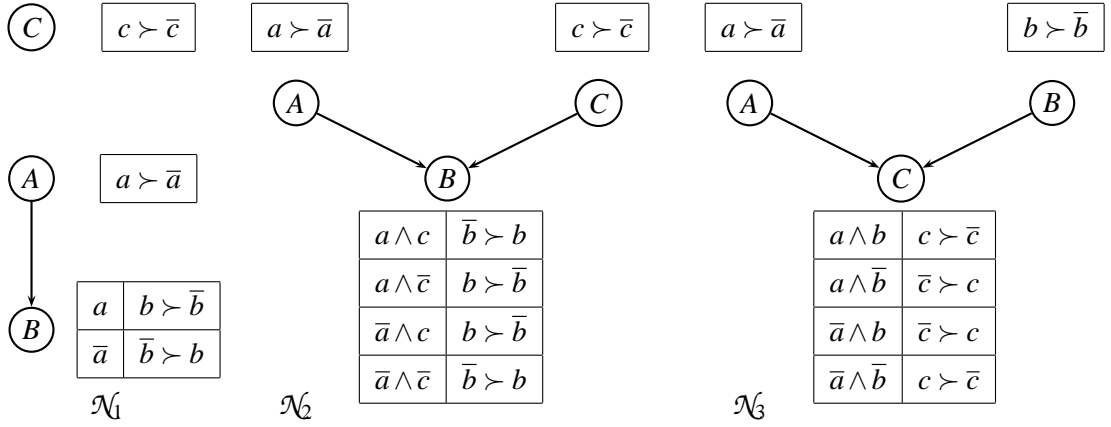
G has no PNE.

Example 14 *Let $G = (N,V,\pi,\Phi)$ be a CP-Boolean game, with $N = \{1,2\}$, $V = \{a,b\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$. Players' preferences are represented on the following figure.*

Player 1   Player 2
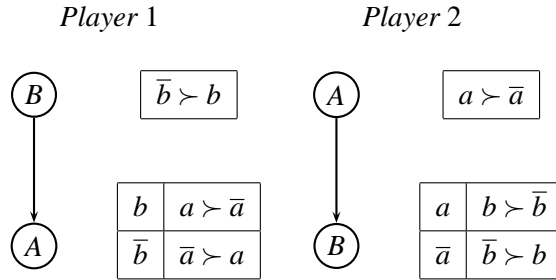
This game has 2 PNEs: $NE = \{ab, \bar{a}\bar{b}\}$

The point is that in general the graphs $\mathcal{G}_i$ for $i \in \{1,\ldots,n\}$ may not be identical. However, they may be *made* identical, once it is noticed that a CP-net $\langle \mathcal{G}, \mathcal{T}\rangle$ can be expressed by a CP-net $\langle \mathcal{G}', \mathcal{T}'\rangle$ as soon as the set of edges in $\mathcal{G}$ is contained in the set of edges in $\mathcal{G}'$. We may then take as a common graph $\mathcal{G}$ (to all players) the graph whose set of edges is the *union* of the set of edges of $\mathcal{G}_1,\ldots,\mathcal{G}_n$. The only problem is that the resulting graph may not be acyclic, in which case Proposition 13 is not applicable. Formally:

Definition 18 (Union graph)
*Let $G = (N,V,\pi,\Phi)$ be a CP-Boolean game. For each player i, $\mathcal{G}_i$ is denoted by (V, Edge_i), with Edge_i being the set of edges of i's CP-net. The **union graph** of G is defined by $\mathcal{G} = (V, \mathrm{Edge}_1 \cup \ldots \cup \mathrm{Edge}_n)$.*

We now define the normalized game equivalent to *G*, which is the game obtained from *G* by rewriting each player's preferences: the graph of each player's CP-net are replaced by the graph of the union of CP-nets of *G* and the CPTs of each player's CP-net are modified in order to fit with the new graph, keeping the same preferences.

Definition 19 (Normalized game)
Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game and let $\mathcal{G}$ be the union graph of G. The **normalized game equivalent to** *G, denoted by $G^* = (N, V, \pi, \Phi^*)$, is the game such that*

- *for each player i, $\mathcal{G}_i^* = \mathcal{G}$ and,*

- *if $\succ_i^y$ denotes the relation associated with $CPT_i(y)$ for player i's CP-net in G, then we have for G^*: $\forall x \in V$ such that x is a parent of y in G^* but not in G, $\succ_{i,x}^y = \succ_{i,\bar{x}}^y = \succ_i^y$).*

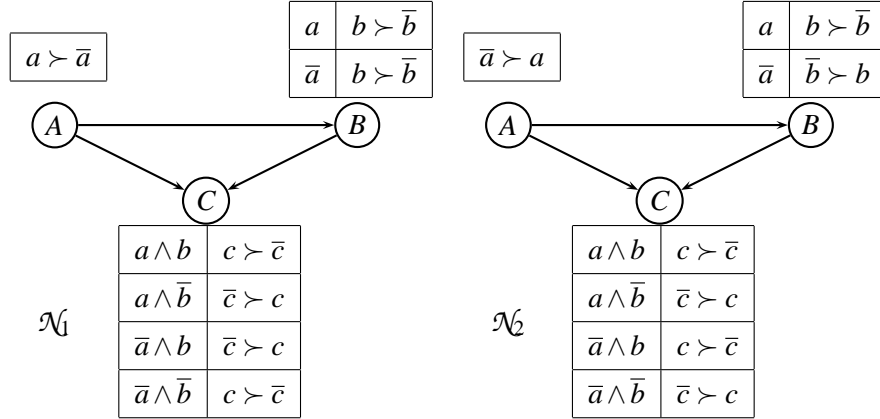The following lemma is straightforward:

Lemma 4
Let G be a CP-Boolean game and G^* its equivalent normalized game. Then G^* and G define the same preference relations on strategy profiles.

Therefore, if G^* is acyclic, then Proposition 13 applies, and G^* has one and only one PNE. Now, since *G* and G^* define the same pre-orders on *S*, the latter is also the only PNE of *G* (on the other hand, if the graph of G^* is cyclic, neither the uniqueness nor the existence of PNEs is guaranted, as you can see on Examples 13 and 14).

Proposition 14
Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game. If the union graph of G is acyclic then G has one and only one PNE.

Example 10, continued: *Players' preferences in the normalized game G^* (equivalent to G) are represented by the CP-nets given on the following figure.*

| | a | $b \succ \overline{b}$ |
| | $\overline{a}$ | $b \succ \overline{b}$ |

$a \succ \overline{a}$

$\overline{a} \succ a$

| | a | $b \succ \overline{b}$ |
| | $\overline{a}$ | $\overline{b} \succ b$ |

$\mathcal{N}_1$

$a \wedge b$	$c \succ \overline{c}$
$a \wedge \overline{b}$	$\overline{c} \succ c$
$\overline{a} \wedge b$	$\overline{c} \succ c$
$\overline{a} \wedge \overline{b}$	$c \succ \overline{c}$

$\mathcal{N}_2$

$a \wedge b$	$c \succ \overline{c}$
$a \wedge \overline{b}$	$\overline{c} \succ c$
$\overline{a} \wedge b$	$c \succ \overline{c}$
$\overline{a} \wedge \overline{b}$	$\overline{c} \succ c$

The union graph is acyclic, therefore Proposition 13 can be applied and G has one and only one PNE (abc).

>From Proposition 11, we know that if *G* is a CP-Boolean game such that the union graph is acyclic, then deciding whether a given strategy profile *s* is a PNE of *G* is polynomial.

There is a last condition (less interesting in practice because it is quite strong) guaranteeing the existence and the uniqueness of a PNE. This condition states that any variable controlled by an agent is preferentially independent from variables controlled by other agents (in other words, the parents of any variable controlled by a player *i* are also controlled by *i*), and that the graphs of players' CP-nets are all acyclic. In this case, each agent is able to instantiate her variables in an unambiguously optimal way, according to her preferences.

Proposition 15
Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game such that all graphs are acyclic and for every player $i \in N$, for every $v \in \pi_i$, we have $Pa(v) \in \pi_i$. Then G has one and only one PNE.

In the framework of CP-Boolean games, each agent's preferences are represented by a CP-net. Yet, to compute Nash equilibria, we are only interested in variables controlled by a player on her CP-net. The idea here is to introduce a new CP-net, called the *global CP-net of G*, unique for the set of players, gathering for each player the set of variables she controls[12]. Since $\{\pi_1, \ldots, \pi_n\}$ forms a partition of *V*, each variable will be present once and only once in this CP-net. Working on a unique CP-net will allow us to use classical tools from the CP-net framework.

---

[12] As pointed out by a referee, this construction can be viewed as a form of preference aggregation, where every player is a dictator for the variables she controls. The collective preference relation is well-defined only when the global CP-net is consistent, therefore this aggregation function applies only to a specific family of preference profiles and not all possible preference profiles, which explains why it escapes Arrow's theorem. See Section 4 of [28] for more details on this issue.

Definition 20 (global CP-net)
Let $G = (N,V,\pi,\Phi)$ be a CP-Boolean game.
*The **global CP-net** of G denotes the CP-net $\mathcal{N}^+ = \langle \mathcal{G}^+, \mathcal{T}^+ \rangle$ such that:*

- $\forall i \in N$, $\forall v \in \pi_i$, $CPT_i(v) \in \mathcal{T}^+$.

- $\mathcal{G}^+ = \langle V, \text{Edge}_1^+ \cup \ldots \cup \text{Edge}_n^+ \rangle$, *with each* Edge_i^+ *defined by*

$$\text{Edge}_i^+ = \{(w,v) \in \text{Edge}_i | v \in \pi_i\}$$

where Edge_i *is the set of edges in* $\mathcal{G}_i$.

Example 15 *Consider the same CP-Boolean game as in Example 10:*
$G = (N,V,\pi,\Phi)$ where $N = \{1,2\}$, $V = \{a,b,c\}$, $\pi_1 = \{a,b\}$, $\pi_2 = \{c\}$, $\mathcal{N}_1$ and $\mathcal{N}_2$ are represented on the following figure. On the figure, each box contains the variables controlled by the player concerned, therefore it contains the CPTs of the global CP-net of G.

$$\mathcal{N}_1 \qquad\qquad\qquad \mathcal{N}_2$$
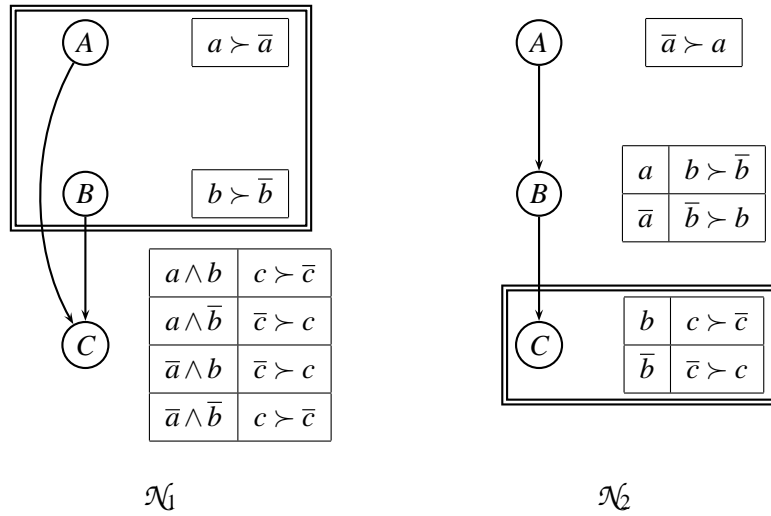
The global CP-net $\mathcal{N}^+$ of G is represented on the following figure (we keep the boxes, the edges entering in each box, and the internal edges of each box).
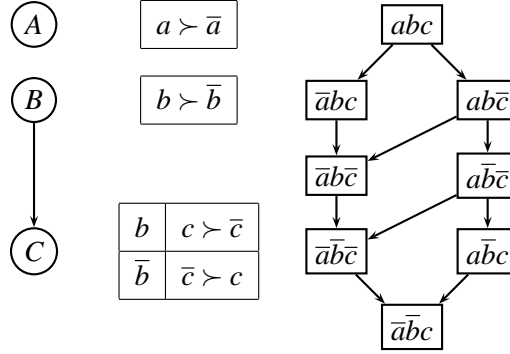
Since computing the global CP-net of a game allows us to have a unique CP-net, computing the best outcome of this CP-net, if it is acyclic, can give us interesting results.

Proposition 16 [13]
Let G be a CP-Boolean game such that the graphs G_i are all acyclic and $\mathcal{N}^+$ be the global CP-net of G. Let s be a strategy profile. If $\mathcal{N}^+$ is an acyclic CP-net, we have the following equivalence:
s is a PNE of G if and only if s is the best outcome of $\mathcal{N}^+$.

To prove this proposition, we will need the following lemma :

Lemma 5
Let G be a CP-Boolean game, $\mathcal{N}$ be an acyclic CP-net of G, and s be a strategy profile.
If $\exists s': s' \succ s$, then $\exists s'': s'' \succ s$ such that s and s'' differ from only one variable.

This proposition allows us to compute more easily the PNE of a CP-Boolean game, using the "forward sweep" procedure.

Example 15, continued: *G has one and only one PNE: abc.*
And the global CP-net $\mathcal{N}^+$ has one best outcome: abc.

Proposition 17
Deciding whether an acyclic CP-Boolean game $G = (N, V, \pi, \Phi)$ (i.e., where each CP-net $\mathcal{N}_i \in \Phi$ is acyclic) has a PNE is NP-complete. Hardness holds even if the number of players is restricted to 2.

The complexity of the problem of deciding if a general CP-Boolean game has a PNE is still open.

---

[13]Thanks to Nic Wilson for pointing this to us.

6 Related work

The related work section is structured in the following way. First we discuss a few previous works on the representation of static games in propositional logic formalisms and give computational characterizations of Nash equilibria. Then we discuss work which relates CP-nets to pure-strategy Nash equilibria. Finally, we compare Boolean games to other frameworks for the compact representation of games, especially graphical games.[14]

6.1 Expressing static games in propositional logic

In [16], a game in normal form is mapped into a *logic program with ordered disjunction* (LPOD) where each player owns a set of clauses that encodes 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 LPODs are induced from games and could be overwhelmed by allowing to express the players' preferences by any LPOD (in the same spirit as our Section 5).

In [13], a strategic game is represented using a *choice logic program*, where a set of rules expresses 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 for computing Nash equilibria for finite strategic games.

An earlier work [37] uses logic programming as well for specifying games: the *independent choice logic* represents games (possibly dynamic and under incomplete knowledge) using a logic program to model the agent and the environment and allows for expressing Nash equilibria as well as other solution concepts.

6.2 Games and CP-nets

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 actions available to the player. Preferences over a player's actions given the

---

[14]We do not discuss previous papers on Boolean games [23, 22, 14] in this section. They are obviously related to our work (this is a euphemism...) but they have been discussed in Sections 3 and 4.

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 [16] and [13], is that players can express arbitrary binary preferences, including extreme cases where the satisfaction of a player's goal may depend only on 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.

6.3 Other classes of compactly represented games

In the last few years a lot of work has been done on defining compact representation languages for games and studying the computation of Nash equilibria given a compact representation of a game. Boolean games and CP-Boolean games are just two families of compactly represented games, which relate to other classes of compactly represented games but are significantly different, as we will see.

The closest framework to the two models developed in this paper is that of *graphical games*. A graphical game specifies, for each player i, the set of all players that have an influence on i (which is represented by a directed graph on the set of players, where an edge from i to j means that the utility of i depends on the action chosen by j). Then, the utility of player i is compactly represented by a *utility table* that specifies a value for each combination of actions of the players on which i depends. This graphical representation for games has been independently proposed by [25] and [26][15]. The complexity of the existence of pure-strategy Nash equilibria in graphical games was extensively studied in [20]; these results were strengthened in [15].

One may wonder whether Boolean games are just a specific case of graphical games (restricted to dichotomous preferences). This is actually not the case, because there is no polynomial-size translation of propositional formulae into utility tables, which implies that even the simple propositional preference representation

---

[15]The latter framework, called *multi-agent influence diagrams* (MAIDs), also allows for dynamicity and incomplete knowledge.

framework used in Section 3 can sometimes be exponentially more compact than a representation by utility tables.[16]

The next example gives a family of Boolean games that are represented in space polynomial in the number of players, and whose graphical representation is exponential in the number of agents.

Example 16 *For every number of players $n \geq 0$, let $G_n = (N, V, \pi, \Phi)$ be a Boolean game, with $N = \{1, 2, \ldots n\}$, $V = \{C_1, \ldots, C_n\}$, for all $i \in n$, $\pi_i = \{C_i\}$ and for all $i \in n$, $\varphi_i = C_i \leftrightarrow (\bigwedge_{j<i} \neg C_j \wedge \bigwedge_{k>i} C_k).$[17]*

In the graphical representation of G_n, first notice that the utility of each agent depends on the actions of all *other agents, which implies that the dependency graph between agents is complete, and the utility tables are exponentially large (for the sake of illustration, we give the utility table when $n = 3$):*
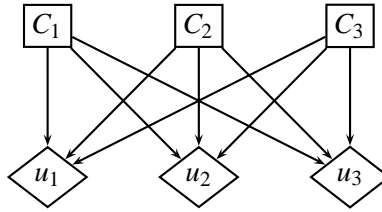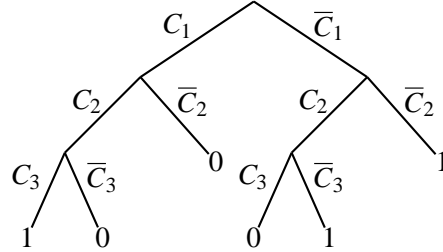

Utility tables:

Strategy profiles	u_1	u_2	u_3
$C_1 C_2 C_3$	1	0	0
$C_1 C_2 \overline{C}_3$	0	0	1
$C_1 \overline{C}_2 C_3$	0	1	0
$C_1 \overline{C}_2 \overline{C}_3$	0	1	1
$\overline{C}_1 C_2 C_3$	0	1	0
$\overline{C}_1 C_2 \overline{C}_3$	1	0	1
$\overline{C}_1 \overline{C}_2 C_3$	1	0	1
$\overline{C}_1 \overline{C}_2 \overline{C}_3$	1	1	0

---

[16]More precisely, the representation of utility functions by sum of local utilities, each of which being expressed by utility tables, is polynomial if and only if there exists a constant K bounding the number of variables involved in each table, *i.e.*, if and only if players' utilities are all K-additive. Now, the utility functions expressable by means of propositional formulae are K-additive if and only if the size of the formulas used in the representation is bounded by K (see for instance [10]).

[17]With the following convention: for every formula Ψ, $\bigwedge_{x \in D} \Psi_x = \top$ if $D = \varnothing$.

Notice that the representation of utilities remains exponentially large in n even if the utilities are represented by decision trees instead of tables; for example, for player 1*:*

$$C_1 \qquad \overline{C}_1$$
$$C_2 \quad \overline{C}_2 \qquad C_2 \quad \overline{C}_2$$
$$C_3 \quad \overline{C}_3 \qquad 0 \qquad C_3 \quad \overline{C}_3 \qquad 1$$
$$1 \qquad 0 \qquad\qquad 0 \qquad 1$$

Comparing graphical games and CP-Boolean games is less easy, because graphical games are tailored for numerical preferences whereas CP-Boolean games are tailored for ordinal preferences (where the preference relation induced by a CP-net may not be complete). However, we want to stress the informal proximity between both frameworks: CP-nets are the most well-known graphical model for ordinal preferences, therefore CP-Boolean games can be viewed as the ordinal counterpart of graphical games.[18]

A specificity of Boolean games (and CP-Boolean games) is that a player may be able to control several variables. Of course, Boolean games can be translated in such a way that each player controls a single variable: if player i controls a set of variables π_i, we consider a new variable v_i whose domain is 2^{π_i}. But this translation is clearly exponentially large. Another idea would consist in duplicating a player controlling p variables into p different players with identical preferences; clearly, the game obtained would be very different from the initial game, because the duplicated players cannot coordinate when choosing their actions.

Another difference between Boolean games and graphical games is that Boolean games allow only for binary variables. This is not really a loss of generality, because it is well-known that each p-ary variable can be expressed by $\lceil \log p \rceil$ binary variables.

Another class of compactly-represented games is that of *circuit games* [38], where the utility of each player is represented by a Boolean circuit computing the payoffs. Because any propositional formula can be translated into a polynomial-size Boolean circuit, circuit games are more succinct than Boolean games. Unsurpris-

---

[18]Another graphical framework for numerical utilities is that of G-nets [27], whose goal is rather to represent causality relations in a graphical way and therefore is rather far from Boolean games.

ingly, deciding the existence of a pure Nash equilibrium in a circuit game is Σ_2^p-complete (Theorem 6.1 in [38]).

Lastly, local-effect games and action-graph games [31, 2], which are dedicated to classes of games where players may share some actions and the utility of a player depends only on the number of players who choose each action, are much further from Boolean games (and CP-Boolean games).

6.4 Controllability in propositional logic

In Section 3.1 we mentioned a relationship to propositional controllability, as studied by [5] and [30]. A recent line of work within this alley [40] 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, [40] focus 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. Once this framework has been 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. These frameworks allow for representing compactly either *numerical* preferences (utility functions on 2^V) or *ordinal* preferences (partial or complete orderings on 2^V). Here, we focused on a specific representation language, namely CP-nets. Our conference paper [3] also includes an extension of Boolean games with prioritized goals (evoked here very briefly in Section 4.3).

Boolean games can be seen as the logical counterpart of graphical games. In this setting, our results show the impact of the structure of players' preferences on the properties of the game (e.g., existence of pure-strategy Nash equilibria) and on the computational complexity of some specific problems. In particular, Proposition 14 shows that if the players' preferences share enough preferential independence properties, then the existence and the uniqueness of a pure-strategy Nash equilibrium is guaranteed and that the latter can be computed in polynomial time.

Two extensions to our work should not present any particular difficulty (we omitted them for the sake of simplicity). First, extending Boolean games so as to handle variables with nonbinary domains, by the usual way of binarizing nonbinary variables. Second, adding constraints restricting the set of individual strategies as suggested in Section 3.

On the contrary, some other extensions demand much more additional work and are left for further research:

- in Section 3.2 we characterized dominated strategies in Boolean games; however, we did not investigate complexity issues for the iterative elimination of dominated strategies[19].

- integrating Boolean games with languages for representing numerical preferences (for instance using sets of weighted formulas, e.g. as in [10]), and for addressing the computation of mixed-strategy Nash equilibria in this setting.

- defining and studying dynamic extensions of Boolean games, as well as allowing for incomplete knowledge states.

Yet another interesting issue for further research stems from the observation that is often unnatural to require players to express preference relations over strategy profiles. Rather, they have natural preferences over a space of *possible outcomes*, and each strategy profile is then mapped to an outcome. Expressing preferences over outcomes rather than strategy profiles is not only more natural in many cases, but it may also make the description of players' preference much more succinct, since there might generally be far less outcomes than strategy profiles. The point that deserves then some attention is how the mapping from strategy profiles to outcomes can be described succinctly; this can be done for instance using a propositional language for concurrent actions, such as *C* [18].

We end this paper by some positioning with respect to a large stream of work that aims at bridging logic and games (see for instance [39]). We remark that in Boolean games logic plays a role only in the description of the players' preferences, but the control function remains extralogical. We could think of "modalizing" Boolean games, which can be done in many ways. First, we could modalize preferences, by expressing preferences directly in the language as in modal logics of preferences (e.g. [21]). Then, we may also want to modalize the "control" component of

---

[19]For games represented in normal form, deciding whether there is some path that eliminates a given strategy is polynomial with strict dominance and NP-complete with weak dominance [19, 11]. We are not aware of any extension of these results to compactly represented games.

Boolean games; this could be done for instance by expressing the control function π using group ability modalities [40].[20]

Acknowledgment

References

[1] Krzysztof R. Apt, Fancesca Rossi, and Kristen Brent Venable. CP-nets and Nash equilibria. In Elsevier, editor, *Proceedings of Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS'05)*, Singapore, December 13-16 2005.

[2] Navin Bhat and Kevin Leyton-Brown. Computing Nash equilibria of Action-Graph Games. In *Proceedings of Conference on Uncertainty in Artificial Intelligence (UAI'04)*, pages 35–42, 2004.

[3] Elise Bonzon, Marie-Christine Lagasquie-Schiex, and Jérôme Lang. Compact preference representation for Boolean games. In *Ninth Pacific Rim International Conference on Artificial Intelligence (PRICAI'06)*, volume 4099, pages 41–50. Springer-Verlag, 2006.

[4] Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang, and Bruno Zanuttini. Boolean games revisited. In *17th European Conference on Artificial Intelligence (ECAI'06)*, pages 265–269. Springer-Verlag, 2006.

[5] Craig Boutilier. Toward a logic for qualitative decision theory. In *Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 75–86, 1994.

[6] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. CP-nets : A Tool for Representing and Reasoning with Conditional *Ceteris Paribus* Preference Statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

---

[20]As pointed out by a referee, this has the disadvantage of losing expressivity with respect to Nash equilibria, except if there is a one-to-one correspondence between strategy profiles and outcomes (see [12]).

[7] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Preference-Based Constrained Optimization with CP-nets. *Computational Intelligence*, 20(2):137–157, 2004. Special Issue on Preferences.

[8] Craig Boutilier, Ronen I. Brafman, Holger H. Hoos, and David Poole. Reasoning with Conditional Ceteris Paribus Preference Statements. In *Proceedings of Uncertainty in Artificial Intelligence (UAI'99)*, 1999.

[9] Ronen I. Brafman, Yannis Dimopoulos. Extended semantics and optimization algorithms for CP-networks *Computational Intelligence*, 20(2):218–245, 2004.

[10] Yann Chevaleyre, Ulle Endriss, and Jérôme Lang. Expressive Power of Weighted Propositional Formulas for Cardinal Preference Modelling. In P. Doherty, J. Mylopoulos, and C. Welty, editors, *Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 145–152. AAAI Press, June 2006.

[11] Vincent Conitzer and Tuomas Sandholm. Complexity of (Iterated) Dominance. In *Proceedings of the 6th ACM Conference on Electronic Commerce (EC'05)*, pages 88–97, 2005.

[12] Effectivity and Noncooperative Solution Concepts. In *Seventh Conference on Logic and the Foundations of Game and Decision Theory (LOFT'06)*, 2006.

[13] Marina De Vos and Dirk 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.

[14] Paul E. Dunne and Wiebe van der Hoek. Representation and Complexity in Boolean Games. In *Proceedings of the Ninth European Conference on Logics in Artificial Intelligence (JELIA'04)*, volume LNCS 3229, pages 347–359. José Júlio Alferes et João Alexandre Leite (eds), 2004.

[15] Felix Fischer, Markus Holzer, and Stefan Katzenbeisser. The influence of neighbourhood and choice on the complexity of finding pure Nash equilibria. *Inf. Process. Lett.*, 99(6):239–245, 2006.

[16] Norman Foo, Thomas Meyer, and Gerhard Brewka. LPOD Answer Sets and Nash Equilibria. In M. Maher, editor, *Proceedings of the 9th Asian Computer Science Conference (ASIAN'04)*, pages 343–351. Chiang Mai, Thailand, Springer LNCS 3321, 2004.

[17] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman and Company, New York, 1979.

[18] Enrico Giunchiglia, Jooyung Lee, Vladimir Lifschitz, Norman McCain and Hudson Turner. Nonmonotonic causal theories *Artificial Intelligence*, Vol. 153, 2004, pp. 49-104.

[19] Itzhak Gilboa, Ehud Kalai, and Eitan Zemel. The complexity of eliminating dominated strategies. *Math. Oper. Res.*, 18(3):553–565, 1993.

[20] Georg Gottlob, Gianluigi Greco, and Francesco Scarcello. Pure Nash Equilibria: Hard and Easy Games. *Journal of Artificial Intelligence Research*, 24:357–406, 2005.

[21] Sven Ove Hansson. *Handbook of Philosophical Logic*, volume 4, chapter Preference Logic, pages 319–393. D. Gabbay and F. Guenthner, 2001.

[22] Paul Harrenstein. *Logic in Conflict*. PhD thesis, Utrecht University, 2004.

[23] Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer, and Cees Witteveen. Boolean Games. In J. van Benthem, editor, *Proceedings of the 8th International Conference on Theoretical Aspects of Rationality and Knowledge (TARK'01)*, volume Theoretical Aspects of Rationality and Knowledge, pages 287–298. San Francisco, Morgan Kaufmann, 2001.

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

[25] Michael Kearns, Michael L. Littman, and Satinder Singh. Graphical Models for Game Theory. In *Uncertainty in Artificial Intelligence (UAI'01)*, 2001.

[26] Daphne Koller and Brian Milch. Multi-Agent Influence Diagrams for Representing and Solving Games. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 1027–1034, 2001.

[27] Pierfrancesco La Mura. Game Networks. In *Uncertainty in Artificial Intelligence (UAI'00)*, pages 335–342, 2000.

[28] Jérôme Lang. Voting and aggregation on combinatorial domains with structured preferences. *Proceedings of the Twenthieth Joint International Conference on Artificial Intelligence (IJCAI'07)*, 1366-1371, 2007.

[29] Jérôme Lang, Paolo Liberatore, and Pierre Marquis. Propositional Independence - Formula-Variable Independence and Forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.

[30] Jérôme Lang and Pierre Marquis. Two forms of dependence in propositional logic: controllability and definability. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'98)*, pages 268–273, 1998.

[31] Kevin Leyton-Brown and Moshe Tennenholtz. Local-Effect Games. In *International Joint Conferences on Artificial Intelligence (IJCAI'03)*, pages 772–777, 2003.

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

[33] Fangzhen Lin and Raymond Reiter. Forget it. In *Proceedings of the National Conference on Artificial Intelligence (AAAI'94)*, pages 154–159, 1994.

[34] Martin J. Osborne. *An Introduction to Game Theory*. Oxford University Press, 2004.

[35] Martin J. Osborne and Ariel Rubinstein. *A course in game theory*. MIT Press, 1994.

[36] Christos Papadimitriou. *Computational Complexity*. Addison-Wesley, 1994.

[37] David Poole. The Independent Choice Logic for modelling multiple agents under uncertainty. *Artificial Intelligence*, 94(1-2):7–56, 1997. Special issue on economic principles of multi-agent systems.

[38] Grant Schoenebeck and Salil Vadhan. The Computational Complexity of Nash Equilibria in Concisely Represented Games. In *Proceedings of Conference on Electronic Commerce (EC'06)*, 2006.

[39] Johan van Benthem. Open problems in logic and games. In *We Will Show Them! (1)*, pages 229–264, 2005.

[40] Wiebe van der Hoek and Michael Wooldridge. On the Logic of Cooperation and Propositional Control. *Artificial Intelligence*, 164(1-2):81–119, 2005.

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

A Proofs

Proposition 1

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

Proof:

Intuitively, if a player's goal has a prime implicant α in which all the variables are controlled by this player, then she has a winning strategy. Indeed, this term α can be satisfied, hence the goal which contains it is also satisfied, and the player wins.

- If $PI_{\pi_i}(\varphi_i) \neq \varnothing$, then $\exists \alpha \in 2^V$ such that: $\alpha \models \varphi_i$ and $Var(\alpha) \subseteq \pi_i$.

  So, as i controls all variables in α, i has a strategy $s_i \in 2^{\pi_i}$ such that $s_i \models \alpha$. Then, $\forall s_{-i} \in 2^{\pi_{-i}}(s_{-i}, s_i) \models \varphi_i$.

- According to Definition 3, i has a winning strategy iff:

$$\exists s_i \in 2^{\pi_i}, \forall s_{-i} \in 2^{\pi_{-i}}(s_{-i}, s_i) \models \varphi_i$$

  So, player i can satisfy her goal whatever the choices of her opponents. Therefore, there exists $\alpha \in 2^{\pi_i}$ such that: $\alpha \models \varphi_i$ and $Var(\alpha) \subseteq \pi_i$, i.e., α is a π_i-implicant of φ_i. Thus there exists a π_i-prime implicant α' of φ_i. Therefore $PI_{\pi_i}(\varphi_i) \neq \varnothing$.

$\blacksquare$

Proposition 2

Let $G = (N, V, \pi, \Phi)$ be a Boolean game and let $s \in 2^V$. s is a pure-strategy Nash equilibrium for G iff for all $i \in N$, $s \not\models \varphi_i$ implies $s_{-i} \models \neg \varphi_i$.

Proof: s is a PNE for G iff

$\forall i \in N, \forall s_i' \in 2^{\pi_i}, u_i(s) \geq u_i(s_{-i}, s_i')$, i.e., $\forall i \in N, \forall s_i' \in 2^{\pi_i}, u_i(s) = 1$ or $u_i(s_{-i}, s_i') = 0$,

i.e., $\forall i \in N, u_i(s) = 1$ 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., $s_{-i} \models \neg \varphi_i$.

So, we have $s \models \varphi_i$ or $s_{-i} \models \neg \varphi_i$, that is $s \not\models \varphi_i$ implies $s_{-i} \models \neg \varphi_i$. $\blacksquare$

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: $(\forall i \in N, s \models \varphi_i$ or $s_{-i} \models \neg\varphi_i) \Leftrightarrow (\forall i \in N, s \models \varphi_i$ or $s \models \neg\exists i : \varphi_i) \Leftrightarrow \forall i \in N, s \models \varphi_i \vee (\neg\exists i : \varphi_i) \Leftrightarrow s \models \bigwedge_i (\varphi_i \vee (\neg\exists i : \varphi_i))$.

∎

Proposition 4

If G is a 2-player 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.

∎

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 2-player zero-sum games.

Proof: Membership in Σ_2^p comes from the following algorithm. Guess a strategy profile and check that no player has a better strategy. This is as many checks as players, and each check is the complement of guessing a better strategy and deciding whether it is better by evaluating the player's goal in the new strategy profile. Thus each check is in coNP, hence checking a strategy profile can be done by a polynomial number of calls to a coNP oracle (one for each player), hence deciding whether there is one is in $NP^{coNP} = \Sigma_2^P$.
Hardness is obtained by a reduction from the problem of deciding the validity of a $\text{QBF}_{2,\exists}$. Given $Q = \exists A, \forall B, \Phi$, where A and B are disjoint sets of variables and Φ is a formula of $L_{A \cup B}$, we define a 2-player zero-sum Boolean game by $\varphi_1 = \Phi \vee (x \leftrightarrow y)$, where x, y are new variables ($x, y \notin A \cup B$) 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 $(M_A, \bar{x})$ 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 $\bar{y}$ (resp. y) if 1 plays x (resp. $\bar{x}$), 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.

∎

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 we restrict the number of players to 2.

Proof: If φ_i is in DNF, then $\exists i : \varphi_i$ can be computed in linear time [29, 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 2-player 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) $\neg \exists X \cup \{x\} : (x = y) \equiv \bot$.
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.

∎

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 $(s_i', s_{-i}) \models$ φ_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) $\Leftrightarrow (\varphi_i)_{s_i} \not\models (\varphi_i)_{s_i'}$.

∎

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 comes from the following algorithm. First guess a candidate strategy. To check whether it weakly dominates s_i', consider every strategy s_{-i} and check $u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$, and guess one for which dominance is strict. Thus the candidate strategy can be checked with one call to a coNP oracle and one to an NP oracle, hence deciding whether s_i' is weakly dominated is in $NP^{NP \cup coNP} = \Sigma_2^P$.

Hardness is obtained again by a reduction from the problem of deciding the validity of a $\text{QBF}_{2,\exists}$. 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 $(M_A', \bar{a})$. 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 $s_1 = (M_A, \bar{a})$. 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 (by s_1) 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.

∎

Proposition 9 *Let $G = (N, V, \pi, \Phi)$ be a distributed evaluation game and let $s \in S$. If s is a SPNE then for all $i \in N$, $Sat(s, \Sigma_i)$ is maximal s_{-i}-consistent.*

Proof: Assume that s is a SPNE, which means that $\forall i \in N$, $\forall s_i' \in 2^{\pi_i}$, $(s_i', s_{-i}) \preceq_i (s_i, s_{-i})$. This is equivalent to $\forall i \in N$, $\forall s_i' \in 2^{\pi_i}$, $Sat((s_i', s_{-i}), \Sigma_i) \subseteq_i Sat((s_i, s_{-i}), \Sigma_i)$, which implies that $\forall i \in N$, $\nexists s_i' \in 2^{\pi_i}$ such that $Sat((s_i', s_{-i}), \Sigma_i) \supset_i Sat((s_i, s_{-i}), \Sigma_i)$. Therefore, $Sat(s, \Sigma_i)$ is maximal s_{-i}-consistent.

∎

Proposition 10 *Let G be a CP-Boolean game such that the graph $\mathcal{G}_i$ for the player i is acyclic.*

- *i has a **dominated strategy** if and only if there is a variable controlled by i and independent (for $\succeq_i$) from all other variables, i.e. if and only if there is a variable $v \in \pi_i$ such that $Pa(v)$ could be simplified to $Pa(v) = \varnothing$ without changing preferences.*
  Moreover, in this case, s_i dominates s'_i if and only if $s_i[v] \succ_i s'_i[v]$, where $s_i[v]$ denotes the value of v in the strategy s_i.

- *i has a **dominant strategy**, that is a strategy which dominates all the others, if and only if every variable controlled by i is independent (for $\succeq_i$) from all other variables, i.e. if and only if for all variables $v \in \pi_i$, $Pa(v) \subseteq \pi_i$.*

Proof:

- $\Rightarrow$ Suppose that $\forall v \in \pi_i$, $Pa(v) \neq \varnothing$. As $\mathcal{G}_i$ is acyclic, we know that exists $v \in \pi_i$ such that $Pa(v) \subseteq \pi_{-i}$. So, $\exists u_v$ an instantiation of all parents of v such that $v \succ_{i,u_v} \bar{v}$, and $\exists u'_v$ another instantiation of all parents of v such that $\bar{v} \succ_{i,u'_v} v$ (otherwise, the graph could be simplified in order to have $Pa(v) = \varnothing$). $\forall s_i \in 2^{\pi_i}$, exists $s'_i \in 2^{\pi_i}$ and exists $s_{-i} \in 2^{\pi_{-i}}$ such that either $(s_i, s_{-i}) \prec_{i,u_v} (s'_i, s_{-i})$, or $(s_i, s_{-i}) \prec_{i,u'_v} (s'_i, s_{-i})$. i has no dominated strategies.

  $\Leftarrow$ Let i a player, and $\exists v \in \pi_i$ such that $Pa(v) = \varnothing$.
  So either $v \succ_i \bar{v}$ or $\bar{v} \succ_i v$. So, strategy s_i, defined by assigning the best value to v (i.e. v in the first case) is better for i to all other s'_i, defined by assigning the other value to v (i.e. $\bar{v}$ in the first case). As $Pa(v) = \varnothing$, we know than $\forall s_{-i} \in 2^{\pi_{-i}}$, $(s_i, s_{-i}) \succ_i (s'_i, s_{-i})$. s_i dominates s'_i, and we have $s_i[v] \succ_i s'_i[v]$.

- $\Rightarrow$ Suppose that $\exists v \in \pi_i$, such that $\exists w \in Pa(v)$ such that $w \notin \pi_i$. So, $w \in \pi_{-i}$. So, $\exists u_v$ an instantiation of all parents of v such that $v \succ_{i,u_v} \bar{v}$, and $\exists u'_v$ another instantiation of all parents of v such that $\bar{v} \succ_{i,u'_v} v$. $\forall s_i \in 2^{\pi_i}$, exists $s'_i \in 2^{\pi_i}$ and exists $s_{-i} \in 2^{\pi_{-i}}$ such that either $(s_i, s_{-i}) \prec_{i,u_v} (s'_i, s_{-i})$, or $(s_i, s_{-i}) \prec_{i,u'_v} (s'_i, s_{-i})$. i has no dominant strategies.

  $\Leftarrow$ Let i a player, and $\forall v \in \pi_i$, $Pa(v) \subseteq \pi_i$.
  Let $(x_1, \ldots, x_k)$ be a topological order on π_i with respect to $\mathcal{G}_i$. We define a strategy s_i as follows.
  As $\mathcal{G}_i$ is acyclic, by construction we have $Pa(x_1) = \varnothing$, and either $x_1 \succ_i \overline{x_1}$ or $\overline{x_1} \succ_i x_1$. Define strategy s_{x_1} as assigning the best value to x_1 (i.e., x_1 in the first case and $\overline{x_1}$ in the second case). This process can be

iterated along the topological order, since by definition, at any time the value of each parent of the current variable x_i will have been fixed by i. We thus end up with a strategy profile s_i.

We now show that s_i is a dominant strategy. Indeed, assume it is not one, and let a strategy s_i' and a strategy s_{-i} such that $(s_i', s_{-i}) \succ_i (s_i, s_{-i})$. Let x be the first variable in π_i, with respect to the topological order above, such that $s_i[x] \neq s_i'[x]$. Then by construction of s_i we have $s_i[x] \succ_{i,u_x} s_i'[x]$, where u_x is the instantiation of all parents of x. Now this instantiation is the same in (s_i, s_{-i}) and (s_i', s_{-i}), because we have chosen the first variable over which s_i' and s_i disagree w.r.t. the topological order. It follows that replacing $s_i'[x]$ with $s_i[x]$ in s_i' yields a better strategy, and iterating this reasoning we end up with showing that replacing every $s_i'[x]$ with $s_i[x]$ in s_i' yield ever better strategies. Since this process ends up with s_i, we finally have $(s_i, s_{-i}) \succ_i (s_i', s_{-i})$, which together with the assumption $(s_i', s_{-i}) \succ_i (s_i, s_{-i})$ contradicts the fact that acyclic CP-nets are always satisfiable.

■

Lemma 2

Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game such that the graphs $\mathcal{G}_i$ are all acyclic, and $i \in N$. For all $s_{-i} \in 2^{\pi_{-i}}$ there exists a best response for i, that is, a strategy $s_i^* \in 2^{\pi_i}$ such that for all $s_i' \neq s_i^*$, $(s_i^*, s_{-i}) \succ_i (s_i', s_{-i})$.

Proof: Suppose that s_i^* does not exist. So, $\exists s_{-i} \in 2^{\pi_{-i}}$ such that $\forall s_i \in 2^{\pi_i}$, $\exists s_i' \neq s_i$ such that $(s_i', s_{-i}) \succ_i (s_i, s_{-i})$. And, there exists a $s_i'' \neq s_i'$ such that $(s_i'', s_{-i}) \succ_i (s_i', s_{-i})$, and so on. As each agent has a finite number of strategies, and $\succ_i$ is transitive, $\exists s_i, s_i' \in 2^{\pi_i}$ such that $(s_i', s_{-i}) \succ_i (s_i, s_{-i})$, $(s_i'', s_{-i}) \succ_i (s_i', s_{-i})$ and $(s_i, s_{-i}) \succ_i (s_i'', s_{-i})$, which contradicts the fact than $\mathcal{G}_i$ is acyclic.

■

Lemma 3

s is a SPNE if and only if for each i, $s_i = r_i(s_{-i})$.

Proof:

$\Rightarrow$ Let $s = (s_1, \ldots, s_n)$ be a SPNE: $\forall i \in N$, $\forall s_i' \in 2^{\pi_i}$, $(s_i, s_{-i}) \succeq_i (s_i', s_{-i})$. But, by definition of a CP-net, we cannot have egality between two strategy profiles, so we know that: $(s_i, s_{-i}) \neq_i (s_i', s_{-i})$ if $s_i \neq s_i'$. Thus, we have $\forall i \in N$, $\forall s_i' \neq s_i \in 2^{\pi_i}$, $(s_i, s_{-i}) \succ_i (s_i', s_{-i})$, and then $\forall i \in N$, $s_i = r_i(s_{-i})$.

$\Leftarrow$ $\forall i \in N$, let $s_i = r_i(s_{-i})$. So, by definition, $\forall s'_i \neq s_i \in 2^{\pi_i}$, $(s_i, s_{-i}) \succ_i (s'_i, s_{-i})$. Then, we obviously have $\forall i \in N$, $\forall s'_i \in 2^{\pi_i}$, $(s_i, s_{-i}) \succeq_i (s'_i, s_{-i})$: s is a SPNE.

$\blacksquare$

Proposition 11 *Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game such that the graphs $\mathcal{G}_i$ are all acyclic. Deciding whether a given strategy profile s is a SPNE of G is polynomial.*

Proof: From Lemma 3, we know that s is a SPNE if and only if for all player i, s_i is a best response for her. Moreover, Lemma 2 shows than we can efficiently verify for each i if s_i is a best response. Then, deciding if s is a SPNE is polynomial.

$\blacksquare$

Proposition 12
In a CP-Boolean game where the graph associated to each player's CP-net is acyclic, SPNE and WPNE coincide.

Proof: It is obvious that $NE_s \subseteq NE_w$. Let us verify than $NE_w \subseteq NE_s$.
Assume that s is a WPNE but is not a SPNE. Then, $\exists i \in N$ such that $s_i \neq r_i(s_{-i})$. But $(r_i(s_{-i}), s_{-i}) \succ_i (s'_i, s_{-i})$ for all s'_i, in particular for $s'_i = s_i$; therefore s cannot be a WPNE.

$\blacksquare$

Proposition 13
Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game such that the graphs $\mathcal{G}_i$ are all identical ($\forall i, j$, $\mathcal{G}_i = \mathcal{G}_j$) and acyclic. Then G has one and only one PNE.

This proof is inspired of the *forward sweep* procedure [8, 6] for outcome optimization.
Proof:

- **Existence** :

    Let $(x_1, \ldots, x_k)$ be a topological order on V with respect to $\mathcal{G}_i$ (for any i since all graphs are identical). We define a PNE $s = (s_1, \ldots, s_n)$ for G as follows.

    Let a be the agent controlling variable x_1. By construction we have $Pa(x_1) = \varnothing$, and either $x_1 \succ_a \overline{x_1}$ or $\overline{x_1} \succ_a x_1$. Define strategy s_a as assigning the best value to x_1 (i.e., x_1 in the first case and $\overline{x_1}$ in the second case). This process

45

can be iterated along the topological order, since by definition, at any time the value of each parent of the current variable x_i will have been fixed (by the same player's strategy or by another one's). We thus end up with a strategy profile s.

We now show that s is a PNE. Indeed, assume it is not one, and let an agent i and a strategy s_i' such that $(s_i', s_{-i}) \succ_i (s_i, s_{-i})$. Let x be the first variable in π_i, with respect to the topological order above, such that $s_i[x] \neq s_i'[x]$. Then by construction of s_i we have $s_i[x] \succ_{i,u_x} s_i'[x]$, where u_x is the instantiation of all parents of x. Now this instantiation is the same in (s_i, s_{-i}) and (s_i', s_{-i}), because we have chosen the first variable over which s_i' and s_i disagree w.r.t. the topological order. It follows that replacing $s_i'[x]$ with $s_i[x]$ in s_i' yields a better strategy, and iterating this reasoning we end up with showing that replacing every $s_i'[x]$ with $s_i[x]$ in s_i' yield ever better strategies. Since this process ends up with s_i, we finally have $(s_i, s_{-i}) \succ_i (s_i', s_{-i})$, which together with the assumption $(s_i', s_{-i}) \succ_i (s_i, s_{-i})$ contradicts the fact that acyclic CP-nets are always satisfiable.

- **Uniqueness** :
  Uniqueness derives easily from the construction above. Indeed, assume G has two different Nash equilibria, and let $(x_1, \ldots, x_k)$ be a topological order on V with respect to any G_i. Since once this order is fixed, the construction above is deterministic (resulting in a "canonical" PNE s_c), there is at least one Nash equilibrium s for G which is not built according to it. Let x be the first variable (w.r.t. to the topological order) such that s and s_c differ, and let i be the agent controlling x. Let u be the instantiation of all previous variables w.r.t. the order. By construction, u is common to s and s_c and instanciates all variables in $Pa(x)$. Assume $\bar{x} \succ_{i,u} x$ (the dual case is similar). Then by construction, s_c assigns $\bar{x}$ to x and thus, s assigns x to it. It follows that replacing x with $\bar{x}$ in s yields a (strictly) better strategy, contradicting the assumption that s is a PNE for G.

■

Lemma 4
Let G be a CP-Boolean game and G^* its equivalent normalized game. Then G^* and G define the same preference relations on strategy profiles.

Proof: For each player $i \in N$, let $\succeq_i$ be the preference relation satisfying i's CP-net for the game G. We show that $\succeq_i$ satisfies i's CP-net for G^*.
Two cases are possible $\forall x \in V$:

1. x has the same parents in CP-nets of G^* and G. In this case, $\succeq_i^x$ is the same in G and in G^* (idem for $\succeq_i^{\bar{x}}$).

2. x has not the same parents in the CP-net of G^* as in the CP-net of G. An edge has been added, denoted by $\langle y, x \rangle$. We have thus $\succeq_{i,\bar{y}}^x = \succeq_{i,y}^x = \succeq_i^x$. Thus this is the same relation.

$\blacksquare$

Proposition 14
Let $G = (N, V, \pi, \Phi)$ be a CP-Boolean game such that all graphs are acyclic and for every player $i \in N$, for every $v \in \pi_i$, we have $Pa(v) \in \pi_i$. Then G has one and only one PNE.

The proof of this proposition is also given by the use of the forward sweep procedure, and follows the same scheme as that of Proposition 13.

Proposition 15
Let G be a CP-Boolean game such that the graphs G_i are all acyclic, and $\mathcal{N}^+$ be the global CP-net of G. Let s be a strategy profile. If $\mathcal{N}^+$ is an acyclic CP-net, we have the following equivalence:
s is a PNE of G if and only if s is a best outcome of $\mathcal{N}^+$.

To prove this proposition, we will need the following lemma :

Lemma 5
Let G be a CP-Boolean game, $\mathcal{N}$ be an acyclic CP-net of G, and s be a strategy profile.
If $\exists s': s' \succ s$, then $\exists s'': s'' \succ s$ such that s and s'' differ from only one variable.

Proof:
Let s and s' be two strategy profiles such that $s' \succ s$. We suppose that there is no $s'' \succ s$ such that s and s'' differ from only one variable: in this case, indifference being impossible, for all such s'' we have $s \succ s''$.

Consider $s = v_1v_2 \ldots v_k \bar{v}_{k+1} \ldots \bar{v}_p$, with $v_1, \ldots, v_p \in V$, and $\forall v_j \in V$, let u_j denote the assignment to the parents of v_j. Then, we consider all possible s'':

$$s \succ \bar{v}_1 v_2 \ldots v_k \bar{v}_{k+1} \ldots \bar{v}_p \qquad \Rightarrow v_1 \succ_{u_1} \bar{v}_1$$
$$s \succ v_1 \bar{v}_2 \ldots v_k \bar{v}_{k+1} \ldots \bar{v}_p \qquad \Rightarrow v_2 \succ_{u_2} \bar{v}_2$$
$$\vdots$$
$$s \succ v_1 v_2 \ldots \bar{v}_k \bar{v}_{k+1} \ldots \bar{v}_p \qquad \Rightarrow v_k \succ_{u_k} \bar{v}_k$$
$$s \succ v_1 v_2 \ldots v_k v_{k+1} \ldots \bar{v}_p \quad \Rightarrow \bar{v}_{k+1} \succ_{u_{k+1}} v_{k+1}$$
$$\vdots$$
$$s \succ v_1 v_2 \ldots v_k \bar{v}_{k+1} \ldots v_p \qquad \Rightarrow \bar{v}_p \succ_{u_p} v_p$$

Thus each variable is instanciated to its best value according to its parents' instantiation. This assignment corresponds to the "forward sweep" procedure, s is thus the best outcome of $\mathcal{N}$ according to Lemma 1, which is in contradiction with the assumption $\exists s'$ such that $s' \succ s$. So $\exists s'' : s'' \succ s$ such that s and s'' differ from only one variable.

∎

We can now prove Proposition 16.

Proof:

$\Rightarrow$ Let s be a PNE of G. Let us suppose that s is not the best outcome of $\mathcal{N}^+$.

>From Definition 17, $\exists s' \in S$ such that $\mathcal{N}^+ \not\models s \succ^+ s'$.
So, there exists a preference relation such that $s' \succ^+ s$ which satisfies $\mathcal{N}^+$. From Lemma 5, we know that $\exists s'' \succ^+ s$ such that s and s'' differ from only one variable: $\exists! v \in V$ such that $v_{s''} \succ_u^+ v_s$, where u is the assignment in s and s'' of $Pa(v)$, all others being identical. Let i be the player such that $v \in \pi_i$. Then, we have $(s_i'', s_{-i}) \succ^+ (s_i, s_{-i})$. Let us now show what happens for G.

$s'' \succ^+ s$ satisfies $CPT^+(v)$ for all $v \in V$. By construction of $\mathcal{N}^+$ from G, $\forall i \in N, \forall v \in \pi_i, CPT_i(v) = CPT^+(v)$.

Thus for i in G, we have $v_{s''} \succ_i v_s$, and $\forall w \in V \setminus \{v\}, \forall j \in N$ such that $w \in \pi_j$, $w_{s''} =_j w_s$. Then: $(s_i'', s_{-i}) \succ_i (s_i, s_{-i})$.

Thus, s is not a PNE, contradiction.

$\Leftarrow$ Let s be the best outcome of $\mathcal{N}^+$. Let us suppose that s is not a PNE. $\exists i \in \{1, \ldots, n\}, \exists s_i' \in 2^{\pi_i}$ such that $(s_i', s_{-i}) \succ_i (s_i, s_{-i})$. Thus, $\exists W \subseteq \pi_i$ such that $\forall w \in W, \forall p \in 2^{Pa(w)}, w_{s'} \succ_{i,p} w_s$, and $\forall x \in \pi_i \setminus W, \forall q \in 2^{Pa(x)}, x_{s'} =_{i,q} x_s$.

By construction of $\mathcal{N}^+$ from G, $\forall i \in N$, $\forall v \in \pi_i$, $CPT_i(v) = CPT^+(v)$. We always have $\forall w \in W$, $\forall p \in 2^{Pa(w)}$, $w_{s'} \succ_p^+ w_s$, and $\forall x \in \pi_i \setminus W$, $\forall q \in 2^{Pa(x)}$, $x_{s'} =_q^+ x_s$. Then $(s_i', s_{-i}) \succ^+ (s_i, s_{-i})$. So $s' \succ^+ s$, and $\mathcal{N}^+ \not\models s \succ^+ s'$.

Thus s is not a best outcome of $\mathcal{N}^+$. Contradiction.

$\blacksquare$

Proposition 16

Deciding whether an acyclic CP-Boolean game $G = (N, V, \pi, \Phi)$ (i.e., where each CP-net $\mathcal{N}_i \in \Phi$ is acyclic) has a PNE is NP-*complete. Hardness holds even if the number of players is restricted to* 2.

Proof: Thanks to Proposition 11, we know that the problem is in NP. As for hardness, we give a reduction from the satisfiability problem for CNF formulas in which every clause contains at most three literals and each variable occurs in at most three clauses [17, comments for problem LO1].

The intuition behind the reduction is the following. Given a CNF formula φ as above, a player 1 will try to satisfy φ by assigning values to its variables; more precisely, he will be happy only if all clauses are satisfied, and otherwise he will always prefer flipping the value of at least one variable in the current assignment. Another player 2 will tag each variable x_i with a literal $(\neg) p_i$ telling whether x_i is positive or not in 1's strategy.

The precise construction is as follows. Given a CNF formula $\varphi = \bigwedge_{i=1}^k C_i$ in which every clause contains at most three literals and each variable occurs in at most three clauses, let $1, 2$ be two players. Let $\{x_1, \ldots, x_n\}$ be the set of variables of φ. For each clause C_i of φ let γ_i be a new variable (intuitively meaning that C_i is satisfied). Finally, for each variable x_i occurring in φ, let p_i be a new variable. We build a CP-Boolean game $G = (N, V, \pi, \Phi)$ where $N = \{1, 2\}$, $V = \{x_1, \ldots, x_n\} \cup \{\gamma_1, \ldots, \gamma_k\} \cup \{p_1, \ldots, p_n\}$, $\pi_1 = \{x_1, \ldots, x_n\} \cup \{\gamma_1, \ldots, \gamma_k\}$, and $\pi_2 = \{p_1, \ldots, p_n\}$; the CP-net of each player is defined below.

For each variable x_i of φ, if x_i occurs (negated or not) in three clauses $C_{i_1}, C_{i_2}, C_{i_3}$ then the following CP-table is added to 1's CP-net:

$\gamma_{i_1}\gamma_{i_2}\gamma_{i_3}p_i$	$x_i \succ \bar{x}_i$
$\gamma_{i_1}\gamma_{i_2}\gamma_{i_3}\bar{p}_i$	$\bar{x}_i \succ x_i$
$\bar{\gamma}_{i_1}\gamma_{i_2}\gamma_{i_3}p_i$	$\bar{x}_i \succ x_i$
$\bar{\gamma}_{i_1}\gamma_{i_2}\gamma_{i_3}\bar{p}_i$	$x_i \succ \bar{x}_i$
$\gamma_{i_1}\bar{\gamma}_{i_2}\gamma_{i_3}p_i$	$\bar{x}_i \succ x_i$
$\gamma_{i_1}\bar{\gamma}_{i_2}\gamma_{i_3}\bar{p}_i$	$x_i \succ \bar{x}_i$
$\cdots$	
$\bar{\gamma}_{i_1}\bar{\gamma}_{i_2}\bar{\gamma}_{i_3}p_i$	$\bar{x}_i \succ x_i$
$\bar{\gamma}_{i_1}\bar{\gamma}_{i_2}\bar{\gamma}_{i_3}\bar{p}_i$	$x_i \succ \bar{x}_i$

If x_i appears in only one or two clauses, a similar CP-table is added (with only one or two γ-variables appearing in the table). In other words, 1 prefers to change the value of x_i (w.r.t. what 2 reports about that value) as soon as at least one clause containing x_i is not satisfied.

Now for each clause C_i in φ, say, $C_i = (x_1 \vee \neg x_2 \vee \neg x_3)$ for the example, the following CP-table is added:

$p_1\bar{p}_2\bar{p}_3$	$\gamma_i \succ \bar{\gamma}_i$
$p_1 p_2\bar{p}_3$	$\gamma_i \succ \bar{\gamma}_i$
$\cdots$	
$\bar{p}_1 p_2\bar{p}_3$	$\gamma_i \succ \bar{\gamma}_i$
$\bar{p}_1 p_2 p_3$	$\bar{\gamma}_i \succ \gamma_i$

In other words, 1 prefers γ_i as soon as C_i is satisfied (by his strategy, as reported by 2) and $\bar{\gamma}_i$ otherwise. It follows that

(1) in any PNE of G, γ_i is true if and only if the clause C_i is satisfied

(otherwise 1 would have a better strategy, since 1 controls γ_i).

1's CP-net is then defined to be the set of all these tables together with any preference on the values of p_i's (with no parents). The construction is unambiguous since only one table is built per x_i or γ_i, and the graph is acyclic since the only dependencies are from p_j's to γ_i's and from γ_k's and p_j's to x_i's.

Finally, 2's CP-net is built as follows. For each variable x_i, the following CP-table is added:

x_i	$p_i \succ \bar{p}_i$
$\bar{x}_i$	$\bar{p}_i \succ p_i$

Obviously, 2's CP-net is acyclic, and

(2) in any PNE of G, for all i, p_i is true if and only if x_i is true

(otherwise 2 would have a better strategy, since 2 controls p_i).

Now, we have the following:

(3) in any PNE of G, if for some i C_i is false then there is a j such that $x_j \equiv \neg p_j$.

Indeed, suppose C_i is false. Then, by (1), γ_i is false. Let now x_j be one of the variables of C_i. The CP-table of 1 attached to x_j contains two conditional preference statements of the form $\overline{\gamma_i}(...)p_j : \overline{x_j} \succ x_j$ and $\overline{\gamma_i}(...)\overline{p_j} : x_j \succ \overline{x_j}$. Because γ_i is false and 1 controls x_j, in a PNE of G, x_j will have the opposite value of that of p_j.

>From (2) and (3), we get that in a PNE of G, each clause C_i is satisfied. Therefore, if G has a PNE then φ is satisfiable.

Conversely, assume φ is satisfiable and let $\vec{x} = (x_1,\ldots,x_n) \models \varphi$. Consider the strategy profile for G defined by assigning each x_i as in $\vec{x}$, each p_i as x_i, and each γ_i to true. It is easily checked that this strategy profile is a PNE for G. Therefore, if φ is satisfiable then G has a PNE.

∎

Observe that cyclicity in the union graph of the CP-game built in the proof allows 1 to get information on its own variables via 2, and hence to decide to play, e.g., x_3 ($\overline{x_3}$) if x_3 is currently false (true) and a clause containing it is not satisfied; thus her decisions about x_3 are partly based on the value of x_3 itself.