# Dependencies between players in Boolean games

Elise Bonzon, Marie-Christine Lagasquie-Schiex, and Jérôme Lang

IRIT, UPS, F-31062 Toulouse Cédex 9, France
{bonzon,lagasq,lang}@irit.fr

**Abstract.** Boolean games are a logical setting for representing static games in a succinct way, taking advantage of the expressive power and conciseness of propositional logic. A Boolean game consists of a set of players, each of them controls a set of propositional variables and has a specific goal expressed by a propositional formula. There is a lot of graphical structures hidden in a Boolean game: the satisfaction of each player's goal depends on players whose actions have an influence on these goals. Even if these dependencies are not specific to Boolean games, in this particular setting they give a way of finding simple characterizations of Nash equilibria and computing them.

## 1 Introduction

The framework of Boolean games [1–4] allows for expressing compactly static games with binary preferences: each player of a Boolean game controls a set of propositional variables, and a player's preferences is expressed by a plain propositional formula.[1] Thus, a player in a Boolean game has a dichotomous preference relation: either her goal is satisfied or it is not. This restriction may appear at first glance unreasonable. However, many concrete situations can be modelled as games where agents have dichotomous preferences. Furthermore, Boolean games can easily be extended to allow for non-dichotomous preferences, represented in some compact language for preference representation (see [5]).

Using the syntactical nature of goals, we can represent graphically the dependencies between players: if the goal (and hence the satisfaction) of a player $i$ depends on some variables controlled by a player $j$, then $i$ may need some action of $j$ to see her goal satisfied. This dependency betweeen players is a central notion in *graphical games* [6, 7] as well as in [8] – see Section 6. Representing these dependencies on a graph will allow us to compute pure-strategy Nash equilibria in a much simpler way, without enumerating all combinations of strategies.

After recalling some background on Boolean games in Section 2, we introduce in Section 3 the notion of dependency graph between players in Boolean games. In Section 4 we show how this graph can be exploited so as to find simple characterizations Nash equilibria in Boolean games, and we generalize some of these results for non-dichotomous preferences in Section 5. Related work and further issues are discussed in Section 6.

[1] We refer here to the version of Boolean games defined in [4], which generalizes the initial proposal [1].

## 2 $n$-players Boolean games

For any finite set $V = \{a, b, \ldots\}$ of propositional variables, $L_V$ denotes the propositional language built up from $V$, the Boolean constants $\top$ and $\bot$, and the usual connectives. Formulas of $L_V$ are denoted by $\varphi, \psi$ etc. A *literal* is a variable $x$ of $V$ or the negation of a variable. A *term* is a consistent conjunction of literals. A *clause* is a disjunction of literals. If $\alpha$ is a term, then $Lit(\alpha)$ is the set of literals appearing in $\alpha$. If $\varphi \in L_V$, then $Var(\varphi)$ denotes the set of propositional variables appearing in $\varphi$.
$2^V$ is the set of the interpretations for $V$, with the usual convention that for $M \in 2^V$ and $x \in V$, $M$ gives the value *true* to $x$ if $x \in M$ and *false* otherwise. $\models$ denotes the consequence relation of classical propositional logic.
Let $V' \subseteq V$. A $V'$-interpretation[2], also said partial interpretation, is a truth assignement to each variable of $V'$, that is, an element of $2^{V'}$. $V'$- interpretations are denoted by listing all variables of $V'$, with a $^-$ symbol when the variable is set to false: for instance, let $V' = \{a, b, d\}$, then the $V'$-interpretation $M = \{a, d\}$ assigning $a$ and $d$ to true and $b$ to false is denoted by $a\overline{b}d$. $\subseteq X$, then
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 $M_1 \cup \ldots \cup M_p$.
Finally, we denote the partial instantiation of a formula $\varphi$ by an $X$-interpretation $M_X$ by: $(\varphi)_{M_X} = \varphi_{v \in M_X \leftarrow \top, v \in X \setminus M_X \leftarrow \bot}$.

Given a set of propositional variables $V$, a Boolean game on $V$ is a $n$-players game[3], where the actions available to each player consist in assigning a truth value to each variable in a given subset of $V$. The preferences of each player $i$ are represented by a propositional formula $\varphi_i$ formed upon the variables in $V$.

**Definition 1** *A $n$-player Boolean game is a 4-tuple $(N, V, \pi, \Phi)$, where $N = \{1, 2, \ldots, n\}$ is a finite set of players (also called agents); $V$ is a finite set of propositional variables; $\pi : N \mapsto 2^V$ is a control assignment function; $\Phi = \{\varphi_1, \ldots, \varphi_n\}$ is a set of goals, where each $\varphi_i$ is a satisfiable formula of $L_V$.*

The control assignment function $\pi$ maps each player to the variables she controls. For the ease of notation, the set of all the variables controlled by $i$ is written $\pi_i$ instead of $\pi(i)$. Each variable is controlled by one and only one agent, that is, $\{\pi_1, \ldots, \pi_n\}$ forms a partition of $V$.

**Definition 2** *Let $G = (N, V, \pi, \Phi)$ be a Boolean game. A* **strategy** *for player $i$ in $G$ is a $\pi_i$-interpretation. The set of strategies for player $i$ in $G$ is $S_i = 2^{\pi_i}$ A* **strategy profile** *$s$ for $G$ is a $n$-uple $s = (s_1, s_2, \ldots, s_n)$ where for all $i$, $s_i \in S_i$. $S = S_1 \times \ldots \times S_n$ is the set of all strategy profiles.*

Note that since $\{\pi_1, \ldots, \pi_n\}$ forms a partition of $V$, a strategy profile $s$ is an interpretation for $V$, i.e., $s \in 2^V$. The following notations are usual in game theory. Let

[2] Note that a $V$-interpretation is an interpretation.

[3] In the original proposal [1], Boolean games are two-players zero-sum games. However the model can easily be generalized to $n$ players and non necessarily zero-sum games [4].

$s = (s_1, \ldots, s_n)$ be a strategy profile. For any non empty set of players $I \subseteq N$, the projection of $s$ on $I$ is defined by $s_I = (s_i)_{i \in I}$ and $s_{-I} = s_{N \setminus I}$. If $I = \{i\}$, we denote the projection of $s$ on $\{i\}$ by $s_i$ instead of $s_{\{i\}}$; similarly, we note $s_{-i}$ instead of $s_{-\{i\}}$. $\pi_I$ denotes the set of the variables controlled by $I$, and $\pi_{-I} = \pi_{N \setminus I}$. The set of strategies for $I \subseteq N$ is $S_I = \times_{i \in I} S_i$. If $s$ and $s'$ are two strategy profiles, $(s_{-I}, s'_I)$ denotes the strategy profile obtained from $s$ by replacing $s_i$ with $s'_i$ for all $i \in I$.
The goal $\varphi_i$ of player $i$ is a compact representation of a dichotomous preference relation, or equivalently, of a binary utility function $u_i : S \to \{0,1\}$ defined by $u_i(s) = 0$ if $s \models \neg\varphi_i$ and $u_i(s) = 1$ if $s \models \varphi_i$. $s$ is at least as good as $s'$ for $i$, denoted by $s \succeq_i s'$, if $u_i(s) \geq u_i(s')$, or equivalently, if $s \models \neg\varphi_i$ implies $s' \models \neg\varphi_i$; $s$ is strictly better than $s'$ for $i$, denoted by $s \succ_i s'$, if $u_i(s) > u_i(s')$, or, equivalently, $s \models \varphi_i$ and $s' \models \neg\varphi_i$.
This choice of binary utilities implies a loss of generality, even if some interesting problems have naturally dichotomous preferences. We relax this assumption in Section 5, where we consider generalized Boolean games with nondichotomous preferences expressed in some logical language for compact preference representation, as in [5].

## 3 Dependencies between players

We now focus on the syntactical nature of goals, which may help us identifying some game-theoretical notions, as pure-strategy Nash equilibria. Intuitively, if the goal $\varphi_i$ of player $i$ does not involve any variable controlled by player $j$, then the satisfaction of $i$ does not depend directly on $j$. This is only a sufficient condition: it may be the case that the syntactical expression of $\varphi_i$ mentions a variable controlled by $j$, but that this variable plays no role whatsoever in the satisfaction of $\varphi_i$, as variable $y$ in $\varphi_i = x \wedge (y \vee \neg y)$. We therefore use a stronger notion of formula-variable independence [9].

**Definition 3** *A propositional formula $\varphi$ is* **independent from** *a propositional variable $x$ if there exists a formula $\psi$ logically equivalent to $\varphi$ and in which $x$ does not appear.*[4]

**Definition 4** *Let $G = (N, V, \pi, \Phi)$ be a Boolean game. The set of* **relevant variables** *for a player $i$, denoted by $RV_G(i)$, is the set of all variables $v \in V$ such that $\varphi_i$ is not independent from $v$.*

For the sake of notation, the set of relevant variables for a given Boolean game $G$ will be denoted by $RV_i$ instead of $RV_G(i)$. We now easily define the *relevant players* for a given player $i$ as the set of players controlling at least one variable of $RV_i$.

**Definition 5** *Let $G = (N, V, \pi, \Phi)$ be a Boolean game. The set of* **relevant players** *for a player $i$, denoted by $RP_i$, is the set of agents $j \in N$ such that $j$ controls at least one relevant variable of $i$: $RP_i = \bigcup_{v \in RV_i} \pi^{-1}(v)$*[5].

[4] We have this equivalent semantical characterization of formula-variable independence [9]: $\varphi$ is independent from $x$ if there exists an interpretation $s$ such that $s \models \varphi$ and switch$(s,x) \models \varphi$, where switch$(s,x)$ is obtained by switching the value of $x$ in $s$, and leaving the values of other variables unchanged.

[5] Again, the set of relevant players for a Boolean game $G$ should be denoted by $RP_G(i)$: for the ease of notation we simply write $RP_i$.

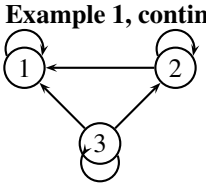
**Example 1** 3 *friends (*1, 2 *and* 3*) are invited at a party.* 1 *wants to go at this party.* 2 *wants to go at the party if and only if* 1 *goes, whereas* 3 *wants to go there, and prefers that* 2 *goes to, and* 1 *doesn't. This situation can be modelled by the following Boolean game* $G = (N, V, \pi, \Phi)$*, defined by* $V = \{a, b, c\}$*, with* $a$ *means "*1 *goes at the party", the same for* $b$ *and* 2*; and for* $c$ *and* 3*;* $N = \{1, 2, 3\}$*,* $\pi_1 = \{a\}$*,* $\pi_2 = \{b\}$*,* $\pi_3 = \{c\}$*,* $\varphi_1 = a$*,* $\varphi_2 = a \leftrightarrow b$ *and* $\varphi_3 = \neg a \wedge b \wedge c$*.*
*We can see that* 1*'s satisfaction depends only on herself, 2's depends on 1 and herself, whereas 3's depends on 1, 2 and herself. So, we have:* $RV_1 = \{a\}$*,* $RV_2 = \{a, b\}$*,* $RV_3 = \{a, b, c\}$*,* $RP_1 = \{1\}$*,* $RP_2 = \{1, 2\}$*,* $RP_3 = \{1, 2, 3\}$*.*

This relation between players can be seen as a directed graph containing a vertex for each player, and an edge from $i$ to $j$ whenever $j \in RP_i$, i.e. if $j$ is a relevant player of $i$.

**Definition 6** *Let* $G = (N, V, \pi, \Phi)$ *be a Boolean game. The* **dependency graph of a Boolean game** $G$ *is the directed graph (but not necessariyly acyclic)* $\mathcal{P} = \langle N, R \rangle$*, with* $\forall i, j \in N$*,* $(i, j) \in R$ *(denoted by* $R(i, j)$*) if* $j \in RP_i$*.*

Thus, $R(i)$ is the set of players from which $i$ may need some action in order to be satisfied: $j \in R(i)$ if and only if $j \in RP_i$. Remark however that $j \in R(i)$ does not imply that $i$ *needs* some action by $j$ to see her goal satisfied. For instance, if $\pi_1 = \{a\}, \pi_2 = \{b\}$ and $\varphi_1 = a \vee b$, then $1 \in R(2)$; however, 1 has a strategy for satisfying her goal (setting $a$ to true) and therefore does not have to rely on 2.
We denote by $R^*$ the transitive closure of $R$. $R^*(i, j)$ means that there exists a path from $i$ to $j$ in $R$. Then, $R^*(i)$ represents all players who have a direct or indirect influence on $i$. $R^{*-1}(i)$ represents all players on which $i$ has a direct or indirect influence.

**Example 1, continued:** *The dependence graph* $\mathcal{P}$ *induced by* $G$ *is depicted as follows:*




*We have* $R^{-1}(1) = \{1, 2, 3\}$*,* $R^{-1}(2) = \{2, 3\}$*,* $R^{-1}(3) = \{3\}$*.*
$R^*(1) = \{1\}$*,* $R^*(2) = \{1, 2\}$ *and* $R^*(3) = \{1, 2, 3\}$*.*
$R^{*-1}(1) = \{1, 2, 3\}$*,* $R^{*-1}(2) = \{2, 3\}$ *and* $R^{*-1}(3) = \{3\}$*.*

We easily obtain the following:

**Proposition 1** *Every dependency graph represents at least one Boolean game.*

We now introduce the notion of stable set. A stable set is a set $B$ of nodes such that all the edges from nodes in $B$ get to another node in $B$. The set of relevant players of a stable set $B$ are the players in $B$.

**Definition 7** *Let* $G = (N, V, \pi, \Phi)$ *be a Boolean game.* $B \subseteq N$ *is* **stable** *for* $R$ *if and only if* $R(B) \subseteq B$*, i.e.* $\forall j \in B$*,* $\forall i$ *such that* $i \in R(j)$*, then* $i \in B$*.*

Clearly, $\varnothing$ and $N$ are stable, and the set of stable sets for a Boolean game is closed under union and intersection. These four properties actually fully characterize the set of coalitions that correspond to the set of stable coalitions for a Boolean game (recall that a coalition is a subset of $N$).

**Proposition 2** *Let $\mathcal{C} \subset 2^N$. There exists a Boolean game $G$ such that $\mathcal{C}$ is the set of stable sets for $G$ if and only if $\mathcal{C}$ satisfies the following four properties: 1. $\varnothing \in \mathcal{C}$; 2. $N \in \mathcal{C}$; 3. If $B, B' \in \mathcal{C}$ then $B \cup B' \in \mathcal{C}$; 4. If $B, B' \in \mathcal{C}$ then $B \cap B' \in \mathcal{C}$.*

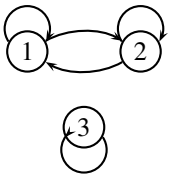
We now define the projection of a Boolean game $G$ on the set of players $B \subseteq N$:

**Definition 8** *Let $G = (N, V, \pi, \Phi)$ be a Boolean game, and $B \subseteq N$ a stable set for $R$. The* **projection** *of $G$ on $B$ is defined by $G_B = (B, V_B, \pi_B, \Phi_B)$, where $V_B = \cup_{i \in B} \pi_i$, $\pi_B : B \to V_B$ such that $\pi_B(i) = \{v | v \in \pi_i\}$, and $\Phi_B = \{\varphi_i | i \in B\}$.*

**Proposition 3** *If $B$ is a stable set, $G_B = (B, V_B, \pi_B, \Phi_B)$ is a Boolean game.*

*Proof:* Let $G_B = (B, V_B, \pi_B, \Phi_B)$. We have to check that every goal $\varphi_i$ for $i \in B$ is a formula of $L_{V_B}$, or can be rewritten equivalently as a formula of $L_{V_B}$. Suppose than $\exists i \in B$, $\exists v \in Var(\varphi_i)$ such that $v \notin V_B$. So, $\forall j \in B$, $v \notin \pi_j$. Let $k \in N \setminus B$ such that $v \in \pi_k$. We know that $v \in Var(\varphi_i)$, so either $\varphi_i$ is independent from $v$, and then is logically equivalent to a formula in which $v$ does not appear; or $\varphi_i$ is not independent from $v$, and in this case $v \in RV_i$ and by definition $k \in RP_i$. So, $k \in R(i)$, but $k \notin B$: this is in contradiction with the fact that $B$ is stable. ■

**Example 2** *Let $G = (N, V, \pi, \Phi)$ be the Boolean game defined by $V = \{a, b, c\}$, $N = \{1, 2, 3\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = a \leftrightarrow \neg b$ and $\varphi_3 = \neg c$.*
*We have: $RV_1 = \{a, b\}$, $RV_2 = \{a, b\}$, $RV_3 = \{c\}$, $RP_1 = \{1, 2\}$, $RP_2 = \{1, 2\}$, $RP_3 = \{3\}$.*
*The dependency graph $\mathcal{P}$ of $G$ follows. The sets of players $B = \{1, 2\}$ and $C = \{3\}$ are stable. We can decompose $G$ in 2 Boolean games:*




- $G_B = (B, V_B, \pi_B, \Phi_B)$, *with* $B = \{1, 2\}$, $V_B = \{a, b\}$, $\pi_1 = a$, $\pi_2 = b$, $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = a \leftrightarrow \neg b$.
- $G_C = (C, V_C, \pi_C, \Phi_C)$, *with* $C = \{3\}$, $V_C = \{c\}$, $\pi_3 = c$, $\varphi_3 = \neg c$.

## 4 Nash equilibria

Pure-strategy Nash equilibria (PNE) for $n$-players Boolean games are defined exactly as usual in game theory (see for instance [10]), having in mind that utility functions are induced from players' goals $\varphi_1, \ldots, \varphi_n$. A PNE is a strategy profile such that each player's strategy is an optimal response to the other players' strategies.

**Definition 9** *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 $\forall i \in \{1, \ldots, n\}$, $\forall s'_i \in S_i$, $u_i(s) \geq u_i(s_{-i}, s'_i)$.*

The following simple characterization of PNEs is straightforward from this definition ([4]): a strategy profile $s$ is a pure-strategy Nash equilibrium for $G$ iff for all $i \in N$, either $s \models \varphi_i$ or $s_{-i} \models \neg\varphi_i$ holds.

These definitions lead to some obvious properties of pure-strategy Nash equilibria:

**Proposition 4** *Let $G$ be a Boolean game. If $\forall i \in N$, $i \notin RP_i$ then every $s \in S$ is a PNE.*

If the irreflexive part of the players' dependency graph $\mathcal{P}$ of a game $G$ is acyclic, (i.e. if there is no cycle of length $\geq 2$), then we can use a procedure inspired by the "forward sweep procedure" [11] to find the pure-strategy Nash equilibria. Let us see this on an example.

**Example 1, continued:** *The irreflexive part of the dependency graph $\mathcal{P}$ of $G$ is acyclic. $RP_1 = \{1\}$, so a strategy profile $s = (s_1, s_2, s_3)$ is a PNE only if 1's goal is satisfied, i.e., $s_1 = \overline{a}$. Then, 2 can choose her strategy, because her goal depends only on her and on 1. Thus, $s$ is a PNE only if $(s_1, s_2) \models (\varphi_2)_{s_1}$, i.e., $s_2 = \overline{b}$. Finally, 3 knows the strategies of 1 and 2, and therefore she knows her goal will never be satisfied whatever she does. Therefore, $G$ has 2 PNEs: $\{\overline{a}\overline{b}c, \overline{a}\overline{b}\overline{c}\}$.*

**Proposition 5** *Let $G$ be a Boolean game such that the irreflexive part of the dependency graph $\mathcal{P}$ of $G$ is acyclic. Then, $G$ has a PNE. Moreover, $s$ is a PNE of $G$ if and only if for every $i \in N$, either $(s_{R^*(i)\setminus\{i\}}, s_i) \models \varphi_i$ or $s_{R^*(i)\setminus\{i\}} \models \neg\varphi_i$.*

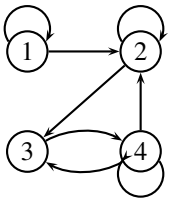
Obviously, when the irreflexive part of the dependency graph is not acyclic, the existence of PNE is no longer guaranteed (still, a game with a cyclic dependency graph may have a PNE, as shown in Example 3).

Proposition 5 leads to the following corollary:

**Corollary 1** *If $G$ is a Boolean game such that $\forall i \in N$, $RP_i = \{i\}$, then $s$ is a PNE if and only if $\forall i$, $s \models \varphi_i$.*
*If $G$ is a Boolean game such that $\forall i \in N$, $\exists j \in N$ such that $RP_i = \{j\}$, then $s$ is a PNE if and only if $s \models \varphi_j$.*

**Proposition 6** *Let $G = (N, V, \pi, \Phi)$ be a Boolean game, $B \subseteq N$ a stable set for $R$, and $G_B$ the projection of $G$ on $B$. If $s$ is a PNE for $G$, then $s_B$ is a PNE for $G_B$.*

**Example 3** *Let $G = (N, V, \pi, \Phi)$ be the Boolean game defined by $V = \{a, b, c, d\}$, $N = \{1, 2, 3, 4\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\pi_4 = \{d\}$, $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = b \leftrightarrow c$, $\varphi_3 = \neg d$, and $\varphi_4 = d \leftrightarrow (b \wedge c)$. We have: $RP_1 = \{1, 2\}$, $RP_2 = \{2, 3\}$, $RP_3 = \{4\}$, $RP_4 = \{2, 3, 4\}$. The dependency graph $\mathcal{P}$ of $G$ is the following:*

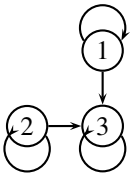



*The set of players $B = \{2, 3, 4\}$ is stable. $G_B = (B, V_B, \pi_B, \Phi_B)$ is a Boolean game, with $V_B = \{b, c, d\}$, $\pi_2 = b$, $\pi_3 = c$, $\pi_4 = d$, $\varphi_2 = b \leftrightarrow c$, $\varphi_3 = \neg d$, and $\varphi_4 = d \leftrightarrow (b \wedge c)$.*
*$G$ has 2 PNEs : $\{abcd, \overline{a}\overline{b}\overline{c}\overline{d}\}$, and $\{bcd, \overline{b}\overline{c}\overline{d}\}$ are 2 PNEs of $G_B$ (and in this case, $G_B$ has no other PNEs).*

As we can see on Example 2, the converse is not always true: $C = \{3\}$ is stable, and the Boolean game $G_C = (C, V_C, \pi_C, \Phi_C)$ has a PNE : $\{\overline{c}\}$, but the game $G$ has no PNE. However, there exist simple cases for which the converse is true.

**Proposition 7** *Let $B$ and $B'$ be two stable sets of players, and let $G_B$ and $G_{B'}$ be the two Boolean games associated. Suppose than $s_B$ is a PNE for $G_B$ and $s_{B'}$ is a PNE for $G_{B'}$ such that $\forall i \in B \cap B'$, $s_{B,i} = s_{B',i}$, where $s_{B,i}$ represents the strategy of player $i$ for the game $G_B$. Then, $s_{B\cup B'}$ is a PNE for $G_{B\cup B'}$.*

**Example 4** *Let $G = (N,V,\pi,\Phi)$ be the Boolean game defined by $V = \{a,b,c\}$, $N = \{1,2,3\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\varphi_1 = a \leftrightarrow c$, $\varphi_2 = b \leftrightarrow \neg c$, and $\varphi_3 = c$. We have: $RP_1 = \{1,3\}$, $RP_2 = \{2,3\}$, $RP_3 = \{3\}$. The dependency graph $\mathcal{P}$ of $G$ is drawn below. The sets of players $B = \{1,3\}$ and $C = \{2,3\}$ are stable. We have two new Boolean games.*




- *$G_B = (B,V_B,\pi_B,\Phi_B)$, with $B = \{1,3\}$, $V_B = \{a,c\}$, $\pi_1 = a$, $\pi_3 = c$, $\varphi_1 = a \leftrightarrow c$ and $\varphi_3 = c$. $G_B$ has one PNE : $\{ac\}$ (denoted by $s_B = (s_{B,1}, s_{B,3})$).*
- *$G_C = (C,V_C,\pi_C,\Phi_C)$, with $C = \{2,3\}$, $V_C = \{b,c\}$, $\pi_2 = b$, $\pi_3 = c$, $\varphi_2 = b \leftrightarrow \neg c$, $\varphi_3 = c$. $G_C$ has one PNE : $\{\overline{b}c\}$ (denoted by $s_C = (s_{C,2}, s_{C,3})$).*

*$B \cap C = \{3\}$. But we have $s_{B,3} = s_{C,3} = c$: $G_{B\cup C}$ has one PNE: $\{a\overline{b}c\}$.*

We can easily generalize Proposition 7, with $p$ stable sets covering the set of players:

**Proposition 8** *Let $G = (N,V,\pi,\Phi)$ be a Boolean game, and let $B_1 \ldots B_p$ be $p$ stable sets of players, such that $B_1 \cup \ldots \cup B_p = N$. Let $G_{B_1}, \ldots, G_{B_p}$ be the $p$ Boolean games associated. If $\exists s_{B_1} \ldots s_{B_p}$ PNEs of $G_{B_1}, \ldots, G_{B_p}$ such that $\forall i,j \in \{1,\ldots p\}$, $\forall k \in B_i \cap B_j$, $s_{B_i,k} = s_{B_j,k}$, then $s = (s_{B_1}, \ldots, s_{B_p})$ is a PNE of $G$.*

As shown in Example 4, splitting a Boolean game makes the computation of Nash equilibria easier. If we try to compute Nash equilibria in the original game, we have to check if either $s \models \varphi_i$ or $s_{-i} \models \neg\varphi_i$ for each of the 8 strategy profiles $s$ and for each of the 3 players $i$. So, we have to make 12 verifications for each player (8 for each strategy profile in order to verify $s \models \varphi_i$, and 4 for each $s_{-i}$ to verify $s_{-i} \models \neg\varphi_i$), then 36 for the game in the worst case. Meanwhile, the computation of PNEs once the game is split is much easier: for $G_B$, from Proposition 5, we have to make 6 verifications for player 1 (4 to compute $(s_1,s_3) \models \varphi_1$, and 2 to compute $s_3 \models \neg\varphi_1$); and only 2 for player 3 (because $R^*(3) \setminus \{3\} = \varnothing$). So, we only have to do 8 verifications in the worst case to find the PNEs of $G_B$, and the same for $G_C$, which has an equivalent configuration. As we have to check if the instanciation of player 3's variables are the same for PNEs of the 2 games, we have to make 17 verifications to compute PNEs of the game $G$.

## 5 Generalization to non-dichotomous preferences

This choice of binary utilities (where agents can express only plain satisfaction or plain dissatisfaction, with no intermediate levels) is a loss of generality. We would like now to allow for associating an arbitrary preference relation on $S$ with each player. A *preference relation* $\succeq$ is a reflexive, transitive and complete binary relation on $S$. The strict preference $\succ$ associated with $\succeq$ is defined as usual by $s_1 \succ s_2$ if and only if $s_1 \succeq s_2$ and not $(s_2 \succeq s_1)$. Given a propositional language $L$ for compact preference representation, a $L$-Boolean game is defined a 4-uple $G = (N,V,\pi,\Phi)$, where $N = \{1,\ldots,n\}$, $V$, and $\pi$

are as before and $\Phi = \langle \Phi_1, \ldots, \Phi_n \rangle$, where for each $i$, $\Phi_i$ is a compact representation, in $L$, of the preference relation $\succeq_i$ of agent $i$ on $S$. We let $Pref_G = \langle \succeq_1, \ldots, \succeq_n \rangle$. Remark that if $L_P$ is the purely propositional preference representation language, where a (dichotomous) preference is represented by a propositional formula, then $L_P$-Boolean games are just standard Boolean games as defined in Section 2. See [5] for several families of $L$-Boolean games.

We now have to generalize the dependency graph between players from Boolean games to $L$-Boolean games, for an arbitrary language $L$. Recall that, in Section 3, a player $i$ was dependent on a player $j$ if her propositional goal $\varphi_i$ was dependent of one of the variables that $j$ controls. Therefore, what we have to start with is generalizing formula-variable dependence to a dependency notion between a preference relation (or a syntactical input in a compact representation language from which this preference relation can be induced) and a variable. Several definitions have been considered in [12], in the more general case where preference relations are partial preorders. In the specific case where preference relations are complete preorders, however, there seems to be only one suitable definition: a preference relation $\succeq$ depends on a propositional variable $x$ if there exists at least one state where the agent is not indifferent between this state and the state obtained by switching the value of $x$:

**Definition 10** *A preference relation $\succeq$ on $2^V$ depends on a propositional variable $x \in V$ if there exists a $s \in S$ such that $switch(s,x) \succ_i s$ or $switch(s,x) \prec_i s$.*

This definition extends naturally to inputs of preference representation languages: an input $\Phi$ of a preference representation language $L$ depends on $x$ if the preference relation $\succeq$ induced by $\Phi$ depends on $x$.

We are now in position of defining the dependency graph for a $L$-Boolean game:

**Definition 11** *Let $G = (N, V, \pi, \Phi)$ a $L$-Boolean game. The set of* **relevant variables** *for a player $i$, denoted by $RV_i$, is the set of all variables $v \in V$ such that $\Phi_i$ is dependent on $v$. The set of* **relevant players** *for a player $i$, denoted by $RP_i$, is the set of agents $j \in N$ such that $j$ controls at least one relevant variable of $i$: $RP_i = \bigcup_{v \in RV_i} \pi^{-1}(v)$*

The dependency graph of a $L$-Boolean game is defined exactly as in Section 3.

These definitions do not depend on the language chosen for compact preference representation. However, for the sake of illustration we give an example in which preferences are represented with prioritized goals (see [5]):

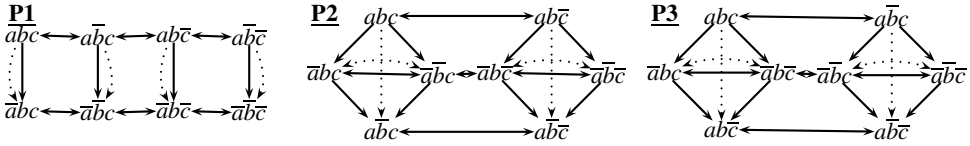
**Definition 12** *A* **prioritized goal base** *$\Sigma$ is a collection $\langle \Sigma^1; \ldots; \Sigma^p \rangle$ of sets of propositional formulas. $\Sigma^j$ represents the set of goals of priority $j$, with the convention that the smaller $j$, the more prioritary the formulas in $\Sigma^j$.*

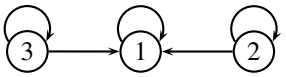
In this context, several criteria can be used in order to generate a preference relation $\succeq$ from $\Sigma$. We choose here to stick to the leximin criterion (see [13–15]). In the following, if $s$ is an interpretation of $2^V$ then we let $Sat(s, \Sigma^j) = \{\varphi \in \Sigma^j \mid s \models \varphi\}$.

**Definition 13** *Let $\Sigma = \langle \Sigma^1; \ldots; \Sigma^p \rangle$, and let $s$ and $s'$ be two interpretations of $2^V$. The leximin preference relation is defined by: $s \succ^{lex} s'$ iff $\exists k \in \{1, \ldots, p\}$ such that: $|Sat(s, \Sigma^k)| > |Sat(s', \Sigma^k)|$ and $\forall j < k$, $|Sat(s, \Sigma^j)| = |Sat(s', \Sigma^j)|$.*

Note that $\succeq^{lex}$ is a complete preference relation. Here is now an example within this preference representation language:

**Example 5** *$G = (N,V,\pi,\Phi)$ where $N = \{1,2,3\}$, $V = \{a,b,c\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\Sigma_1 = \langle a \rangle$, $\Sigma_2 = \langle (b \vee \neg a); a \rangle$ and $\Sigma_3 = \langle (c \vee \neg a); a \rangle$. We draw below the preference relations $Pref_G^{lex} = \langle \succeq_1^{lex}, \succeq_2^{lex}, \succeq_3^{lex} \rangle$[6].*



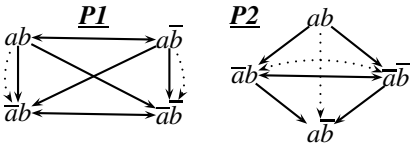

*We have: $RV_1 = \{a\}$, $RV_2 = \{a,b\}$, $RV_3 = \{a,c\}$, $RP_1 = \{1\}$, $RP_2 = \{1,2\}$, $RP_3 = \{1,3\}$.*

**Definition 14** *Let $G = (N,V,\pi,\Phi)$ be a L-Boolean game, and $B \subseteq N$ a stable set for $R$. The* **projection** *of $G$ on $B$ is defined by $G_B = (B,V_B,\pi_B,\Phi_B)$, where $V_B = \cup_{i \in B} \pi_i$, $\pi_B(i) = \{v | v \in \pi_i\}$, and $\Phi_B$ are the goals of players in $B$.*

We can now generalize some properties found previously to these non-dichotomous preferences. For example, Propositions 3, 5, 6, 7 and 8 can be easily generalized in this framework.

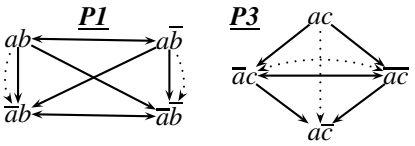
**Example 5, continued:** *The sets of players $B = \{1,2\}$ and $C = \{1,3\}$ are stable. We have two new Boolean games:*

*$G_B = (B,V_B,\pi_B,\Phi_B)$, with $B = \{1,2\}$, $V_B = \{a,b\}$, $\pi_1 = a$, $\pi_2 = b$, $\Sigma_1 = \langle a \rangle$, and $\Sigma_2 = \langle (b \vee \neg a); a \rangle$. The preference relations $Pref_G^{lex} = \langle \succeq_1^{lex}, \succeq_2^{lex} \rangle$ are drawn on the right.*




*$G_B$ has one PNE : $\{ab\}$ (denoted by $s_B = (s_{B,1}, s_{B,2})$).*

*$G_C = (C,V_C,\pi_C,\Phi_C)$, with $C = \{1,3\}$, $V_C = \{a,c\}$, $\pi_1 = a$, $\pi_3 = c$, $\Sigma_1 = \langle a \rangle$ and $\Sigma_3 = \langle (c \vee \neg a); a \rangle$. The preference relations $Pref_G^{lex} = \langle \succeq_1^{lex}, \succeq_3^{lex} \rangle$ are drawn on the right.*




*$G_C$ has one PNE : $\{ac\}$ (denoted by $s_C = (s_{C,1}, s_{C,3})$).*
*$B \cap C = \{1\}$. But we have $s_{B,1} = s_{C,1} = a$: $G_{B \cup C}$ has one PNE: $\{abc\}$.*

[6] Arrows are oriented from more preferred to less preferred strategy profiles ($s_1$ is preferred to $s_2$ is denoted by $s_1 \rightarrow s_2$). To make the figures clearer, we do not draw edges that are obtained from others by transitivity. The dotted arrows indicate the links taken into account in order to compute Nash equilibria. For example, player 2 prefers $abc$ to $\overline{a}bc$ because $|Sat(ab, \Sigma_2^1)| = 1$, $|Sat(ab, \Sigma_2^2)| = 1$ (both stratas of $\Sigma^2$ are satisfied), and $|Sat(\overline{a}b, \Sigma_2^1)| = 1$, $|Sat(ab, \Sigma_2^2)| = 0$ (only the first strata of $\Sigma_2$ is satisfied).

## 6 Conclusion

We have shown how the intuitive notion of dependency between players in a Boolean game can be exploited so as to give simpler characterizations of pure-strategy Nash equilibria. Moreover, our properties not only hold for the standard version of Boolean game (with propositional goals and dichotomous preferences) but also for generalized Boolean games, where players' preferences are expressed in a compact representation language. Another class of games with dichotomous preferences shares a lot with Boolean games: Qualitative Coalitional Games (QCG), introduced by [16]. In a QCG, each agent has a set of goals, and is satisfied if one of her goals is achieved, but is indifferent on which goal is, and on the number of goals achieved[7]. Thus agents have dichotomous preferences (as in the standard version of Boolean games - cf. Sections 2–4). A characteristic function associates with each agent, or set of agents, the set of goals they can achieve. The main difference between QCGs and BG is that characteristic functions in QCGs are not monotonic, whereas utility functions are in Boolean games. However, we can represent a QCG with monotonic characteristic function by a Boolean games.

Boolean games take place in a larger stream of work, that we may gather under the generic name of *compactly represented games*. All frameworks for compactly represented games make use of notions of dependencies between players and/or actions that have a lot in common with ours. Most of these frameworks, including [6, 7, 18], share the following mode of representation of players' utilities: the utility of a player $i$ is described by a table specifying a numerical value for each combination of values to each of the set of variables that are relevant to $i$[8]. The representation of games with such utility tables is called *graphical normal form* (GNF) in [8]. Dependency between players and variables in such games naturally induce a dependency relation between players, in the same way as we do ($i$ depends on $j$ if $i$'s utility table refers to a variable that is controlled by $j$).

Boolean games are very similar to these graphical games, except that the form chosen for expressing compactly players' preferences is *logical*. The logical form is sometimes exponentialy more compact than the graphical form: consider for instance the dichotomous preference relation corresponding to the goal $\varphi = x_1 \oplus \ldots \oplus x_p$, where $\oplus$ is exclusive or. While the logical representation of $u_\varphi$ is linear in $p$, its representation by utility tables would be exponential in $p$, since each of the $p$ variables is relevant to the utility of the player. In the general case of non-dichotomous utility functions or preference relations, the Boolean game framework, by allowing some flexibility on the choice of the language for preference representation, is more general than that of graphical games, where the format for expressing preferences is fixed. Moreover, solving games in logical form may benefit from the huge literature on SAT and related algorithms for propositional logic.

The notion of dependency between players and variables in graphical games is used for the very same purpose as our dependency graph, namely, to split up a game into a set

[7] In [17], QCGs are extended by allowing agents to have preferences over goals.

[8] In multi-agent influence diagrams [6], a players' utility is actually express in a more compact way as the sum of local utilities, each corresponding to a smaller set of variables.

of interacting smaller games, which can solved more or less independently. [8] study speficic restrictions on graphical games, either by bounding the size of players' neighbourhoods (the neighbourhood of a player $i$ in a graphical game is the set of players who potentially influence the utility of $i$), or by imposing that the dependency relation between players should be acyclic. They study the impact of such restrictions on the complexity of checking the existence of a Nash equilibrium (or their computation). Clearly, similar structural restrictions on Boolean games would probably allow for a complexity fall with respect to the complexity results for the general case in [4]. This is left for further study.

The work reported here is still preliminary and can be pursued in many other directions.

First, apart of the *structural* restrictions mentioned just above, we may study the impact of *syntactical* restrictions on propositional goals on the computation of Nash equilibria and on the construction of the dependency graph. In [19], Sichman and Conte introduced dependence graphs which can represent and/or dependencies[9] on actions needed to achieve an agent's goal and on the agents who control these actions. In the first case, this is similar to our set of relevant variables, and in the second case this corresponds to our set of relevant players. Sichman and Conte's ideas can be used for introducing and/or dependencies in our framework, but using the syntactical form of the goals. In [20], 3 notions of dependance are defined: the weak one is the same than our (an agent $i$ is dependent from a set of agents $C$ if $C$ can achieve $i$'s goal). The second one, normal dependence, adds to weak dependence the condition that $i$ cannot achieve her goal by herself. Finally, the third one adds the fact that agents in $C$ are the only ones able to achive $i$'s goal. Following [19], [20] use an and-graph to reprensent weak/strong dependence: for every coalition $C$, there is an and-edge from agent $i$, $i \in C$, to agent $j \in N$ if the agents in $C$ can achieve the goal desired by the agent $j$. This notion of dependence is the basis of their computation of admissible coalition under the do-ut-des criterion (see [21]).

Second, while our Section 5 does not focus on particular language (prioritized goals we used in an example just for the sake of illustration), we may want to study in further detail the computation of Nash equilibria (using the structural properties of the game) for some specific languages for preference representation (see [5] for the case of CP-nets and prioritized goals). A particularly appealing language is that of *weighted goals*, where a player's utility function is represented using several propositional formulas, each being attached with a numerical value (see [22]). This is especially interesting because this language generalizes the representation by utility tables in graphical games.

So far, Boolean games allow only for expressing *static* games (with simultaneous moves by the players) and with *complete information*. Enriching Boolean games with dynamicity and nature-driven uncertainty, as in multi-agent influence diagrams, is not as simple as it looks at first glance, and is a challenging issue. Computing *mixed strategy* Nash equilibria in Boolean games is another challenging issue.

[9] The or-dependence means that several actions allow an agent to achieve a in several ways, and the and-dependence means that this agent needs all these actions to achieve her goal.

**Acknowledgment**

We thank the anonymous reviewers for their useful comments and suggestions.

## References

1. Harrenstein, P., van der Hoek, W., Meyer, J., Witteveen, C.: Boolean Games. In: Proc. of TARK'01, Morgan Kaufmann (2001) 287–298
2. Harrenstein, P.: Logic in Conflict. PhD thesis, Utrecht University (2004)
3. Dunne, P., van der Hoek, W.: Representation and Complexity in Boolean Games. In: Proc. of JELIA2004. Volume LNCS 3229. (2004) 347–359
4. Bonzon, E., Lagasquie-Schiex, M., Lang, J., Zanuttini, B.: Boolean games revisited. In: Proc. of ECAI'06, Springer-Verlag (2006) 265–269
5. Bonzon, E., Lagasquie-Schiex, M., Lang, J.: Compact preference representation for Boolean games. In: Proc. of PRICAI'06. Volume 4099., Springer-Verlag (2006) 41–50
6. Koller, D., Milch, B.: Multi-Agent Influence Diagrams for Representing and Solving Games. Games and Economic Behavior **45**(1) (2003) 181–221
7. Kearns, M., Littman, M.L., Singh, S.: Graphical Models for Game Theory. In: Procs. of UAI'01. (2001)
8. Gottlob, G., Greco, G., Scarcello, F.: Pure Nash Equilibria: Hard and Easy Games. Journal of Artificial Intelligence Research **24** (2005) 357–406
9. Lang, J., Liberatore, P., Marquis, P.: Propositional Independence - Formula-Variable Independence and Forgetting. Journal of Artificial Intelligence Research **18** (2003) 391–443
10. Osborne, M., Rubinstein, A.: A course in game theory. MIT Press (1994)
11. Boutilier, C., Brafman, R., Hoos, H., Poole, D.: Reasoning with Conditional Ceteris Paribus Preference Statements. In: Proc. of UAI'99. (1999)
12. Besnard, P., Lang, J., Marquis, P.: Variable Forgetting in Preference Relations over Propositional Domains. In: Proc. of ECAI'06, Springer-Verlag (2006) 763–764
13. Dubois, D., Lang, J., Prade, H.: Inconsistency in possibilistic knowledge bases: To live with it or not live with it. In: Fuzzy Logic for the Management of Uncertainty. (1992) 335–351
14. Benferhat, S., Cayrol, C., Dubois, D., Lang, J., Prade, H.: Inconsistency management and prioritized syntax-based entailment. In: Proc. of IJCAI'93. (1993) 640–645
15. Lehmann, D.: Another Perspective on Default Reasoning. Annals of Mathematics and Artificial Intelligence **15** (1995) 61–82
16. Wooldridge, M., Dunne, P.: On the computational complexity of qualitative coalitional games. Artificial Intelligence (2004)
17. Dunne, P., Wooldridge, M.: Preferences in Qualitative Coalitional Games. In: Proc. of GTDT'04. (2004) 29–38
18. La Mura, P.: Game Networks. In: Procs. of UAI'00. (2000) 335–342
19. Sichman, J., Conte, R.: Multi-Agent Dependence by Dependence Graphs. In: Proc. of AAMAS'02. (2002)
20. Boella, G., Sauro, L., van der Torre, L.: From Social Power to Social Importance. Web Intelligence and Agent Systems Journal (2006) To appear.
21. Boella, G., Sauro, L., van der Torre, L.: Admissible agreements among goal-directed agents. In IEEE, ed.: Proc. of IAT'05. (2005)
22. Chevaleyre, Y., Endriss, U., Lang, J.: Expressive Power of Weighted Propositional Formulas for Cardinal Preference Modelling. In: Proc. of KR'06, AAAI Press (2006) 145–152