

THÈSE

présentée devant

l'Université Paul Sabatier de Toulouse III

U.F.R. MATHÉMATIQUES, INFORMATIQUE ET GESTION

pour obtenir le titre de

DOCTEUR DE L'UNIVERSITÉ PAUL SABATIER

Mention INFORMATIQUE

par

ELISE BONZON

École doctorale : Informatique et Télécommunication

Laboratoire d'accueil : Institut de Recherche en Informatique de Toulouse

Équipe d'accueil : Raisonnements Plausibles, Décision, Méthodes de Preuves

Modélisation des interactions entre agents rationnels : les jeux booléens

soutenue le 13 Novembre 2007 devant la commission d'examen :

Nicholas ASHER	Directeur de recherche, Université Paul Sabatier (membre)
Annie ASTIÉ-VIDAL	Professeur, Université Paul Sabatier (invité)
Olivier GASQUET	Professeur, Université Paul Sabatier (membre)
Andreas HERZIG	Directeur de recherche, Université Paul Sabatier (invité)
Marie-Christine LAGASQUIE-SCHIEX	Maître de conférences, Université Paul Sabatier (directrice de thèse)
Jérôme LANG	Chargé de recherche, Université Paul Sabatier (directeur de thèse)
Pierre MARQUIS	Professeur, Université d'Artois (rapporteur)
Nicolas MAUDET	Maître de conférences, Université Paris Dauphine (membre)
Leon van der TORRE	Professeur, Université du Luxembourg (rapporteur)
Bruno ZANUTTINI	Maître de conférences, Université de Caen (membre)

Elise Bonzon

**MODÉLISATION DES INTERACTIONS ENTRE AGENTS RATIONNELS :
LES JEUX BOOLÉENS**

Directeurs de thèse :
Jérôme Lang, CR CNRS, HdR
Marie-Christine Lagasque-Schiex, Maître de conférences
Université Paul Sabatier

Résumé

Les jeux booléens permettent de représenter les jeux stratégiques d'une manière succincte en tirant profit du pouvoir d'expression et de la concision de la logique propositionnelle. Informellement, un jeu booléen est un jeu à deux joueurs, chacun d'entre eux contrôlant un ensemble de variables propositionnelles, et à somme nulle. La fonction d'utilité du joueur 1 (et donc celle du joueur 2 qui est son opposé) est représentée par une formule de la logique propositionnelle, appelée *forme booléenne* du jeu. Ainsi, un joueur dans un jeu booléen a des préférences dichotomiques : son but est satisfait ou ne l'est pas.

Ces trois restrictions (deux joueurs, somme nulle, préférences binaires) limitent fortement l'expressivité de ce cadre. Les deux premières restrictions peuvent être facilement résolues en définissant les préférences des agents comme étant un n -uplet de formules propositionnelles (une pour chaque agent). Des outils simples issus de la logique propositionnelle nous permettent ainsi de caractériser certaines propriétés du jeu. Deux autres notions ont alors été étudiées : la dépendance entre joueurs (si le but, et donc la satisfaction, d'un joueur i dépend de variables contrôlées par le joueur j , alors i aura besoin de j pour satisfaire son but) et les coalitions de joueurs (une coalition dans un jeu booléen est efficace si elle peut garantir à tous ses membres que leurs buts sont satisfaits). Dans les deux cas, l'objectif est de faciliter le calcul des concepts de solution tels que les équilibres de Nash en stratégies pures.

Lever la troisième restriction consiste à exprimer des préférences (non binaires) dans un cadre propositionnel. Cela est possible en utilisant un langage de représentation compacte de préférences. Nous avons intégré donc deux de ces langages aux jeux booléens : tout d'abord, les *buts à priorité* puis les *CP-nets*.

Institut de Recherche en Informatique de Toulouse - UMR 5505
Université Paul Sabatier, 118 route de Narbonne. 31062 TOULOUSE cedex 9

Elise Bonzon

BOOLEAN GAMES

Supervisors :

Jérôme Lang, CR CNRS, HdR

Marie-Christine Lagasquie-Schiex, Maître de conférences

Université Paul Sabatier

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. Boolean games allow to express compactly two-players 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 three restrictions (two-players, zero-sum, binary preferences) strongly limit the expressivity of the framework. The first two can be easily encompassed by defining the agents' preferences as an arbitrary n -uple of propositional formulas. Two others notions have been studied : 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) and efficient coalitions (a coalition in a Boolean game is efficient if it has the power to guarantee that all goals of the members of the coalition are satisfied). We give simple characterizations of Nash equilibria and dominated strategies, and investigate the computational complexity of the related problems.

Then, we relax the last restriction by coupling Boolean games with propositional languages for compact preference representation ; we consider generalized Boolean games where players' preferences are expressed within the two following languages : propositionalized CP-nets, and then prioritized goals.

Institut de Recherche en Informatique de Toulouse - UMR 5505

Université Paul Sabatier, 118 route de Narbonne. 31062 TOULOUSE cedex 9

Remerciements

Je voudrais en premier lieu remercier Jérôme LANG et Marie-Christine LAGASQUIE-SCHIEX pour m'avoir aidée, accompagnée, encadrée et encouragée tout au long de ces trois années. Leur présence, leurs conseils et leur complémentarité m'ont permis non seulement de mener ce travail à bien, mais également de découvrir et d'aimer le monde de la recherche, comme celui de l'enseignement.

J'aimerais ensuite remercier Pierre MARQUIS et Leon van der TORRE pour avoir accepté d'être rapporteurs de ma thèse, et pour l'avoir lue avec autant d'attention. Merci pour leurs conseils, remarques et commentaires, qui m'ont permis d'améliorer ce manuscrit tant sur le fond que sur la forme.

Merci également à Nicholas ASHER, Annie ASTIÉ-VIDAL, Olivier GASQUET, Andreas HERZIG, Nicolas MAUDET et Bruno ZANUTTINI pour avoir accepté de participer à mon jury de thèse, et pour leurs commentaires qui ouvrent de nombreuses pistes de recherche. Je tiens particulièrement à remercier Pierre, Leon, Nicolas et Bruno qui ont accepté de se déplacer à Toulouse malgré les grèves.

Merci à Florence BOUÉ et Martine LABRUYÈRE qui m'ont permis de traverser sans encombres toutes les embûches administratives semées sur le chemin de la soutenance. Merci Florence d'avoir été aussi efficace pour gérer tous les problèmes de dernières minutes.

Merci à Ulle ENDRISS pour m'avoir accueillie 6 semaines au sein de l'ILLC à Amsterdam, pour m'avoir ainsi permis de découvrir un autre environnement de travail, et pour m'avoir consacré du temps et de l'énergie.

Merci à Marie-Christine LAGASQUIE-SCHIEX, Florence BANNAY, Martin STRECKER, et tous les moniteurs de la promo 2004 pour m'avoir accompagné dans les divers enseignements que j'ai accompli. Ils sont pour beaucoup dans le plaisir que j'ai aujourd'hui à enseigner.

Merci à Jorge CHAM qui m'a permis de découvrir de façon ludique les côtés obscurs du milieu académique.

Merci aux membres du troisième étage de l'Irit qui, de pauses café en pauses repas, m'ont appris que le milieu de la recherche peut être extrêmement convivial.

Merci à mes copines de bureau, Marie et Sihem, pour avoir grandement contribué à rendre l'ambiance de travail très agréable.

Merci à Nicolas, Axel, Kevin, Virginie, Alexia, Sylvain, Florian, Mylen, Filou, Caroline, Cédric, Marie, Julie, Amélie, Elodie, Manue, Xavier, Dominique, Marjolaine et tant d'autres pour m'avoir permis de ne pas oublier le monde extérieur pendant ces trois ans.

Merci à ma famille, mes parents, mes frères et sœurs, pour m'avoir soutenue tout au long de ces années, et d'avoir toujours été là pour moi.

Table des matières

Remerciements	v
Introduction	1
Notations	7
1 Éléments de théorie des jeux	9
1.1 Taxonomie partielle des jeux	10
1.1.1 Jeux statiques et dynamiques	10
1.1.1.1 Jeux statiques	10
1.1.1.2 Jeux dynamiques	11
1.1.2 Jeux coopératifs et non coopératifs	12
1.1.2.1 Jeux coopératifs	12
1.1.2.2 Jeux non coopératifs	14
1.1.3 Récapitulatif	16
1.2 Représentation des jeux	16
1.2.1 Forme extensive	16
1.2.2 Forme normale	17
1.3 Concepts de solution	18
1.3.1 Equilibres de Nash	19
1.3.1.1 Equilibre de Nash en stratégies pures	19
1.3.1.2 Equilibres de Nash en stratégies mixtes	20
1.3.2 Stratégies dominées	21
1.3.3 Cœur	25
1.3.3.1 Utilités transférables	25
1.3.3.2 Utilités non transférables	26
1.3.4 Equilibres parfaits de Selten	27

1.3.5	Récapitulatif	29
2	Jeux booléens	31
2.1	Introduction aux jeux booléens	32
2.2	Jeux booléens à n joueurs : définitions et exemples	33
2.3	Graphe de dépendance entre les joueurs	37
2.4	Concepts de solution : équilibres de Nash et stratégies dominées	41
2.4.1	Equilibres de Nash	41
2.4.2	Stratégies dominées	50
2.5	Cas particulier : Jeux à deux joueurs et à somme nulle	55
2.6	Jeux booléens et duopole de Stackelberg	60
2.6.1	2 joueurs, 1 variable chacun	60
2.6.2	2 joueurs, 3 variables	62
2.6.3	2 joueurs, 2 variables chacun	63
2.6.4	2 joueurs, n variables chacun	64
3	Jeux booléens et préférences non dichotomiques	69
3.1	Préférences ordinales et théorie des jeux	71
3.2	Jeux booléens et CP-nets	74
3.2.1	Ceteris Paribus	75
3.2.2	CP-nets	75
3.2.2.1	Définitions générales	75
3.2.2.2	Sémantique des CP-nets	76
3.2.2.3	Utilisation des CP-nets dans les jeux booléens	78
3.2.3	Propriétés des CP-jeux booléens	82
3.2.3.1	Stratégies dominées	82
3.2.3.2	Equilibres de Nash en stratégies pures	83
3.2.3.3	CP-jeux booléens avec graphe acyclique commun à tous les joueurs	85
3.2.3.4	CP-net global	91
3.2.3.5	Complexité	94
3.2.4	Introduction d'une relation d'indifférence	96
3.3	Buts à priorité	99
3.3.1	Etat de l'art	100
3.3.1.1	Ordre Discrimin	101
3.3.1.2	Ordre Leximin	103
3.3.1.3	Ordre Best-out	103

3.3.2	Utilisation des buts à priorité dans les jeux booléens	105
3.3.3	Quelques propriétés	110
3.4	Préférences non dichotomiques et graphe de dépendance	116
4	Jeux, logique propositionnelle et représentation compacte	121
4.1	Jeux et programmation logique	121
4.1.1	Programmes logiques	121
4.1.2	Programme logique de choix	122
4.1.3	Programme logique avec des disjonctions ordonnées	124
4.1.3.1	Présentation et définitions	124
4.1.3.2	LPOD et équilibres de Nash	125
4.2	Jeux et représentation graphique	127
4.2.1	CP-nets et équilibres de Nash	127
4.2.2	Forme normale graphique	128
4.2.2.1	Diagrammes d'influence multi-agents	129
4.2.2.2	Formes normales graphiques et restrictions	131
4.2.2.3	G nets	133
4.2.3	Symétries dans les fonctions d'utilité	133
4.2.3.1	Jeux de congestion	133
4.2.3.2	Jeux à effets locaux	134
4.2.3.3	Jeux graphiques d'actions	135
5	Coalitions efficaces dans les jeux booléens	137
5.1	Fonctions d'effectivité	137
5.2	Coalitions et fonctions d'effectivité dans les jeux booléens	138
5.3	Coalitions efficaces	143
5.3.1	Définition et caractérisation	143
5.3.2	Coalition efficace et noyau	149
5.3.3	Lien avec les graphes de dépendance	154
5.4	Travaux connexes	157
5.4.1	Graphe de dépendance et coalitions admissibles	157
5.4.1.1	Graphe de dépendance	157
5.4.1.2	Coalitions admissibles	158
5.4.2	Jeux qualitatifs coalitionnels	161
5.4.3	Logique des jeux coalitionnels	163

Conclusion	165
Bibliographie	173
Index	183
Liste des symboles	185

Introduction

Contexte

L'intelligence artificielle permet à des machines d'effectuer des tâches réputées intelligentes, en concevant et en réalisant des fonctions cognitives artificielles, et en utilisant la puissance calculatoire des ordinateurs. Parmi ces tâches se trouvent, entre autres, l'acquisition et la représentation des connaissances, la formalisation et la mécanisation de différents types de raisonnement, l'aide à la décision collective, la planification, les systèmes multi-agents, etc.

Objet de nombreuses recherches en intelligence artificielle, les systèmes multi-agents s'inspirent des sciences humaines pour modéliser des groupes d'agents. Un système multi-agent est un ensemble d'agents situés dans un certain environnement et interagissant selon une certaine organisation. Un agent rationnel est une entité caractérisée par le fait qu'elle est, au moins partiellement, autonome. Ce peut être un processus, un robot, un être humain, etc. La création de systèmes multi-agents conduit à cinq problématiques principales :

- * Problématique de l'action : comment plusieurs agents peuvent-ils agir de manière simultanée dans un environnement commun, et comment cet environnement interagit en retour avec les agents ? Les questions sous-jacentes à celle-ci sont entre autres celles de la représentation de l'environnement par les agents, de la collaboration entre agents et de la planification multi-agents.
- * Problématique de l'agent et de sa relation au monde : de quel modèle cognitif dispose l'agent pour représenter le monde ? Comment un agent peut-il mettre en œuvre les actions qui répondent au mieux à ses objectifs ? Cette capacité à la décision est liée à "l'état mental" de l'agent, qui reflète ses perceptions, ses représentations, ses croyances, ses désirs, ses préférences, etc.
- * Problématique de l'interaction : quelles formes d'interaction existent entre les agents ? Comment est-elle représentée ? Quels moyens de communication existent entre les agents ?
- * Problématique de l'adaptation : quelles sont les possibilités et les moyens d'adaptation individuelle et d'apprentissage pour chaque agent ? Existe-t'il un modèle d'adaptation collective ou d'évolution ?
- * Problématique de l'implémentation : comment implémenter des systèmes multi-agents ? Et notamment comment choisir le langage de communication entre agents, le langage de description des lois de l'environnement, celui de représentation des connaissances des agents et de leurs préférences ?

En reprenant ces cinq problématiques, il est possible de décrire quelques éléments de l'architecture

d'un système multi-agent :

- * Les agents doivent être dotés de systèmes de décision. Les théories de la décision sont un domaine à part entière d'étude à ce sujet (voir par exemple [Hansson, 1991] pour un aperçu du domaine).
- * Les agents doivent être dotés d'un modèle cognitif. Là-aussi, plusieurs modèles existent, l'un des plus classiques étant le modèle BDI (*Beliefs-Desires-Intentions*) [Rao et Georgeff, 1991]. Ce modèle considère d'une part l'ensemble de croyances (*Beliefs*) de l'agent sur son environnement, qui sont le résultat de ses connaissances et de ses perceptions ; l'ensemble de ses objectifs (*Desires*), qui sont les états possibles envers lesquels l'agent peut vouloir s'engager ; et enfin l'ensemble des intentions (*Intentions*) qui regroupe l'ensemble des projets qu'il a l'intention de mener à bien.
- * Les agents doivent être dotés d'un système de communication. Plusieurs langages spécialisés ont vu le jour à cette fin : le *Knowledge Query and Manipulation Language* (KQML, [Finin *et al.*, 1994; Labrou et Finin, 1997]), et plus récemment, le standard *FIPA-ACL* créé par la *Foundation for Intelligent Physical Agents FIPA*. Ce dernier standard repose en particulier sur la théorie des actes de langage [Searle, 1969].
- * La problématique de l'adaptation est un sujet épineux, objet de nombreuses recherches à l'heure actuelle. On pourrait toutefois citer l'exemple de certains virus, aussi bien biologiques qu'informatiques, capables de s'adapter à leur environnement en mutant.
- * Enfin, l'implémentation effective du système multi-agent, si elle ne fait pas à proprement parler partie de l'architecture du système, mérite d'être évoquée à travers l'exemple des nombreux langages de programmation qui ont été développés à des fins de recherche en intelligence artificielle (en particulier le langage LISP). Les langages de représentation utiles à ces systèmes, que ce soit pour représenter les connaissances des agents, leurs préférences, ou encore la façon dont ils communiquent, ont également fait l'objet de nombreuses recherches en intelligence artificielle.

Bien que toutes ces problématiques soient liées, nous avons choisi de nous intéresser plus spécifiquement dans ce manuscrit à la *problématique de l'interaction*, et à la *façon de représenter les préférences* de chacun des agents. Comme nous l'avons évoqué plus haut, les agents d'un système peuvent avoir besoin de communiquer afin de résoudre des différences d'opinions et des conflits d'intérêts, de travailler ensemble afin de résoudre des dilemmes, de trouver des preuves, ou tout simplement d'échanger des informations. De plus, chaque agent a des préférences au sein d'un système multi-agent, des désirs sur les états du monde qu'il souhaite atteindre, et ceux qu'il souhaite éviter. Le "sort" de chacun des agents, c'est-à-dire la satisfaction de ses préférences, dépend alors non seulement de ses propres décisions, mais aussi des décisions prises par les autres membres du système. La décision "optimale" pour un individu dépend donc généralement de ses propres actions, mais également de ce que font les autres agents. Comme chacun n'est pas totalement maître de son sort, on dit que les agents sont en *interaction stratégique*. Plusieurs cadres de travail existent pour traiter cette problématique.

Cadre de travail choisi

La théorie des jeux est probablement le modèle formel le plus abouti pour l'étude des interactions stratégiques entre agents. Plusieurs raisons nous ont amené à choisir ce cadre de travail. Tout d'abord les jeux permettent de décrire des situations sociales très différentes : les marchés en économie peuvent

être vus comme des jeux dans lesquels les participants sont des producteurs ou des consommateurs ; et, plus généralement, une partie d'échecs, la formation d'une coalition gouvernementale ou encore une négociation sont autant de jeux différents obéissant à des règles spécifiques. Ensuite, ce cadre de travail est, et a été, l'objet de nombreuses recherches, et est très riche en résultats. C'est également un cadre de travail très intuitif à manipuler : il est en effet facile de visualiser les interactions entre agents.

Informellement, un jeu consiste en un ensemble d'agents (ou joueurs), et pour chaque agent, la donnée d'un ensemble de stratégies possibles et une fonction d'utilité associant une valeur réelle à chaque combinaison possible de stratégies, représentant ses préférences. Des hypothèses sur les croyances de l'agent au cours du jeu doivent être en outre spécifiées pour les jeux dynamiques à information incomplète.

Nous étudierons ici les interactions entre joueurs et la représentation de leurs préférences en nous plaçant dans un cadre simple : les jeux statiques à information complète. Un jeu est statique si les agents choisissent leur stratégie en parallèle et en une seule étape, sans observer les choix des autres joueurs. Il est à information complète si chaque joueur connaît exactement l'état du monde, les préférences et les actions disponibles pour chacun des joueurs.

Plusieurs modes de représentation sont utilisés en théorie des jeux, notamment les formes extensives et les formes normales qui coïncident dans le cas des jeux statiques. Les fonctions d'utilité sont généralement décrites explicitement dans ces représentations, en énumérant toutes les valeurs pour chaque combinaison de stratégies. Le nombre de valeurs numériques à spécifier, c'est-à-dire le nombre de combinaisons de stratégies possibles, est alors exponentiel en fonction du nombre de joueurs, ce qui rend cette représentation explicite des préférences des joueurs déraisonnable lorsque le nombre de joueurs est grand. Ceci devient encore plus problématique lorsque l'ensemble des stratégies disponibles pour un agent consiste à assigner une valeur à chacune de ses variables à partir d'un domaine fini donné (ce qui est le cas dans beaucoup de domaines réalistes). Dans ce cas, la représentation explicite des fonctions d'utilité est de taille exponentielle en fonction du nombre d'agents ($n \times 2^n$ valeurs pour n agents ayant chacun deux stratégies disponibles) et en fonction du nombre de variables contrôlées par les agents ($2 \times 2^p \times 2^p$ valeurs pour deux agents contrôlant chacun p variables booléennes). Ainsi, spécifier les préférences des joueurs de façon explicite est clairement peu raisonnable, tout d'abord car cela nécessiterait une quantité d'espace exponentielle, puis parce qu'étudier ces jeux (en calculant par exemple des concepts de solution comme les équilibres de Nash en stratégies pures) exigerait d'accéder à toutes ces valeurs d'utilité au moins une fois, et serait donc exponentiel en temps en fonction du nombre d'agents et du nombre de variables.

Une solution pour répondre à ces besoins consiste à utiliser un langage permettant une représentation concise des relations de préférences, ou des fonctions d'utilité, sur un ensemble de conséquences qui possède une structure combinatoire (c'est-à-dire un produit cartésien de domaines de valeurs finis pour un ensemble fini de variables). Ces langages ont été activement étudiés ces dernières années, spécialement dans la communauté d'intelligence artificielle. Ils exploitent dans une large mesure des propriétés structurelles des relations de préférences (comme l'indépendance préférentielle entre variables). Partant de là, puisque la spécification d'un jeu statique nécessite la description des préférences des agents, il apparaît naturel de représenter de tels jeux en utilisant des langages de représentation compacte de préférences. Il existe déjà plusieurs cadres répondant aux problèmes que nous avons posés plus haut, notamment les langages graphiques que nous présenterons dans le chapitre 4

(page 121).

Proposition et méthodologie

Comme nous l'avons vu, notre objectif ici est de spécifier de façon concise et efficace les interactions entre agents rationnels au sein d'un système multi-agents. Pour cela, nous avons choisi de nous appuyer sur la théorie des jeux, qui est un modèle formel abouti pour l'étude de ces interactions. Pourtant, une des principales lacunes de cette théorie est la façon de représenter les utilités des joueurs, coûteuse en place mémoire et en temps d'exécution. Nous avons vu qu'une solution pour pallier ces problèmes est d'utiliser un langage de représentation compacte de préférences. La première étape consiste donc à choisir un tel langage, puis d'étudier les jeux pouvant être formalisés dans ce cadre : étude des différents concepts de solution de ces jeux, étude de la complexité des problèmes associés. L'étape suivante consiste à essayer de généraliser ces jeux en introduisant de nouvelles représentations compactes de préférences.

Choix d'un langage de représentation compacte de préférences

Afin de trouver un langage de représentation compacte de préférence répondant à nos besoins, nous devons commencer par répondre à une question préliminaire : les buts des agents doivent-ils être exprimés avec des préférences numériques ou ordinales ? Un langage de représentation de préférences doit être aussi proche que possible de la façon dont les individus "connaissent" leurs préférences et les expriment en langage naturel. Dans ce cadre, le problème conceptuel de l'utilité cardinale est qu'il n'existe pas d'échelle objective de la mesure de l'utilité, et qu'il est donc difficile pour un individu d'évaluer numériquement l'utilité apportée par la satisfaction de chacun de ses désirs. Dans le cadre de l'utilité ordinale, il est demandé au consommateur de pouvoir classer raisonnablement les mondes disponibles en fonction de l'utilité apportée, ce qui semble a priori plus aisé à effectuer pour de nombreux individus.

Les notions que nous voulons étudier dans un jeu représentent un autre critère de choix entre préférences numériques ou ordinales. En effet, quelques notions (telles que équilibres de Nash en stratégies pures et stratégies dominées) peuvent être définies avec des préférences ordinales, tandis que d'autres (telles que équilibres de Nash en stratégies mixtes) ont besoin de préférences cardinales. Ici, nous avons choisi de nous restreindre aux calculs de stratégies pures, et *nous avons donc choisi de représenter les préférences des joueurs de façon ordinale*.

Le langage de représentation compacte de préférences ordinal le plus simple et le plus intuitif paraît être la *logique propositionnelle*. En effet, utiliser la logique propositionnelle pour représenter les préférences des joueurs permet non seulement de simplifier les jeux, mais aussi d'utiliser les propriétés et les outils bien connus de cette logique pour étudier les caractéristiques de tels jeux. Nous avons donc choisi ici d'étudier le cas où chaque agent contrôle un ensemble fini de variables *binaires*. Ces jeux, appelés *jeux booléens*, ont été introduits par [Harrenstein *et al.*, 2001; Harrenstein, 2004a; Dunne et van der Hoek, 2004]. Un jeu booléen est un jeu à deux joueurs, chacun d'entre eux contrôlant un ensemble de variables propositionnelles, et à somme nulle. La fonction d'utilité du joueur 1 (et donc celle du joueur 2 qui est son opposé) est représentée par une formule de la logique propositionnelle, appelée *forme booléenne* du jeu, qui doit être satisfaite pour que le joueur soit également

satisfait. Chaque joueur a donc des préférences *dichotomiques* (soit il est satisfait, soit il ne l'est pas, sans niveau intermédiaire).

Ces trois restrictions (deux joueurs, somme nulle, préférences binaires) limitent fortement l'expressivité des jeux booléens. Nous avons donc commencé par généraliser ces jeux en étudiant des jeux à n joueurs et à somme non nulle, puis nous avons introduit des préférences non binaires.

Etude des jeux booléens

Les jeux booléens peuvent être facilement transformés en des jeux à n joueurs à somme non nulle en définissant les préférences des agents comme étant un n -uplet de formules propositionnelles. Des outils simples issus de la logique propositionnelle nous permettent alors de caractériser certaines propriétés du jeu, comme les équilibres de Nash en stratégies pures et les stratégies dominées, et de calculer la complexité des problèmes associés.

Deux autres notions ont alors été étudiées : les dépendances entre joueurs (si le but, et donc la satisfaction, d'un joueur i dépend de variables contrôlées par le joueur j , alors i aura besoin de j pour satisfaire son but), et les coalitions de joueurs (une coalition dans un jeu booléen est efficace si elle peut garantir à tous ses membres que leurs buts sont satisfaits). Ces deux nouvelles notions nous ont permis de faciliter encore le calcul des concepts de solution tels que les équilibres de Nash en stratégies pures.

Introduction de nouvelles préférences

Tandis qu'une simple formule propositionnelle ϕ ne peut pas exprimer plus qu'une relation de préférence binaire sur les interprétations (les modèles de ϕ sont strictement meilleurs que les modèles de $\neg\phi$), exprimer des préférences (non binaires) dans un cadre propositionnel est possible en utilisant un autre langage de représentation compacte de préférences. Nous avons choisi de nous restreindre encore aux préférences ordinales, et nous avons intégré deux de ces langages aux jeux booléens : tout d'abord, les *buts à priorité* puis les *CP-nets*.

Plan du manuscrit

Après avoir donné dans le chapitre 1 quelques éléments sur la théorie des jeux, nous donnerons dans le chapitre 2 une description (simplifiée) des jeux booléens puis nous montrerons qu'ils peuvent facilement être généralisés de manière à représenter des jeux avec un nombre arbitraire de joueurs et à somme non nulle, mais en gardant l'hypothèse que les préférences de chaque joueur sont représentées par une formule propositionnelle unique, ce qui ne permet de représenter que des utilités binaires. Nous verrons comment introduire la notion de dépendance entre les joueurs, et comment des outils simples issus de la logique propositionnelle permettent de caractériser certaines propriétés du jeu, puis nous donnerons quelques résultats de complexité algorithmique. Nous introduirons ensuite dans le chapitre 3 deux langages de représentation de préférences afin d'enrichir encore les jeux booléens avec des préférences non dichotomiques : les buts à priorité puis les CP-nets. Dans le chapitre 4, nous exposerons quelques-uns des travaux utilisant des langages de représentation compacte de préférences dans des jeux, en montrant le lien avec nos solutions. Nous étudierons ensuite dans le chapitre 5 les

coalitions efficaces dans les jeux booléens. Nous étudierons tout d'abord les propriétés des fonctions d'effectivité associées aux jeux booléens, puis la notion de coalition efficace : nous donnerons une caractérisation exacte des ensembles de coalitions correspondant à des ensembles de coalitions efficaces associées à un jeu booléen, et nous ferons le lien entre coalition efficace et noyau avant de conclure.

La figure 1 donne un guide de lecture des chapitres : une flèche d'un chapitre vers un autre indique que la lecture du premier est nécessaire à la compréhension du second.

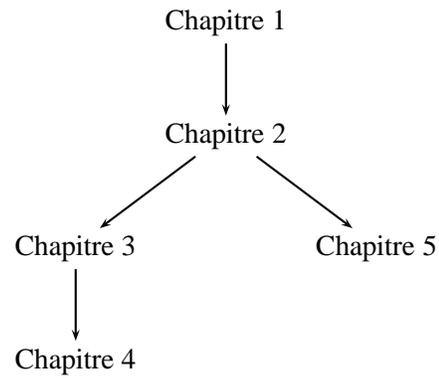


Figure 1 — Guide de lecture

Notations

Soit $V = \{a, b, \dots\}$ un ensemble fini de variables propositionnelles et L_V le langage propositionnel construit à partir de V , des connecteurs habituels et des constantes booléennes \top (*vrai*) et \perp (*faux*). Les formules de L_V seront notées φ, ψ etc. On note $Var(\varphi)$ l'ensemble des variables présentes dans la formule φ .

Un littéral est, soit une variable de V , soit sa négation. Une conjonction finie de littéraux est appelée *terme* ou *cube*, et une disjonction finie de littéraux est appelée une *clause*. On note $Lit(\varphi)$ l'ensemble des littéraux formant la formule φ . Une formule φ est en DNF si c'est une disjonction de termes.

2^V est l'ensemble des interprétations pour V avec la convention suivante : soit M une interprétation pour V et pour tout $x \in V$, M donne la valeur *vrai* à x si $x \in M$ et *faux* sinon. Soit M une interprétation pour V et $\psi \in L_V$, la conséquence logique $M \models \psi$ est définie de la manière usuelle.

Soit $X \subseteq V$. 2^X est l'ensemble des X -interprétations. Une *interprétation partielle* de L_V est une X -interprétation pour $X \subseteq V$. Les interprétations partielles sont représentées par une liste de variables de X , le symbole $-$ représentant la négation d'une variable. Par exemple, si $X = \{a, b, d\}$, la X -interprétation $M = \{a, d\}$ sera notée $a\bar{b}d$. Si $Var(\varphi) \subseteq X$, alors $Mod_X(\varphi)$ représente l'ensemble des X -interprétations satisfaisant φ .

Si $\{V_1, \dots, V_p\}$ est une partition de V et si $\{M_1, \dots, M_p\}$ sont des interprétations partielles, avec $M_i \in 2^{V_i}$, (M_1, \dots, M_p) représente alors l'interprétation $M_1 \cup \dots \cup M_p$.

Rappelons que, quelle que soit la formule φ , une interprétation qui rend φ vrai est un *modèle* de φ .

L'interprétation partielle d'une formule φ par une X -interprétation M_X sera noté : $(\varphi)_{M_X} = \bigwedge_{v \in M_X} v \leftarrow \top, v \in X \setminus M_X \leftarrow \perp$.

Nous aurons également besoin dans ce document de plusieurs notions d'impliquants premiers. Les définitions suivantes sont reprises du rapport de synthèse [Marquis, 2000].

Intuitivement, un impliquant premier d'une formule propositionnelle ψ est un des plus petits termes dont tous les modèles sont des modèles de ψ .

Définition 1. Soit ψ une formule propositionnelle.

- * Un terme α est un impliquant de ψ ssi $\alpha \models \psi$.
- * Un terme α est un impliquant premier de ψ ssi
 - * α est un impliquant de ψ , et
 - * pour chaque impliquant α' de ψ , si $\alpha \models \alpha'$, alors $\alpha' \models \alpha$.

On notera $PI(\psi)$ l'ensemble des impliquants premiers de ψ .

Un L -impliquant (resp. L -impliquant premier) est un impliquant (resp. impliquant premier) dont tous les littéraux appartiennent à l'ensemble L .

Définition 2. Soit $L \subseteq V$ et soit ψ une formule propositionnelle de L_V .

- * Un terme α est un L -impliquant de ψ ssi $\alpha \models \psi$ et $\text{Lit}(\alpha) \subseteq L$.
- * Un terme α est un L -impliquant premier de ψ ssi
 - * α est un L -impliquant de ψ , et
 - * pour chaque L -impliquant α' de ψ , si $\alpha \models \alpha'$, alors $\alpha' \models \alpha$.

On notera $PI_L(\psi)$ l'ensemble des L -impliquants premiers de ψ .

La notion de projection d'une formule sur un ensemble de variables correspond à l'utilisation de l'opérateur *forget*, qui a été étudié par [Lang et al., 2002a, 2003].

On dit que l'on “oublie complètement” la variable x dans une formule ϕ si et seulement si on s'intéresse à la formule (notée $\forall x : \phi$) : $\phi_{x \leftarrow \top} \wedge \phi_{x \leftarrow \perp}$.

On peut aussi “oublier partiellement” x dans ϕ en s'intéressant à la formule (notée $\exists x : \phi$) : $\phi_{x \leftarrow \top} \vee \phi_{x \leftarrow \perp}$.

On note que l'on a :

$$\forall x : \phi \equiv \neg \exists x : \neg \phi$$

Par exemple, soit la formule ϕ suivante :

$$\phi = (a \wedge \neg b \wedge c) \vee (b \wedge \neg c) \vee (a \wedge b \wedge d)$$

La projection de ϕ sur la variable c sera calculée comme suit :

$$\begin{aligned} \exists c : \phi &= [(a \wedge \neg b \wedge \top) \vee (b \wedge \perp) \vee (a \wedge b \wedge d)] \vee [(a \wedge \neg b \wedge \perp) \vee (b \wedge \top) \vee (a \wedge b \wedge d)] \\ &= (a \wedge \neg b) \vee b \vee (a \wedge b \wedge d) \end{aligned}$$

$$\begin{aligned} \forall c : \phi &= [(a \wedge \neg b \wedge \top) \vee (b \wedge \perp) \vee (a \wedge b \wedge d)] \wedge [(a \wedge \neg b \wedge \perp) \vee (b \wedge \top) \vee (a \wedge b \wedge d)] \\ &= ((a \wedge \neg b) \vee (a \wedge b \wedge d)) \wedge (b \vee (a \wedge b \wedge d)) \\ &= a \wedge b \wedge d \end{aligned}$$

1

Eléments de théorie des jeux

La théorie des jeux est un outil mathématique permettant d'étudier les comportements - prévus, réels, ou justifiés a posteriori - d'individus face à des situations d'antagonisme.

Si ses précurseurs furent Cournot [Cournot, 1838] et Edgeworth [Edgeworth, 1897], c'est la parution en 1944 de l'ouvrage de John Von Neumann et Oskar Morgenstern, *The Theory of Games and Economic Behaviour* [von Neumann et Morgenstern, 1944], qui instaura véritablement la théorie des jeux comme étant une nouvelle discipline. Dans cet ouvrage, Von Neumann et Morgenstern proposent une solution dans le cas particulier d'un *jeu à somme nulle* (ce qui est gagné par l'un est perdu par l'autre, et réciproquement). En 1950, John Nash [Nash, 1950] a montré comment les idées développées par Cournot dès 1838 pouvaient servir de base pour construire une théorie de l'équilibre pour des jeux à somme non nulle, qui généralise la solution proposée par Von Neumann et Morgenstern. Les économistes, premiers à s'approprier cet outil, ont été depuis rejoints par, entre autres, les sociologues, les chercheurs en sciences politiques, les philosophes, ou encore les informaticiens.

La théorie des jeux étudie des situations dans lesquelles le sort de chaque participant dépend non seulement des décisions qu'il prend, mais également des décisions prises par d'autres participants. Le choix optimal pour un agent (appelé **joueur**) dépend donc généralement des choix des autres agents. Comme chaque joueur n'est pas totalement maître de son sort, on dit que les agents sont en situation d'*interaction stratégique*. On suppose dans un jeu en interaction stratégique que les joueurs se connaissent : ils savent combien il y a de joueurs, et qui ils sont. Du fait que le gain de chacun dépend en partie des actions des autres, un joueur ne peut pas se contenter de choisir ses propres plans d'actions, en négligeant ce que font les autres. Il doit au contraire se faire une idée aussi précise que possible des stratégies choisies, ou susceptibles d'être choisies, par les autres joueurs. Pour cela, on admet que **les agents sont rationnels**, c'est-à-dire que chaque joueur s'efforce de prendre les meilleures décisions pour lui-même, et sait que les autres joueurs font de même.

Notre objectif ici n'est pas de donner un état de l'art exhaustif sur la théorie des jeux, nous voulons juste introduire quelques concepts qui nous seront utiles dans la suite de ce manuscrit. Nous allons donc tout d'abord présenter une taxonomie partielle des jeux en Section 1.1, puis les types de représentation des jeux en Section 1.2, et enfin nous présenterons en Section 1.3 quelques concepts de solution.

1.1 Taxonomie partielle des jeux

Nous allons présenter ici quatre types de jeux, les jeux statiques, dynamiques, coopératifs et non coopératifs. Un jeu peut réunir plusieurs de ces caractéristiques : il peut être statique et coopératif, statique et non coopératif, dynamique et coopératif ou encore dynamique et non coopératif.

1.1.1 Jeux statiques et dynamiques

La première distinction que nous allons faire est celle entre jeu statique et dynamique. Un jeu est statique lorsque tous les joueurs jouent simultanément en une seule étape, alors qu'il est dynamique lorsque le jeu se déroule en plusieurs étapes (un ou plusieurs joueurs peuvent jouer à chaque étape).

1.1.1.1 Jeux statiques

Un jeu est dit **statique** lorsque les joueurs choisissent **simultanément** leurs actions, et reçoivent ensuite leurs gains respectifs.

Définition 1.1. *Un jeu statique est un ensemble de règles qui encadre le comportement des joueurs et qui détermine les gains des joueurs selon les actions entreprises. Formellement, un jeu G est constitué de :*

- * un ensemble de joueurs $N = \{1, \dots, n\}$,
- * l'ensemble des profils de stratégies ou issues du jeu $S = S_1 \times \dots \times S_n$, où S_i représente l'ensemble des choix possibles (stratégies) du joueur i . On suppose dans la suite que les S_i sont finis.
- * $\forall i \in N$, une fonction d'utilité qui représente les gains du joueur en fonction de l'issue du jeu : $u_i : S \rightarrow \mathbb{R}$. Le joueur i préfère strictement l'issue s à l'issue s' si $u_i(s) > u_i(s')$. Si $u_i(s) = u_i(s')$, i est indifférent entre ces deux issues.

$\forall i \in N$, S_i représente toutes les stratégies s_i disponibles pour le joueur i . Un profil de stratégies, appelé aussi issue du jeu, est une combinaison de stratégies individuelles $s = (s_1, \dots, s_n) \in S_1 \times \dots \times S_n$.

Soit G un jeu statique, avec $N = \{1, \dots, n\}$, $s = (s_1, \dots, s_n)$ et $s' = (s'_1, \dots, s'_n)$ deux profils de stratégies. On note s_{-i} le profil de stratégies s privé de la stratégie du joueur i : $s_{-i} = (s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$. On note (s_{-i}, s'_i) le profil de stratégies s dans lequel on a remplacé la stratégie du joueur i par celle du profil s' : $(s_{-i}, s'_i) = (s_1, s_2, \dots, s_{i-1}, s'_i, s_{i+1}, \dots, s_n)$. On note $S_I = \times_{i \in I} S_i$ l'ensemble des stratégies pour $I \subseteq N$.

Le dilemme du prisonnier, que nous allons présenter maintenant, est un exemple célèbre de la théorie des jeux.

Exemple 1.1. *Deux suspects sont retenus dans des cellules séparées, et ne peuvent donc pas communiquer. La police ne dispose pas d'éléments de preuve suffisants pour obtenir leur condamnation, l'aveu d'au moins un des deux est donc indispensable. La police propose à chacun d'entre eux le marché suivant :*

- * si vous avouez et que votre complice n'avoue pas, vous aurez une remise de peine, tandis que votre complice aura la peine maximale (10 ans) ;

- * si vous avouez tous les deux, vous serez condamnés à une peine plus légère (5 ans) ;
- * si aucun de vous n'avoue, la peine sera minimale (6 mois), faute d'éléments au dossier.

On peut formaliser cette situation par un jeu à 2 joueurs, chaque joueur ayant 2 stratégies possibles : avouer (dénotée par A) ou se taire (dénotée par T).

- * $N = \{1, 2\}$,
- * le prisonnier 1 a deux stratégies possibles : $s_{1_1} = A$ et $s_{1_2} = T$. Il en est de même pour le joueur 2 : $s_{2_1} = A$ et $s_{2_2} = T$.
- * Ce jeu a donc 4 profils de stratégies possibles : AA , AT , TA et TT .
- * On peut donc calculer les utilités de chacun des joueurs pour chacune des issues possibles du jeu :
 - * $u_1(AA) = u_2(AA) = -5$,
 - * $u_1(TT) = u_2(TT) = -0.5$,
 - * $u_1(AT) = u_2(TA) = 0$,
 - * $u_2(AT) = u_1(TA) = -10$.

La bataille des sexes est également un jeu célèbre en théorie des jeux :

Exemple 1.2. *Lucas et Elsa veulent aller au cinéma. Ils ont le choix entre un film d'horreur et une comédie romantique. Pour les deux, ce qui compte avant tout, c'est d'être ensemble. Néanmoins, Elsa a une préférence pour le film d'horreur et Lucas pour la comédie romantique.*

On peut formaliser cette situation par un jeu à 2 joueurs, chaque joueur ayant 2 stratégies possibles : aller voir un film d'horreur (dénotée par H) ou une comédie romantique (dénotée par R).

- * $N = \{1, 2\}$,
- * Elsa (1) a 2 stratégies possibles : $s_{1_1} = H$ et $s_{1_2} = R$. Lucas (2) a les mêmes possibilités : $s_{2_1} = H$ et $s_{2_2} = R$.
- * On peut donc calculer les utilités de chacun des joueurs pour chacune des issues possibles du jeu :
 - * $u_1(H, H) = u_2(R, R) = 2$,
 - * $u_1(R, R) = u_2(H, H) = 1$,
 - * $u_1(H, R) = u_1(R, H) = u_2(H, R) = u_2(R, H) = 0$.

1.1.1.2 Jeux dynamiques

Un jeu dynamique est un jeu qui se déroule en **plusieurs étapes**. On se place ici dans le cadre des **jeux dynamiques en information complète**, c'est-à-dire que l'on admet que toutes les actions passées sont *observables* et *connues* de tous les joueurs. Dans ce cadre, en intervenant à des étapes antérieures du jeu, certains joueurs ont le pouvoir d'affecter directement les gains d'autres joueurs de manière irréversible.

Un jeu est **en information parfaite** si chaque joueur connaît l'ensemble des actions choisies par *tous* les joueurs qui sont intervenus avant qu'il ne sélectionne sa stratégie¹, et qu'il connaît toutes leurs stratégies possibles. Il est le seul joueur à prendre une décision à cette étape². Si plusieurs joueurs

¹Dans le cadre des jeux statiques, la notion d'information parfaite n'a aucun sens : les joueurs jouent simultanément et en une seule étape, et n'ont donc pas à connaître les actions déjà réalisées.

²Par exemple, le jeu d'échecs est un jeu à information parfaite.

choisissent leurs actions simultanément à une étape donnée, ou si les joueurs ne connaissent pas toutes les stratégies des autres joueurs³, le jeu est dit **en information imparfaite**. Ces actions ne sont pas connues et chacun des joueurs intervenant à cette étape se comporte un peu comme dans un jeu statique, à la différence que dans ce cas, l'histoire du jeu influence le choix de chacun.

Définissons le cadre conceptuel général des jeux dynamiques. Soit G un jeu dynamique. On désigne par a^t le **vecteur des actions choisies à l'étape t du jeu** par les participants qui interviennent à cette étape (un seul joueur dans le cas d'un jeu en information parfaite). Soit t une **étape** quelconque du jeu. On définit l'**histoire** $h^t = (a^0, a^1, \dots, a^{t-1})$ du jeu à l'étape t par la séquence de toutes les décisions prises par les joueurs intervenant lors des étapes antérieures. On suppose que tous les joueurs connaissent l'histoire du jeu à chaque étape, c'est-à-dire que toutes les actions passées sont observables et connues par tous les participants. Le reste du jeu (toutes les étapes r telles que $r > t$) est appelé **sous-jeu** de G , et est noté $G(h^t)$. Puisque l'histoire h^t du jeu à chaque étape t est connue, le sous-jeu se déroulant à partir de t peut être vu comme un jeu à part entière induit par l'histoire h^t . L'histoire h^t impose des restrictions sur les choix offerts au joueur i . Soit $A_i(h^t)$ l'**ensemble des actions** auxquelles le joueur i a accès à l'étape t du jeu lorsque l'histoire est donnée par h^t . Si $A_i(h^t)$ est vide, le joueur i n'intervient pas à l'étape considérée⁴. Soit H^t l'**ensemble de toutes les histoires possibles** jusqu'à l'étape t . $A_i(H^t) = \bigcup_{h^t \in H^t} A_i(h^t)$ désigne alors l'ensemble de toutes les actions possibles pour le joueur i à l'étape t selon les histoires possibles.

Nous pouvons à présent définir une stratégie pure d'un jeu dynamique admettant T étapes. Une **stratégie pure pour le joueur i** est définie par une suite de T applications S_i^t de H^t vers $A_i(H^t)$. En d'autres termes, une stratégie pure est une suite de règles de sélection d'une action particulière par le joueur i à chaque étape du jeu compte tenu de l'histoire qui s'est déroulée jusqu'alors. Par exemple, les actions du joueur i à l'étape 0 sont $a_i^0 = s_i^0(h^0)$, celles de l'étape 1 sont $a_i^1 = s_i^1(h^1)$, de l'étape 2 $a_i^2 = s_i^2(h^2)$, et ainsi de suite.

Nous donnerons un exemple de jeu dynamique en Section 1.3.4 (page 27).

1.1.2 Jeux coopératifs et non coopératifs

Comme nous l'avons vu, une caractéristique fondamentale des jeux est que le gain obtenu par un joueur dépend de ses choix, mais aussi des choix effectués par les autres joueurs. Il convient alors de distinguer deux grandes familles de jeux : les jeux coopératifs et les jeux non coopératifs.

Un jeu est **coopératif** lorsque les joueurs peuvent passer entre eux des accords qui les lient de manière contraignante. On dit alors qu'ils forment une **coalition** dont les membres agissent de concert. Dans le cas contraire, c'est-à-dire lorsque les joueurs n'ont pas la possibilité de former des coalitions, le jeu est **non coopératif**.

1.1.2.1 Jeux coopératifs

Un jeu coopératif (appelé aussi jeu coalitionnel) est un jeu dans lequel les joueurs peuvent former des coalitions et agir de concert.

³Les jeux de cartes sont généralement des jeux à information imparfaite : à la belote par exemple, un joueur ne connaît pas toutes les stratégies possibles de ses adversaires, il n'a pas une connaissance parfaite du jeu.

⁴La description des ensembles $A_i(h^t)$ à chaque étape t pour tout joueur i fait partie de la spécification des règles du jeu.

Définition 1.2. Une **coalition** est un sous-ensemble de joueurs : $C \subseteq N = \{1, \dots, N\}$. Si C est une coalition d'un seul joueur ($C = \{i\}$), C est appelé **singleton**. Si C est la coalition formée de tous les joueurs ($C = N$), C est appelé **grande coalition**.

On dit qu'un jeu coopératif est à **utilité non transférable** s'il n'est pas possible d'additionner les utilités des joueurs et de les redistribuer aux membres d'une coalition. Chaque membre d'une coalition essaie d'optimiser le montant obtenu par chacun individuellement.

Définition 1.3. Un **jeu coopératif à utilité non transférable** est une paire (N, v) où

- * N est un ensemble de joueurs
- * v est une **fonction caractéristique** qui associe un vecteur $v(C) \in \mathbb{R}^C$ à chaque coalition C de N , chaque élément v_i du vecteur $v(C)$ correspondant à l'utilité obtenue par le joueur i dans la coalition C .

La **bataille des sexes** peut être transformée en un jeu coopératif à utilité non transférable [Luce et Raiffa, 1957] :

Exemple 1.2 (page 11) – suite : La bataille des sexes peut être formalisée par le jeu coopératif à utilité non transférable à 2 joueurs suivant :

- * $N = \{1, 2\}$,
- * v définie par :
$$\begin{cases} v(12) = \{(v_1, v_2); 1 \leq v_1 \leq 2, 1 \leq v_2 \leq 2\} \\ v(1) = \{(v_1); 0 \leq v_1 \leq 2\} \\ v(2) = \{(v_2); 0 \leq v_2 \leq 2\} \end{cases}$$

En effet, si Lucas et Elsa se coordonnent, ils sont sûrs d'aller au même endroit, et donc auront chacun au moins une utilité de 1. Par contre, s'ils ne se coordonnent pas, ils ont toutes les chances de ne pas être au même endroit, et donc de n'en retirer aucune satisfaction (même s'ils peuvent avoir de la chance et que l'un des deux se dévoue, en espérant que l'autre n'ait pas eu la même idée).

On dit qu'un jeu coopératif est à **utilité transférable** s'il est possible d'additionner les utilités des joueurs et de les redistribuer aux membres d'une coalition (il existe une "monnaie" commune à tous avec laquelle on peut effectuer des transferts).

Définition 1.4. Un **jeu coopératif à utilité transférable** est une paire (N, v) où

- * N est un ensemble de joueurs
- * v est une **fonction caractéristique** qui associe une valeur $v(C) \in \mathbb{R}$ à chaque coalition C de N .

Pour chaque coalition C , $v(C)$ est le paiement total que peuvent se partager les joueurs appartenant à C , indépendamment du comportement des joueurs n'appartenant pas à C .

Un jeu coopératif à utilité transférable est :

- * **symétrique** si la valeur d'une coalition ne dépend que de sa taille : il existe une fonction f telle que $\forall C \subseteq N, v(C) = f(|C|)$;
- * **monotone** si $B \subseteq C \Rightarrow v(B) \leq v(C)$;
- * **superadditif** si $B \cap C = \emptyset \Rightarrow v(B \cup C) \geq v(B) + v(C)$;

- * **simple** si pour toute coalition C , soit $v(C) = 1^5$ (**coalition gagnante**), soit $v(C) = 0$ (**coalition perdante**), et $v(N) = 1$.

Un joueur i dans un jeu coopératif à utilité transférable :

- * a un **droit de veto** s'il appartient à toutes les coalitions gagnantes ($v(C) = 1 \Rightarrow i \in C$);
- * est un **dictateur** si une coalition est gagnante si et seulement si il en fait partie ($v(C) = 1 \Leftrightarrow i \in C$).

Exemple 1.3. *Présentons ici plusieurs jeux coopératifs à utilité transférable à 3 joueurs satisfaisant différentes propriétés⁶ :*

- * Majorité simple. *Une coalition est gagnante si et seulement si elle comprend au moins deux membres*
 $\Rightarrow \begin{cases} v(1) = v(2) = v(3) = 0 \\ v(1,2) = v(2,3) = v(1,3) = v(1,2,3) = 1 \end{cases}$
- * Unanimité. *Une coalition est gagnante si et seulement si elle comprend tous les membres*
 $\Rightarrow \begin{cases} v(1,2,3) = 1 \\ \forall C \subset N, v(C) = 0 \end{cases}$
- * Le joueur 2 a un droit de veto. *Une coalition est gagnante si et seulement si elle comprend au moins deux membres, dont le joueur 2 : 2 peut empêcher une coalition de gagner, mais ne peut pas pour autant gagner seul*
 $\Rightarrow \begin{cases} v(1) = v(2) = v(3) = v(1,3) = 0 \\ v(1,2) = v(2,3) = v(1,2,3) = 1 \end{cases}$
- * Le joueur 2 est dictateur. *Une coalition est gagnante si et seulement si elle comprend le joueur 2*
 $\Rightarrow \begin{cases} v(1) = v(3) = v(1,3) = 0 \\ v(2) = v(1,2) = v(2,3) = v(1,2,3) = 1 \end{cases}$

1.1.2.2 Jeux non coopératifs

Les jeux non coopératifs se divisent en deux grandes familles : les jeux à somme nulle, et ceux à somme non nulle. En économie, cette notion simplificatrice de jeu à somme nulle est importante : ces jeux correspondent à l'absence de production, ou de destruction, de produits.

Les **jeux à somme nulle** sont tous les jeux où la somme "algébrique" des gains des joueurs est constante : ce que gagne l'un est nécessairement perdu par un autre. Stricto sensu, il est possible que les jeux ne soient pas à somme nulle, mais à somme constante, et cela n'a aucune importance en pratique : l'enjeu est de répartir entre tous les joueurs un total de gains préalablement fixé. Les échecs, le poker ou encore le jeu **matching pennies**, présenté ci-dessous, sont des jeux à somme nulle, les gains d'un joueur étant très exactement les pertes d'un autre joueur, tandis que le dilemme du prisonnier est un jeu non coopératif à somme non nulle⁷.

Exemple 1.4. *Deux joueurs, Robin et Annelise, annoncent simultanément pile ou face.*

- * *Si les annonces sont identiques, Robin donne 15 euros à Annelise.*

⁵La valeur 1 est choisie arbitrairement, il faut juste que ce soit la même pour toutes les coalitions gagnantes.

⁶Par souci de simplifier les notations, on écrira $v(i, j)$ au lieu de $v(\{i, j\})$.

⁷Les deux prisonniers sont enfermés dans des cellules séparées, ne peuvent pas communiquer, et donc ne peuvent pas passer un accord et former une coalition, mais ce que gagne l'un n'est pas forcément perdu par l'autre.

* Si les annonces ne concordent pas, c'est Annelise qui doit donner 15 euros à Robin.

On peut formaliser cette situation par un jeu à 2 joueurs, chaque joueur ayant 2 stratégies possibles : annoncer pile (dénnotée par P) ou face (dénnotée par F).

* $N = \{1, 2\}$,

* Annelise (1) a 2 stratégies possibles : $s_{11} = P$ et $s_{12} = F$. Robin (2) a les mêmes possibilités : $s_{21} = P$ et $s_{22} = F$.

* On peut donc calculer les utilités de chacun des joueurs pour chacune des issues possibles du jeu :

$$* u_1(P, P) = u_1(F, F) = u_2(P, F) = u_2(F, P) = 15, \text{ et}$$

$$* u_1(P, F) = u_1(F, P) = u_2(P, P) = u_2(F, F) = -15.$$

Ce jeu est bien un jeu à somme nulle : tout ce qui est gagné par Annelise est perdu par Robin, et vice-versa.

En 1944, John von Neumann et Oskar Morgenstern [von Neumann et Morgenstern, 1944] ont démontré que tout jeu à somme nulle à n joueurs est une forme généralisée des jeux à somme nulle à 2 joueurs, et qu'il est possible de ramener tout jeu à somme non nulle à n joueurs à un jeu à somme nulle à $n + 1$ joueurs, le $n + 1$ ^{ème} joueur représentant le gain ou la perte globale.

Les jeux à somme nulle à 2 joueurs constituent donc une partie essentielle de la théorie mathématique des jeux.

On peut noter ici que si les jeux à somme nulle et à deux joueurs sont des jeux non coopératifs, les jeux à somme nulle et à n joueurs peuvent être coopératifs, si par exemple $n - 1$ joueurs se liguent contre le n ^{ème} joueur pour le faire perdre, et se partager les gains.

Exemple 1.5. *Tristan, Aguirre et Matisse jouent à Pile ou Face. Si deux d'entre eux annoncent la même chose, ils gagnent et le troisième perd. Si les trois ont la même annonce, aucun d'entre eux ne gagne.*

On peut formaliser cette situation par le jeu à somme nulle et à utilités non transférables à 3 joueurs suivant :

* $N = \{1, 2, 3\}$,

* Tristan (1) a 2 stratégies possibles : $s_{11} = P$ et $s_{12} = F$. Aguirre (2) et Matisse (3) ont les mêmes possibilités : $s_{21} = s_{31} = P$ et $s_{22} = s_{32} = F$.

* On peut donc calculer les utilités de chacun des joueurs pour chacune des issues possibles du jeu :

$$* u_1(P, P, P) = u_1(F, F, F) = u_2(P, P, P) = u_2(F, F, F) = u_3(P, P, P) = u_3(F, F, F) = 0,$$

$$* u_1(P, P, F) = u_2(P, P, F) = u_1(P, F, P) = u_3(P, F, P) = u_2(F, P, P) = u_3(F, P, P) =$$

$$u_1(F, F, P) = u_2(F, F, P) = u_1(F, P, F) = u_3(F, P, F) = u_2(P, F, F) = u_3(P, F, F) = 1.$$

$$* u_3(P, P, F) = u_2(P, F, P) = u_1(F, P, P) = u_3(F, F, P) = u_2(F, P, F) = u_1(P, F, F) = -2.$$

Ce jeu peut être vu comme un jeu coopératif, 2 joueurs ont intérêt à se mettre d'accord pour avoir une chance de battre le 3^{ème}. Il est à utilités non transférables car si un joueur gagne, il ne peut pas donner une partie de sa victoire, ou de ce qu'il a gagné, à un autre joueur.

Ce jeu peut aussi être vu comme un jeu à somme nulle et à utilités transférables si on suppose par exemple que Tristan, Aguirre et Matisse misent à l'origine 25 euros chacun ; que s'ils ont tous les trois la même annonce ils récupèrent leur mise, mais si l'annonce d'un joueur est différente de celle des deux autres, il perd sa mise tandis que les deux autres gagnent 37.5 euros chacun.

1.1.3 Récapitulatif

Nous avons donc vu dans cette section une taxonomie de quelques jeux, que nous récapitulons dans le tableau représenté Figure 1.1, dans lequel nous avons placé les quelques exemples déjà étudiés.

	Somme nulle			Somme non nulle		
	¬Coop.	Coop.		¬Coop.	Coop.	
		Util. transf.	Util. ¬transf.		Util. transf.	Util. ¬transf.
Statique	Ex. 1.4	Ex. 1.5	Ex. 1.5	Ex. 1.1	Ex. 1.3	Ex. 1.2
Dynamique				Ex. 1.12		

Figure 1.1 — Taxonomie des jeux

Comme c'est visible sur ce tableau, certains cas ne sont pas illustrés par des exemples. Ces derniers concernent les jeux dynamiques, que nous n'étudierons pas dans la suite de ce rapport. Nous avons donc omis de présenter exhaustivement cette catégorie de jeux.

Nous allons à présent voir comment l'on peut représenter ces jeux.

1.2 Représentation des jeux

Un jeu stratégique peut être représenté de deux façons différentes mais équivalentes : sous forme normale (dite aussi stratégique) et sous forme extensive (dite aussi développée).

1.2.1 Forme extensive

Un jeu sous **forme extensive** est défini par un **arbre** de décision décrivant les actions possibles des joueurs à chaque étape du jeu, la séquence de tours de jeu des joueurs ainsi que l'information dont ils disposent à chaque étape pour prendre leur décision. Chaque nœud de l'arbre spécifie le joueur qui doit choisir une action (ou stratégie) à ce moment du jeu, ainsi que l'information dont il dispose. Les gains que chaque joueur peut réaliser après avoir suivi un des chemins possibles au sein de l'arbre, correspondant à chaque profil de stratégies, sont associés à chaque feuille de l'arbre. Par exemple, la forme extensive du dilemme du prisonnier est représentée Figure 1.2 (page suivante).

Exemple 1.1 (page 10) – suite : *Une forme extensive du dilemme du prisonnier est la suivante :*

Chaque nœud de cet arbre correspond à un joueur, et chaque branche partant de ce nœud correspond à une stratégie possible de ce joueur.

Dans les feuilles, le premier élément de chaque couple représente l'utilité du prisonnier 1, tandis que le second élément représente celle du prisonnier 2.

Le cercle en pointillé entourant les deux occurrences du joueur 2 signifie que ce dernier ne sait pas dans quelle situation il se trouve, il ne sait pas si son complice a choisi d'avouer ou de se taire.

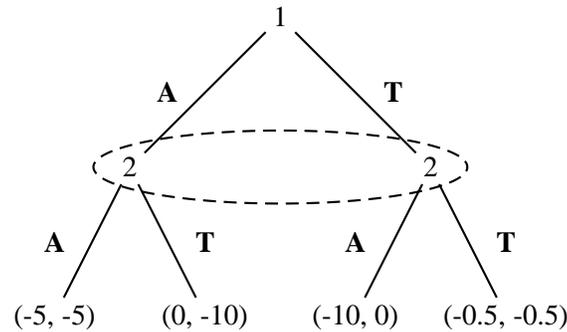


Figure 1.2 — Forme extensive du dilemme du prisonnier

Ce jeu a une seconde forme extensive, plaçant le second joueur en haut de l'arbre. Ces deux représentations sont équivalentes car les deux joueurs jouent simultanément.

La forme extensive permet une description “dynamique” du jeu parce qu'elle spécifie les séquences de décisions prises par les joueurs. Pourtant, lorsqu'un jeu implique de multiples joueurs et de multiples choix, l'arbre peut devenir complexe à représenter. Or, lorsque les stratégies des agents sont choisies en parallèle, c'est-à-dire sans que l'un des joueurs observe les décisions des autres, comme c'est le cas dans le dilemme du prisonnier, cette construction basée sur un modèle dynamique n'est pas nécessaire. Dans ce cas, on simplifiera la présentation du jeu en utilisant une représentation sous forme normale, absolument équivalente à la forme extensive associée.

1.2.2 Forme normale

Un jeu sous **forme normale** est la donnée de l'ensemble des joueurs, de l'ensemble des stratégies pour chaque joueur et des paiements associés à toute combinaison possible de stratégies. On peut alors représenter ces jeux **sous forme matricielle**, en associant à chaque profil de stratégies s un n -uplet donnant l'utilité obtenue par chaque joueur dans l'ordre : $(u_1(s), u_2(s), \dots, u_n(s))$. Par exemple, la forme normale du dilemme du prisonnier est représentée Figure 1.3.

Exemple 1.1 (page 10) – suite : *La forme normale du dilemme du prisonnier est représentée Figure 1.3.*

		2	A	T
1	A	(-5, -5)	(0, -10)	
	T	(-10, 0)	(-0.5, -0.5)	

Figure 1.3 — Forme normale du dilemme du prisonnier

Comme précédemment, le premier élément de chaque couple représente l'utilité du prisonnier 1, tandis que le second élément représente celle du prisonnier 2.

Cette représentation sous forme matricielle permet de représenter des jeux ayant un nombre de joueurs et un nombre de stratégies pour chaque joueur raisonnable : la taille de la matrice est exponentielle

en fonction du nombre d'agents, et du nombre de choix possible pour chaque agent. Par exemple, si n agents ont chacun le choix entre deux actions possibles, il faudra spécifier $n \times 2^n$ valeurs numériques. Ces descriptions, sous forme extensive ou normale, supposent que les ensembles d'actions sont finis. A chaque jeu sous forme extensive correspond un jeu sous forme normale dans lequel les joueurs choisissent simultanément les stratégies qu'ils mettront en œuvre. En revanche, un jeu sous forme normale peut correspondre à plusieurs jeux sous forme extensive différente, comme nous l'avons vu pour le dilemme du prisonnier.

1.3 Concepts de solution

Etudions un autre exemple classique de jeu statique, qui suit la même logique que le dilemme du prisonnier, le **jeu de la tirelire** :

Exemple 1.6. *On propose à Jérémie et Léonore le jeu suivant : ils ont chacun la possibilité de mettre 0 ou 100 euros dans une tirelire. Une fois qu'ils ont tous deux pris une décision, sans connaître la décision de l'autre, le contenu de la tirelire est multiplié par 1.5 et est réparti en part égale entre les deux joueurs.*

Une fois ce jeu formalisé, si on choisit Jérémie comme étant le joueur 1, Léonore le joueur 2, on obtient la forme normale représentée Figure 1.4.

	2	0	100
1			
0	(0, 0)	(75, -25)	
100	(-25, 75)	(50, 50)	

Figure 1.4 — Forme normale du jeu de la tirelire

Dans ce cas de figure, que vont faire Jérémie et Léonore ? Mettons nous à la place de Jérémie, qui tient le raisonnement suivant : “ Si Léonore ne met rien dans la tirelire, il est optimal pour moi que je ne mette rien, sinon je perdrai 25 euros. Si elle dépose 100 euros, je gagne 75 euros si je ne mets rien, et 50 si je mets 100 euros. Donc, dans les deux cas de figure, j'ai intérêt à ne rien mettre dans la tirelire.” Si Léonore tient le même raisonnement, le résultat sera un gain nul pour chacun d'entre eux. Même s'ils se mettent d'accord au début du jeu pour mettre tous les deux 100 euros, il reste “optimal” de ne rien mettre dans la tirelire s'ils sont motivés par la recherche de leur seul intérêt personnel.

La logique qui se trouve derrière ce jeu, et qui est la même que celle se trouvant derrière le dilemme du prisonnier, montre qu'un groupe d'individus ne va pas nécessairement se comporter dans l'intérêt du groupe si chacun peut obtenir pour lui-même un résultat meilleur en choisissant pour son propre compte. Sachant cela, nous allons à présent présenter quelques concepts permettant de prédire l'issue d'un jeu (ce que l'on appelle des “concepts de solution”).

L'analyse d'un jeu permet de prédire l'**équilibre** qui émergera si les joueurs sont rationnels. Par équilibre, nous entendons un état ou une situation dans lequel aucun joueur ne souhaite modifier son

comportement compte tenu du comportement des autres joueurs. De façon plus précise, un équilibre est une combinaison de stratégies telle qu'aucun des joueurs n'a d'intérêt à changer sa stratégie compte tenu des stratégies des autres joueurs. Une fois que l'équilibre a été atteint dans un jeu (et peu importe la manière dont il a été obtenu), il n'y a aucune raison de le quitter.

Nous allons présenter ici plusieurs concepts de solution : les deux premiers, équilibres de Nash et stratégies dominées, s'appliquent à des jeux statiques non coopératifs ; le concept de noyau (core) s'applique à des jeux statiques coopératifs, et enfin les équilibres parfaits de Selten pour les jeux dynamiques.

1.3.1 Équilibres de Nash

L'**équilibre de Nash**, introduit par John Nash en 1950 [Nash, 1950], est un concept fondamental en théorie des jeux. Il décrit une issue du jeu dans laquelle aucun joueur ne souhaite modifier sa stratégie étant donnée la stratégie de chacun de ses rivaux.

Les jeux pour lesquels il est possible de calculer les équilibres de Nash sont représentés Figure 1.5.

		Somme nulle		Somme non nulle		
		¬Coop.	Coop.	¬Coop.	Coop.	
		Util. transf.	Util. ¬transf.		Util. transf.	Util. ¬transf.
		Statique	Nash			Nash
Dynamique						

Figure 1.5 — Taxonomie des jeux : Domaine d'application des équilibres de Nash

1.3.1.1 Équilibre de Nash en stratégies pures

Définition 1.5. Soit G un jeu non coopératif à n joueurs, avec $N = \{1, \dots, n\}$ l'ensemble de joueurs. Un profil de stratégie $s = (s_1, s_2, \dots, s_n)$ est un **équilibre de Nash en stratégies pures (PNE)** si et seulement si

$$\forall i \in N, \forall s'_i \in S_i, u_i(s) \geq u_i(s'_i, s_{-i})$$

En d'autres termes, un équilibre de Nash est un profil de stratégie dont aucun joueur n'a intérêt à dévier s'il suppose que les autres joueurs ne dévieront pas non plus.

Exemple 1.1 (page 10) – suite : Le profil de stratégies AA est un équilibre de Nash en stratégies pures du dilemme du prisonnier. En effet, on peut vérifier dans la matrice des paiements représentée Figure 1.3 (page 17) que l'on a : $u_1(AA) \geq u_1(TA)$ et $u_2(AA) \geq u_2(AT)$.

AA est le seul équilibre de Nash de ce jeu. En effet, AT ne peut pas être un PNE car AA est une meilleure stratégie pour le joueur 2 : $u_2(AA) > u_2(AT)$; TA non plus car AA est une meilleure stratégie pour le joueur 1 : $u_1(AA) > u_1(TA)$; et de même pour TT car AT est une meilleure stratégie pour le joueur 1 : $u_1(TT) < u_1(AT)$.

Ce jeu n'a qu'un équilibre de Nash en stratégies pures. Pourtant, cette unicité n'est pas toujours garantie :

Exemple 1.2 (page 11) – suite : *Le jeu de la bataille des sexes a deux équilibres de Nash en stratégies pures : HH et RR. En effet, on peut vérifier dans la matrice des paiements représentée Figure 1.6 que l'on a : $u_1(H,H) \geq u_1(R,H)$, $u_2(H,H) \geq u_2(H,R)$; $u_1(R,R) \geq u_1(H,R)$ et $u_2(R,R) \geq u_2(R,H)$.*

		2	
		H	R
1	H	(2, 1)	(0, 0)
	R	(0, 0)	(1, 2)

Figure 1.6 — Forme normale de la bataille des sexes

De la même façon, l'existence d'un équilibre de Nash en stratégies pures n'est pas garantie non plus, comme on peut le constater facilement dans le jeu matching pennies de l'exemple 1.4 (page 14).

1.3.1.2 Équilibres de Nash en stratégies mixtes

Dans ce genre de situation, il existe une autre méthode pour obtenir un équilibre de Nash, et donc un équilibre : il est possible d'élargir la définition d'une stratégie, et d'y inclure non seulement les actions pures (telles qu'annoncer pile ou face), mais aussi les probabilités de choisir l'une ou l'autre de ces actions. Chaque joueur associe une probabilité p_i (positive ou nulle) à la stratégie s_i , et vise à maximiser ses gains espérés en choisissant la meilleure loterie possible, c'est-à-dire la meilleure **stratégie mixte**. Une stratégie mixte est donc une stratégie définissant les probabilités avec lesquelles les joueurs choisissent chacune de leurs stratégies pures.

Définition 1.6. *Une stratégie mixte pour le joueur i est une distribution de probabilité sur S_i . $\Sigma_i = \Delta(S_i)$ représente l'ensemble des stratégies mixtes du joueur i . La fonction $\sigma_i : S_i \rightarrow \mathbb{R}$ associée à la stratégie pure s_i sa probabilité d'être jouée.*

On peut noter ici qu'une stratégie pure s_i correspond à la stratégie mixte s_i associée à une probabilité de 1.

Définition 1.7. *Un équilibre de Nash en stratégies mixtes est un profil de stratégies mixtes $\sigma \in \Sigma$ tel que*

$$\forall i \in N, \forall \sigma'_i \in \Sigma_i, u_i(\sigma) \geq u_i(\sigma'_i, \sigma_{-i})$$

Un équilibre en stratégies mixtes est donc une situation dans laquelle tous les joueurs choisissent leur stratégies mixtes de façon à rendre leurs adversaires indifférents entre les gains espérés de chacune de leurs stratégies pures.

Exemple 1.4 (page 14) – suite : *Le jeu matching pennies n'a aucun équilibre de Nash en stratégies pures.*

Supposons que la distribution de probabilités pour Annelise est la suivante : $\sigma_1(P) = x$ et $\sigma_1(F) = 1 - x$; et que celle de Robin est $\sigma_2(P) = y$ et $\sigma_2(F) = 1 - y$.

Dans ce cas, si on suppose que Robin choisit $\sigma_2(P) = 1$ (il choisit toujours P), d'après la Figure 1.7 son gain espéré est de : $15(1 - x) - 15x = 15(1 - 2x)$. De même, si Robin choisit $\sigma_2(F) = 1$, son gain espéré est de : $15(2x - 1)$. On constate alors que si Annelise choisit une probabilité $\sigma_1(P) = x = 1$ (elle jouera toujours P), alors Robin a tout intérêt de choisir toujours $\sigma_2(F) = 1$ ($y = 0$). Par contre, si Robin choisit $\sigma_2(F) = 1$, Annelise choisira $\sigma_1(F) = 1$ ($x = 0$). Cette situation n'est donc pas un équilibre.

En raisonnant de cette façon, on constate que pour ne pas influencer le choix de Robin, Annelise a donc tout intérêt à choisir $x = 1/2$. Le gain espéré de Robin sera alors toujours 0; et de même, Robin a intérêt à choisir $y = 1/2$.

On vérifie que ce jeu est un équilibre de Nash en stratégies mixtes : si Annelise choisit $\sigma_1(P) = x = 1/2$, l'utilité espérée de Robin est de 0 qu'il choisisse $\sigma_2(P) = 1$ ($15(1 - 2x)$) ou $\sigma_2(F) = 1$ ($15(2x - 1)$). Annelise n'a donc pas intérêt à dévier. En effectuant le même raisonnement pour Robin, on constate que s'il choisit $y = 1/2$, l'utilité espérée d'Annelise est de 0 également, et Robin n'a pas non plus intérêt à dévier.

Vérifions à présent que cet équilibre de Nash en stratégies mixtes est le seul : supposons que $y = 1/2$ et $x \neq 1/2$. Dans ce cas, si $x > 1/2$, Robin a intérêt à choisir $y = 0$, et si $x < 1/2$, Robin choisira $y = 1$. Donc Annelise est sûre que Robin déviera. De même, si $x = 1/2$ et $y \neq 1/2$, Annelise aura tout intérêt à dévier.

Ce jeu a donc un équilibre de Nash mixte : $\sigma_1(F) = \sigma_1(P) = 1/2$, $\sigma_2(F) = \sigma_2(P) = 1/2$.

	2	P	F
1			
	P	(15, -15)	(-15, 15)
	F	(-15, 15)	(15, -15)

Figure 1.7 — Forme normale du jeu matching pennies

Quelques propriétés, classiques en théorie des jeux, découlent de ces définitions (voir par exemple [Osborne et Rubinstein, 1994; Hillas et Kohlberg, 2002]) :

- * Tout équilibre de Nash en stratégies pures est un équilibre de Nash en stratégies mixtes.
- * Tout jeu fini a au moins un équilibre de Nash en stratégies mixtes.

1.3.2 Stratégies dominées

Les jeux pour lesquels il est possible de calculer les stratégies dominées sont représentés Figure 1.8 (page suivante).

Comme le montre l'exemple suivant, qui présente le **jeu de la soirée**, les conditions d'existence d'un équilibre de Nash en stratégies pures sont parfois trop faibles.

		Somme nulle		Somme non nulle			
		\neg Coop.	Coop.		\neg Coop.	Coop.	
			Util. transf.	Util. \neg transf.		Util. transf.	Util. \neg transf.
Statique	Strat. Dom				Strat. Dom		
Dynamique							

Figure 1.8 — Taxonomie des jeux : Domaine d'application des stratégies dominées

Exemple 1.7. *Agathe et Iwan sont invités à une soirée. La raison principale qui les motive à y aller est de pouvoir se voir là-bas : si l'autre n'y va pas, il leur est indifférent d'y aller (dénnoté par A) ou pas (dénnoté par P).*

Une fois ce jeu formalisé, si on choisit Agathe comme étant le joueur 1 et Iwan le joueur 2, on obtient la forme normale représentée Figure 1.9.

		2	
		A	P
1	A	(1, 1)	(0, 0)
	P	(0, 0)	(0, 0)

Figure 1.9 — Forme normale du jeu de la soirée

Ce jeu a 2 équilibres de Nash en stratégies pures : AA et PP. Pourtant, un seul de ces équilibres est intéressant : si Agathe et Iwan veulent se voir, et qu'ils ont l'occasion de le faire à cette soirée, ils iront tous les deux.

On voit sur cet exemple que l'un des équilibres de Nash en stratégies pures, PP , n'est pas une stratégie optimale. En effet, pour chacun des deux joueurs, la stratégie A permet toujours d'obtenir une utilité au moins aussi bonne que la stratégie P . Il semble alors assez naturel qu'Agathe et Iwan choisissent cette stratégie. On dit que A est une stratégie (faiblement) dominante pour chacun des deux joueurs de ce jeu.

Définissons les notions de stratégies strictement et faiblement dominées :

Définition 1.8. *La stratégie s_i du joueur i est dite **strictement dominée** s'il existe une autre stratégie s'_i telle que, quelles que soient les stratégies des autres joueurs, s'_i assure au joueur i une utilité strictement plus grande que s_i . Donc :*

$s_i \in S_i$ est **strictement dominée** si

$$\exists s'_i \in S_i \text{ telle que } \forall s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) < u_i(s'_i, s_{-i})$$

Définition 1.9. *La stratégie s_i du joueur i est dite **faiblement dominée** s'il existe une autre stratégie s'_i telle que, quelles que soient les stratégies des autres joueurs, s'_i assure au joueur i une utilité au*

moins aussi grande que s_i , et qu'il existe au moins une combinaison des stratégies des autres joueurs telle que l'utilité du joueur i avec s'_i soit strictement plus grande que celle avec s_i . Donc : $s_i \in S_i$ est **faiblement dominée** si $\exists s'_i \in S_i$ telle que

$$\forall s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) \leq u_i(s'_i, s_{-i})$$

et que

$$\exists s'_{-i} \in S_{-i} \text{ telle que } u_i(s_i, s'_{-i}) < u_i(s'_i, s'_{-i})$$

Dans l'exemple 1.7 (page ci-contre), chaque joueur a une stratégie faiblement dominée (P) et, Agathe et Iwan étant deux joueurs rationnels, il est facile de voir qu'ils joueront leur stratégie dominante, à savoir A . Pourtant, l'existence d'une solution aussi simple est rare. Il est souvent nécessaire de faire appel à d'autres manières de raisonner dans l'espoir de trouver un équilibre au jeu. Par exemple, si le joueur 1 possède une stratégie strictement ou faiblement dominante, on peut s'attendre à ce qu'il choisisse cette stratégie. Comme le joueur 2 est capable d'anticiper ce choix, il choisit alors sa meilleure stratégie contre la stratégie dominante du premier.

Exemple 1.8. Soit un jeu G à deux joueurs, chacun des joueurs ayant 3 stratégies possibles, représenté par la forme normale donnée Figure 1.10.

	2			
1		G	M	D
H		(4, 3)	(5, 1)	(6, 2)
M		(2, 1)	(8, 4)	(3, 6)
B		(3, 0)	(9, 6)	(2, 8)

Figure 1.10 — Forme normale du jeu présentant l'élimination des stratégies dominées

Il est tout d'abord possible de remarquer que M est une stratégie strictement dominée par D pour le joueur 2. Dans ce cas, il est possible de penser que le joueur 2 ne retiendra jamais cette stratégie. On peut donc éliminer la stratégie M de la matrice.

Ceci étant fait, on remarque alors que, dans la matrice résultante, H est "devenue" une stratégie strictement dominante pour le joueur 1. Dès lors, celui-ci devrait jouer H . Le joueur 2 étant rationnel, il est capable d'anticiper ce raisonnement, il sait donc qu'il est optimal pour lui de choisir G .

En conséquence, une fois que l'on a éliminé toutes les stratégies dominées, on obtient un profil de stratégies résultat : HG .

Le processus d'élimination qui vient d'être appliqué dans cet exemple est appelé **processus d'élimination des stratégies dominées**. Il demande un comportement assez sophistiqué, dans la mesure où chaque joueur doit être capable de reconstituer les opérations auxquelles les autres joueurs procèdent et d'en déduire de nouvelles implications pour lui-même.

Toutefois, le processus de dominance successive admet également des limites :

Exemple 1.9. Soit le jeu G à deux joueurs représenté par la forme normale donnée Figure 1.11 (page suivante).

	2	<i>G</i>	<i>D</i>
1		<i>G</i>	<i>D</i>
<i>H</i>		(3, 6)	(7, 1)
<i>M</i>		(5, 1)	(8, 0)
<i>B</i>		(6, 0)	(6, 2)

Figure 1.11 — Forme normale du jeu présentant les limites de l'élimination des stratégies dominées

Dans ce jeu, il est clair que *H* est strictement dominé par *M* pour le joueur 1. On élimine donc la stratégie *H*. On ne peut pas aller plus loin car il n'y a plus de stratégie strictement ou faiblement dominée dans la matrice résultante. Cela revient à dire que, dans ce cas, le processus d'élimination des stratégies dominées ne conduit pas à un résultat unique.

Nous pouvons énoncer ici quelques propriétés, classiques en théorie des jeux (voir par exemple [Osborne et Rubinstein, 1994; Hillas et Kohlberg, 2002]) :

- * Comme nous l'avons vu dans l'exemple 1.9 (page précédente), le processus d'élimination des stratégies dominées ne conduit pas nécessairement à une solution unique ;
- * L'ordre d'élimination des stratégies strictement dominées n'affecte pas le résultat final ;
- * L'ordre d'élimination des stratégies faiblement dominées peut affecter le résultat final ;
- * Une stratégie strictement dominée ne peut jamais être présente dans un équilibre de Nash en stratégies pures ;
- * Une stratégie faiblement dominée peut apparaître dans un équilibre de Nash en stratégies pures.

On a pu constater que le concept de stratégies dominées pouvait conduire à des résultats alors que celui d'équilibre de Nash en stratégies pures aboutissait à une impasse. Mais l'inverse peut être vrai aussi : le processus d'élimination des stratégies dominées ne conduit pas forcément à un résultat, alors que l'on peut obtenir un équilibre de Nash, comme le montre l'exemple suivant :

Exemple 1.10. Soit le jeu *G* à deux joueurs représenté par la forme normale donnée Figure 1.12.

	2	<i>A</i>	<i>B</i>	<i>C</i>
1		<i>A</i>	<i>B</i>	<i>C</i>
<i>D</i>		(2, 3)	(1, 2)	(0, 2)
<i>E</i>		(1, 2)	(2, 3)	(1, 2)
<i>F</i>		(1, 1)	(3, 0)	(0, 2)

Figure 1.12 — Forme normale du jeu de l'exemple 1.10

Dans ce jeu, aucun joueur n'a de stratégie strictement ou faiblement dominée. Pourtant, il y a un équilibre de Nash : le profil de stratégies *DA*.

Étudions à présent des concepts de solution pour des jeux coopératifs (notion de cœur, appelé aussi noyau ou core).

1.3.3 Cœur

Les jeux pour lesquels il est possible de calculer le cœur sont représentés Figure 1.13.

	Somme nulle			Somme non nulle		
	¬Coop.	Coop.		¬Coop.	Coop.	
		Util. transf.	Util. ¬transf.		Util. transf.	Util. ¬transf.
Statique		cœur	cœur		cœur	cœur
Dynamique						

Figure 1.13 — Taxonomie des jeux : Domaine d'application du cœur

Le cœur, ou noyau, d'un jeu est un concept de solution pour les jeux coopératifs. Informellement, une issue d'un jeu appartient au cœur de ce jeu si aucune coalition ne peut améliorer le paiement (utilité) de *tous* ses membres, et donc aucune coalition n'a intérêt à dévier (c'est-à-dire changer sa stratégie commune).

On peut introduire ici deux définitions du cœur d'un jeu coalitionnel, une pour les jeux à utilités transférables, et une pour les jeux dont les utilités ne sont pas transférables :

1.3.3.1 Utilités transférables

On note $(x_i)_{i \in N}$ le **profil de paiement (ou répartition)** d'un jeu coopératif, x_i représentant le paiement obtenu par le joueur i . On note $x(C) = \sum_{i \in C} x_i$ la somme des paiements des membres de la coalition C (voir par exemple [Osborne et Rubinstein, 1994]).

On dit qu'un profil de paiement $(x_i)_{i \in N}$ est **C-réalisable** si $x(C) = v(C)$ (rappelons que la fonction v représente la fonction caractéristique du jeu, voir définition 1.4 (page 13)). Il est **réalisable** s'il est N -réalisable.

Définition 1.10. Le **cœur (ou noyau)** d'un jeu coalitionnel à utilités transférables (N, v) est l'ensemble des répartitions $(x_i)_{i \in N}$ réalisables telles que $\forall C \subseteq N, x(C) \geq v(C)$, ou, de manière équivalente, telles qu'il n'existe pas de coalition C et de répartition C -réalisable $(y_i)_{i \in N}$ où $y_i > x_i$ pour tout $i \in C$.

Exemple 1.3 (page 14) – suite :

* Majorité simple. Le profil de paiement doit respecter les équations suivantes :

$$\left\{ \begin{array}{l} \forall i, x_i \geq 0 \\ x_1 + x_2 + x_3 \geq 1 \text{ car on a ainsi } x(C) = x_1 + x_2 + x_3 \geq v(C) \\ x_1 + x_2 \geq 1 \\ x_1 + x_3 \geq 1 \\ x_2 + x_3 \geq 1 \end{array} \right. \Rightarrow \text{Cœur} = \emptyset$$

* Unanimité. Le cœur réunit les profils de paiements tels que :

$$\text{Cœur} = \{(x_1, x_2, x_3) : x_1 + x_2 + x_3 = 1, \forall i, x_i \geq 0\}$$

* Le joueur 2 a un droit de veto. *Le profil de paiement doit respecter les équations suivantes :*

$$\begin{cases} \forall i, x_i \geq 0 \\ x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 \geq 1 \\ x_2 + x_3 \geq 1 \end{cases} \Rightarrow C_{\text{cœur}} = \{(0, 1, 0)\}$$

* Le joueur 2 est dictateur. *Le profil de paiement doit respecter les équations suivantes :*

$$\begin{cases} \forall i, x_i \geq 0 \\ x_1 + x_2 + x_3 \geq 1 \\ x_1 + x_2 \geq 1 \\ x_2 + x_3 \geq 1 \\ x_2 \geq 1 \end{cases} \Rightarrow C_{\text{cœur}} = \{(0, 1, 0)\}$$

1.3.3.2 Utilités non transférables

Il existe plusieurs façons de définir le noyau d'un jeu coopératif à utilités non transférables : la première définition est semblable à celle du cœur pour les jeux avec utilités transférables. La seconde est la suivante :

Un profil de stratégies s est dans le noyau d'un jeu coalitionnel si et seulement si il n'existe pas de coalition C telle que tous les membres de cette coalition ont une stratégie commune qui leur permet à *tous* d'obtenir une meilleure utilité qu'avec s (voir par exemple [Aumann, 1967; Owen, 1982; Myerson, 1991]).

Définition 1.11. *Le cœur (ou noyau) d'un jeu coalitionnel à utilités non transférables est l'ensemble des profils de stratégies $s = (s_1, \dots, s_n)$ tels qu'il n'existe pas de coalition $C \subset N$ et de $s_C \in S_C$ tels que $\forall i \in C, \forall s_{-C} \in S_{-C}, u_i(s_C, s_{-C}) > u_i(s)$.*

Exemple 1.11. *Soit le jeu G à trois joueurs représenté par la forme normale donnée Figure 1.14.*

		3 : E		3 : F	
		2	C	D	2
1	A	(3, 3, 3)	(4, 1, 0)	(3, 2, 1)	(1, 3, 2)
	B	(5, 0, 0)	(2, 5, 2)	(1, 0, 1)	(4, 2, 1)

Figure 1.14 — Forme normale d'un jeu coopératif à utilités non transférables

On a ici : $C_{\text{cœur}} = \{(ACE)\}$. En effet, il n'existe pas de coalition qui permette à tous les joueurs de cette coalition d'obtenir une meilleure utilité :

- * $C = \{1\}$. Si 1 dévie et joue B, il existe un $s_{-C} = DE$ tel que $u_1(ACE) > u_1(BDE)$.
- * $C = \{2\}$. Si 2 dévie et joue D, il existe un $s_{-C} = AE$ tel que $u_1(ACE) > u_1(ADE)$.
- * $C = \{3\}$. Si 3 dévie et joue F, il existe un $s_{-C} = AC$ tel que $u_1(ACE) > u_1(ACF)$.
- * On raisonne de même pour $C = \{1, 2\}$, $C = \{1, 3\}$, $C = \{2, 3\}$ et $C = \{1, 2, 3\}$.

1.3.4 Equilibres parfaits de Selten

Les jeux pour lesquels il est possible de calculer l'équilibre parfait de Selten sont représentés Figure 1.15.

	Somme nulle			Somme non nulle		
	¬Coop.	Coop.		¬Coop.	Coop.	
		Util. transf.	Util. ¬transf.		Util. transf.	Util. ¬transf.
Statique						
Dynamique	Selten	Selten	Selten	Selten	Selten	Selten

Figure 1.15 — Taxonomie des jeux : Domaine d'application de l'équilibre parfait de Selten

Exemple 1.12. Soit le jeu dynamique en information parfaite représenté sous forme extensive par la Figure 1.16.

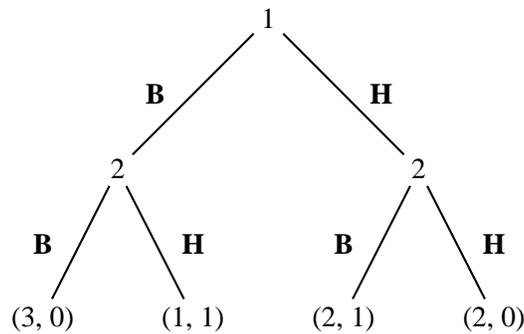


Figure 1.16 — Forme extensive d'un jeu dynamique en information parfaite

Le joueur 2 a deux stratégies possibles : H et B . Le joueur 2, qui joue après 1, a 2 stratégies possibles selon le choix de 1 : $s_2 = [s_2(H), s_2(B)]$.

Comment doit-on jouer ce jeu ? Le principe consiste à raisonner en remontant par **induction vers l'amont** (backward induction) : on raisonne en sens contraire de la manière dont le jeu va effectivement se dérouler.

Dans l'exemple 1.12, le joueur 1 se dit "si je joue H , alors 2 va jouer B de sorte que mon gain sera égal à 2 ; si je joue B , 2 jouera H et mon gain sera de 1". Dès lors, si 1 suppose que 2 choisira sa meilleure réponse à la seconde étape, il choisira H à la première. On admet que lorsque c'est au tour du joueur 2 de jouer, il choisit la meilleure réponse pour lui-même, à savoir $s_2(H) = B$ et $s_2(B) = H$. En conséquence, sachant que le joueur 2 se comporte de manière optimale à la seconde étape, le joueur 1 intègre cette information, et choisit une action optimale conditionnellement au comportement optimal de 2.

A chaque étape t , les stratégies du sous-jeu sont définies comme celles du jeu initial. La seule différence est que l'histoire à considérer pour les étapes précédentes est donnée par $h^t : s_i|h^t$ est la

restriction de s_i imposée par l'histoire h^t . On dit alors que $s = (s_1, s_2, \dots, s_n)$ est un **équilibre parfait de Selten** (ou subgame perfect Nash equilibrium, [Selten, 1965]) si, pour toute histoire h^t , la restriction $s|h^t$ est un équilibre de Nash du sous-jeu $G(h^t)$.

Exemple 1.13. Soit le jeu dynamique représenté sous forme extensive par la Figure 1.17.

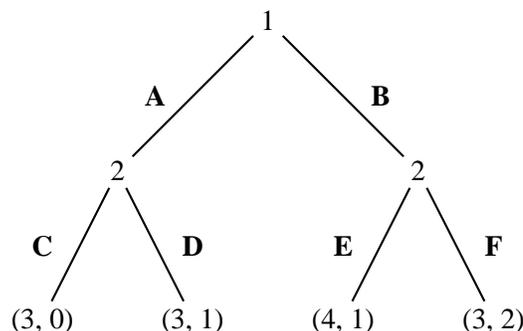
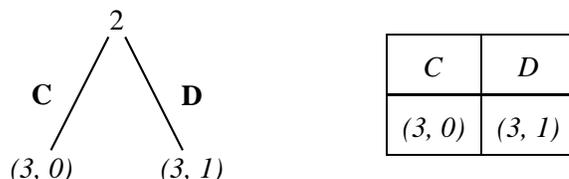


Figure 1.17 — Forme extensive d'un jeu dynamique

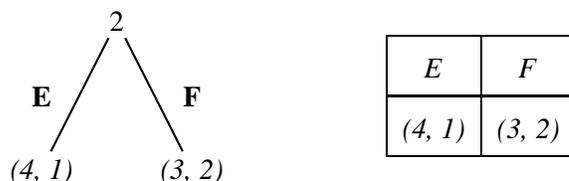
Ce jeu sous forme extensive a trois sous-jeux : le premier est le jeu lui-même, puisque tout jeu est un sous-jeu, le second commence après A, et le dernier après B.

La forme extensive du sous-jeu commençant après A, et sa forme normale associée sont les suivantes :



Ce sous-jeu a un équilibre de Nash : D, que l'on notera $D(A)$, car 2 jouera D si 1 joue A auparavant.

La forme extensive du sous-jeu commençant après B, et sa forme normale associée sont les suivantes :



Ce sous-jeu a un équilibre de Nash : F(B).

Dans le jeu complet, le joueur 1 a 2 stratégies possibles : A et B. Par contre, le joueur 2 en a 4 : $C(A)E(B)$ qui signifie que 2 joue C si 1 joue A, et E si 1 joue B, $C(A)F(B)$, $D(A)E(B)$ et $D(A)F(B)$.

La forme normale de ce jeu est :

1 \ 2	$C(A)E(B)$	$C(A)F(B)$	$D(A)E(B)$	$D(A)F(B)$
A	(3, 0)	(3, 0)	(3, 1)	(3, 1)
B	(4, 1)	(3, 2)	(4, 1)	(3, 2)

Ce jeu a 3 équilibres de Nash :

- * $(A, D(A)F(B))$ est un équilibre parfait de Selten : $D(A)$ et $F(B)$ sont des équilibres de Nash des sous-jeux associés.
- * $(B, C(A)F(B))$ n'est pas un équilibre parfait de Selten : $C(A)$ ne sera pas joué dans le sous jeu commençant après A , ce n'est donc pas un équilibre de Nash de ce sous-jeu.
- * $(B, D(A)F(B))$ est un équilibre parfait de Selten : $D(A)$ et $F(B)$ sont des équilibres de Nash des sous-jeux associés.

Ce jeu a donc 2 équilibres parfaits de Selten : $(A, D(A)F(B))$ et $(B, D(A)F(B))$.

1.3.5 Récapitulatif

Nous pouvons à présent récapituler les concepts de jeu que nous avons vu en fonction des types de jeux auxquels ils peuvent être appliqués dans le tableau représenté Figure 1.18.

	Somme nulle			Somme non nulle		
	\neg Coop.	Coop.		\neg Coop.	Coop.	
		Util. transf.	Util. \neg transf.		Util. transf.	Util. \neg transf.
Statique	Nash / Dom	cœur	cœur	Nash / Dom	cœur	cœur
Dynamique	Selten	Selten	Selten	Selten	Selten	Selten

Figure 1.18 — Taxonomie des jeux : Domaine d'application des concepts de solution

2

Jeux booléens

Comme nous l'avons vu dans la section 1.2 (page 16), les deux modes de représentation usuels des jeux (forme extensive et forme normale) coïncident pour les jeux statiques. Cette représentation ne fait pas l'économie de la description explicite de la fonction d'utilité de chaque agent. Or, cette description est de taille exponentielle en fonction du nombre d'agents : par exemple, si n agents ont chacun un choix entre deux actions possibles, il faudra spécifier $n \times 2^n$ valeurs numériques ; si deux agents contrôlent chacun un ensemble de p variables booléennes (il suffit de penser à de telles variables comme à des boutons que l'agent peut choisir d'enfoncer ou non), chaque agent a 2^p stratégies possibles et il faudra donc expliciter $2 \times (2^p)^2 = 2^{2p+1}$ valeurs numériques.

Cette explosion combinatoire est encore plus flagrante lorsqu'à la fois l'ensemble des agents et l'ensemble des stratégies pour chacun des agents sont de taille importante. Il devient alors déraisonnable de spécifier les fonctions d'utilité de manière explicite, en listant les valeurs pour chaque combinaison de stratégies. Il est tout aussi déraisonnable de penser pouvoir calculer des propriétés du jeu en appliquant un algorithme nécessitant une énumération explicite des combinaisons de stratégies. Pensons par exemple au calcul des équilibres de Nash en stratégies pures : ce calcul nécessite, dans le cas des jeux précédents, et dans le pire des cas, un temps de calcul de l'ordre de $n \times 2^n$ (pour le jeu à n joueurs avec 2 actions chacun) et de $2 \times 2^p \times 2^{2p} = 2^{3p+1}$ (pour le jeu à deux joueurs contrôlant chacun p variables booléennes).

D'un autre côté, une sous-branche de l'intelligence artificielle s'intéresse aux langages de représentation compacte de préférences (ordinales ou numériques). Ces langages permettent une représentation concise de relations de préférences, ou de fonctions d'utilité, sur un ensemble de conséquences qui possède une structure combinatoire (c'est-à-dire un produit cartésien de domaines de valeurs finis pour un ensemble fini de variables), en exploitant dans une large mesure des propriétés structurelles des relations de préférences (comme l'indépendance préférentielle entre variables). En particulier, lorsque les variables en jeu sont binaires, ces langages sont fondés sur la logique propositionnelle, dont ils héritent l'expressivité et les méthodes algorithmiques (pour la déduction et la recherche de modèles, notamment). L'expressivité et le pouvoir de concision des langages de représentation logique de préférences sont étudiés dans [Coste-Marquis *et al.*, 2004] et leur complexité algorithmique dans [Lang, 2004].

Partant de là, puisque la spécification d'un jeu statique nécessite la description des préférences des agents, il apparaît naturel de représenter de tels jeux en utilisant des langages de représentation compacte de préférences. Il existe déjà plusieurs cadres répondant aux problèmes que nous avons posés plus haut, notamment les langages graphiques que nous présenterons dans le chapitre 4 (page 121). Nous avons choisi ici d'étudier le cas où chaque agent contrôle un ensemble fini de variables *binaires* :

les *jeux booléens* [Harrenstein *et al.*, 2001; Harrenstein, 2004a]. Cela permet non seulement de simplifier les jeux, mais aussi d'utiliser un langage logique de représentation compacte des préférences.

Un jeu booléen est un jeu à deux joueurs et à somme nulle, la fonction d'utilité du joueur 1 (et donc celle du joueur 2 qui est son opposé) est représentée par une formule de la logique propositionnelle, appelée *forme booléenne* du jeu. Après avoir donné une description (simplifiée) des jeux booléens, nous montrerons que ces jeux booléens peuvent facilement être généralisés de manière à représenter des jeux avec un nombre arbitraire de joueurs et à somme non nulle, mais en gardant l'hypothèse que les préférences de chaque joueur sont représentées par une formule propositionnelle unique, ce qui ne permet de représenter que des utilités binaires. Nous verrons alors comment des outils simples issus de la logique propositionnelle permettent de caractériser certaines propriétés du jeu, et nous donnerons quelques résultats de complexité algorithmique¹.

2.1 Introduction aux jeux booléens

Dans cette section, nous allons faire un rapide état de l'art des jeux booléens tels qu'ils ont été introduits par Harrenstein, van der Hoek, Meyer et Witteveen dans [Harrenstein *et al.*, 2001; Harrenstein, 2004a].

Un jeu booléen sur un ensemble de variables propositionnelles V est un jeu à deux joueurs, 1 et 2, à somme nulle, ayant les spécificités suivantes :

- * les actions que peuvent entreprendre les deux joueurs consistent à donner une valeur de vérité à des variables de V ;
- * les fonctions d'utilité des deux joueurs sont représentées au moyen d'une formule propositionnelle φ formée sur les variables V , appelée **forme booléenne** du jeu.

φ représente le but du joueur 1 : l'utilité de celui-ci est $+1$ ² lorsque φ est satisfaite³ (et alors le joueur 1 gagne), et -1 sinon (et c'est alors le joueur 2 qui gagne). Les **utilités** sont donc **binaires** : il n'y en a que 2 possibles, 1 et -1 .

Le jeu étant à somme nulle, nous avons $u_2 = -u_1$ et le but du joueur 2 n'est autre que $\neg\varphi$. Le jeu n'a donc que deux issues possibles : la victoire de 1 ou celle de 2.

Pour construire ces jeux booléens, [Harrenstein *et al.*, 2001; Harrenstein, 2004a] commencent par définir deux jeux booléens atomiques, dénotés par **1** et **0**. Le premier est gagné par 1, tandis que le second est gagné par 2, sans qu'aucun des joueurs n'ait à prendre la moindre décision. Des jeux booléens plus complexes sont ensuite construits récursivement à partir de ces jeux atomiques et d'un ensemble de variables propositionnelles, que l'on appellera variables de décision binaires. Chaque variable de décision est contrôlée de manière exclusive par l'un des deux joueurs. Pour tous jeux booléens g_0 et g_1 , et pour toute variable de décision a , il existe un autre jeu booléen dénoté $a(g_0, g_1)$. Dans le jeu $a(g_0, g_1)$, c'est au joueur qui contrôle la variable a de décider de la valeur à laquelle il veut l'assigner. S'il choisit d'assigner a à vrai (resp. faux), alors le jeu continue avec g_1 (resp. g_0). Un jeu booléen est alors défini formellement comme suit :

¹Les résultats donnés dans ce chapitre ont fait l'objet de plusieurs publications : [Bonzon *et al.*, 2005, 2006b, 2007a].

²On notera alors $u_1 = 1$.

³Une formule booléenne est satisfaite si et seulement si la formule est vraie.

Définition 2.1. Soit A un ensemble fini de variables propositionnelles, et $\{0,1\}$ les deux joueurs. L'ensemble des **formes booléennes sur A** est le plus petit ensemble $B(A)$ tel que :

- * $\{0,1\} \subseteq B(A)$
- * si $a \in A$ et $g, h \in B(A)$, alors $a(g, h) \in B(A)$.

Un **jeu booléen sur A** est un couple (g, π) , où g est une forme booléenne, et π est une **fonction d'affectation de contrôle** ($\pi : A \rightarrow \{0,1\}$), qui associe à chaque variable le joueur qui la contrôle (chaque variable étant contrôlée par un et un seul joueur).

Exemple 2.1. Soit $V = \{a, b, c\}$ un ensemble de variables propositionnelles. Soit 1 et 2 deux joueurs ayant pour buts : $\varphi_1 = (a \leftrightarrow b) \vee (\neg a \wedge b \wedge \neg c)$, $\varphi_2 = \neg \varphi_1 = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b)$. Le joueur 1 contrôle les variables a et c , tandis que 2 contrôle la variable b .

La représentation proposée par Harrenstein dans [Harrenstein, 2004a] de ce jeu booléen est donnée en figure 2.1. Sur cette figure, la flèche gauche partant du nœud x représente la mise à faux de x , tandis que la flèche droite représente la mise à vrai de x .

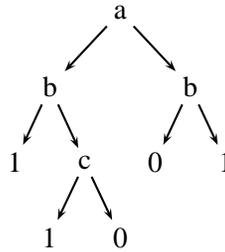


Figure 2.1 — Jeu booléen $(a, (b(1, c(1, 0)), b(0, 1)))$

Comme l'ont constaté Dunne et Van der Hoek [Dunne et van der Hoek, 2004], cette construction basée sur un modèle dynamique n'est pas nécessaire. En effet, l'hypothèse disant que les stratégies des agents sont choisies en parallèle (c'est-à-dire sans que l'un observe la décision de l'autre) est implicite. Cette forme extensive, sous forme d'arbre, est donc inutile. La représentation sous forme normale, qui correspond à celle d'un jeu statique, est ici plus simple, et a l'avantage de montrer clairement quel joueur gagnera à chaque issue du jeu.

Exemple 2.1 – suite La forme normale du jeu booléen $(a, (b(1, c(1, 0)), b(0, 1)))$ est représentée Figure 2.2 (page suivante).

[Harrenstein, 2004a, Section 8.4 (page 191)] introduit également des “jeux d'évaluation distribués” qui généralisent les jeux booléens en introduisant un nombre quelconque de joueurs et des préférences non binaires représentées par des ensembles de formules. Nous ne considérerons pas cette généralisation dans ce chapitre, mais elle sera étudiée plus en avant dans le chapitre 3 (page 69).

2.2 Jeux booléens à n joueurs : définitions et exemples

Avant de revenir plus en détail aux jeux booléens tels qu'ils ont été définis dans [Harrenstein *et al.*, 2001; Harrenstein, 2004a], nous allons d'abord les généraliser en nous intéressant à des jeux à n

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

Figure 2.2 — Forme normale du jeu booléen $(a, (b(\mathbf{1}, c(\mathbf{1}, \mathbf{0})), b(\mathbf{0}, \mathbf{1})))$

joueurs et à somme non nulle. Nous verrons ensuite que le cadre étudié par [Harrenstein *et al.*, 2001; Harrenstein, 2004a] est un cas particulier de ce cadre plus général.

Commençons par formaliser la notion de jeu booléen à n joueurs. Etant donné un ensemble de variables propositionnelles V , un jeu booléen sur V est un jeu à n joueurs pour lequel les actions disponibles de chaque joueur consistent à assigner une valeur de vérité à toutes les variables d'un sous-ensemble donné de V . Les préférences de chaque joueur i sont représentées par une formule propositionnelle φ_i formée sur les variables de V .

Définition 2.2. *Un jeu booléen à n joueurs est un 5-uplet $(N, V, \pi, \Gamma, \Phi)$, avec*

- * $N = \{1, 2, \dots, n\}$ l'ensemble des joueurs (appelés aussi agents) ;
- * V un ensemble de variables propositionnelles ;
- * $\pi : N \mapsto V$ une fonction d'affectation de contrôle ;
- * $\Gamma = \{\gamma_1, \dots, \gamma_n\}$ l'ensemble des contraintes, chaque γ_i étant une formule propositionnelle de $L_{\pi(i)}$ satisfiable ;
- * $\Phi = \{\varphi_1, \dots, \varphi_n\}$ l'ensemble des buts, chaque φ_i étant une formule de L_V satisfiable.

Un 4-uplet (N, V, π, Γ) , avec N, V, π, Γ définis comme ci-dessus est appelé un **pré-jeu booléen**.

La fonction d'affectation de contrôle π associe à chaque joueur les variables qu'il contrôle. On note π_i l'ensemble des variables contrôlées par le joueur i . Chaque variable est contrôlée par un et un seul joueur. Ainsi, $\{\pi_1, \dots, \pi_n\}$ forme une partition de V .

Chaque γ_i représente ici les contraintes d'un agent sur l'ensemble des variables qu'il contrôle. Ce choix de représentation est assez intuitif : en règle générale, les contraintes d'une personne reposent sur les actions qu'il lui est possible d'effectuer. Si une de mes contraintes pèse sur une action d'un autre joueur, je ne peux pas être assurée de la respecter, ce n'est pas de mon ressort. Ainsi, cette représentation des contraintes permet de respecter l'indépendance des agents : chaque agent gère ses variables, et les contraintes qui y pèsent, sans être tributaire d'un autre agent pour cela.

L'utilisation des jeux booléens permet d'avoir une représentation compacte du problème. Pour illustrer ce propos, nous allons utiliser une variante de l'exemple 1.1 (page 10) du dilemme des prisonniers : nous considérons ici n prisonniers qui ne peuvent bénéficier que d'un seul type de remise de peine afin de simplifier le problème.

Exemple 2.2. *Dans le jeu simplifié du prisonnier à n joueurs, n détenus (notés $1, \dots, n$) sont emprisonnés dans des cellules séparées. La police fait à chacun d'eux le même marché :*

"Tu as le choix entre couvrir tes complices en te taisant (noté T_i , $i = 1, \dots, n$) ou les trahir en les dénonçant (noté $\neg T_i$, $i = 1, \dots, n$). Si tu les dénonces, tu auras une remise de peine et tes partenaires purgeront le maximum (sauf si l'un d'eux t'a dénoncé aussi, auquel cas il bénéficiera comme toi d'une remise de peine). Mais si vous vous couvrez mutuellement, vous aurez tous une remise de peine⁴."

La représentation de ce jeu en forme normale pour $n = 3$ est la suivante :

3 : T_3			3 : $\neg T_3$		
1	2		1	2	
	T_2	$\neg T_2$		T_2	$\neg T_2$
T_1	(1, 1, 1)	(0, 1, 0)	T_1	(0, 0, 1)	(0, 1, 1)
$\neg T_1$	(1, 0, 0)	(1, 1, 0)	$\neg T_1$	(1, 0, 1)	(1, 1, 1)

Dans cette forme normale, les n -uplets (ici des triplets) donnent le résultat obtenu par les n joueurs dans l'ordre : (résultat joueur 1, résultat joueur 2, ...). Le 0 (resp. 1) signifie que le joueur concerné perd (resp. gagne).

On constate ici que pour n prisonniers, on aura une matrice à n dimensions, chaque dimension étant égale à 2, donc 2^n n -uplets à spécifier⁵.

Or, ce jeu peut être traduit très simplement par le jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$ suivant :

- * $N = \{1, 2, \dots, n\}$,
- * $V = \{T_1, \dots, T_n\}$,
- * $\forall i \in \{1, \dots, n\}, \pi_i = \{T_i\}$,
- * $\forall i \in \{1, \dots, n\}, \gamma_i = \top$, et
- * $\forall i \in \{1, \dots, n\}, \phi_i = (T_1 \wedge T_2 \wedge \dots \wedge T_n) \vee \neg T_i$.

L'utilisation des jeux booléens permet donc de réduire de manière très significative la taille de la représentation du problème.

Définition 2.3. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen. Une **stratégie** s_i pour un joueur i de G est une π_i -interprétation satisfaisant γ_i . L'**ensemble des stratégies du joueur i** est représenté par $S_i = \{s_i \in 2^{\pi_i} \mid s_i \models \gamma_i\}$.

Un **profil de stratégies** s pour G est un n -uplet $s = (s_1, \dots, s_n)$, avec pour tout i , $s_i \in S_i$. $S = S_1 \times \dots \times S_n$ est l'ensemble de tous les profils de stratégies.

En d'autres mots, une stratégie pour le joueur i est une affectation à vrai ou faux des variables qu'il contrôle. Pour chaque joueur i , γ_i est l'ensemble des contraintes restreignant l'ensemble des profils de stratégies possibles pour ce joueur.

Pour tout ensemble non vide de joueurs (appelé **coalition**) $I \subseteq N$, la projection de s sur I est définie par $s_I = (s_i)_{i \in I}$. Si $I = \{i\}$, la projection de s sur $\{i\}$ est dénotée par s_i au lieu de $s_{\{i\}}$.

⁴L'emploi de préférences binaires ne nous permet pas ici d'exprimer le fait qu'un prisonnier préfère la situation dans laquelle il dénonce ses complices tandis que les autres le couvrent à la situation dans laquelle tout le monde couvre tout le monde, elle-même préférée à la situation dans laquelle tout le monde dénonce. Pour faire cela, nous aurons besoin d'un langage plus sophistiqué, cf. Chapitre 3 (page 69).

⁵Ou un arbre binaire à 2^n feuilles si on utilise une représentation sous forme extensive.

Comme décrit en section 1.1.1.1 (page 10), les notations suivantes sont usuelles en théorie des jeux. Soit $s = (s_1, \dots, s_n)$ et $s' = (s'_1, \dots, s'_n)$ deux profils de stratégies.

On note s_{-i} le *profil de stratégies s privé de la stratégie du joueur i* : $s_{-i} = (s_1, s_2, \dots, s_{i-1}, s_{i+1}, \dots, s_n)$. On note (s_{-i}, s'_i) le *profil de stratégies s dans lequel on a remplacé la stratégie du joueur i par celle du profil s'* : $(s_{-i}, s'_i) = (s_1, s_2, \dots, s_{i-1}, s'_i, s_{i+1}, \dots, s_n)$. Similairement, si $I \subseteq N$, on note $s_{-I} = s_{N \setminus I}$.

π_I représente l'ensemble des variables contrôlées par I , et $\pi_{-I} = \pi_{N \setminus I}$. Si $I = \{i\}$, on note π_{-i} l'ensemble des variables contrôlées par tous les joueurs sauf le joueur i : $\pi_{-i} = V \setminus \pi_i$. $\{\pi_1, \dots, \pi_n\}$ formant une partition de V , un profil de stratégies s est une interprétation pour V , i.e. $s \in 2^V$. π^{-1} représente la fonction inverse de π .

L'ensemble des stratégies de $I \subseteq N$ est dénoté par $S_I = \times_{i \in I} S_i$, et l'ensemble des buts de $I \subseteq N$ par $\Phi_I = \bigwedge_{i \in I} \Phi_i$.

Etudions l'exemple suivant pour illustrer ces notions.

Exemple 2.3. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen, avec

- * $V = \{a, b, c\}$, $N = \{1, 2\}$,
- * $\pi_1 = \{a, b\}$ et $\pi_2 = \{c\}$.
- * $\gamma_1 = \neg a \vee \neg b$, $\gamma_2 = \top$
- * $\Phi_1 = (a \leftrightarrow b) \vee (\neg a \wedge b \wedge \neg c)$,
- * $\Phi_2 = (\neg a \wedge b \wedge c) \vee (a \wedge \neg b)$,

Le joueur 1 a trois stratégies possibles : $s_{11} = \bar{a}\bar{b}$, $s_{12} = \bar{a}b$, $s_{13} = \bar{a}\bar{c}$. La stratégie ab ne satisfait pas les contraintes du joueur 1, et donc n'est pas une stratégie acceptable.

Le joueur 2 a deux stratégies possibles : $s_{21} = c$ ou $s_{22} = \bar{c}$.

G possède donc 6 profils de stratégies : $S = \{\bar{a}\bar{b}c, \bar{a}\bar{b}\bar{c}, \bar{a}bc, \bar{a}b\bar{c}, \bar{a}\bar{b}\bar{c}, \bar{a}\bar{b}c\}$.

Les profils de stratégies $\bar{a}\bar{b}c$, $\bar{a}b\bar{c}$ et $\bar{a}\bar{b}\bar{c}$ donnent la victoire à 1, tandis que $\bar{a}bc$, $\bar{a}\bar{b}c$ et $\bar{a}bc$ permettent à 2 de gagner.

Le but Φ_i du joueur i est une relation de préférence dichotomique et compacte, correspondant donc à une fonction d'utilité binaire⁶ : un joueur i est satisfait, et a une utilité de 1, si et seulement si son but Φ_i est satisfait. Il a une utilité de 0 sinon. Les buts $\{\Phi_i, i = 1, \dots, n\}$ jouent donc le rôle des fonctions d'utilité.

Définition 2.4. Pour chaque joueur i , la **fonction d'utilité** induite par le but de ce joueur est la fonction $u_i : S \rightarrow \{0, 1\}$ telle que :

$$u_i(s) = \begin{cases} 0 & \text{si } s \models \neg \Phi_i \\ 1 & \text{si } s \models \Phi_i \end{cases}$$

Nous avons :

- * s est au moins aussi bon que s' pour i , dénoté par $s \succeq_i s'$, si $u_i(s) \geq u_i(s')$, ou de façon équivalente si $s \models \neg \Phi_i$ implique $s' \models \neg \Phi_i$;
- * s est strictement meilleur que s' pour i , dénoté par $s \succ_i s'$, si $u_i(s) > u_i(s')$, ou, de façon équivalente, $s \models \Phi_i$ et $s' \models \neg \Phi_i$.

⁶Nous verrons comment lever cette contrainte dans le chapitre 3 (page 69).

- * i est indifférent entre s et s' , dénoté par $s \sim_i s'$, si $s \geq_i s'$ et $s' \geq_i s$, ou de façon équivalente si $s \models \varphi_i$ si et seulement si $s' \models \varphi_i$.

Définissons à présent la notion de stratégie gagnante pour un joueur.

Définition 2.5. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen, avec $\Phi = \{\varphi_1, \dots, \varphi_n\}$, et $N = \{1, \dots, n\}$. La stratégie s_i est une **stratégie gagnante** pour le joueur i si, quels que soient les choix effectués par ses adversaires, i est sûr de gagner en choisissant cette stratégie.

$$\forall s_{-i} \in S_{-i}, (s_{-i}, s_i) \models \varphi_i$$

2.3 Graphe de dépendance entre les joueurs

De nombreuses structures graphiques sont cachées dans un jeu booléen : la satisfaction de chaque joueur i dépend des joueurs qui contrôlent des variables de φ_i . Il est donc possible de représenter graphiquement les dépendances entre joueurs en utilisant la nature syntactique des buts. Intuitivement, si le but d'un joueur i ne dépend d'aucune variable contrôlée par le joueur j , alors la satisfaction de i ne dépendra pas directement de j . Toutefois, ce n'est qu'une condition suffisante : il se peut que l'expression syntactique de φ_i contienne une variable contrôlée par j , mais que cette variable ne joue aucun rôle dans la satisfaction de φ_i , comme c'est le cas pour la variable y dans $\varphi_i = x \wedge (y \vee \neg y)$. Nous utiliserons donc une notion plus forte d'indépendance entre une formule et une variable que la simple notion d'indépendance syntactique [Lang et Marquis, 1998a; Lang *et al.*, 2003].

Définition 2.6. Une formule propositionnelle φ est **indépendante** d'une variable propositionnelle x s'il existe une formule ψ logiquement équivalente à φ et dans laquelle x n'apparaît pas⁷.

Définition 2.7. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen. L'ensemble des **variables utiles** pour un joueur i , dénoté par $RV_G(i)$, est l'ensemble de toutes les variables $v \in V$ telles que φ_i n'est pas indépendante de v .

Pour faciliter les notations, l'ensemble des variables utiles pour un joueur i dans un jeu booléen G sera noté RV_i au lieu de $RV_G(i)$. On peut à présent facilement définir l'ensemble des *joueurs utiles* pour un joueur i comme étant l'ensemble des joueurs contrôlant au moins une variable de RV_i .

Définition 2.8. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen. L'ensemble des **joueurs utiles** pour un joueur i , dénoté par⁸ RP_i , est l'ensemble des joueurs $j \in N$ tels que j contrôle au moins une variable utile de i : $RP_i = \bigcup_{v \in RV_i} \pi^{-1}(v)$.

Exemple 2.4. Axel, Kevin et Virginie sont invités à une fête. Axel veut y aller. Kevin aussi, mais si et seulement si Axel y va. Quant à Virginie, elle voudrait y aller, que Kevin y aille mais pas Axel. Cette situation peut être modélisée par le jeu booléen $G = (N, V, \Gamma, \pi, \Phi)$, défini par

- * $N = \{1, 2, 3\}$, Axel étant le joueur 1, Kevin le 2 et Virginie 3 ;

⁷Il existe une caractérisation sémantique équivalente de l'indépendance entre une formule et une variable [Lang *et al.*, 2003] : φ est indépendante de x s'il existe une interprétation s telle que $s \models \varphi$ et $\text{switch}(s, x) \models \varphi$, où $\text{switch}(s, x)$ est obtenue en changeant la valeur de x dans s , et en laissant les valeurs des autres variables inchangées.

⁸Comme précédemment, l'ensemble des joueurs utiles pour un joueur i dans un jeu booléen G devrait être noté $RP_G(i)$. Pour faciliter les notations nous écrirons simplement RP_i .

- * $V = \{a, b, c\}$, où a signifie “1 va à la fête”, et de même pour b et 2, et pour c et 3 ;
- * pour chaque i , $\gamma_i = \top$;
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$;
- * $\varphi_1 = a$,
- * $\varphi_2 = a \leftrightarrow b$ et
- * $\varphi_3 = \neg a \wedge b \wedge c$.

On peut voir que la satisfaction d’Axel ne dépend que de lui-même, celle de Kevin dépend de la décision d’Axel et de la sienne, tandis que celle de Virginia dépend des décisions de tous les joueurs.

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

Cette relation entre les joueurs peut être vue comme un graphe orienté contenant un nœud pour chaque joueur, et un arc de i à j si $j \in RP_i$, c’est-à-dire si j est un joueur utile pour i .

Définition 2.9. Le **graphe de dépendance d’un jeu booléen** $G = (N, V, \Gamma, \pi, \Phi)$ est un graphe orienté $\mathcal{P} = \langle N, R \rangle$, avec $\forall i, j \in N$, $(i, j) \in R$ (dénnoté par $R(i, j)$) si $j \in RP_i$.

$R(i)$ est donc l’ensemble des joueurs nécessaires à i pour satisfaire son but : $j \in R(i)$ si et seulement si $j \in RP_i$. Il faut pourtant remarquer que $j \in R(i)$ n’implique pas que i a besoin de j pour que son but soit satisfait. Par exemple, si $\pi_1 = \{a\}$, $\pi_2 = \{b\}$ et $\varphi_1 = a \vee b$, alors $2 \in R(1)$. Pourtant, 1 a une stratégie lui permettant de satisfaire son but (en mettant a à vrai) et n’a donc pas besoin de 2.

La fermeture transitive de R est notée R^* . $R^*(i, j)$ représente le chemin de i à j dans R . $R^*(i)$ représente donc tous les joueurs qui ont une influence directe ou indirecte sur i . $R^{*-1}(i)$ représente tous les joueurs sur lesquels i a une influence directe ou indirecte.

Exemple 2.4 (page précédente), suite : Le graphe de dépendance \mathcal{P} induit par G est représenté figure 2.3.

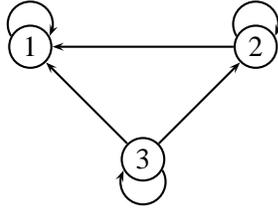


Figure 2.3 — Graphe de dépendance du jeu de la fête

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

Propriété 2.1. Tout graphe de dépendance $\mathcal{P} = \langle N, R \rangle$ représente au moins un jeu booléen $G = (N, V, \Gamma, \pi, \Phi)$.

Preuve :

Soit $\mathcal{P} = \langle N, R \rangle$ un graphe de dépendance. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen, avec $V = \{v_1, \dots, v_n\}$, $\forall i \in N$, $\pi_i = \{v_i\}$, et pour tout i , $\gamma_i = \top$.

Les buts de G sont construits comme suit : $\forall i \in N$, $\forall j$ tel que $j \in R(i)$, alors $\varphi_i =$

$\bigwedge_j v_j$. Si $\nexists j$ tel que $j \in R(i)$, alors $\varphi_i = \top$.
Le jeu construit de cette façon est un jeu booléen.

■

On introduit à présent la notion d'ensemble stable afin d'exploiter au mieux ce graphe de dépendance. Un ensemble de nœuds B est stable pour une relation R si tous les arcs de R partant d'un nœud de B atteignent un autre nœud de B . L'ensemble des joueurs utiles d'un ensemble stable B sont les joueurs dans B .

Définition 2.10. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen. $B \subseteq N$ est **stable**⁹ pour R si et seulement si $R(B) \subseteq B$, c'est-à-dire $\forall j \in B, \forall i$ tels que $i \in R(j)$, alors $i \in B$.

On remarque facilement que \emptyset et N sont des ensembles stables, et que l'ensemble des ensembles stables pour un jeu booléen est fermé pour l'union et l'intersection. Ces quatre propriétés caractérisent complètement l'ensemble des coalitions stables pour un jeu booléen.

Propriété 2.2. Soit $C \subset 2^N$. Il existe un jeu booléen G tel que C est l'ensemble des ensembles stables de G si et seulement si C satisfait les quatre propriétés suivantes :

1. $\emptyset \in C$;
2. $N \in C$;
3. Si $B, B' \in C$ alors $B \cup B' \in C$;
4. Si $B, B' \in C$ alors $B \cap B' \in C$.

Preuve : Grâce à la propriété 2.1 (page ci-contre), il suffit de montrer qu'il existe une relation R sur N telle que C est l'ensemble des ensembles stables pour R si et seulement si C satisfait (1)–(4).

\Rightarrow \emptyset et N sont évidemment stables pour R .

Si B et B' sont stables, alors $R(B) \subseteq B$ et $R(B') \subseteq B'$. Donc, $R(B) \cup R(B') \subseteq B \cup B'$. Par définition, $R(B) = \{j | \exists i \in B : j \in RP_i\}$ et $R(B') = \{j | \exists i \in B' : j \in RP_i\}$. Donc, $R(B) \cup R(B') = \{j | \exists i \in B \cup B' : j \in RP_i\} = R(B \cup B')$. Alors, $R(B \cup B') \subseteq B \cup B'$, et $B \cup B'$ est un ensemble stable.

Si B et B' sont stables, alors $R(B) \subseteq B$ et $R(B') \subseteq B'$. Donc, $R(B) \cap R(B') \subseteq B \cap B'$. Par définition, $R(B) = \{j | \exists i \in B : j \in RP_i\}$ et $R(B') = \{j | \exists i \in B' : j \in RP_i\}$. Donc, $R(B) \cap R(B') = \{j | \exists i \in B \cap B' : j \in RP_i\} = R(B \cap B')$. Alors, $R(B \cap B') \subseteq B \cap B'$, et $B \cap B'$ est un ensemble stable.

\Leftarrow Soit C un ensemble de coalitions satisfaisant (1)–(4). Pour tout $i \in N$, soit X_i le plus petit ensemble de C contenant i : $X_i = \bigcap \{B \in C | i \in B\}$ (puisque C est fermé pour \cap , nous avons $X_i \in C$). Construisons à présent R tel que pour tout $i, j \in N$, $(i, j) \in R$ si et seulement si $j \in X_i$.

On veut à présent vérifier que pour tout $B \in C$, B est stable pour R . Supposons que $B \in C$ et que B ne soit pas stable (il existe un $i \in B$ tel qu'il existe un

⁹La notion d'ensemble stable que nous définissons ici n'est pas la même que celle utilisée en théorie de graphe : nous ne parlons pas ici des ensembles de sommets deux-à-deux non adjacents.

$j \in R(i)$ et $j \notin B$. Comme $j \in R(i)$, on sait par construction de R que $j \in X_i$, et donc $j \in \bigcap \{B \in \mathcal{C} \mid i \in B\}$, donc $j \in B$, et donc B est stable pour R , ce qui est en contradiction avec l'hypothèse effectuée.

On montre ensuite que si $B \notin \mathcal{C}$, alors B n'est pas stable pour R . Puisque \mathcal{C} est fermé pour \cup , $\bigcup_{i \in B} X_i \neq B$. Donc il existe un $i \in B$ tel qu'il existe $k \in X_i$ et que $k \notin B$. Par construction de R , $R(B) \not\subseteq B$, et B n'est pas stable pour R .

■

On définit à présent la projection d'un jeu booléen G sur un ensemble stable de joueurs $B \subseteq N$, afin de pouvoir décomposer un jeu booléen en plusieurs jeux :

Définition 2.11. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen, et $B \subseteq N$ un ensemble stable pour R . La **projection** de G sur B est définie par $G_B = (B, V_B, \pi_B, \Gamma_B, \Phi_B)$, où $V_B = \bigcup_{i \in B} \pi_i$, $\pi_B : B \rightarrow V_B$ telle que $\pi_B(i) = \{v \mid v \in \pi_i\}$, $\Gamma_B = \{\gamma_i \mid i \in B\}$, et $\Phi_B = \{\varphi_i \mid i \in B\}$.

Propriété 2.3. Si B est un ensemble stable, $G_B = (B, V_B, \pi_B, \Gamma_B, \Phi_B)$ est un jeu booléen.

Preuve : Soit $G_B = (B, V_B, \pi_B, \Gamma_B, \Phi_B)$.

- * $B \subseteq N$ est l'ensemble des joueurs,
- * $V_B = \bigcup_{i \in B} \pi_i$. Nous avons évidemment $V_B \subseteq V$, et donc V_B est l'ensemble des variables.
- * Nous devons vérifier que chaque but φ_i pour $i \in B$ est une formule de L_{V_B} , ou est logiquement équivalente à une formule de L_{V_B} .
Supposons que $\exists i \in B, \exists v \in \text{Var}(\varphi_i)$ tel que $v \notin V_B$. Donc, $\forall j \in B, v \notin \pi_j$. Soit $k \in N \setminus B$ tel que $v \in \pi_k$. Nous savons que $v \in \text{Var}(\varphi_i)$, donc soit φ_i est indépendante de v , et est donc logiquement équivalente à une formule dans laquelle v n'apparaît pas, soit φ_i n'est pas indépendante de v , et dans ce cas $v \in RV_i$, et par définition $k \in RP_i$. Donc, $k \in R(i)$, mais $k \notin B$: c'est en contradiction avec le fait que B est stable.
- * $\pi_B : B \rightarrow V_B$ tel que $\pi_B(i) = \{v \mid v \in \pi_i\}$ est l'ensemble des fonctions d'affectation de contrôle des joueurs de B .
- * $\Gamma_B = \{\gamma_i \mid i \in B\}$ est l'ensemble des contraintes des joueurs de B . Comme chaque γ_i est une formule propositionnelle de L_{π_i} , Γ_B est bien défini pour le jeu G_B .

■

Exemple 2.5. Soit $G = (N, V, \pi, \Gamma, \Phi)$ le jeu booléen défini par $V = \{a, b, c\}$, $N = \{1, 2, 3\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, pour chaque i , $\gamma_i = \top$, $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = a \leftrightarrow \neg b$ et $\varphi_3 = \neg c$.

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

Le graphe de dépendance \mathcal{P} de G est représenté figure 2.4 (page ci-contre).

Les ensembles de joueurs $B = \{1, 2\}$ et $C = \{3\}$ sont stables. On peut décomposer G en 2 jeux booléens :

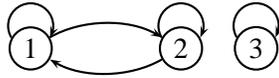


Figure 2.4 — Graphe de dépendance de l'exemple 2.5

- * $G_B = (B, V_B, \pi_B, \Gamma_B, \Phi_B)$, avec $B = \{1, 2\}$, $V_B = \{a, b\}$, $\pi_1 = a$, $\pi_2 = b$, $\gamma_1 = \gamma_2 = \top$, $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = a \leftrightarrow \neg b$.
- * $G_C = (C, V_C, \pi_C, \Gamma_C, \Phi_C)$, avec $C = \{3\}$, $V_C = \{c\}$, $\pi_3 = c$, $\gamma_3 = \top$, $\varphi_3 = \neg c$.

Cette décomposition nous permet ainsi d'étudier deux jeux plus simples, G_B et G_C , plutôt que d'étudier le jeu G . Nous verrons dans la suite que nous pouvons ainsi calculer les équilibres de Nash sur les sous-jeux plutôt que sur le jeu complet.

2.4 Concepts de solution : équilibres de Nash et stratégies dominées¹⁰

2.4.1 Équilibres de Nash

La définition des équilibres de Nash en stratégie pures (PNE) pour les jeux booléens à n joueurs est la même que la définition classique en théorie des jeux (voir Chapitre 1, Définition 1.5 (page 19)), en ayant à l'esprit que les fonctions d'utilité sont induites par les buts des joueurs $\varphi_1, \dots, \varphi_n$.

Rappelons qu'un équilibre de Nash est un profil de stratégies tel que la stratégie de chaque joueur est une réponse optimale aux stratégies des autres joueurs, et étudions un exemple simple.

Exemple 2.6. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen avec

- * $V = \{a, b, c\}$, $N = \{1, 2, 3\}$, pour tout i , $\gamma_i = \top$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$,
- * $\varphi_1 = (\neg a \vee (a \wedge b \wedge \neg c))$,
- * $\varphi_2 = (a \leftrightarrow (b \leftrightarrow c))$ et
- * $\varphi_3 = ((a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge c))$.

On peut à présent construire la forme normale de G :

		stratégie de 3 : c		stratégie de 3 : \bar{c}	
		2	\bar{b}	b	2
1	\bar{a}	$(1, 1, 0)$	$(1, 0, 1)$	$(1, 0, 0)$	$(1, 1, 0)$
	a	$(0, 0, 0)$	$(0, 1, 0)$	$(0, 1, 1)$	$(1, 0, 0)$

On constate tout d'abord que le joueur 1 a une stratégie gagnante. En effet, s'il choisit d'instancier la variable a à faux, alors il est sûr de gagner quels que soient les choix de 2 et 3.

¹⁰Certains des résultats de cette section ont été obtenus en collaboration avec Bruno Zanuttini.

Étudions à présent les équilibres de Nash. Pour cela, il faut étudier chaque profil de stratégies.

abc : 1, qui contrôle a , préfère $\bar{a}bc$

$a\bar{b}c$: 2, qui contrôle b , préfère $a\bar{b}c$

$\bar{a}bc$: 2, qui contrôle b , préfère abc

$\bar{a}\bar{b}c$: 2, qui contrôle b , préfère $\bar{a}\bar{b}c$

$ab\bar{c}$: 1, qui contrôle a , préfère $\bar{a}\bar{b}c$

$\bar{a}b\bar{c}$: 3, qui contrôle c , préfère $\bar{a}bc$

$\bar{a}\bar{b}\bar{c}$: Équilibre de Nash (aucun joueur ne préfère un autre profil)

$\bar{a}b\bar{c}$: 2, qui contrôle b , préfère $\bar{a}\bar{b}c$

Le profil de stratégies $\bar{a}\bar{b}\bar{c}$ est donc le seul équilibre de Nash du jeu G .

Donnons à présent une caractérisation simple des équilibres de Nash aux stratégies pures.

Propriété 2.4. Soit $s \in S$. s est un équilibre de Nash en stratégies pures de G si et seulement si pour tout $i \in N$, on a :

- * soit $s \models \varphi_i$,
- * soit $s_{-i} \models \neg\varphi_i$.

Cette caractérisation, simple, permet de calculer plus facilement les équilibres de Nash en stratégies pures des jeux booléens. Ainsi, un profil de stratégies s sera un PNE si et seulement si, pour chacun des joueurs i , soit s permet d'obtenir le but du joueur i , soit le joueur i ne peut obtenir son but si les autres joueurs maintiennent leurs stratégies.

Preuve : s est un équilibre de Nash en stratégies pures de G si et seulement si pour tout $i \in N$ et pour tout $s'_i \in S_i$ on a $u_i(s) \geq u_i(s_{-i}, s'_i)$. Or, puisque $u_i(s)$ et $u_i(s_{-i}, s'_i)$ ne peuvent prendre que deux valeurs, cette inégalité est équivalente à

$$\forall i \in N \text{ et } \forall s'_i \in S_i, u_i(s) = 1 \text{ ou } u_i(s_{-i}, s'_i) = 0$$

c'est-à-dire

$$\forall i \in N \text{ soit } u_i(s) = 1 \tag{2.1}$$

$$\text{soit } \forall s'_i \in S_i, u_i(s_{-i}, s'_i) = 0 \tag{2.2}$$

2.1 est équivalente à $s \models \varphi_i$.

2.2 est équivalente à $\forall s'_i \in S_i, (s_{-i}, s'_i) \models \neg\varphi_i$, donc $s_{-i} \models \neg\varphi_i$.

■

Exemple 2.6 (page précédente), suite : Nous avons vu que le seul équilibre de Nash de ce jeu est $s = \bar{a}\bar{b}\bar{c}$. En effet, nous avons :

- * $s = \bar{a}\bar{b}c \models \varphi_1 = (\neg a \vee (a \wedge b \wedge \neg c))$
- * $s = \bar{a}b\bar{c} \models \varphi_2 = (a \leftrightarrow (b \leftrightarrow c))$
- * $s_{-3} = \bar{a}\bar{b} \models \neg\varphi_3 = ((\neg a \vee b \vee c) \wedge (a \vee \neg b \vee \neg c))$

On peut vérifier qu'aucun autre profil de stratégies ne satisfait l'une ou l'autre de ces conditions.

Cette caractérisation des équilibres de Nash en stratégies pures peut être encore simplifiée. En effet, vérifier que $s_{-i} \models \neg\varphi_i$ revient à vérifier que s_{-i} peut inférer $\neg\varphi_i$ *quelles que soient* les valeurs prises par les variables contrôlées par le joueur i . Cela évoque la notion de projection d'une formule sur un ensemble de variables, qui correspond à l'utilisation de l'opérateur *forget*, notion qui a été étudiée par [Lin et Reiter, 1994; Lang *et al.*, 2002a, 2003], et qui est rappelée dans le chapitre de notations (page 7).

On généralise ici cette notion d'oubli en utilisant la notation suivante : $\forall i : \varphi \equiv \neg\exists i : \neg\varphi$ qui représente l'oubli de toutes les variables contrôlées par le joueur i .

Si l'on revient à la seconde équation de la propriété 2.4 (page précédente), on voit que l'on a :

$$s_{-i} \models \neg\varphi_i \Leftrightarrow s_{-i} \models \forall i : \neg\varphi_i \quad (2.3)$$

$$\Leftrightarrow (s_i, s_{-i}) \models \forall i : \neg\varphi_i \quad (2.4)$$

$$\Leftrightarrow s \models \forall i : \neg\varphi_i \quad (2.5)$$

L'équation 2.3 est équivalente à l'équation 2.4 car les variables contrôlées par le joueur i ont disparu de $\forall i : \neg\varphi_i$.

Cela nous permet d'établir le corollaire suivant, qui simplifie la caractérisation des PNE que nous avons donnée dans la propriété 2.4 (page ci-contre) :

Corollaire 2.1. *Soit $s \in S$. Alors s est un équilibre de Nash en stratégies pures pour G si et seulement si :*

$$s \models \bigwedge_{i \in N} (\varphi_i \vee (\forall i : \neg\varphi_i))$$

Preuve :

$$\begin{aligned} & \forall i \in N, s \models \varphi_i \text{ ou } s_{-i} \models \neg\varphi_i \\ \Leftrightarrow & \forall i \in N, s \models \varphi_i \text{ ou } s \models \forall i : \neg\varphi_i \\ \Leftrightarrow & \forall i \in N, s \models \varphi_i \vee (\forall i : \neg\varphi_i) \\ \Leftrightarrow & s \models \bigwedge_{i \in N} (\varphi_i \vee (\forall i : \neg\varphi_i)) \end{aligned}$$

■

Exemple 2.6 (page 41), suite : *Comme précédemment, le seul équilibre de Nash de ce jeu est $\bar{a}\bar{b}c$. Calculons $\forall 3 : \neg\varphi_3$:*

$$\begin{aligned} \forall 3 : \neg\varphi_3 &= \neg\varphi_{3_{c \leftarrow \top}} \wedge \neg\varphi_{3_{c \leftarrow \perp}} \\ &= ((\neg a \vee b \vee \top) \wedge (a \vee \neg b \vee \perp)) \wedge ((\neg a \vee b \vee \perp) \wedge (a \vee \neg b \vee \top)) \\ &= (a \vee \neg b) \wedge (\neg a \vee b) \end{aligned}$$

On a donc bien :

$$\bar{a}\bar{b}c \models \varphi_1 \wedge \varphi_2 \wedge \forall 3 : \neg\varphi_3$$

Le graphe de dépendance entraîne également quelques propriétés sur les équilibres de Nash.

Propriété 2.5. Soit G un jeu booléen. Si $\forall i \in N, i \notin RP_i$ alors tout $s \in S$ est un PNE.

Si un joueur ne contrôle aucune des variables qui lui sont utiles, il n'a pas d'influence sur son propre but, et donc aucune préférence pour une ou l'autre de ses stratégies. Si c'est le cas pour tous les agents, aucun profil de stratégies n'est préféré, et tous sont donc des équilibres de Nash en stratégies pures.

Preuve : Puisque $\forall i \in N, i \notin RP_i$, aucun joueur i n'a d'influence sur son propre but :
 $\forall s_i \in S_i, (s_i, s_{-i}) \models \varphi_i$ ou $s_{-i} \models \neg \varphi_i$. Donc, $\forall i \in N$, soit $s_{-i} \models \varphi_i$, soit $s_{-i} \models \neg \varphi_i$.
 La propriété 2.4 (page 42) s'applique pour tout s , et s est donc un PNE.

■

Propriété 2.6. Soit G un jeu booléen tel que $\forall i \in N, |RP_i| = 1$, et $\exists i \in N$ tel que $RP_i = \{i\}$.

s est un PNE si et seulement si $\forall i \in N$ tels que $RP_i = \{i\}$, $s \models \varphi_i$.

Si chaque joueur d'un jeu booléen ne dépend que d'un seul joueur, et qu'au moins un joueur ne dépend que de lui-même, les joueurs tels que $RP_i = \{i\}$ seront les seuls à avoir une influence sur leurs buts, et donc à pouvoir s'assurer que ces derniers sont satisfaits. Un profil de stratégies s sera donc un PNE si et seulement s'il satisfait les buts de ces joueurs.

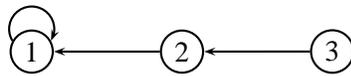
Preuve :
 \Rightarrow Supposons que s est un PNE et $\exists i$ tel que $RP_i = \{i\}$ et $s \models \neg \varphi_i$.
 Par définition d'un PNE, nous avons $s_{-i} \models \neg \varphi_i$. Mais nous savons que $RP_i = \{i\}$, donc $\forall v \in \text{Var}(s_{-i}), v \notin \text{Var}(\varphi_i)$. $s_{-i} \models \neg \varphi_i$ n'est alors pas possible. Nous avons une contradiction.
 \Leftarrow Supposons à présent que pour tout i tel que $RP_i = \{i\}$ alors $s \models \varphi_i$, et s n'est pas un PNE.
 Si s n'est pas un PNE, alors $\exists i \in N$ tel que $s \models \neg \varphi_i$ et $s_{-i} \models \varphi_i$. Sachant que $s \models \neg \varphi_i$, on a donc $RP_i \neq \{i\}$. $\forall v \in \pi_i, v \in \text{Var}(\varphi_i)$. Donc si $s \models \neg \varphi_i$, alors $s_{-i} \models \neg \varphi_i$ et s est un PNE ; nous avons une contradiction.

■

Exemple 2.7. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen défini par :

- * $V = \{a, b, c\}, N = \{1, 2, 3\}$,
- * pour tout $i, \gamma_i = \top$,
- * $\pi_1 = \{a\}, \pi_2 = \{b\}, \pi_3 = \{c\}$,
- * $\varphi_1 = a, \varphi_2 = \neg a$ et $\varphi_3 = b$.

Nous avons : $RV_1 = \{a\}, RV_2 = \{a\}, RV_3 = \{b\}, RP_1 = \{1\}, RP_2 = \{1\}$ et $RP_3 = \{2\}$.



Ce jeu a 4 PNEs : $\{abc, a\bar{b}c, ab\bar{c}, a\bar{b}\bar{c}\}$. s est un PNE si et seulement si $s \models \varphi_1$.

Propriété 2.7. Soit G un jeu booléen tel que les buts des joueurs soient tous des clauses. Alors G a toujours un équilibre de Nash en stratégies pures.

De plus, s est un PNE si et seulement si pour tout $i \in N$ tel que $Lit(\varphi_i) \cap \pi_i \neq \emptyset$, $s \models \varphi_i$.

Si les buts des joueurs sont tous des clauses, un joueur peut donc soit satisfaire son but seul, s'il contrôle une variable de son but, soit il n'a aucune influence sur sa satisfaction. Tout profil de stratégies qui satisfait tous les joueurs contrôlant une variable de leur but est donc un équilibre de Nash en stratégies pures.

Preuve : Soit $i \in N$. Si les buts de tous les joueurs sont des clauses, deux cas sont possibles :

- * $Lit(\varphi_i) \cap \pi_i \neq \emptyset$. Dans ce cas, i est sûr de pouvoir satisfaire son but, en affectant comme il le souhaite une des variables qu'il contrôle.
- * $Lit(\varphi_i) \cap \pi_i = \emptyset$. i ne peut rien faire pour satisfaire son but, il est totalement dépendant des autres joueurs.

Il existe donc $s \in S$ tel qu'on ait pour tout $i \in N$ soit $s \models \varphi_i$, soit $s_{-i} \models \neg \varphi_i$. s est donc un PNE, et il satisfait pour tout $i \in N$ tel que $Lit(\varphi_i) \cap \pi_i \neq \emptyset$, $s \models \varphi_i$.

Supposons qu'il existe $i \in N$ tel que $Lit(\varphi_i) \cap \pi_i \neq \emptyset$ et $s \models \neg \varphi_i$. Dans ce cas le joueur i a intérêt à changer sa stratégie pour satisfaire son but, s n'est donc pas un PNE.

■

Propriété 2.8. Soit G un jeu booléen tel que les buts des joueurs soient tous des cubes. G a toujours alors un équilibre de Nash en stratégies pures.

Si les buts des joueurs sont tous des cubes, un joueur peut instancier les variables qu'il contrôle de façon à satisfaire son but au mieux, et il n'a pas d'influence sur le reste. Le profil de stratégies construit de cette façon est un PNE.

Preuve : Soit $s \in S$ tel que pour tout $i \in N$, $s \models (\varphi_i)_{\pi_i}$. On a alors pour tout $i \in N$, soit $s \models \varphi_i$, soit $s_{-i} \models \neg \varphi_i$. En effet, si un joueur i a instancié toutes ses variables au mieux et que son but n'est quand même pas satisfait, c'est qu'un autre joueur contrôle une variable de son but, et l'instancie d'une façon non satisfaisante pour i .

■

Si pour un jeu G , la partie irréflexive d'un graphe de dépendance \mathcal{P} est acyclique (c'est-à-dire qu'il n'y a pas de cycles de longueur ≥ 2), nous pouvons utiliser une procédure inspirée de la "forward sweep procedure" (procédure de recherche en avant) [Boutilier *et al.*, 1999] pour trouver un équilibre de Nash en stratégies pures. Regardons comment procéder sur un exemple.

Exemple 2.4 (page 37), suite : La partie irréflexive du graphe de dépendance \mathcal{P} de G , représenté figure 2.3 (page 38), est acyclique.

$RP_1 = \{1\}$, et donc un profil de stratégies $s = (s_1, s_2, s_3)$ sera un PNE seulement si le but du joueur 1 est satisfait, et donc si $s_1 = a$.

2 peut alors choisir sa stratégie, car son but dépend seulement de 1 et de lui-même ($RP_2 = \{1, 2\}$).

Donc s sera un PNE seulement si $(s_1, s_2) \models (\varphi_2)_{s_1}$, c'est-à-dire si $s_2 = b$.

Enfin, le joueur 3 a fait le même raisonnement que 1 et 2, et donc sait que son but ne sera jamais satisfait quoi qu'il fasse.

Donc G a 2 PNEs : $\{abc, ab\bar{c}\}$.

Propriété 2.9. Soit G un jeu booléen tel que la partie irréflexive du graphe de dépendance \mathcal{P} de G soit acyclique. On sait alors que :

1. G a un PNE.
2. s est un PNE de G si et seulement si pour tout $i \in N$,

$$* \text{ soit } (s_{R^*(i) \setminus \{i\}}, s_i) \models \varphi_i,$$

$$* \text{ soit } s_{R^*(i) \setminus \{i\}} \models \neg \varphi_i.$$

Preuve :

1. On peut supposer sans perte de généralité que les joueurs sont numérotés de sorte que pour tout $i \in N$, si i dépend de j , alors $j < i$. Soit $s = (s_1, \dots, s_N)$ défini inductivement comme suit : pour $i = 1, \dots, N$, si $(\varphi_i)_{s_1, \dots, s_{i-1}}$ est satisfiable, alors on prend s_i tel que $(s_1, \dots, s_i) \models \varphi_i$. Un tel s_i existe car i ne dépend pas de k pour tout $k > i$. Si $(\varphi_i)_{s_1, \dots, s_{i-1}}$ n'est pas satisfiable, on prend alors un s_i quelconque.

On a donc construit un profil de stratégies s tel que pour tout i , $s \models \varphi_i$ ou $s_{-i} \models \neg \varphi_i$. D'après la propriété 2.4 (page 42), s est donc un PNE.

2. Il nous reste à prouver que s est un PNE si et seulement si $\forall i \in N$, soit $(s_{R^*(i) \setminus \{i\}}, s_i) \models \varphi_i$, soit $s_{R^*(i) \setminus \{i\}} \models \neg \varphi_i$.

\Rightarrow Supposons que s est un PNE et qu'il existe un joueur $i \in N$ tel que

$$(s_{R^*(i) \setminus \{i\}}, s_i) \not\models \varphi_i \text{ et } s_{R^*(i) \setminus \{i\}} \not\models \neg \varphi_i.$$

Nous avons alors $s \models \neg \varphi_i$, et, comme $\forall k \notin R^*(i)$, $(\varphi_i)_{s_k} = \varphi_i$, nous avons $s_{-i} \not\models \neg \varphi_i$. D'après la propriété 2.4 (page 42), s n'est pas un PNE.

\Leftarrow Supposons à présent que $(s_{R^*(i) \setminus \{i\}}, s_i) \models \varphi_i$ ou $s_{R^*(i) \setminus \{i\}} \models \neg \varphi_i$.

Si i est un joueur tel que $(s_{R^*(i) \setminus \{i\}}, s_i) \models \varphi_i$, on a évidemment $s \models \varphi_i$.

Si $s_{R^*(i) \setminus \{i\}} \models \neg \varphi_i$, alors, comme $\forall k \notin RP_i$, $(\varphi_i)_{s_k} = \varphi_i$, on a $s_{-i} \models \neg \varphi_i$.

Comme $\forall i \in N$, soit $s \models \varphi_i$, soit $s_{-i} \models \neg \varphi_i$, s est un PNE. ■

Exemple 2.4 (page 37), suite : Dans cet exemple, nous avons $R^*(1) = \{1\}$, $R^*(2) = \{1, 2\}$, $R^*(3) = \{1, 2, 3\}$. Pour $s = abc$, on a bien

$$* (s_{R^*(1) \setminus \{1\}}, s_1) = s_1 \models \varphi_1 = a$$

$$* (s_{R^*(2) \setminus \{2\}}, s_2) = (s_1, s_2) \models \varphi_2 = a \leftrightarrow b$$

$$* s_{R^*(3) \setminus \{3\}} = (s_1, s_2) \models \neg \varphi_3 = a \vee \neg b \vee \neg c$$

Le raisonnement est le même pour $s = ab\bar{c}$.

Quand la partie irréflexive du graphe de dépendance \mathcal{P} de G n'est pas acyclique, l'existence d'un équilibre de Nash en stratégies pures n'est plus garanti, comme le montre l'exemple suivant.

Exemple 2.8. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen défini par :

- * $V = \{a, b\}, N = \{1, 2\},$
- * $\pi_1 = \{a\}, \pi_2 = \{b\},$
- * $\forall i, \gamma_i = \top,$
- * $\varphi_1 = a \leftrightarrow b,$
- * $\varphi_2 = (a \wedge \neg b) \vee (\neg a \wedge b).$

On a $RV_1 = \{a, b\}, RV_2 = \{a, b\}, RP_1 = \{1, 2\}, RP_2 = \{1, 2\}.$

Le graphe de dépendance \mathcal{P} de G est le suivant :



Ce jeu n'a aucun PNE.

La propriété 2.9 (page ci-contre) entraîne le corollaire suivant :

Corollaire 2.2. Si G est un jeu booléen tel qu'il existe un $j \in N$ tel que pour tous les $i \in N, RP_i = \{j\},$ alors s est un PNE si et seulement si $s \models \varphi_j.$

Si tous les joueurs ne dépendent que d'un et un seul joueur $j,$ tous les joueurs (sauf j) n'ont aucune influence sur leurs buts. Seul $j,$ qui contrôle son but, a une préférence sur ses stratégies, et donc une influence sur le choix des PNEs.

Preuve : Soit G est un jeu booléen tel que $\exists j \in N$ tel que $\forall i \in N RP_i = \{j\}.$
 \Rightarrow Soit s un PNE de $G.$ Comme, $\forall i \in N, RP_i = \{j\},$ alors $\forall i \in N, R^*(i) = \{j\}.$
 D'après la propriété 2.9 (page précédente), on doit avoir $s_j \models \varphi_j,$ et pour tout $i \neq j,$ on peut avoir soit $(s_j, s_i) \models \varphi_i,$ soit $s_j \models \neg \varphi_i.$ La seule condition est donc que $s \models \varphi_j.$
 \Leftarrow Soit $s \models \varphi_j.$ Si $\forall i, RP_i = \{j\},$ on sait que i n'a aucun pouvoir sur son but, et donc que s est équivalent à $s_j.$ Donc, on a pour tout $i,$ soit $s \models \varphi_i,$ soit $s_{-i} \models \neg \varphi_i,$ et s est un PNE.

■

Propriété 2.10. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen, $B \subseteq N$ un ensemble stable pour $R,$ et $G_B = (B, V_B, \pi_B, \Gamma_B, \Phi_B)$ la projection de G sur $B.$

Si s est un PNE pour $G,$ alors s_B est un PNE pour $G_B.$

Preuve : Soit s un PNE de $G.$ Alors, $\forall i \in N,$ soit $s \models \varphi_i,$ soit $s_{-i} \models \neg \varphi_i.$
 Nous savons que $s = (s_B, s_{-B}),$ et nous voulons vérifier que s_B est un PNE de $G_B,$ c'est-à-dire que $\forall i \in B,$ soit $s_B \models \varphi_i,$ soit $s_{B-i} \models \neg \varphi_i.$ Soit $i \in B.$
 – $s = (s_B, s_{-B}) \models \varphi_i.$ Comme $i \in B,$ et que B est stable, nous savons que $\forall j \in R(i), j \in B.$ Donc tous les joueurs qui ont une influence sur i sont dans $B,$ et donc s_{-B} n'a aucune influence sur $\varphi_i :$ on a $s_B \models \varphi_i.$
 – $s_{-i} \models \neg \varphi_i.$ Comme précédemment, on sait que seuls les joueurs de B ont une influence sur $\varphi_i.$ On a donc $s_{(B \setminus \{i\})} \models \neg \varphi_i,$ ce qui correspond à $s_{B-i} \models \neg \varphi_i.$

■

Exemple 2.9. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen défini par

- * $V = \{a, b, c, d\}$, $N = \{1, 2, 3, 4\}$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\pi_4 = \{d\}$,
- * $\forall i, \gamma_i = \top$,
- * $\varphi_1 = a \leftrightarrow b$, $\varphi_2 = b \leftrightarrow c$, $\varphi_3 = \neg d$, et $\varphi_4 = d \leftrightarrow (b \wedge c)$.

On a : $RP_1 = \{1, 2\}$, $RP_2 = \{2, 3\}$, $RP_3 = \{4\}$, $RP_4 = \{2, 3, 4\}$.

Le graphe de dépendance \mathcal{P} de G est représenté figure 2.5.

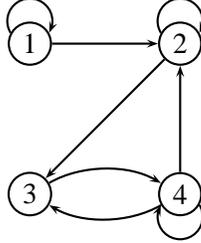


Figure 2.5 — Graphe de dépendance de l'exemple 2.9

L'ensemble de joueurs $B = \{2, 3, 4\}$ est stable. $G_B = (B, V_B, \Gamma_B, \pi_B, \Phi_B)$ est un jeu booléen, avec

- * $V_B = \{b, c, d\}$,
- * $\pi_2 = b$, $\pi_3 = c$, $\pi_4 = d$,
- * $\gamma_2 = \gamma_3 = \gamma_4 = \top$,
- * $\varphi_2 = b \leftrightarrow c$, $\varphi_3 = \neg d$, et $\varphi_4 = d \leftrightarrow (b \wedge c)$.

G a 2 PNEs : $\{abcd, \bar{a}\bar{b}\bar{c}\bar{d}\}$, et $\{bcd, \bar{b}\bar{c}\bar{d}\}$ sont les 2 PNEs de G_B (et dans cet exemple, G_B n'a aucun autre PNE).

Comme il est possible de le voir dans l'exemple 2.5 (page 40), la réciproque n'est pas toujours vraie : $C = \{3\}$ est stable, et le jeu booléen $G_C = (C, V_C, \pi_C, \Gamma_C, \Phi_C)$ a un équilibre de Nash en stratégies pures : $\{\bar{c}\}$, mais pourtant le jeu G n'a aucun PNE.

Toutefois, il existe des cas simples pour lesquels la réciproque est vraie.

Propriété 2.11. Soit B et C deux ensembles stables de joueurs, et soit G_B et G_C deux jeux booléens associés.

Si s_B est un PNE pour G_B et s_C un PNE pour G_C tels que¹¹ $\forall i \in B \cap C, s_{B,i} = s_{C,i}$, alors $s_{B \cup C}$ est un PNE pour le jeu $G_{B \cup C}$.

Preuve : B et C sont stables, donc d'après la propriété 2.2 (page 39), $B \cup C$ est un ensemble stable, et d'après la propriété 2.3 (page 40), $G_{B \cup C}$ est un jeu booléen.

Soit $i \in B \cup C$. 3 cas sont possibles (2 étant symétriques) :

- * $i \in B \setminus C$ (ou $i \in C \setminus B$). s_B est un PNE pour G_B , donc on a :
 - * soit $s_B \models \varphi_i$, et comme $i \notin C$, on a $(s_B, s_C) \models \varphi_i$;
 - * soit $s_{B-i} \models \neg \varphi_i$. Comme $i \notin C$, on a également $(s_{B-i}, s_C) \models \neg \varphi_i$, ce qui correspond à $s_{B \cup C - i} \models \neg \varphi_i$.

¹¹ $s_{B,i}$ représente la stratégie du joueur i pour le jeu G_B .

- * $i \in B \cap C$. On a alors $\forall k \in B \cap C, s_{B,k} = s_{C,k}$. Plusieurs cas sont possibles :
 - * $s_B \models \varphi_i$ et $s_C \models \varphi_i$. Puisque $\forall k \in B \cap C, s_{B,k} = s_{C,k}$, on a $(s_B, s_C) \models \varphi_i$, donc $s_{B \cup C} \models \varphi_i$.
 - * $s_{B-i} \models \neg \varphi_i$ et $s_{C-i} \models \neg \varphi_i$. Pour les mêmes raisons que précédemment, on a $(s_{B-i}, s_{C-i}) \models \neg \varphi_i$, et donc $s_{B \cup C - i} \models \neg \varphi_i$.
 - * $s_B \models \varphi_i$ et $s_{C-i} \models \neg \varphi_i$ (ou vice versa). On sait que $\forall k \in B \cap C, s_{B,k} = s_{C,k}$, et puisque les résultats obtenus par B et C sont différents, ces stratégies n'interviennent pas dans φ_i . Les joueurs dans $B \cap C$ ne contrôlent aucune variable de φ_i : $\forall v \in \text{Var}(\varphi_i), \forall k \in B \cap C, v \notin RV_k$. Dans ce cas, puisque $s_B \models \varphi_i$, on sait que toutes les variables de φ_i sont contrôlées par des joueurs dans $B \setminus C$; et puisque $s_{C-i} \models \neg \varphi_i$, on sait que toutes les variables de φ_i sont contrôlées par des joueurs dans $C \setminus B$. Comme chaque variable est contrôlée par un et un seul joueur, c'est impossible.

On sait donc que soit $s_{B \cup C} \models \varphi_i$, soit $s_{B \cup C - i} \models \neg \varphi_i$. $s_{B \cup C}$ est un équilibre de Nash de $G_{B \cup C}$.

■

Exemple 2.10. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen défini par

- * $V = \{a, b, c\}, N = \{1, 2, 3\}$,
- * $\pi_1 = \{a\}, \pi_2 = \{b\}, \pi_3 = \{c\}$,
- * $\forall i, \gamma_i = \top$,
- * $\varphi_1 = a \leftrightarrow c, \varphi_2 = b \leftrightarrow \neg c$, et $\varphi_3 = c$.

On a $RP_1 = \{1, 3\}, RP_2 = \{2, 3\}, RP_3 = \{3\}$. Le graphe de dépendance \mathcal{P} de G est représenté figure 2.6.

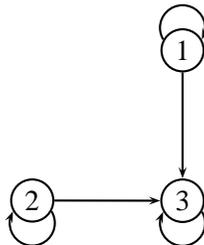


Figure 2.6 — Graphe de dépendance de l'exemple 2.10

L'ensemble des joueurs $A = \{1, 3\}$ et $B = \{2, 3\}$ sont stables. On a deux nouveaux jeux booléens :

- * $G_A = (A, V_A, \Gamma_A, \pi_A, \Phi_A)$, avec
 - * $A = \{1, 3\}, V_A = \{a, c\}$,
 - * $\pi_1 = a, \pi_3 = c$,
 - * $\gamma_1 = \gamma_3 = \top$,
 - * $\varphi_1 = a \leftrightarrow c$ et $\varphi_3 = c$.

G_A a un PNE : $\{ac\}$ (dénnoté par $s_A = (s_{A,1}, s_{A,3})$).

- * $G_B = (B, V_B, \pi_B, \Gamma_B, \Phi_B)$, avec
 - * $B = \{2, 3\}, V_B = \{b, c\}$,

- * $\pi_2 = b, \pi_3 = c,$
- * $\gamma_1 = \gamma_3 = \top,$
- * $\varphi_2 = b \leftrightarrow \neg c, \varphi_3 = c.$

G_B a un PNE : $\{\bar{b}c\}$ (dénnoté par $s_B = (s_{B,2}, s_{B,3})$).

$A \cap B = \{3\}$. Mais nous avons $s_{A,3} = s_{B,3} = c$: $G_{A \cup B}$ a un PNE : $\{a\bar{b}c\}$.

On peut facilement généraliser la propriété 2.11 (page 48), avec p ensembles stables couvrant l'ensemble de tous les joueurs :

Propriété 2.12. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen, et soit $B_1 \dots B_p$ p ensembles stables de joueurs, tels que $B_1 \cup \dots \cup B_p = N$. Soit G_{B_1}, \dots, G_{B_p} les p jeux booléens associés.

Si $\exists s_{B_1} \dots s_{B_p}$ des PNEs de G_{B_1}, \dots, G_{B_p} tels que $\forall i, j \in \{1, \dots, p\}, \forall k \in B_i \cap B_j, s_{B_i,k} = s_{B_j,k}$, alors $s = (s_{B_1}, \dots, s_{B_p})$ est un PNE de G .

Preuve : Soit B_1 et B_2 les deux premiers ensembles stables, et G_{B_1} et G_{B_2} les deux jeux booléens associés. On peut appliquer la propriété 2.11 (page 48), et montrer ainsi que $B_1 \cup B_2$ est un ensemble stable, que $G_{B_1 \cup B_2}$ est un jeu booléen, et que $s_{B_1 \cup B_2}$ est un PNE de $G_{B_1 \cup B_2}$.
On peut faire de même pour $B_1 \cup B_2$ et B_3 , et ainsi de suite, jusqu'à trouver le résultat voulu. ■

Comme montré dans l'exemple 2.10 (page précédente), diviser un jeu booléen permet de faciliter le calcul des équilibres de Nash.

Si on essaie de calculer les équilibres de Nash dans le jeu original, nous devons vérifier soit $s \models \varphi_i$, soit $s_{-i} \models \neg \varphi_i$ pour chacun des 8 profils de stratégies s , et pour chacun des 3 joueurs i . Il faut donc faire 12 vérifications par joueur (8 pour chaque profil de stratégies pour vérifier $s \models \varphi_i$, et 4 pour chaque s_{-i} pour vérifier $s_{-i} \models \neg \varphi_i$), et donc 36 pour le jeu dans le pire des cas.

Le calcul des équilibres de Nash une fois le jeu divisé est beaucoup plus facile : d'après la propriété 2.9 (page 46), le calcul du PNE de G_B nécessite 6 vérifications pour le joueur 1 (4 pour calculer $(s_1, s_3) \models \varphi_1$, et 2 pour calculer $s_3 \models \neg \varphi_1$); et seulement 2 pour le joueur 3 (car $R^*(3) \setminus \{3\} = \emptyset$). On a donc à faire seulement 8 vérifications dans le pire des cas pour trouver les PNEs de G_B , et il en est de même pour G_C , qui a une configuration équivalente. Comme il faut également vérifier que l'instanciation des variables du joueur 3 sont les mêmes pour les 2 jeux, il faut faire 17 vérifications pour calculer les PNEs du jeu G .

2.4.2 Stratégies dominées

Comme nous l'avons vu dans le chapitre 1, section 1.3.2 (page 21), il arrive parfois que le calcul des équilibres de Nash donne des résultats mal adaptés. Nous allons donc étudier ici le principe de *dominance*. Dans le cadre des jeux booléens, la définition des stratégies dominées est la même que dans la littérature (Définitions 1.8 et 1.9 (page 22)).

L'idée principale de ce concept est que les stratégies dominées ne sont pas utiles et peuvent être éliminées itérativement, jusqu'à ce qu'un point fixe soit atteint. Ce processus repose sur l'hypothèse que chaque joueur se comporte de manière rationnelle et sait que les autres joueurs sont rationnels.

Comme nous l'avons vu dans la section 1.3.2 (page 21), une stratégie strictement dominée ne peut jamais être présente dans un équilibre de Nash, tandis qu'une stratégie faiblement dominée peut apparaître dans un équilibre de Nash. Ce résultat reste valable dans le cas simple des jeux booléens.

Propriété 2.13.

- * Une stratégie strictement dominée n'est présente dans aucun équilibre de Nash.
- * Une stratégie faiblement dominée peut être présente dans un équilibre de Nash.

Preuve :

- * Montrons qu'une stratégie strictement dominée n'est présente dans aucun équilibre de Nash. Soit s_i une stratégie du joueur i strictement dominée. Alors

$$\exists s'_i \in S_i \text{ telle que } \forall s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) < u_i(s'_i, s_{-i})$$

Donc, pour tout profil de stratégies $s = \{s_1, \dots, s_i, \dots, s_n\}$, on aura : $u_i(s) < u_i(s'_i, s_{-i})$.

Or, s_i appartiendra à un équilibre de Nash si et seulement si il existe s tel que :

$$\forall s'_{-i} \in S_{-i}, u_i(s) > u_i(s'_{-i}, s_i)$$

ce qui est en contradiction avec l'hypothèse précédente. s_i ne peut donc pas appartenir à un équilibre de Nash.

- * Montrons à présent sur un contre-exemple qu'une stratégie faiblement dominée peut être présente dans un équilibre de Nash. Pour cela, étudions le jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$ avec $V = \{a, b\}$, $N = \{1, 2\}$, pour tout i , $\gamma_i = \top$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$ et $\phi_1 = \phi_2 = a \wedge \neg b$.

Ce jeu a deux équilibres de Nash : les profils de stratégies $a\bar{b}$ et $\bar{a}b$. Or, la stratégie $\neg a$, présente dans l'équilibre de Nash $\bar{a}b$, est faiblement dominée par la stratégie a .

■

La propriété suivante est également issue de la théorie classique des jeux.

Propriété 2.14.

- * L'ordre d'élimination des stratégies strictement dominées n'affecte pas le résultat final.
- * L'ordre d'élimination des stratégies faiblement dominées peut affecter le résultat final.

Le premier point de cette propriété, sur la dominance stricte, est évidemment applicable dans le cas simple des jeux booléens [Hillas et Kohlberg, 2002].

On aurait pu croire que l'ordre d'élimination des stratégies faiblement dominées dans le cas simple des jeux booléens n'affecte pas le résultat final, mais il n'en n'est rien. Nous en donnons ici un contre-exemple.

Exemple 2.11. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen avec

- * $V = \{a, b\}$, $N = \{1, 2\}$,
- * pour tout i , $\gamma_i = \top$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$,
- * $\varphi_1 = a \wedge b$ et
- * $\varphi_2 = a \wedge \neg b$

On peut à présent construire la forme normale de G :

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

Étudions les stratégies dominées de ce jeu :

- * Pour le joueur 1, la stratégie a domine faiblement la stratégie \bar{a} .
- * Pour le joueur 2, la stratégie \bar{b} domine faiblement la stratégie b .

Éliminons dans un premier temps la stratégie dominée du joueur 1. On élimine donc \bar{a} . Après cette élimination, la stratégie \bar{b} du joueur 2 domine faiblement la stratégie b . On obtient donc une seule stratégie résultat : $a\bar{b}$.

Éliminons à présent la stratégie dominée du joueur 2 en premier. On élimine donc la stratégie b . Après cette élimination, le joueur 1 n'a plus de stratégie dominée. On obtient donc deux stratégies résultat : $a\bar{b}$ et $a\bar{b}$.

On voit ici que nous n'avons pas obtenu le même résultat suivant l'ordre d'élimination des stratégies faiblement dominées.

Reprenons l'exemple 2.6 (page 41) pour calculer l'élimination des stratégies dominées après avoir calculé les équilibres de Nash.

Exemple 2.6 – suite Étudions les stratégies dominées de ce jeu :

- * Pour le joueur 1, la stratégie \bar{a} domine faiblement la stratégie a . Éliminons cette stratégie.
- * Une fois cette stratégie éliminée, la stratégie c du joueur 3 domine faiblement sa stratégie \bar{c} . On élimine donc cette dernière.
- * Enfin, pour le joueur 2, la stratégie \bar{b} domine faiblement la stratégie b . On élimine également cette stratégie.

On obtient ainsi un seul profil de stratégies résultat, $a\bar{b}c$ qui permet aux joueurs 1 et 2 de gagner.

Nous avons vu que l'ordre d'élimination des stratégies faiblement dominées a un impact sur le résultat. On est donc en droit de se demander si on aurait pu trouver un autre résultat en éliminant les stratégies des joueurs dans un ordre différent.

Ici, c'est le seul ordre d'élimination possible. En effet, le joueur 1 est le seul joueur qui a une stratégie dominante avant élimination d'autres stratégies. Et, une fois la stratégie dominée du joueur 1 éliminée, le joueur 3 est également le seul à avoir une stratégie dominante.

Une seconde propriété se dégage immédiatement :

Propriété 2.15. Dans un jeu booléen, si la stratégie s_i du joueur i domine strictement une autre stratégie s'_i , alors s_i est une stratégie gagnante pour ce joueur.

Preuve : s_i domine strictement s'_i . Donc,

$$\begin{aligned} & \forall s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) > u_i(s'_i, s_{-i}) \\ \Leftrightarrow & \forall s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) = 1 \text{ et } u_i(s'_i, s_{-i}) = 0 \\ \Rightarrow & \forall s_{-i} \in S_{-i}, (s_i, s_{-i}) \models \varphi_i \\ \Rightarrow & s_i \text{ est une stratégie gagnante pour le joueur } i. \end{aligned}$$

■

L'utilisation de la notion de projection (voir le chapitre Notations (page 7)) nous permet ici aussi de donner une caractérisation logique des stratégies strictement dominées.

Propriété 2.16. Soit G un jeu booléen. La stratégie s_i du joueur i domine strictement la stratégie s'_i si et seulement si :

$$s_i \models (\forall -i : \varphi_i) \text{ et } s'_i \models (\forall -i : \neg \varphi_i)$$

Preuve : Supposons que la stratégie s_i du joueur i domine strictement s'_i .

$$\begin{aligned} & \forall s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) > u_i(s'_i, s_{-i}) \\ \Leftrightarrow & \forall s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) = 1 \text{ et } u_i(s'_i, s_{-i}) = 0 \\ \Leftrightarrow & \forall s_{-i} \in S_{-i}, (s_i, s_{-i}) \models \varphi_i \text{ et } (s'_i, s_{-i}) \models \neg \varphi_i \\ \Leftrightarrow & s_i \models (\forall -i : \varphi_i) \text{ et } s'_i \models (\forall -i : \neg \varphi_i) \end{aligned}$$

■

Etudions cette caractérisation sur un exemple.

Exemple 2.12. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen. On donne

- * $V = \{a, b, c\}$, $N = \{1, 2\}$,
- * pour tout i , $\gamma_i = \top$,
- * $\pi_1 = \{a, b\}$, $\pi_2 = \{c\}$,
- * $\varphi_1 = (\neg a \wedge \neg b \wedge c) \vee (a \wedge b)$, et
- * $\varphi_2 = (a \wedge c) \vee (\neg b \wedge \neg c)$

On peut à présent construire la forme normale de G :

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

Étudions les stratégies dominées du joueur 1 en utilisant les caractérisations de la propriété 2.16. Pour cela, commençons par calculer les formules qui nous seront utiles.

- * $\varphi_1 = (\neg a \wedge \neg b \wedge c) \vee (a \wedge b)$
- * $\neg\varphi_1 = (a \vee b \vee \neg c) \wedge (\neg a \vee \neg b)$
- * $\forall - 1 : \varphi_1 = a \wedge b = \varphi'_1$
- * $\forall - 1 : \neg\varphi_1 = (a \vee b) \wedge (\neg a \vee \neg b) = \varphi''_1$

Nous devons ensuite étudier chaque stratégie du joueur 1 pour voir si elle permet d'inférer φ'_1 ou φ''_1 .

- * $\bar{a}\bar{b} \not\models \varphi'_1$ et $\bar{a}\bar{b} \not\models \varphi''_1$
- * $a\bar{b} \not\models \varphi'_1$ et $a\bar{b} \models \varphi''_1$
- * $\bar{a}b \not\models \varphi'_1$ et $\bar{a}b \models \varphi''_1$
- * $ab \models \varphi'_1$ et $ab \not\models \varphi''_1$

On peut à présent calculer les stratégies strictement dominées de 1.

ab domine strictement $a\bar{b}$ et $\bar{a}b$. En effet, on a bien :

- * pour $a\bar{b}$: $ab \models \varphi'_1$ et $a\bar{b} \models \varphi''_1$
- * pour $\bar{a}b$: $ab \models \varphi'_1$ et $\bar{a}b \models \varphi''_1$

Essayons à présent de caractériser les stratégies faiblement dominées. Cette fois ce n'est pas la notion de projection qui est utilisée mais celle d'interprétation partielle.

Propriété 2.17. Soit G un jeu booléen. La stratégie s_i du joueur i domine faiblement la stratégie s'_i si et seulement si :

$$(\varphi_i)_{s'_i} \models (\varphi_i)_{s_i} \text{ et } (\varphi_i)_{s_i} \not\models (\varphi_i)_{s'_i}$$

$(\varphi_i)_{s'_i} \models (\varphi_i)_{s_i}$ signifie que $\forall s_{-i}$, si l'instanciation partielle de φ_i par s'_i est satisfaite, alors il en est de même pour celle de φ_i par s_i . Donc, si $\forall s_{-i}$, (s'_i, s_{-i}) permet de satisfaire φ_i , alors (s_i, s_{-i}) également.

$(\varphi_i)_{s_i} \not\models (\varphi_i)_{s'_i}$ signifie que $\exists s_{-i}$ tel que (s_i, s_{-i}) satisfait φ_i , tandis que (s'_i, s_{-i}) non.

Preuve : La stratégie s_i du joueur i domine faiblement s'_i si et seulement si :

$$\forall s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) \geq u_i(s'_i, s_{-i}) \quad (2.6)$$

$$\text{et } \exists s_{-i} \in S_{-i}, u_i(s_i, s_{-i}) > u_i(s'_i, s_{-i}) \quad (2.7)$$

L'équation 2.6 nous permet d'obtenir les équivalences suivantes :

$$\begin{aligned} 2.6 &\Leftrightarrow \forall s_{-i} \in S_{-i}, (u_i(s_i, s_{-i}) = 1 \text{ ou } u_i(s'_i, s_{-i}) = 0) \\ &\Leftrightarrow \forall s_{-i} \in S_{-i}, (s_i, s_{-i}) \models \varphi_i \text{ ou } (s'_i, s_{-i}) \models \neg\varphi_i \\ &\Leftrightarrow \forall s_{-i} \in S_{-i}, \text{ si } (s'_i, s_{-i}) \models \varphi_i \text{ alors } (s_i, s_{-i}) \models \varphi_i \\ &\Leftrightarrow \forall s_{-i} \in S_{-i}, \text{ si } s_{-i} \models (\varphi_i)_{s'_i} \text{ alors } s_{-i} \models (\varphi_i)_{s_i} \\ &\Leftrightarrow (\varphi_i)_{s'_i} \models (\varphi_i)_{s_i} \end{aligned}$$

Ensuite, il est possible de remarquer que l'on a : $(2.7) \Leftrightarrow \neg(2.6)$ si l'on intervertit s_i et s'_i . Donc :

$$2.7 \Leftrightarrow (\varphi_i)_{s_i} \not\models (\varphi_i)_{s'_i}$$

■

2.5 Cas particulier : Jeux à deux joueurs et à somme nulle

Dans cette section, on montre que les jeux booléens étudiés dans [Harrenstein *et al.*, 2001; Harrenstein, 2004a] qui nous ont inspirés au départ, sont un cas particulier des jeux booléens à n joueurs présentés dans la section 2.2 (page 33). Les définitions sont les mêmes, mis à part la notion de coalition qui n'a aucun sens ici.

Certains paramètres sont simplifiés. En effet, comme chaque variable de décision de V ne peut être contrôlée que par un seul des deux joueurs, on a $\pi_2 = V \setminus \pi_1$. D'autre part, on a $\varphi_2 \equiv \neg\varphi_1$ et $\gamma_1 = \gamma_2 = \top$.

Étudions à présent un exemple simple.

Exemple 2.13. Soit $V = \{a, b, c, d\}$, $N = \{1, 2\}$, $\pi_1 = \{a, c\}$ et $\varphi_1 = (a \wedge \neg b) \vee (b \wedge d)$.

Le jeu booléen $G = (N, V, \pi_1, \Gamma, \Phi)$ est totalement défini.

En effet, on sait que $\pi_2 = V \setminus \pi_1$, que $\varphi_2 = \neg\varphi_1 = (\neg a \vee b) \wedge (\neg b \vee \neg d)$, et que $\gamma_1 = \gamma_2 = \top$.

On peut à présent construire la forme normale de G ¹² :

1 \ 2	$\bar{b}\bar{d}$	$b\bar{d}$	$\bar{b}d$	bd
$\bar{a}\bar{c}$	0	0	0	1
$a\bar{c}$	1	0	1	1
$\bar{a}c$	0	0	0	1
ac	1	0	1	1

On constate ici que le joueur 2, qui contrôle les variables b et d , a une stratégie gagnante. En effet, s'il choisit de mettre b à vrai et d à faux, il est sûr de gagner quels que soient les choix de 1.

La propriété suivante donne une caractérisation simple des équilibres de Nash en stratégies pures dans les jeux booléens à deux joueurs et à somme nulle qui est obtenue par une simple transposition de résultats issus de la théorie des jeux à somme nulle.

Propriété 2.18. Si G est un jeu booléen à deux joueurs et à somme nulle, $s = (s_1, s_2)$ est un équilibre de Nash en stratégies pures si et seulement si s_1 est une stratégie gagnante pour le joueur 1, ou s_2 est une stratégie gagnante pour le joueur 2.

Preuve :

1. Soit $s = (s_1, s_2)$ un équilibre de Nash en stratégies pures.

* Supposons que l'on a $u_1(s) = 1$. Le jeu étant à somme nulle, on a $u_2(s) = 0$. Comme s est un équilibre de Nash, $\forall s'_2, u_2(s) \geq u_2(s_1, s'_2)$ ce qui entraîne $\forall s'_2, u_2(s_1, s'_2) = 0$. Donc, $\forall s'_2, (s_1, s'_2) \models \neg\varphi_2$, ce qui implique que $\forall s'_2, (s_1, s'_2) \models \varphi_1$. s_1 est donc une stratégie gagnante pour le joueur 1.

¹²1 signifie que 1 gagne et 2 perd, et vice-versa pour 0

- * On a $u_1(s) = 0$ et $u_2(s) = 1$. En faisant le même raisonnement que précédemment, on montre que s_2 est donc une stratégie gagnante pour le joueur 2.
2. Supposons que s_1 est une stratégie gagnante pour le joueur 1. On a donc : $\forall s_2, u_1(s_1, s_2) = 1$ et $\forall s_2, u_2(s_1, s_2) = 0$. Posons $s = (s_1, s_2)$. On a bien $\forall s'_1, u_1(s) \geq u_1(s'_1, s_2)$ et $\forall s'_2, u_2(s) \geq u_2(s_1, s'_2)$. s est donc bien un équilibre de Nash.
- On raisonne de la même façon si l'on suppose que s_2 est une stratégie gagnante pour le joueur 2.

■

Par conséquent, dans un jeu booléen à deux joueurs et à somme nulle, il existe un équilibre de Nash en stratégies pures si et seulement si l'un des deux joueurs a une stratégie gagnante.

La propriété 2.18 (page précédente) permet de déterminer facilement la complexité algorithmique du problème d'existence d'un équilibre de Nash en stratégies pures. On rappelle que $\Sigma_2^P = \text{NP}^{\text{NP}}$ est la classe des langages reconnaissables en temps polynomial par une machine de Turing non-déterministe munie d'oracles NP (voir [Papadimitriou, 1994a]). Les résultats de complexité suivants ont été trouvés avec l'aide de Bruno Zanuttini.

Propriété 2.19. *Le problème de l'existence d'un équilibre de Nash en stratégies pures dans un jeu booléen à deux joueurs et à somme nulle est Σ_2^P -complet.*

Preuve : L'appartenance à Σ_2^P est immédiate. La difficulté est obtenue par la réduction du problème consistant à décider la validité de $\text{QBF}_{2,\exists}$. Etant donné une formule $Q = \exists A, \forall B \Phi$, où A et B sont des ensembles disjoints de variables et Φ est une formule propositionnelle de $L_{A \cup B}$, on définit le jeu booléen à 2 joueurs et à somme nulle suivant : $\varphi_1 = \Phi \vee (x \leftrightarrow y)$, où x, y sont des nouvelles variables (*i.e.* $x, y \notin A \cup B$), $\varphi_2 = \neg \varphi_1$, $\pi_1 = A \cup \{x\}$, $\pi_2 = B \cup \{y\}$ et $\gamma_1 = \gamma_2 = \top$. Ce jeu peut être évidemment construit en temps polynomial étant donné Q . D'après la propriété 2.18 (page précédente), ce jeu a un équilibre de Nash si et seulement si un des joueurs 1 ou 2 a une stratégie gagnante.

Supposons que Q est valide. Soit $M_A \in 2^A$ un modèle de Q . Alors, (M_A, x) et (M_A, \bar{x}) sont des stratégies gagnantes pour 1.

Supposons maintenant que Q n'est pas valide ; alors quel que soit $M_A \in 2^A$ que 1 choisit de jouer, 2 peut jouer $M_B \in 2^B$ tel que $(M_A, M_B) \not\models \Phi$. De plus, si 1 joue x (resp. \bar{x}), alors 2 peut jouer \bar{y} (resp. y), avec dans les deux cas la victoire de 2. D'autre part, 2 n'a pas non plus de stratégie gagnante puisque 1 peut toujours rendre $x \leftrightarrow y$ vrai, et donc gagner.

Il existe donc une stratégie gagnante pour 1 (ou 2, au choix) si et seulement si Q est valide.

■

On en tire le corollaire suivant, concernant cette fois les jeux booléens à n joueurs.

Corollaire 2.3. *Le problème de l'existence d'un équilibre de Nash en stratégies pures dans un jeu booléen à n joueurs est Σ_2^P -complet.*

Preuve : L'appartenance à Σ_2^P est immédiate ; la difficulté est obtenue, dès que $n = 2$ et que le jeu est à somme nulle par le théorème 2.19 (page précédente).

■

Ce résultat est à rapprocher de la complexité de la contrôlabilité – également un problème Σ_2^P -complet [Lang et Marquis, 1998b]. Or, le problème de l'existence d'un équilibre de Nash dans un jeu à plusieurs joueurs et à somme non nulle étant bien plus général que la contrôlabilité, le fait qu'il ne soit pas situé plus haut dans la hiérarchie des classes de complexité est plutôt une bonne nouvelle.

On peut expliquer intuitivement le fait que le problème de l'existence d'un équilibre de Nash soit au second niveau de la hiérarchie polynomiale par le fait que la résolution de ce problème comporte deux sources indépendantes de complexité NP-difficiles : la recherche du “bon” profil de stratégies, et la vérification qu'il constitue un équilibre de Nash en stratégies pures. Par comparaison, l'existence d'un profil de stratégies dont l'utilité cumulée est supérieure à une borne donnée est seulement un problème NP-complet.

Nous allons à présent essayer de simplifier le problème en étudiant les restrictions syntaxiques sur les formules représentant les buts des joueurs. Nous nous intéressons particulièrement aux formules en forme normale disjonctive (DNF). Rappelons que toute formule booléenne peut être mise en DNF, et que c'est donc une restriction syntaxique et non sémantique.

Tant que l'on considère des jeux à 2 joueurs et à somme nulle, et sachant que décider la validité de $\exists A, \forall B, \Phi$, une formule $\text{QBF}_{2,\exists}$, est Σ_2^P -complet même si Φ est en DNF, la propriété 2.19 (page ci-contre) reste correcte même si le but du joueur 1 est en DNF (le but du joueur 2 étant implicite).

Toutefois, lorsque nous représentons explicitement les buts de chaque joueur en DNF, la complexité du problème descend en NP, comme le montre la propriété suivante :

Propriété 2.20. *Soit G un jeu booléen. Si $\forall i \in N$, φ_i est en DNF, alors le problème de l'existence d'un équilibre de Nash en stratégie pure est NP-complet.*

Preuve :

Si φ_i est en DNF, alors $\exists i : \varphi_i$ peut être calculé en temps linéaire [Lang *et al.*, 2003, Propositions 17–18]. Alors, si chaque φ_i est en DNF, la formule $\psi \equiv \bigwedge_i (\varphi_i \vee (\neg \exists i : \varphi_i))$ peut être calculée en temps linéaire. Comme la propriété 2.4 (page 42) permet de trouver un profil de stratégies s et de vérifier que $s \models \psi$, le problème est en NP.

La difficulté est obtenue par la réduction du complément du problème consistant à décider si une DNF $\Phi = \bigvee_{i=1}^k T_i$ est une tautologie, ce qui est un problème coNP-complet bien connu. Soit X l'ensemble des variables de Φ et soit $x, y \notin X$. On définit un jeu G à deux joueurs par :

- * $\varphi_1 = \bigvee_{i=1}^k (T_i \wedge x \wedge \neg y) \vee (T_i \wedge \neg x \wedge y)$,
- * $\varphi_2 = (x \wedge y) \vee (\neg x \wedge \neg y)$,
- * $\pi_1 = \{y\}$,
- * $\pi_2 = X \cup \{x\}$.

G peut être construit en temps linéaire et φ_1, φ_2 sont en DNF.

On a $\varphi_1 \equiv \Phi \wedge (x \neq y)$ et $\varphi_2 \equiv (x = y)$. Grâce à la propriété 2.4 (page 42) et à son corollaire 2.1 (page 43), nous savons que G a un équilibre de Nash si et seulement si $((\Phi \wedge (x \neq y)) \vee \neg\Phi) \wedge (x = y)$ est satisfiable.

En effet, comme y n'apparaît pas dans Φ , nous avons

$$\neg\exists y : (\Phi \wedge x \neq y) \equiv \neg(\Phi \wedge \exists y : x \neq y) \equiv \neg(\Phi \wedge \top) \equiv \neg\Phi$$

et nous avons aussi

$$\neg\exists X \cup \{x\} : (x = y) \equiv \perp$$

Comme $\Phi \wedge (x \neq y) \wedge (x = y)$ n'est pas satisfiable, le jeu G a un équilibre de Nash si et seulement si $\neg\Phi \wedge (x = y)$ est satisfiable, c'est-à-dire si et seulement si $\neg\Phi$ est satisfiable, étant donné que x et y n'apparaissent pas dans Φ .

Donc, G a un équilibre de Nash si et seulement si Φ n'est pas une tautologie. ■

Lorsqu'on considère uniquement des jeux à deux joueurs, la complexité de ce problème peut descendre à P pour quelques classes non triviales de formules : si les buts des joueurs sont représentés par des formules en DNF renommables en Horn, affines, 2CNF ou CNF monotones.

On ne développe pas ces démonstrations ici, car la résolution de la projection en est le résultat principal, et que les détails sont en grande partie les mêmes que dans [Zanuttini, 2003, Section 6].

Nous pouvons également déterminer la complexité algorithmique du problème de la dominance faible dans un jeu booléen.

Propriété 2.21. *Décider si une stratégie s'_i donnée est faiblement dominée est Σ_2^P -complet. La difficulté reste valable même si φ_i est restreint aux DNF.*

Preuve : L'appartenance à Σ_2^P est immédiate. La difficulté est obtenue cette fois encore par la réduction du problème consistant à décider la validité de $\text{QBF}_{2,\exists}$.

Soit $Q = \exists A \forall B \Phi$, et a, b deux nouvelles variables. On définit :

- * $\varphi_1 = (a \wedge \Phi) \vee (\neg a \wedge b)$, $\pi_1 = A \cup \{a\}$,
- * $\pi_2 = B \cup \{b\}$ (φ_2 est quelconque et n'est pas définie ici car sa valeur n'intervient pas dans la démonstration).

Soit M'_A une A -interprétation, et soit $s'_1 = (M'_A, \bar{a})$. On a $(\varphi_1)_{s'_1} \equiv (b)$.

Supposons que Q est valide, avec $M_A \in 2^A$ un modèle de A , et $s_1 = (M_A, a)$. s_1 est alors une stratégie gagnante pour 1, contrairement à s'_1 . Donc s_1 domine faiblement s'_1 .

Supposons à présent que Q n'est pas valide. Soit $M_A \in 2^A$, et $s_1 = (M_A, \bar{a})$. Alors, $(\varphi_1)_{s_1} \equiv (b) \equiv (\varphi_1)_{s'_1}$, donc, d'après la propriété 2.17 (page 54), s_1 ne domine pas faiblement s'_1 .

À présent, soit $s_1 = (M_A, a)$. Comme Q n'est pas valide, il existe $M_B \in 2^B$ tel que $(M_A, M_B) \not\models \Phi$. Donc $(M_B, b) \models (\varphi_1)_{s'_1}$ mais $(M_B, b) \not\models (\varphi_1)_{s_1}$, et d'après la propriété 2.17 (page 54), s_1 ne domine pas faiblement s'_1 .

Donc, s'_1 est faiblement dominée si et seulement si Q est valide.
 On peut remarquer que si Φ est en DNF, alors $\exists A, \forall B, \Phi$ est toujours Σ_2^P -complet et que φ_1 peut être transformée en DNF efficacement.

■

Le problème de l'existence d'un équilibre de Nash, et son calcul, dans un jeu donné est une question importante, qui a attiré un grand nombre de chercheurs (par exemple [Deng *et al.*, 2002; McKelvey et McLennan, 1996; Koller *et al.*, 1996]). La plupart de ces travaux ont consisté à étudier des problèmes concernant les équilibres de Nash en *stratégies mixtes*. Dans ce contexte, l'existence d'un tel équilibre est toujours garantie, mais le problème de savoir s'il est possible de le calculer en temps polynomial est toujours ouvert [Papadimitriou, 2001].

Les premiers résultats de complexité ont été donnés par [Gilboa et Zemel, 1989], qui ont notamment montré que le problème consistant à trouver plus d'un équilibre de Nash en stratégies mixtes dans un jeu en forme normale est NP-difficile, ou encore que prouver l'existence d'un équilibre de Nash en stratégies mixtes satisfaisant certaines propriétés dans un jeu à deux joueurs en forme normale est NP-difficile. Une réduction plus simple pour ce problème a été trouvée dans [Conitzer et Sandholm, 2003]. Des extensions de ces résultats à des jeux plus généraux ont été trouvés par [Meggido et Papadimitriou, 1991], et par [Papadimitriou, 1994b].

D'autres travaux étudient des problèmes plus proches des nôtres, à savoir l'existence d'un équilibre de Nash en *stratégies pures*. Par exemple, les jeux étudiés dans [Gottlob *et al.*, 2005] sont des jeux stratégiques sur lesquels il est imposé des limitations quantitatives et/ou qualitatives sur l'influence d'un agent sur les décisions des autres agents : par exemple, restreindre le nombre de joueurs ayant une influence sur un joueur donné à k (ce qui correspondrait dans nos jeux booléens à $RP_i \leq k$); ou encore définir un graphe de dépendance, et étudier le cas particulier où il est acyclique. Nous étudierons plus longuement ces cas particuliers, et nous les comparerons avec les nôtres dans le chapitre 4 (page 121). Dans ce contexte, le problème consistant à décider s'il existe un équilibre de Nash en stratégies pures est NP-complet pour chacune des deux restrictions décrites plus haut.

Le problème de l'existence d'un équilibre de Nash en stratégies pures pour plusieurs classes plus spécifiques de jeux a également été étudié. Par exemple, trouver un équilibre de Nash en stratégies pures a été montré polynomial pour quatre notions de jeux symétriques ayant un nombre constant d'actions, mais non-polynomial si le nombre d'actions est linéaire dans le nombre de joueurs [Brandt *et al.*, 2007] (voir chapitre 4 (page 121)).

D'autre part, les jeux graphiques d'actions (action graph games, AGGs, voir chapitre 4 (page 121)) sont des jeux totalement expressifs (tout jeu peut être représenté par un AGG) représentés par des graphes dont les nœuds correspondent à des actions, et les arcs aux dépendances entre les utilités des agents effectuant ces actions [Bhat et Leyton-Brown, 2004; Jiang et Leyton-Brown, 2006]. Si le problème consistant à décider s'il existe un équilibre de Nash en stratégies pures dans un AGG est NP-complet [Jiang et Leyton-Brown, 2006], il existe des classes de jeu pour lesquels ce problème est polynomial [Jiang et Leyton-Brown, 2007]. C'est le cas pour les AGGs ayant un nombre d'actions, et donc de nœuds, limité à k .

Les jeux de congestion (congestion games, voir chapitre 4 (page 121)) [Rosenthal, 1973], sont des jeux dans lesquels les actions disponibles consistent en un ensemble de ressources, et l'uti-

lité de chaque joueur dépend du nombre de joueurs ayant sélectionné les mêmes ressources (c'est-à-dire ayant joué la même action). Les jeux de congestion ont toujours un équilibre de Nash en stratégies pures [Rosenthal, 1973], et il est montré dans [Fabrikant *et al.*, 2004] qu'un PNE peut être calculé en temps polynomial dans le cas où le réseau de ce jeu (c'est-à-dire le graphe dont les nœuds sont les actions possibles, et les arcs les ressources) est symétrique.

2.6 Jeux booléens et duopole de Stackelberg

Von Stackelberg a imaginé en 1934 une situation à deux joueurs dans laquelle un des deux joueurs a une idée précise du comportement de son concurrent : il connaît parfaitement sa fonction de réaction et il l'intègre dans son processus de décision. On appelle alors ce joueur le *leader* ou le *meneur*. Suite à sa décision, son concurrent réagit en maximisant son profit et donc en suivant sa fonction de réaction ; il se contente de "suivre" le comportement du leader et pour cette raison, on l'appelle le *suiveur* (*follower*). Dans ce cas, le suiveur considère que ses décisions n'ont aucun impact sur le comportement du meneur (voir par exemple [Osborne et Rubinstein, 1994]).

Une idée est alors d'étudier la dynamique dans les jeux booléens à deux joueurs en suivant ce principe de "meneur-suiveur". Cette ébauche de travail a été effectuée au laboratoire ILIC à Amsterdam avec Ulle Endriss.

Dans cette section, nous allons procéder par étapes : nous allons commencer par étudier les jeux de type Stackelberg à deux joueurs et une variable par joueur, puis nous chercherons empiriquement les propriétés que doivent respecter les stratégies du meneur et du receveur pour être optimales. Nous généraliserons ensuite ces résultats à des jeux à deux joueurs et 3 variables, puis ainsi de suite jusqu'à prouver ce résultat, trouvé empiriquement, pour des jeux à deux joueurs et n variables par joueur.

2.6.1 2 joueurs, 1 variable chacun

Dans cette section, nous allons étudier des jeux booléens $G = (N, V, \Gamma, \pi, \Phi)$ tels que $N = \{1, 2\}$, $V = \{a, b\}$, $\gamma_1 = \gamma_2 = \top$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, les buts des joueurs étant quelconques. Supposons que le joueur 1 est le meneur, et donc joue en premier. La liste de toutes les issues possibles de ce type de jeux une fois que le joueur 1 a joué la stratégie s_1 ($s_1 = a$ ou $s_1 = \bar{a}$) est représentée sur la figure 2.7 (page suivante).

Chaque paire de couples du type $(x, y)(z, t)$ apparaissant dans la première colonne de ce tableau représente les issues du jeu pour le joueur 1 s'il joue s_1 selon la stratégie choisie par le joueur 2. Donc si le joueur 1 joue s_1 et que le joueur 2 joue s_2 ($s_2 = b$ ou $s_2 = \bar{b}$), l'issue du jeu sera¹³ (x, y) ; si le joueur 2 joue \bar{s}_2 (si $s_2 = b$, alors $\bar{s}_2 = \bar{b}$ et vice-versa), l'issue du jeu sera (z, t) . Dans ces couples, la notation $_$ signifie "0 ou 1".

La seconde colonne du tableau représente l'intérêt de s_1 pour le joueur 1 : c'est soit une stratégie gagnante (si l'issue est $(1, _)(1, _)$ par exemple, quel que soit le choix de 2, 1 est sûr de gagner s'il joue s_1), soit une stratégie perdante (si l'issue est $(0, _)(0, _)$ par exemple, 1 est sûr de perdre s'il joue s_1), soit le joueur 1 ne peut pas prévoir quelle sera son utilité (si l'issue est $(0, 1)(1, 1)$ par exemple, 1 ne sait pas quelle stratégie jouera 2).

¹³Comme précédemment, cela signifie que le joueur 1 aura l'utilité x , et le joueur 2 l'utilité y .

$(1, _)(1, _)$	stratégie gagnante	$s_1 \models \varphi_1$
$(0, 0)(1, 1)$	stratégie gagnante	$s_1 \models \varphi_1 \leftrightarrow \varphi_2$
$(1, 1)(0, 0)$	stratégie gagnante	$s_1 \models \varphi_1 \leftrightarrow \varphi_2$
$(0, _)(0, _)$	stratégie perdante	$s_1 \models \neg\varphi_1$
$(1, 0)(0, 1)$	stratégie perdante	$s_1 \models \neg(\varphi_1 \leftrightarrow \varphi_2)$
$(0, 1)(1, 0)$	stratégie perdante	$s_1 \models \neg(\varphi_1 \leftrightarrow \varphi_2)$
$(1, 0)(0, 0)$	imprévisible	$s_1 \models \neg\varphi_2$
$(0, 0)(1, 0)$	imprévisible	$s_1 \models \neg\varphi_2$
$(0, 1)(1, 1)$	imprévisible	$s_1 \models \varphi_2$
$(1, 1)(0, 1)$	imprévisible	$s_1 \models \varphi_2$

Figure 2.7 — Issues possibles du jeu de type Stackelberg à 2 joueurs et 1 variable chacun

Enfin, dans la troisième colonne se trouve une description des effets de s_1 sur les buts des joueurs (par exemple, si l'issue est $(1, _)(1, _)$, on sait que $s_1 \models \varphi_1$).

A partir de ce tableau, qui couvre tous les cas possibles, on déduit que la stratégie s_1 est optimale pour le joueur 1 si et seulement si

- * soit c 'est une stratégie gagnante,
- * soit¹⁴ \bar{s}_1 est une stratégie perdante (auquel cas s_1 est forcément meilleure),
- * soit il est impossible de choisir entre s_1 et \bar{s}_1 , les deux étant imprévisibles (dans ce cas, les deux stratégies sont optimales).

Une fois que le joueur 1 a joué, le joueur 2 peut en toute connaissance de cause jouer à son tour, et il choisira une stratégie qui ne le satisfait pas uniquement s'il ne peut pas être satisfait.

On obtient donc l'assertion suivante :

Assertion 2.1. Soit un jeu booléen $G = (N, V, \Gamma, \pi, \Phi)$ tel que $N = \{1, 2\}$, $V = \{a, b\}$, $\gamma_1 = \gamma_2 = \top$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$ du type Stackelberg.

- * **La stratégie s_1 est optimale pour le meneur de G si et seulement si :**

$$\begin{aligned} & \exists s_2 \text{ tq } s_1 \models \varphi_1 \vee ((\varphi_1 \leftrightarrow \varphi_2) \wedge (\varphi_1)_{s_2}) \quad (\text{stratégie gagnante}) \\ \text{ou} & \\ & \exists s_2 \text{ tq } \bar{s}_1 \models \neg\varphi_1 \vee (\neg(\varphi_1 \leftrightarrow \varphi_2) \wedge (\varphi_2)_{s_2}) \quad (\text{stratégie non perdante}) \\ \text{ou} & \\ & \left(\begin{array}{c} s_1 \models \varphi_2 \\ \text{ou} \\ s_1 \models \neg\varphi_2 \end{array} \right) \text{ et } \left(\begin{array}{c} \bar{s}_1 \models \varphi_2 \\ \text{ou} \\ \bar{s}_1 \models \neg\varphi_2 \end{array} \right) \quad (\text{peu importe}) \end{aligned}$$

- * Si le joueur 1 joue s_1 , **la stratégie s_2 est optimale pour le suiveur si et seulement si**

$$s_2 \models \neg(\varphi_2)_{s_1} \Rightarrow \bar{s}_2 \models \neg(\varphi_2)_{s_1}$$

¹⁴Si $s_1 = a$, alors $\bar{s}_1 = \bar{a}$, et vice-versa.

2.6.2 2 joueurs, 3 variables

On étudie à présent des jeux booléens $G = (N, V, \Gamma, \pi, \Phi)$ tels que $N = \{1, 2\}$, $V = \{a, b, c\}$, $\gamma_1 = \gamma_2 = \top$, $\pi_1 = \{a, c\}$, $\pi_2 = \{b\}$. Supposons que le joueur 1 est le meneur, et donc qu'il joue en premier. Le joueur 2 répond en instanciant b , et c'est ensuite encore au joueur 1 de jouer.

Le joueur 2 se retrouve donc ici dans la même position que le joueur 1 dans la section précédente : il est meneur d'un jeu booléen de type Stackelberg à deux joueurs ayant une variable chacune. Sa stratégie optimale est donc donnée par l'assertion 2.1 (page précédente). Il nous reste donc à définir la stratégie optimale du premier joueur pour l'instanciation de sa première variable.

En effectuant le même raisonnement que précédemment, nous trouvons donc l'assertion suivante :

Assertion 2.2. *Une fois la première variable du meneur instanciée, une stratégie optimale pour le joueur 2 est donnée dans l'assertion 2.1 (page précédente), ce joueur se retrouvant meneur du sous-jeu à deux joueurs et deux variables restant. Soit s_1 l'instanciation de la première variable du joueur 1 dans la stratégie s_1 (puisque s_{1_1} , s_{1_2} et s_2 correspondent chacun à l'instanciation d'une seule variable, on peut écrire comme précédemment \bar{s}_{1_1} , \bar{s}_{1_2} et \bar{s}_2).*

* **La stratégie s_2 est optimale pour le suiveur si et seulement si**

$$\begin{aligned} & \exists s_{1_2} \, tq \, s_2 \models (\varphi_2)_{s_{1_1}} \vee (((\varphi_1)_{s_{1_1}} \leftrightarrow (\varphi_2)_{s_{1_1}}) \wedge (\varphi_2)_{s_{1_1}, s_{1_2}}) \quad (\text{stratégie gagnante}) \\ \text{ou} & \\ & \exists s_{1_2} \, tq \, \bar{s}_2 \models \neg(\varphi_2)_{s_{1_1}} \vee (\neg((\varphi_1)_{s_{1_1}} \leftrightarrow (\varphi_2)_{s_{1_1}}) \wedge (\varphi_1)_{s_{1_1}, s_{1_2}}) \quad (\text{stratégie non perdante}) \\ \text{ou} & \\ & \left(\begin{array}{c} s_2 \models (\varphi_1)_{s_{1_1}} \\ \text{ou} \\ s_2 \models \neg(\varphi_1)_{s_{1_1}} \end{array} \right) \text{ et } \left(\begin{array}{c} \bar{s}_2 \models (\varphi_1)_{s_{1_1}} \\ \text{ou} \\ \bar{s}_2 \models \neg(\varphi_1)_{s_{1_1}} \end{array} \right) \quad (\text{peu importe}) \end{aligned}$$

* *Le joueur 1 sait que 2 choisira une stratégie s_2 optimale suivant l'instanciation de sa première variable, c'est-à-dire s_{1_1} . Il peut donc choisir sa stratégie suivant s_2 . **La stratégie s_1 est optimale pour le meneur si et seulement si***

$$\begin{aligned} \forall s_2 \text{ optimale pour 2} : & \left(\begin{array}{c} \exists s_{1_2} \, tq \, (s_{1_1}, s_{1_2}) \models (\varphi_1)_{s_2} \quad (\text{stratégie gagnante}) \\ \text{ou} \\ \bar{s}_{1_1} \models \neg(\varphi_1)_{s_2} \quad (\text{stratégie non perdante}) \end{array} \right) \\ \text{ou} & \\ \text{si } \forall s_{1_2} \text{ optimale pour 1, 1 est indifférent} : & \left(\begin{array}{c} \exists s_{1_2} \, tq \, s_{2_1} \models \neg(\varphi_2)_{s_{1_2}} \\ \text{et} \\ \exists s_{1_2} \, tq \, \bar{s}_{2_1} \models \neg(\varphi_2)_{s_{1_2}} \end{array} \right) \quad (\text{peu importe}) \end{aligned}$$

Exemple 2.14. *Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen tel que $N = \{1, 2\}$, $V = \{a, b, c\}$, $\gamma_1 = \gamma_2 = \top$, $\pi_1 = \{a, c\}$, $\pi_2 = \{b\}$, $\varphi_1 = (a \wedge b) \vee (c \leftrightarrow b)$ et $\varphi_2 = (\neg a \wedge \neg b) \vee c$.*

Le joueur 1 commence par jouer la variable a , et doit tout d'abord calculer les stratégies optimales du joueur 2 pour chacune des instanciations de a .

Soit $s_{1_1} = a$. On a alors $(\varphi_2)_{s_{1_1}} = c$, $(\varphi_1)_{s_{1_1}} = b \vee (c \leftrightarrow b)$.

On a $\neg b \models \neg((\varphi_1)_{s_{1_1}} \leftrightarrow (\varphi_2)_{s_{1_1}})$, ce qui est une stratégie perdante pour 2, et $b \models (\varphi_1)_{s_{1_1}}$ ce qui est une stratégie non perdante pour 2, et donc sa stratégie optimale.

Une fois que 1 a fait ce calcul, il n'a plus besoin de calculer la stratégie optimale de 2 pour $\bar{s}_{1_1} = \neg a$ car il sait déjà qu'il a une stratégie gagnante s'il choisit a : 2 étant rationnel va choisir b , et alors $ab \models \varphi_1$.

2.6.3 2 joueurs, 2 variables chacun

Nous étudions à présent des jeux booléens $G = (N, V, \Gamma, \pi, \Phi)$ tels que $N = \{1, 2\}$, $V = \{a, b, c, d\}$, $\gamma_1 = \gamma_2 = \top$, $\pi_1 = \{a, c\}$, $\pi_2 = \{b, d\}$. Supposons que le joueur 1 commence par instancier la variable a , puis que 2 instancie b , et ainsi de suite.

Nous connaissons déjà la stratégie optimale de 2 une fois que 1 a instancié a , ainsi que la stratégie optimale de 1 pour la variable c une fois que a et b sont instanciées. Il nous manque donc à connaître l'instanciation optimale de a pour le premier joueur.

s_{1_1} est une stratégie optimale de 1 si et seulement si $\forall s_{2_1}$ optimales pour¹⁵ 2 :

1. $\exists s_{1_2}$ une stratégie gagnante pour 1, donc telle que :

$$\exists s_{2_2} \text{ tq } (s_{1_1}, s_{1_2}) \models (\varphi_1)_{s_{2_1}} \vee (((\varphi_1)_{s_{2_1}} \leftrightarrow (\varphi_2)_{s_{2_1}}) \wedge (\varphi_1)_{s_{2_1} s_{2_2}})$$

2. ou, \bar{s}_{1_1} est une stratégie perdante :

$$\exists s_{2_2} \text{ tq } \bar{s}_{1_1} \models \neg(\varphi_1)_{s_{2_1}} \vee (\neg((\varphi_1)_{s_{2_1}} \leftrightarrow (\varphi_2)_{s_{2_1}}) \wedge (\varphi_2)_{s_{2_1} s_{2_2}})$$

3. ou, 1 est indifférent :

$$\begin{aligned} & \left(\begin{array}{c} (s_{1_1}, s_{1_2}) \models (\varphi_2)_{s_{2_1}} \\ \text{ou} \\ (s_{1_1}, s_{1_2}) \models \neg(\varphi_2)_{s_{2_1}} \end{array} \right) \text{ et } \left(\begin{array}{c} (s_{1_1}, \bar{s}_{1_2}) \models (\varphi_2)_{s_{2_1}} \\ \text{ou} \\ (s_{1_1}, \bar{s}_{1_2}) \models \neg(\varphi_2)_{s_{2_1}} \end{array} \right) \\ & \text{et } \left(\begin{array}{c} (\bar{s}_{1_1}, s_{1_2}) \models (\varphi_2)_{s_{2_1}} \\ \text{ou} \\ (\bar{s}_{1_1}, s_{1_2}) \models \neg(\varphi_2)_{s_{2_1}} \end{array} \right) \text{ et } \left(\begin{array}{c} (\bar{s}_{1_1}, \bar{s}_{1_2}) \models (\varphi_2)_{s_{2_1}} \\ \text{ou} \\ (\bar{s}_{1_1}, \bar{s}_{1_2}) \models \neg(\varphi_2)_{s_{2_1}} \end{array} \right) \end{aligned}$$

Assertion 2.3.

* **La stratégie s_{2_1} est optimale pour le suiveur si et seulement si**

$$\forall s_{1_2} \text{ optimale pour 1 : } \left(\begin{array}{c} \exists s_{2_2} \text{ tq } (s_{2_1}, s_{2_2}) \models (\varphi_2)_{s_{1_1}, s_{1_2}} \quad (\text{stratégie gagnante}) \\ \text{ou} \\ \bar{s}_{2_1} \models \neg(\varphi_2)_{s_{1_1}, s_{1_2}} \quad (\text{stratégie non perdante}) \end{array} \right)$$

ou

$$\text{si } \forall s_{1_2} \text{ optimale, 1 est indifférent : } \left(\begin{array}{c} \exists s_{1_2} \text{ tq } s_{2_1} \models \neg(\varphi_2)_{s_{1_2}} \\ \text{et} \\ \exists s_{1_2} \text{ tq } \bar{s}_{2_1} \models \neg(\varphi_2)_{s_{1_2}} \end{array} \right) \quad (\text{peu importe})$$

¹⁵De nouveau, quels que soient $i \in \{1, 2\}$ et $j \in \{1, 2\}$, chaque s_{ij} correspondant à l'instanciation d'une seule variable, il est possible d'écrire \bar{s}_{ij} .

- * **La stratégie s_{1_1} est optimale pour le meneur si et seulement si pour toute stratégie s_{2_1} optimale pour 2, soit il existe s_{1_2} optimale pour 1, soit \bar{s}_{1_1} est une stratégie perdante, soit 1 est indifférent. Donc s_{1_1} est une stratégie optimale pour 1 si et seulement si $\forall s_{2_1}$ optimale pour 2 :**

$$\begin{aligned} \exists s_{1_2, s_{2_2}} tq (s_{1_1}, s_{1_2}) \models & ((\varphi_1)_{s_{2_1}} \vee ((\varphi_1)_{s_{2_1}} \leftrightarrow (\varphi_2)_{s_{2_1}}) \wedge (\varphi_1)_{s_{2_1}, s_{2_2}}) \quad (\text{stratégie gagnante}) \\ & \text{ou} \\ \exists s_{2_2} tq \bar{s}_{1_1} \models & \neg(\varphi_1)_{s_{2_1}} \vee (\neg((\varphi_1)_{s_{2_1}} \leftrightarrow (\varphi_2)_{s_{2_1}}) \wedge (\varphi_2)_{s_{2_1}, s_{2_2}}) \quad (\text{stratégie non perdante}) \\ & \text{ou} \\ \left(\begin{array}{c} (s_{1_1}, s_{1_2}) \models (\varphi_2)_{s_{2_1}} \\ \text{ou} \\ (s_{1_1}, s_{1_2}) \models \neg(\varphi_2)_{s_{2_1}} \end{array} \right) & \text{et} \left(\begin{array}{c} (s_{1_1}, \bar{s}_{1_2}) \models (\varphi_2)_{s_{2_1}} \\ \text{ou} \\ (s_{1_1}, \bar{s}_{1_2}) \models \neg(\varphi_2)_{s_{2_1}} \end{array} \right) \\ & \text{et} \\ \left(\begin{array}{c} (\bar{s}_{1_1}, s_{1_2}) \models (\varphi_2)_{s_{2_1}} \\ \text{ou} \\ (\bar{s}_{1_1}, s_{1_2}) \models \neg(\varphi_2)_{s_{2_1}} \end{array} \right) & \text{et} \left(\begin{array}{c} (\bar{s}_{1_1}, \bar{s}_{1_2}) \models (\varphi_2)_{s_{2_1}} \\ \text{ou} \\ (\bar{s}_{1_1}, \bar{s}_{1_2}) \models \neg(\varphi_2)_{s_{2_1}} \end{array} \right) \quad (\text{peu importe}) \end{aligned}$$

2.6.4 2 joueurs, n variables chacun

Etudions à présent des jeux booléens $G = (N, V, \Gamma, \pi, \Phi)$ tels que $N = \{1, 2\}$, $V = \{v_1, \dots, v_{2n}\}$, $\gamma_1 = \gamma_2 = \top$, $\pi_1 = \{v_1, \dots, v_n\}$, $\pi_2 = \{v_{n+1}, \dots, v_{2n}\}$. Chaque joueur contrôle donc exactement n variables. Supposons que le joueur 1 est le meneur, et qu'il commence par jouer la variable v_1 . Le joueur 2 joue ensuite la variable v_{n+1} , et ainsi de suite.

Propriété 2.22.

- * **La stratégie s_{1_1} est optimale pour le meneur si et seulement si $\forall s_{2_1}, \dots, s_{2_{n-1}}$ optimales¹⁶ pour 2 :**

$$\exists s_{1_2, \dots, s_{1_n}, s_{2_n}} tq \quad (s_{1_1}, \dots, s_{1_n}) \models (\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \vee (((\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \leftrightarrow (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}}) \wedge (\varphi_1)_{s_{2_1} \dots s_{2_n}}) \quad (1)$$

ou

$$\exists s_{2_n} tq \bar{s}_{1_1} \models \neg(\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \vee (\neg((\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \leftrightarrow (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}}) \wedge (\varphi_2)_{s_{2_1} \dots s_{2_n}}) \quad (2)$$

ou

$$\forall s_1 \in S_1, \left(\begin{array}{c} s_1 \models (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}} \\ \text{ou} \\ s_1 \models \neg(\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}} \end{array} \right) \quad (3)$$

- * **La stratégie s_{2_1} est optimale pour le suiveur si et seulement si**

$$\forall s_{1_2}, \dots, s_{1_n} \text{ optimales pour 1} : \left(\begin{array}{c} \exists s_{2_2}, \dots, s_{2_n} tq (s_{2_1}, \dots, s_{2_n}) \models (\varphi_2)_{s_{1_1}, \dots, s_{1_n}} \quad (1) \\ \text{ou} \\ \bar{s}_{2_1} \models \neg(\varphi_2)_{s_{1_1}, \dots, s_{1_n}} \quad (2) \end{array} \right)$$

ou

$$\text{si } \forall s_{1_2}, \dots, s_{1_n} \text{ optimales, 1 est indifférent} : \left(\begin{array}{c} \exists s_1 \in S_1 tq s_{2_1} \models \neg(\varphi_2)_{s_1} \\ \text{et} \\ \exists s_1 \in S_1 tq \bar{s}_{2_1} \models \neg(\varphi_2)_{s_1} \end{array} \right) \quad (3)$$

¹⁶De nouveau, quels que soient $i \in \{1, 2\}$ et $j \in \{1, \dots, n\}$, chaque s_{ij} correspondant à l'instanciation d'une seule variable, il est possible d'écrire \bar{s}_{ij} .

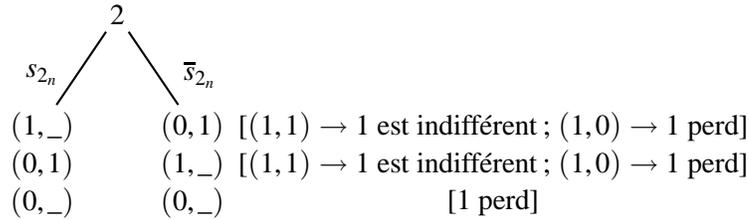
Preuve :

* Joueur 1.

⇒ L'optimalité n'est pas exclusive (s_{1_1} et \bar{s}_{1_1} peuvent être optimales). Nous voulons donc montrer que si s_{1_1} ne respecte pas les équations (1), (2) et (3), alors \bar{s}_{1_1} est optimale.

$$(1) \quad \forall s_{1_2}, \dots, s_{1_n}, (s_{1_1}, s_{1_2}, \dots, s_{1_n}) \not\models (\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \vee (((\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \leftrightarrow (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}}) \wedge (\exists s_{2_n} tq (\varphi_1)_{s_{2_1} \dots s_{2_n}})).$$

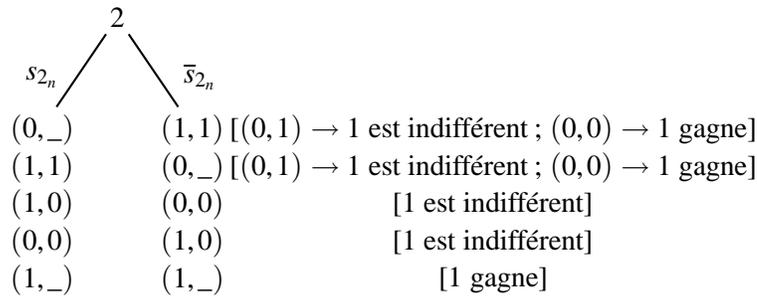
Les possibilités de satisfaire cette formule dépendent de la valeur de s_{2_n} , qui est la seule variable dont on ne connaît pas encore l'instanciation. On obtient les possibilités suivantes : $\forall s_{1_2}, \dots, s_{1_n}$ et $\forall s_{2_1}, \dots, s_{2_{n-1}}$ stratégies optimales pour 2, après un chemin contenant $s_{1_1}, s_{1_2}, \dots, s_{1_n}, s_{2_1}, \dots, s_{2_{n-1}}$:



En effet, si (1) n'est pas satisfaite, on ne peut pas avoir, une fois $s_{1_1}, s_{1_2}, \dots, s_{1_n}, s_{2_1}, \dots, s_{2_{n-1}}$ instanciées, φ_1 satisfaite pour tout s_{2_n} . On ne peut pas avoir non plus pour tout s_{2_n} , $\varphi_1 \leftrightarrow \varphi_2$ et pour un s_{2_n} , φ_1 satisfaite. Les seules options disponibles sont donc celles décrites plus haut.

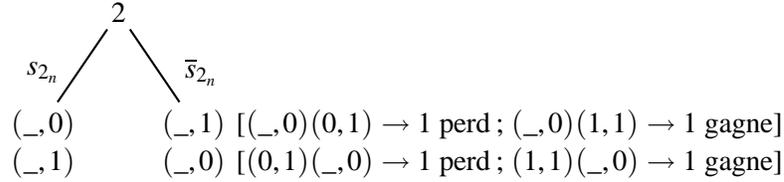
$$(2) \quad \bar{s}_{1_1} \not\models \neg(\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \vee (\neg((\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \leftrightarrow (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}}) \wedge (\exists s_{2_n} tq (\varphi_2)_{s_{2_1} \dots s_{2_n}})).$$

Les possibilités de satisfaire cette formule dépendent une fois encore de la valeur de s_{2_n} . Nous obtenons les possibilités suivantes : $\forall s_{2_1}, \dots, s_{2_{n-1}}$ stratégies optimales pour 2, $\exists s_{1_2}, \dots, s_{1_n}$ telles que le chemin contenant $\bar{s}_{1_1}, s_{1_2}, \dots, s_{1_n}, s_{2_1}, \dots, s_{2_{n-1}}$ satisfait les conditions suivantes pour s_{2_n} :



$$(3) \quad \exists s_1 \in \mathcal{S}_1, \left(\begin{array}{c} s_1 \not\models (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}} \\ et \\ s_1 \not\models \neg(\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}} \end{array} \right)$$

Une fois encore, les possibilités de satisfaire cette formule dépendent de la valeur de s_{2_n} . Nous obtenons les possibilités suivantes : $\forall s_{2_1}, \dots, s_{2_{n-1}}$ stratégies optimales pour 2, $\exists s_{1_1}, s_{1_2}, \dots, s_{1_n}$ telles que le chemin contenant $\bar{s}_{1_1}, s_{1_2}, \dots, s_{1_n}, s_{2_1}, \dots, s_{2_{n-1}}$ satisfait les conditions suivantes pour s_{2_n} :



Plusieurs cas sont à envisager :

* **À partir de (1)** : Si $\forall s_{1_2}, \dots, s_{1_n}$ et $\forall s_{2_1}, \dots, s_{2_{n-1}}$, le résultat obtenu par le joueur 1 est $[(0, 1) \text{ et } (1, 0)]$ ou $[(1, 0) \text{ et } (0, 1)]$ ou $[(0, _) \text{ et } (0, _)]$, alors s_{1_1} est une stratégie perdante, et \bar{s}_{1_1} est une stratégie optimale.

Si $\forall s_{1_2}, \dots, s_{1_n}$ et $\forall s_{2_1}, \dots, s_{2_{n-1}}$, le résultat obtenu par le joueur 1 est $[(1, 1) \text{ et } (0, 1)]$ ou $[(0, 1) \text{ et } (1, 1)]$, on sait d'après (2) que le joueur 1 peut obtenir les mêmes résultats en jouant \bar{s}_{1_1} , auquel cas il est indifférent entre ces deux stratégies (et \bar{s}_{1_1} est une stratégie optimale) ; ou alors la stratégie \bar{s}_{1_1} est une stratégie gagnante.

* **À partir de (2)** : Si $\exists s_{1_2}, \dots, s_{1_n}$ et $\forall s_{2_1}, \dots, s_{2_{n-1}}$, le résultat obtenu par le joueur 1 est $[(0, 0) \text{ et } (1, 1)]$ ou $[(1, 1) \text{ et } (0, 0)]$ ou $[(1, _) \text{ et } (1, _)]$, alors \bar{s}_{1_1} est une stratégie gagnante, donc est une stratégie optimale.

D'après (1), on sait que soit s_{1_1} est une stratégie perdante, soit le joueur 1 est indifférent. Donc si $\exists s_{1_2}, \dots, s_{1_n}$ et $\forall s_{2_1}, \dots, s_{2_{n-1}}$ tels que le résultat obtenu par le joueur 1 est $[(0, 1) \text{ et } (1, 1)]$ ou $[(1, 1) \text{ et } (0, 1)]$ ou $[(1, 0) \text{ et } (0, 0)]$, ou $[(0, 0) \text{ et } (1, 0)]$, \bar{s}_{1_1} est une stratégie optimale (puisque soit s_{1_1} est une stratégie perdante, soit le joueur 1 est indifférent entre s_{1_1} et \bar{s}_{1_1}).

* **À partir de (1), (2) et (3)** : Si la stratégie s_1 obtenue par (3) entraîne un résultat pour le joueur 1 de la forme $[(_, 0) \text{ et } (1, 1)]$ ou $[(1, 1) \text{ et } (_, 0)]$, alors d'après (1) et (2), nous savons que s_1 contient \bar{s}_{1_1} , et que s_1 est une stratégie gagnante, donc \bar{s}_{1_1} est optimale.

* **À partir de (1), (2) et (3)** : Si la stratégie s_1 obtenue par (3) entraîne un résultat pour le joueur 1 de la forme $[(_, 0) \text{ et } (0, 1)]$ ou $[(0, 1) \text{ et } (_, 0)]$, alors d'après (1) et (2), nous savons que s_1 contient s_{1_1} . Dans ce cas, soit s_{1_1} est une stratégie perdante pour tout $s_{1_i}, i \in [1, n]$ et \bar{s}_{1_1} est optimale ; soit le joueur 1 est indifférent entre s_{1_1} et \bar{s}_{1_1} , auquel cas \bar{s}_{1_1} est optimale.

⇐ Supposons à présent que s_{1_1} satisfait les équations (1), (2) et (3), et prouvons alors que s_{1_1} est optimale.

(1) $\forall s_{2_1}, \dots, s_{2_{n-1}}$ optimales pour 2, $\exists s_{1_2}, \dots, s_{1_n}, s_{2_n}$ telles que

$$(s_{1_1}, \dots, s_{1_n}) \models \underbrace{(\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}}}_{\varphi_i \text{ est satisfaite } \forall s_{2_n}} \vee \underbrace{(((\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \leftrightarrow (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}}) \wedge (\varphi_1)_{s_{2_1} \dots s_{2_n}})}_{(*)}$$

(*) : 2 jouant de façon rationnelle, ce joueur choisira s_{2_n} telle que $s_{2_n} \models (\varphi_2)_{s_{1_1}, \dots, s_{1_n}, s_{2_1}, \dots, s_{2_{n-1}}}$, et alors $s_{2_n} \models (\varphi_1)_{s_{1_1}, \dots, s_{1_n}, s_{2_1}, \dots, s_{2_{n-1}}}$. s_{1_1} est donc optimale.

(2) $\forall s_{2_1}, \dots, s_{2_{n-1}}$ optimales pour 2, $\exists s_{2_n}$ telle que $\forall s_{1_2}, \dots, s_{1_n}, s_{2_n}$,

$$\bar{s}_{1_1} \models \underbrace{\neg(\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}}}_{\varphi_i \text{ n'est pas satisfaite } \forall s_{2_n}} \vee \underbrace{(\neg((\varphi_1)_{s_{2_1} \dots s_{2_{n-1}}} \leftrightarrow (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}}) \wedge (\varphi_2)_{s_{2_1} \dots s_{2_n}})}_{(*)}$$

(*) : 2 jouant de façon rationnelle, ce joueur choisira s_{2_n} telle que $s_{2_n} \models (\varphi_2)_{s_{1_1}, \dots, s_{1_n}, s_{2_1}, \dots, s_{2_{n-1}}}$, et alors $s_{2_n} \models \neg(\varphi_1)_{s_{1_1}, \dots, s_{1_n}, s_{2_1}, \dots, s_{2_{n-1}}}$. s_{1_1} est donc optimale.

(3) $\forall s_{2_1}, \dots, s_{2_{n-1}}$ optimales pour 2, $\forall s_1 \in S_1$,

$$\left(\begin{array}{c} s_1 \models (\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}} \\ \text{ou} \\ s_1 \models \neg(\varphi_2)_{s_{2_1} \dots s_{2_{n-1}}} \end{array} \right)$$

$\forall s_1 \in S_1$, 1 ne connaît donc pas le choix de 2, et donc ne peut pas choisir. s_{1_1} et \bar{s}_{1_1} sont donc optimales.

* Joueur 2.

\Rightarrow L'optimalité n'est pas exclusive (s_{2_1} et \bar{s}_{2_1} peuvent être optimales). Nous voulons donc montrer que si s_{2_1} ne respecte pas les équations (1), (2) et (3), alors \bar{s}_{2_1} est optimale.

(1) $\exists s_{1_2}, \dots, s_{1_n}$ optimales pour 1 telles que $\forall s_{2_2}, \dots, s_{2_n}$ telles que $(s_{2_1}, \dots, s_{2_n}) \not\models (\varphi_2)_{s_{1_1}, \dots, s_{1_n}}$, donc $(s_{2_1}, \dots, s_{2_n}) \models \neg(\varphi_2)_{s_{1_1}, \dots, s_{1_n}}$, donc s_{2_1} est une stratégie perdante, et donc \bar{s}_{2_1} est optimale.

(2) $\exists s_{1_2}, \dots, s_{1_n}$ optimales pour 1 telles que $\bar{s}_{2_1} \not\models \neg(\varphi_2)_{s_{1_1}, \dots, s_{1_n}}$. Donc, $\exists s_{2_2}, \dots, s_{2_n}$ telles que $\bar{s}_{2_1}, \dots, s_{2_n} \models (\varphi_2)_{s_{1_1}, \dots, s_{1_n}}$. Donc il existe une stratégie gagnante contenant \bar{s}_{2_1} , et donc cette stratégie est optimale.

(3) si $\forall s_{1_2}, \dots, s_{1_n}$ optimales pour 1, 1 est indifférent :

$$\left(\begin{array}{c} \forall s_1 \in S_1, s_{2_1} \models \neg(\varphi_2)_{s_1} \\ \text{ou} \\ \forall s_1 \in S_1, \bar{s}_{2_1} \models \neg(\varphi_2)_{s_1} \end{array} \right).$$

$$\text{Donc, } \left(\begin{array}{c} \forall s_1 \in S_1, \exists s_{2_2}, \dots, s_{2_n} \text{ telles que } s_{2_1} \models (\varphi_2)_{s_1} \quad (4) \\ \text{or} \\ \forall s_1 \in S_1, \exists s_{2_2}, \dots, s_{2_n} \text{ telles que } \bar{s}_{2_1} \models (\varphi_2)_{s_1} \quad (5) \end{array} \right).$$

Nous savons d'après (1) que (4) n'est pas possible. (5) est donc vrai, et \bar{s}_{2_1} est optimale.

\Leftarrow Supposons à présent que s_{2_1} satisfait les équations (1), (2) et (3), et prouvons que s_{2_1} est optimale.

(1) $\forall s_{1_2}, \dots, s_{1_n}$ optimales pour 1 : $\exists s_{2_2}, \dots, s_{2_n}$ telles que $(s_{2_1}, \dots, s_{2_n}) \models (\varphi_2)_{s_{1_1}, \dots, s_{1_n}}$. φ_2 est satisfaite, et donc s_{2_1} est optimale.

(2) $\forall s_{1_2}, \dots, s_{1_n}$ optimales pour 1 : $\bar{s}_{2_1} \models \neg(\varphi_2)_{s_{1_1}, \dots, s_{1_n}}$. φ_2 n'est pas satisfaite, et donc s_{2_1} est optimale.

(3) si $\forall s_{1_2}, \dots, s_{1_n}$ optimales pour 1, 1 est indifférent :

$$\left(\begin{array}{c} \exists s_1 \in S_1 \text{ st } s_{2_1} \models \neg(\varphi_2)_{s_1} \\ \text{et} \\ \exists s_1 \in S_1 \text{ st } \bar{s}_{2_1} \models \neg(\varphi_2)_{s_1} \end{array} \right).$$

Dans ce cas, 2 ne peut pas prévoir les choix de 1, et peut donc toujours avoir une stratégie perdante. Ce joueur est donc indifférent entre ses deux choix, et s_{1_1} est optimale.

■

3

Jeux booléens et préférences non dichotomiques

Ce choix d'utilités binaires (pour lequel les agents peuvent seulement exprimer leur totale satisfaction ou leur total mécontentement, sans niveaux intermédiaires) est une vraie perte de généralité. En effet, on ne peut par exemple pas exprimer dans un jeu booléen classique des préférences du type “Je préférerais boire un café, si je ne peux pas un thé, et si ce n'est pas possible non plus je prendrai un chocolat chaud”, qui nécessite 3 niveaux d'utilité ; tout comme on ne peut pas exprimer le dilemme du prisonnier classique (présenté exemple 1.1 (page 10)), qui lui nécessite 4 niveaux d'utilité.

On aimerait donc incorporer des préférences non dichotomiques aux jeux booléens afin que chaque joueur puisse représenter ses préférences de manière plus souple. Nous voudrions pouvoir associer une relation de préférence quelconque à l'ensemble des profils de stratégies S .

Définition 3.1. Une **relation de préférence** \succeq est une relation binaire reflexive et transitive (non nécessairement totale¹) sur S . La **relation de préférence stricte** \succ associée à \succeq est définie par $s_1 \succ s_2$ si et seulement si $s_1 \succeq s_2$ et non($s_2 \succeq s_1$).

Ces préférences peuvent être de différents types : elles peuvent être numériques (on parlera alors de préférences *cardinales*), ou encore *ordinales*. Les préférences cardinales consistent à associer à chaque issue du jeu (ou profil de stratégies) une valeur numérique, et sont donc des relations complètes. Les préférences ordinales consistent à associer directement un pré-ordre sur l'ensemble des issues du jeu, et peuvent être partielles.

Le choix entre ces deux représentations dépend des notions que l'on veut étudier : certaines notions (comme par exemple les équilibres de Nash en stratégies mixtes) nécessitent des préférences cardinales, tandis que d'autres (comme les équilibres de Nash en stratégies pures ou les stratégies dominées) peuvent être définies dans un contexte purement ordinal. Les notions qui nous intéressent ici pouvant être définies avec des préférences ordinales, nous choisissons d'étudier tout particulièrement ces dernières.

Il nous faut donc choisir à présent une représentation permettant d'éviter la description explicite de la fonction d'utilité de chaque agent, qui est, comme nous l'avons déjà vu, de taille exponentielle en fonction du nombre d'agents. Pour les préférences dichotomiques, nous avons utilisé la logique propositionnelle, qui est le langage de représentation compacte le plus naturel pour ces préférences. Pour les préférences non dichotomiques, il nous faut donc trouver un langage permettant d'expri-

¹Une relation de préférence totale est une relation transitive, reflexive et qui vérifie : $\forall x, y : x \succeq y$ ou $y \succeq x$

mer les préférences de chacun des joueurs de façon aussi concise, ou compacte, que possible. Ces langages sont appelés *langages de représentation compacte*. On définit formellement un langage de représentation de préférences [Lang, 2006] :

Définition 3.2. *Un langage de représentation de préférences est un couple $\langle L, Ind \rangle$ où*

- * *L est un langage propositionnel formé sur V,*
- * *Ind est une fonction de L dans Pref, Pref étant l'ensemble des relations de préférence sur 2^V , qui associe à chaque élément de L la relation de préférence induite. Dans le cas des préférences cardinales, la relation de préférence est obtenue par l'intermédiaire d'une fonction d'utilité u.*

Exemple 3.1. *Soit $V = \{a, b, c, d\}$ un ensemble de variables propositionnelles. Nous pouvons définir le langage de représentation de préférences dichotomique de la façon suivante :*

- * *L_{dicho} est le langage formé sur V,*
- * *u_{dicho} est définie comme suit : pour toute formule $\varphi \in L_{dicho}$, et pour tout ensemble $w \subseteq V$,*
 - * *$u_{dicho}^\varphi(w) = 1$ si $w \models \varphi$,*
 - * *$u_{dicho}^\varphi(w) = 0$ sinon.*

Ce langage est celui que nous avons utilisé pour définir les jeux booléens tout au long du chapitre 2 (page 31).

Il existe de nombreux langages de représentation compacte de préférences. On peut citer par exemple :

- * **Logique des pondérations** [Pinkas, 1991; Haddawy et Hanks, 1992; Dupin de Saint Cyr *et al.*, 1992; Benferhat *et al.*, 2001]. Le principe de cette famille de langage de représentation de préférences est d'associer des pondérations numériques à des formules propositionnelles. Il n'est pas nécessaire que les poids soient numériques : il est possible d'utiliser une échelle plus ou moins qualitative munie d'une fonction d'agrégation, comme par exemple la logique possibiliste.
- * **Logique des priorités.** Cette famille de langage de représentations est la contrepartie ordinale des logiques à pondérations. Nous la présenterons plus longuement en section 3.3 (page 99).
- * **Logique des distances** [Katsuno et Mendelzon, 1991, 1992; Lafage et Lang, 2000, 2001; Benferhat *et al.*, 2002]. L'idée est d'utiliser des distances entre interprétations propositionnelles : si un agent doit satisfaire un but G , plus l'interprétation est "loin" de G , moins elle est satisfaisante.
- * **Logique des préférences du type "ceteris paribus".** Cette famille de logiques des préférences est construite sur le principe de l'interprétation *ceteris paribus*, ou encore *toutes choses étant égales par ailleurs*, ou plus généralement *toutes choses non pertinentes étant égales*. Le principe consistant à interpréter des préférences ceteris paribus est dû à [von Wright, 1963], et a été revu par [Hansson, 1989, 2001]. Un langage de représentation graphique, les CP-nets, est fondé sur le critère de comparaison ceteris paribus. Nous présentons les CP-nets en section 3.2 (page 74).
- * **Logique des conditionnels.** L'idée est ici d'utiliser la logique des conditionnels [Lewis, 1973] pour représenter des préférences, et permettre de réviser ces préférences [Boutilier, 1994; Lang, 1996; Lang *et al.*, 2002b].

Mis à part le langage de représentation dichotomique présenté au chapitre 2 (page 31), il existe d'autres travaux dans le cadre des jeux booléens : des préférences non dichotomiques ont déjà été introduites dans ces jeux. Il s'agit des jeux d'évaluation distribués, introduits par [Harrenstein, 2004a],

qui sont des jeux booléens à n joueurs dont les préférences sont représentées par des ensembles de formules propositionnelles : plus le nombre de formules satisfaites pour un agent est grand, plus cet agent est satisfait. Cette relation de préférences simple est pourtant déjà assez sophistiquée : elle permet de représenter des jeux plus complexes avec des jeux booléens, comme par exemple le dilemme classique du prisonnier.

Nous allons donc continuer ici dans cette voie, et étudier deux nouveaux langages de représentation de préférences : les CP-nets et les logiques des priorités² qui permettront eux aussi de traiter des préférences non dichotomiques.

Nous pouvons à présent définir un jeu booléen ayant des préférences ordinales.

Définition 3.3. *Soit un langage propositionnel L pour une représentation compacte de préférences. Un L -jeu booléen est défini par un 5-uple $G = (N, V, \pi, \Gamma, \Phi)$, avec*

- * $N = \{1, \dots, n\}$, V , π et Γ étant définis comme précédemment, et
- * $\Phi = \langle \Phi_1, \dots, \Phi_n \rangle$. Pour chaque i , Φ_i est une représentation compacte, dans L , de la relation de préférences³ de \succeq_i de l'agent i sur⁴ S . On note $Pref_G = \langle \succeq_1, \dots, \succeq_n \rangle$.

3.1 Préférences ordinales et théorie des jeux

Nous devons redéfinir ici les équilibres de Nash en utilisant des préférences ordinales. En effet, les équilibres de Nash sont classiquement définis pour des jeux avec des préférences totales, ce qui n'est plus nécessairement le cas ici, les préférences ordinales pouvant être partielles. L'introduction de ces préférences partielles peut être justifiée de façon épistémique : il se peut qu'un agent ne puisse pas donner ses préférences sur deux états, car il n'a pas toutes les informations nécessaires pour le faire. Par exemple, supposons que Mylen doit réserver un billet d'avion pour Florian. Un avion part à 10h, l'autre à 21h. Si Mylen ne connaît pas les disponibilités de Florian, elle est dans l'incapacité de comparer, et de choisir, entre ces deux vols. Ces deux états sont donc incomparables. Un autre type d'incomparabilité est l'incomparabilité intrinsèque : deux états peuvent être incomparables du fait de leur nature. Par exemple la majorité des gens sont incapables de répondre à la question "Préférez-vous avoir sur votre ordinateur un virus qui efface tous vos messages électroniques, ou un virus qui envoie tous ces messages à tout vos contacts ?"

Nous allons donc définir deux notions d'équilibre de Nash, une forte et une faible (elles sont équivalentes aux notions d'équilibre maximal et maximum dans [Harrenstein, 2004a]).

On rappelle qu'un équilibre de Nash en stratégies pures est un profil de stratégies tel que la stratégie de chaque joueur est une réponse optimale aux stratégies des autres joueurs.

Définition 3.4. *Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen, avec $N = \{1, \dots, n\}$ l'ensemble des joueurs, $\Phi = \langle \Phi_1, \dots, \Phi_n \rangle$ l'ensemble des buts des joueurs, et $Pref_G = \langle \succeq_1, \dots, \succeq_n \rangle$ l'ensemble des préférences de chaque joueur. Deux définitions, une faible et une forte, peuvent ici caractériser les équilibres de Nash.*

²Les résultats donnés dans ce chapitre ont fait l'objet de plusieurs publications : [Bonzon *et al.*, 2006a, 2007a]

³La définition des buts Φ correspond à la définition de *Ind* dans la définition 3.2 (page ci-contre).

⁴On rappelle que dans un jeu booléen, $S_i = \{s_i \in 2^{\mathcal{T}_i} \mid s_i \models \gamma_i\}$, et que $S = S_1 \times \dots \times S_n$.

$s = (s_1, \dots, s_n)$ est un **équilibre de Nash faible en stratégies pures** si et seulement si :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \not\prec_i (s_i, s_{-i}) \quad (3.1)$$

$s = (s_1, \dots, s_n)$ est un **équilibre de Nash fort en stratégies pures** si et seulement si :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \preceq_i (s_i, s_{-i}) \quad (3.2)$$

L'ensemble des équilibres de Nash forts (resp. faibles) en stratégie pure sera noté NE_{fort} (resp. NE_{faible}).

Un équilibre de Nash fort en stratégies pures est donc un profil de stratégies dont on a la certitude qu'il est une réponse optimale aux stratégies des autres joueurs. Un équilibre de Nash faible en stratégies pures est lui un profil de stratégies qui *peut* être une réponse optimale aux stratégies des autres joueurs.

Il est clair que tout équilibre de Nash fort est un équilibre de Nash faible. On a donc $NE_{fort}(G) \subseteq NE_{faible}(G)$.

Exemple 3.2. Soit le L -jeu booléen $G = (N, V, \Gamma, \pi, \Phi)$ suivant :

- * $N = \{1, 2\}$,
- * $V = \{a, b\}$,
- * $\gamma_1 = \gamma_2 = \top$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$,
- * \succeq_1 est défini par : $ab \succ_1 a\bar{b} \succ_1 \bar{a}b$ et $ab \succ_1 \bar{a}\bar{b} \succ_1 \bar{a}b$ (les profils de stratégies $a\bar{b}$ et $\bar{a}\bar{b}$ sont incomparables pour le joueur 1), et
- * \succeq_2 est défini par : $\bar{a}\bar{b} \succ_2 ab \succ_2 a\bar{b} \succ_2 \bar{a}b$.

Ce jeu a un équilibre de Nash fort : $NE_{fort}(G) = \{ab\}$ et deux équilibres de Nash faibles : $NE_{faible}(G) = \{ab, \bar{a}\bar{b}\}$.

Nous devons également raffiner la notion de stratégies dominées, définies initialement à partir des préférences totales (définitions 1.8 et 1.9 (page 22)). L'introduction de préférences partielles, et donc de la notion d'incomparabilité, nous permet d'introduire un nouveau cas de stratégie dominée. Cette nouvelle notion est très faible : toutes les stratégies peuvent être partiellement dominées.

Définition 3.5. Soit i un joueur et \succeq_i sa relation de préférence sur S . La stratégie s_i du joueur i est dite **strictement dominée** s'il existe une autre stratégie s'_i telle que, quelles que soient les stratégies des autres joueurs, s'_i est strictement préférée à s_i pour \succeq_i . Donc, $s_i \in S_i$ est **strictement dominée** si et seulement si :

$$\exists s'_i \in S_i \text{ telle que } \forall s_{-i} \in S_{-i}, (s_i, s_{-i}) \prec_i (s'_i, s_{-i})$$

Soit i un joueur et \succeq_i sa relation de préférence sur S . La stratégie s_i du joueur i est dite **faiblement dominée** s'il existe une autre stratégie s'_i telle que, quelles que soient les stratégies des autres joueurs, s'_i est préférée à s_i pour \succeq_i , et qu'il existe au moins une combinaison des stratégies des autres joueurs telle que s'_i soit strictement préférée à s_i pour \succeq_i . Donc, $s_i \in S_i$ est **faiblement dominée** si $\exists s'_i \in S_i$ telle que :

$$\forall s_{-i} \in S_{-i}, (s_i, s_{-i}) \preceq_i (s'_i, s_{-i}) \text{ et que } \exists s_{-i} \in S_{-i} \text{ telle que } (s_i, s_{-i}) \prec_i (s'_i, s_{-i})$$

Soit i un joueur et \succeq_i sa relation de préférence sur S . La stratégie s_i du joueur i est dite **partiellement dominée** s'il existe une autre stratégie s'_i telle que, quelles que soient les stratégies des autres joueurs, s_i n'est pas strictement préférée à s'_i pour \succeq_i , et qu'il existe au moins une combinaison des stratégies des autres joueurs telle que s'_i soit strictement préférée à s_i pour \succeq_i . Donc, $s_i \in S_i$ est **partiellement dominée** si $\exists s'_i \in S_i$ telle que :

$$\forall s_{-i} \in S_{-i}, (s_i, s_{-i}) \not\prec_i (s'_i, s_{-i}) \text{ et que } \exists s_{-i} \in S_{-i} \text{ telle que } (s_i, s_{-i}) \prec_i (s'_i, s_{-i})$$

Toute stratégie strictement dominée est faiblement dominée ; et toute stratégie faiblement dominée est partiellement dominée. Lorsque la relation \succeq_i est un pré-ordre total, il y a équivalence entre stratégies faiblement et partiellement dominées.

Exemple 3.3. Soit le L -jeu booléen $G = (N, V, \Gamma, \pi, \Phi)$ suivant :

- * $N = \{1, 2\}$,
- * $V = \{a, b\}$,
- * $\gamma_1 = \gamma_2 = \top$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$,
- * \succeq_1 est défini par : $ab \succ_1 \bar{a}\bar{b} \succ_1 \bar{a}b$ et $ab \succ_1 \bar{a}\bar{b} \succ_1 \bar{a}b$ (les profils de stratégies $\bar{a}\bar{b}$ et $\bar{a}b$ sont incomparables pour le joueur 1), et
- * \succeq_2 est défini par : $ab \succeq_2 \bar{a}\bar{b} \succeq_2 \bar{a}b \succ_2 \bar{a}\bar{b}$.

Étudions les stratégies dominées de ce jeu :

- * Joueur 1 : La stratégie a du joueur 1 domine partiellement sa stratégie \bar{a} . En effet, $ab \succ_1 \bar{a}b$, et $\bar{a}\bar{b} \not\prec_1 \bar{a}\bar{b}$.
- * Joueur 2 : La stratégie b du joueur 2 domine faiblement (et partiellement) sa stratégie \bar{b} . En effet, par transitivité de \succeq_2 , $ab \succeq_2 \bar{a}\bar{b}$, et $\bar{a}b \succ_2 \bar{a}\bar{b}$.

La propriété 2.14 (page 51), issue de la théorie des jeux, reste valable ici :

Propriété 3.1.

- * l'ordre d'élimination des stratégies strictement dominées n'affecte pas le résultat final.
- * Par contre, l'ordre d'élimination des stratégies faiblement (et donc partiellement) dominées affecte le résultat final.

Preuve :

- * Montrons que l'ordre d'élimination des stratégies strictement dominées n'affecte pas le résultat final. Supposons que nous sommes en présence d'un jeu G ayant au moins deux stratégies strictement dominées. Ces deux stratégies peuvent être :

- * deux stratégies d'un joueur donné.

Dans ce cas, on suppose que la stratégie s_{i_1} du joueur i est dominée strictement par la stratégie s'_i , et que s_{i_2} est dominée strictement par s''_i . On a alors 3 possibilités, 2 d'entre elles étant équivalentes :

- * $s_{i_2} = s'_i$ (ou $s_{i_1} = s''_i$) : la stratégie s_{i_1} est strictement dominée par s_{i_2} . Comme s''_i domine strictement s_{i_2} , alors s''_i domine strictement s_{i_1} . En effet si on a : $\forall s_{-i} \in S_{-i}, (s_{i_2}, s_{-i}) \succ_i (s_{i_1}, s_{-i})$ et $(s''_i, s_{-i}) \succ_i (s_{i_2}, s_{-i})$ alors

on a, par transitivité de \succ : $\forall s_{-i} \in S_{-i}, s_i'' \succ_i s_{i_1}, s_{-i}$. Dans ce cas, si on élimine s_{i_1} en premier, s_{i_2} sera toujours strictement dominée par s_i'' : et si on élimine s_{i_2} en premier, s_{i_1} sera toujours strictement dominée par s_i'' .

* $s_{i_2} \neq s_i'$ et $s_{i_1} \neq s_i''$: dans ce cas, l'élimination de l'une de ces stratégies n'influe pas sur l'élimination de l'autre.

* des stratégies de deux joueurs différents.

Nous supposons ici que la stratégie s_i du joueur i est dominée strictement par la stratégie s_i' , et que la stratégie s_j du joueur j est dominée strictement par s_j' . On a alors :

$$\forall s_{-i} \in S_{-i}, (s_i', s_{-i}) \succ_i (s_i, s_{-i})$$

$$\forall s_{-j} \in S_{-j}, (s_j', s_{-j}) \succ_j (s_j, s_{-j})$$

Si on élimine s_i en premier, la stratégie s_j sera toujours strictement dominée par s_j' . En effet, on aura toujours $\forall s_{-j} \in S_{-j}, (s_j', s_{-j}) \succ_j (s_j, s_{-j})$ car l'ensemble des $s_{-j} \in S_{-j}$ diminue après la suppression de s_i .

On raisonne de même si on élimine s_j en premier.

* On montre sur l'exemple suivant que l'ordre d'élimination des stratégies faiblement dominées affecte le résultat final :

Soit un L -jeu booléen tel que $N = \{1, 2\}$, $V = \{a, b\}$, $\gamma_1 = \gamma_2 = \top$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, \succeq_1 définie par : $ab \succeq_1 \bar{a}b \succeq_1 a\bar{b} \succ_1 \bar{a}\bar{b}$, et \succeq_2 définie par : $ab \succeq_2 \bar{a}b \succeq_2 \bar{a}\bar{b} \succ_2 a\bar{b}$.

* Pour le joueur 1, a domine faiblement \bar{a} . En effet, on a $ab \succeq_1 \bar{a}b$ et $a\bar{b} \succ_1 \bar{a}\bar{b}$. Eliminons cette stratégie. On obtient alors 2 états, ab et $\bar{a}\bar{b}$, et 2 n'a aucune stratégie dominée.

* Pour le joueur 2, b domine faiblement \bar{b} . En effet, on a $ab \succeq_2 \bar{a}b$ et $a\bar{b} \succ_2 \bar{a}\bar{b}$. Eliminons cette stratégie. On obtient alors 2 états, ab et $\bar{a}b$, et 1 n'a aucune stratégie dominée.

On voit ici que si l'on élimine \bar{a} ou \bar{b} en premier, on obtient des résultats différents.

Comme toute stratégie faiblement dominée est partiellement dominée, ce contre-exemple est valable pour les stratégies partiellement dominées. ■

Nous pouvons à présent étudier des langages de représentation compacte de préférences et essayer de les adapter à notre modèle. Nous étudierons dans un premier temps un langage de représentation dit "graphique", les CP-nets ; puis un langage fondé sur la représentation de préférences en logique propositionnelle, les buts à priorité.

3.2 Jeux booléens et CP-nets

Les CP-nets appartiennent à une famille de langages de représentation dit "graphiques". Ce langage est fondé sur le critère de comparaison *Ceteris Paribus* que nous allons présenter dans la section

suivante.

3.2.1 Ceteris Paribus

Quand un agent exprime en langage naturel une préférence telle que “une table ronde sera mieux dans le salon qu’une table carrée”, il ne veut sans doute pas dire que n’importe quelle table ronde sera préférée à n’importe quelle table carrée. Il veut exprimer le fait qu’il préférera une table ronde à une table carrée si ces deux tables ne diffèrent pas significativement dans leurs autres caractéristiques (la taille, la couleur, le bois utilisé, les finitions ou encore le prix). Le principe qui est à l’œuvre dans l’interprétation de telles préférences est que les alternatives doivent être comparées *toutes choses étant égales par ailleurs*, ou encore *Ceteris Paribus*. Les préférences *Ceteris Paribus* ont été tout d’abord étudiées par von Wright dans [von Wright, 1963], puis ont été considérablement revues par Hansson [Hansson, 1989, 2001], qui en a proposé une généralisation fondée sur les fonctions de représentation et qui en a étudié les propriétés logiques. Ces préférences ont été indépendamment redécouvertes dans la communauté IA par [Doyle et Wellman, 1991; Doyle *et al.*, 1991], et étudiées notamment par [Boutilier *et al.*, 1999].

Soit $V = \{X_1, \dots, X_n\}$ un ensemble de variables, chaque variable X_i étant associée à un domaine $D(X_i)$. Si $\{X_1, \dots, X_p\}$ est contenu dans V , alors $D(\{X_1 \dots X_p\}) = D(X_1) \times \dots \times D(X_p)$, produit cartésien des domaines de chaque variable.

Définition 3.6. *Un sous-ensemble de variables X est **préférentiellement indépendant** à son complément $Y = V \setminus X$ si et seulement si $\forall x_1, x_2 \in D(X)$ et $\forall y_1, y_2 \in D(Y)$ on a :*

$$x_1 y_1 \succeq x_2 y_1 \text{ si et seulement si } x_1 y_2 \succeq x_2 y_2$$

On dit que x_1 est **préférée** à x_2 **ceteris paribus**.

En d’autres mots, la relation de préférence sur les instanciations de X , quand toutes les autres variables sont fixées, est la même quelles que soient les valeurs de ces autres variables.

Définition 3.7. *Soit X , Y et Z trois ensembles non vides disjoints formant une partition de V . X et Y sont **conditionnellement préférentiellement indépendants étant donné Z** si et seulement si $\forall z \in D(Z)$, $\forall x_1, x_2 \in D(X)$ et $\forall y_1, y_2 \in D(Y)$ on a :*

$$x_1 y_1 z \succeq x_2 y_1 z \text{ si et seulement si } x_1 y_2 z \succeq x_2 y_2 z$$

Pour une valeur de Z fixée, la relation de préférence sur les instanciations de X est la même quelles que soient les valeurs des instanciations de Y .

3.2.2 CP-nets

3.2.2.1 Définitions générales

Les CP-nets ont été introduits dans [Boutilier *et al.*, 1999] comme un outil pour représenter compactement les relations de préférences qualitatives. Ce modèle graphique utilise l’indépendance préférentielle conditionnelle pour structurer les préférences d’un agent sous l’hypothèse *Ceteris Paribus*. Ils ont été principalement étudiés dans [Domshlak, 2002], [Boutilier *et al.*, 2004a] ou encore [Boutilier *et al.*, 2004b].

Les CP-nets peuvent être utilisés dans le cadre des jeux booléens de façon à représenter les préférences de chaque joueur. On ne considèrera plus que les buts des joueurs sont représentés par une formule logique, mais directement par un CP-net propositionnel (pour lequel les domaines des variables sont binaires). Dans ce contexte, $\forall x_i \in V, D(x_i) = \{x_i, \bar{x}_i\} = 2^{x_i}$ et $D(\{x_1 \dots x_p\}) = 2^{\{x_1, \dots, x_p\}}$. Ainsi, un élément de $D(x_i)$ correspond à une $\{x_i\}$ -interprétation, et un élément de $D(\{x_1 \dots x_p\})$ est une $\{x_1 \dots x_p\}$ -interprétation.

Définition 3.8. Pour chaque variable $X \in V$, un ensemble de **variables parents**, noté $Pa(X)$, est spécifié. Ces variables sont celles qui influent sur les préférences de l'agent entre les différentes valeurs de X . Formellement, X et $V \setminus (\{X\} \cup Pa(X))$ sont conditionnellement préférentiellement indépendantes étant donné $Pa(X)$.

La **table de préférence conditionnelle** (notée CPT) décrit les préférences de l'agent sur les valeurs de la variable X , étant données toutes les combinaisons des valeurs des variables parents.

Pour chaque instantiation de $Pa(X)$, $CPT(X)$ spécifie un ordre total sur $D(X)$ tel que $\forall x_i, x_j \in D(X)$ on ait soit $x_i \succ x_j$, soit $x_j \succ x_i$.

Soit $V = \{X_1, \dots, X_n\}$ un ensemble de variables. $\mathcal{N} = \langle \mathcal{G}, \mathcal{T} \rangle$ est un **CP-net** sur V , \mathcal{G} étant un graphe orienté sur V , et \mathcal{T} étant un ensemble de tables de préférences conditionnelles $CPT(X_i)$ pour chaque $X_i \in V$. Chaque table de préférence conditionnelle $CPT(X_i)$ est associée à un ordre total \succ_p^i , selon chaque instantiation $p \in D(Pa(X_i))$.

Illustrons ces définitions avec un exemple simple :

Exemple 3.4. Soit le CP-net présenté sur la figure 3.1 qui exprime mes préférences sur le menu du dîner. Ce CP-net est composé de deux variables, S et V qui correspondent respectivement à la soupe et au vin. Je préfère strictement manger une soupe de poisson (S_p) plutôt qu'une soupe de légumes (S_l), tandis que mes préférences sur le vin rouge (V_r) ou blanc (V_b) dépendent de la soupe que je mangerai. En effet je préfère du vin rouge avec une soupe de légumes, mais du blanc avec une soupe de poisson. On a donc $D(S) = \{S_p, S_l\}$ et $D(V) = \{V_r, V_b\}$.

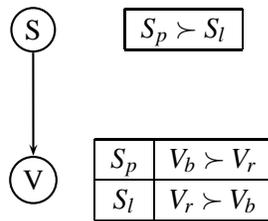


Figure 3.1 — CP-net “Mon dîner simple”

3.2.2.2 Sémantique des CP-nets

La sémantique des CP-nets a été principalement étudiée dans [Domshlak, 2002], [Boutilier *et al.*, 2004a] ou encore [Boutilier *et al.*, 2004b].

Informellement, un CP-net \mathcal{N} est satisfait par \succ si \succ satisfait chacune des préférences conditionnelles exprimées dans les CPTs de \mathcal{N} sous l'interprétation *ceteris paribus*.

Définition 3.9. Soit \mathcal{N} un CP-net sur les variables V . Soit $X \in V$ une variable, $U \subseteq V$ les parents de X dans \mathcal{N} , et $Y = V \setminus (U \cup \{X\})$. Soit \succ_u une relation d'ordre sur $D(X)$ dictée par $CPT(X)$ pour l'instanciation $u \in D(U)$ des parents de X . Soit enfin \succ une relation de préférence sur $D(V)$.

1. Une relation de préférence \succ **satisfait** \succ_u si et seulement si $yux_i \succ yux_j$, pour tout $y \in D(Y)$, à chaque fois que $x_i \succ_u x_j$.
2. Une relation de préférence \succ **satisfait** $CPT(X)$ si et seulement si \succ satisfait \succ_u pour chaque $u \in D(U)$.
3. Une relation de préférence \succ **satisfait** le CP-net \mathcal{N} si et seulement si elle satisfait $CPT(X)$ pour chaque variable X .

Un CP-net \mathcal{N} est **satisfiable** si et seulement si il existe une relation \succ qui le satisfait.

Définition 3.10. Soit \mathcal{N} un CP-net sur les variables V , et $o, o' \in D(V)$ deux états quelconques. \mathcal{N} **entraîne** $o \succ o'$ (i.e. l'état o est préféré à o'), noté $\mathcal{N} \models o \succ o'$, si et seulement si $o \succ o'$ dans chaque relation de préférence qui satisfait \mathcal{N} . On dit alors que $o \succ o'$ est une **conséquence** de \mathcal{N} .

L'ensemble des conséquences $o \succ o'$ d'un CP-net constitue un ordre partiel sur les états : o est préféré à o' dans cette relation d'ordre si et seulement si $\mathcal{N} \models o \succ o'$. Cet ordre partiel peut être représenté par un graphe orienté, que l'on appellera graphe de préférences induit, ou pré-ordre associé à \mathcal{N} :

1. Les nœuds du graphe de préférences correspondent à l'affectation de toutes les variables du CP-net,
2. il y a un arc du nœud o au nœud o' si et seulement si les affectations de o et o' ne diffèrent que pour la valeur d'une seule variable X , et si, étant donné $Pa(X)$, la valeur assignée par o à X est préférée à la valeur assignée par o' à X ⁵.

La fermeture transitive de ce graphe spécifie les ordres partiels sur les états induits par le CP-net. Plus précisément, nous avons $\mathcal{N} \models o \succ o'$ si et seulement si il existe un chemin de o à o' dans le graphe de préférences induit par \mathcal{N} .

Formellement, la relation de préférence induite par \mathcal{N} , représentée par le graphe de préférences induit, est définie comme suit :

Définition 3.11. La relation de préférence sur les états induite par un CP-net \mathcal{N} est dénotée par $\succ_{\mathcal{N}}$, et est définie par $\forall o, o' \in D(V)$, $o \succ_{\mathcal{N}} o'$ si et seulement si $\mathcal{N} \models o \succ o'$.

Exemple 3.4 (page précédente), suite : La figure 3.2 (page suivante) représente la relation de préférence induite par ce CP-net. L'élément du bas ($S_l \wedge V_b$) représente le pire état tandis que l'élément du haut ($S_p \wedge V_b$) est le meilleur état possible.

Nous remarquons ici que nous avons une flèche entre les nœuds ($S_l \wedge V_b$) et ($S_p \wedge V_b$) car nous comparons les états 2 par 2, toutes choses étant égales par ailleurs.

Nous pouvons donc ordonner totalement les états possibles (du plus préféré au moins préféré) :

$$(S_p \wedge V_b) \succ (S_p \wedge V_r) \succ (S_l \wedge V_r) \succ (S_l \wedge V_b)$$

⁵La représentation graphique des préférences que nous utilisons ici dans le cadre des CP-nets est contraire à la représentation utilisée dans la littérature [Boutilier *et al.*, 2004a,b] : dans cette dernière les flèches vont de l'état le moins préféré vers le plus préféré. Afin d'homogénéiser cette section avec la section 3.3 (page 99) nous choisissons ici d'orienter les flèches de l'état le plus préféré vers l'état le moins préféré.

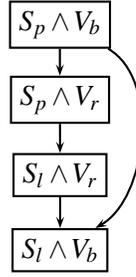


Figure 3.2 — Graphe de préférences induit par le CP-net “Mon dîner simple”

Cette relation \succ est la seule relation de préférence qui satisfasse ce CP-net.

Théorème 3.1 ([Boutilier et al., 2004a]). *Tout CP-net acyclique est satisfiable.*

Lemme 3.1 ([Boutilier et al., 2004a]). *La conséquence préférentielle en ce qui concerne un CP-net est transitive. Donc, si $o \succ_{\mathcal{N}} o'$ et $o' \succ_{\mathcal{N}} o''$ alors $o \succ_{\mathcal{N}} o''$.*

Ce lemme nous permet de simplifier les futurs schémas. Par exemple, sur la figure 3.2 de l'exemple 3.4, nous pouvons supprimer la flèche entre les nœuds $(S_l \wedge V_b)$ et $(S_p \wedge V_b)$ car elle est déductible par transitivité.

Définition 3.12. *Soit \mathcal{N} un CP-net sur les variables V . $D(V)$ dénote l'ensemble de toutes les instances de V .*

*On appelle **ensemble des résultats optimaux de \mathcal{N}** l'ensemble d'états O tel que $\forall o \in O, \mathcal{N} \models o \succ o'$ pour tout $o' \in D(V) \setminus O$.*

Chercher le résultat optimal d'un CP-net \mathcal{N} consiste intuitivement à parcourir le graphe de haut en bas (c'est-à-dire des parents aux descendants), en instanciant chaque variable à sa valeur préférée selon l'instanciation de ses parents. Cette procédure est appelée *recherche avant (forward sweep)*.

Par exemple, sur la figure 3.2 de l'exemple 3.4, le résultat optimal sera $(S_p \wedge V_b)$.

Dans [Boutilier et al., 1999], il est montré que si le CP-net \mathcal{N} est acyclique, le résultat sera unique. Par contre, en cas d'un graphe cyclique, il peut y avoir plusieurs résultats optimaux, comme il peut n'y en avoir aucun.

Lemme 3.2. [Boutilier et al., 1999] *Soit \mathcal{N} un CP-net acyclique sur un ensemble de variables V . La procédure de recherche en avant (forward sweep procedure) construit le résultat optimal dans $D(V)$.*

Les CP-nets sont utilisés dans [Apt et al., 2005], pour représenter des jeux : les CP-nets sont vus comme des jeux en forme normale et vice versa. Il est alors montré dans [Apt et al., 2005] qu'un profil de stratégies est un équilibre de Nash du jeu G si et seulement si c'est un résultat optimal du CP-net $\mathcal{N}(G)$. De même, dans la traduction inverse, il est montré qu'un profil de stratégies est un résultat optimal d'un CP-net si et seulement si c'est un équilibre de Nash du jeu associé. Nous en parlerons plus longuement dans la section 4.2.1 (page 127).

3.2.2.3 Utilisation des CP-nets dans les jeux booléens

Les définitions classiques des CP-nets doivent être adaptées pour pouvoir être appliquées à des jeux booléens. En effet, nous sommes ici dans un contexte particulier où les variables que nous manipulons

sont booléennes, et où nous avons plusieurs joueurs. Dans ce contexte, chaque joueur doit pouvoir choisir la stratégie qui lui permettra d'obtenir le "meilleur résultat possible". Pour ce faire, nous pouvons utiliser les notions définies en section 3.1 (page 71) pour calculer les équilibres de Nash et les stratégies dominées, à condition de disposer d'une relation de préférence pour chaque joueur sur l'ensemble des profils de stratégies. Les CP-nets interviennent ici puisqu'ils constituent un outil de représentation des préférences et fourniront ainsi les relations de préférences de chaque joueur.

Définition 3.13. *La table de préférence conditionnelle pour un jeu booléen (notée $CPT_i(X)$) décrit les préférences du joueur i sur les valeurs de la variable X , étant données toutes les combinaisons des valeurs des variables parents.*

Pour chaque instantiation p de $Pa_i(X)$, $CPT_i(X)$ spécifie un ordre total tel que l'on ait soit $x \succ_{i,p} \bar{x}$, soit $\bar{x} \succ_{i,p} x$.

Définition 3.14. *Un CP-jeu booléen est un 5-tuple $G = (N, V, \Gamma, \pi, \Phi)$, où $N = \{1, \dots, n\}$ est un ensemble de joueurs, $V = \{x_1, \dots, x_n\}$ est un ensemble de variables, $\Phi = \langle \mathcal{N}_1, \dots, \mathcal{N}_n \rangle$, chaque \mathcal{N}_i étant un CP-net sur V et, pour tout $i \in N$, $\succeq_i = \succeq_{\mathcal{N}_i}$.*

La grande différence entre les CP-nets classiques et ceux que nous introduisons dans les jeux booléens est l'introduction de contraintes. Les CP-nets avec contraintes ont été étudiés parallèlement dans [Boutilier *et al.*, 2004b] et [Prestwich *et al.*, 2004].

Dans [Prestwich *et al.*, 2004], les préférences sont exprimées sur toutes les issues du jeu, les états ne respectant pas les contraintes étant considérés comme inaccessibles. S'il existe un état accessible et non dominé, ce dernier est le résultat optimal.

Dans [Boutilier *et al.*, 2004b], les préférences exprimées dans les CPTs du CP-net doivent respecter les contraintes : si par exemple une contrainte est du type $a \leftrightarrow b$, et que $a \in P(b)$, les préférences sur b seront identiques à celles sur a (si a est indépendant et que $a \succ \bar{a}$, alors on aura $b \succ b'$). Un algorithme, appelé Search-CP, permet alors de trouver l'ensemble de résultats optimaux qui sont accessibles (c'est-à-dire qui respectent les contraintes), et qui ne sont pas dominés par un autre état accessible. Nous choisissons ici d'utiliser cette méthode.

Exemple 3.5. *Soit le jeu $G = (N, V, \Gamma, \pi, \Phi)$ suivant :*

- * $N = \{1, 2, 3\}$
- * $V = \{a, b, c\}$, avec $D(a) = \{a, \bar{a}\}$, $D(b) = \{b, \bar{b}\}$ et $D(c) = \{c, \bar{c}\}$.
- * $\gamma_1 = \gamma_2 = \gamma_3 = \top$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$,
- * *Le but du joueur 1 est représenté par le CP-net et la relation de préférence induite figure 3.3 (page suivante),*
- * *Le but du joueur 2 est représenté par le CP-net et la relation de préférence induite figure 3.4 (page suivante),*
- * *Le but du joueur 3 est représenté par le CP-net et la relation de préférence induite figure 3.5 (page 81).*

Pour calculer facilement les équilibres de Nash de ce jeu, nous mettons en exergue les relations qui nous intéressent pour chaque joueur. Ces relations apparaissent sous la forme de flèches en pointillé sur la figure 3.3 (page suivante).

En effet, pour le joueur 1, qui contrôle la variable a , nous comparons les états abc et $\bar{a}bc$, les états $ab\bar{c}$, $\bar{a}b\bar{c}$, les états $a\bar{b}c$ et $\bar{a}\bar{b}c$, et enfin les états $ab\bar{c}$ et $\bar{a}b\bar{c}$.

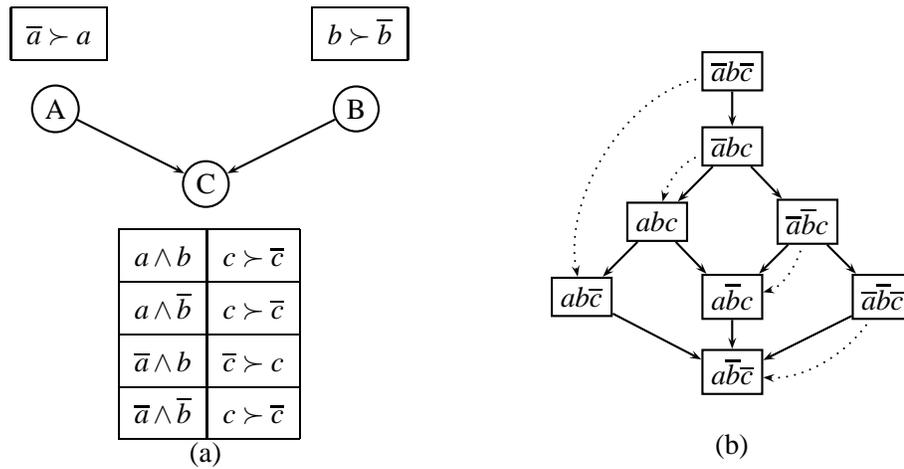


Figure 3.3 — CP-net des préférences et le pré-ordre associé du joueur 1

Cela revient à comparer pour chaque joueur uniquement les stratégies de ce dernier pour chaque combinaison de stratégies des autres joueurs. Notons que ces flèches en pointillé représentent une relation d'ordre transitive.

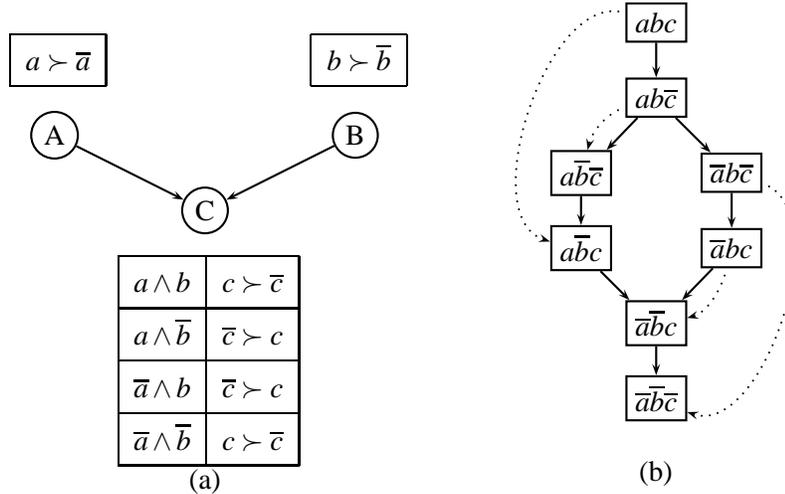


Figure 3.4 — CP-net des préférences et le pré-ordre associé du joueur 2

Le calcul des équilibres de Nash faibles et forts nous donne ici comme résultat :

$$NE_{faible} = NE_{fort} = \{\bar{a}\bar{b}\bar{c}\}$$

Etudions à présent les stratégies dominées de ce jeu :

- * *Joueur 1 : Il est possible d'éliminer la stratégie a de ce joueur, qui est strictement dominée par la stratégie a-bar.*
- * *Joueur 2 : Une fois la stratégie a du joueur 1 éliminée, la stratégie b du joueur 2 domine strictement sa stratégie b-bar. On élimine cette dernière.*
- * *Joueur 3 : Après élimination de ces 2 stratégies, c-bar domine c pour le joueur 3.*

En éliminant itérativement les stratégies strictement dominées, on obtient donc un profil de stratégies résultat : $\{\bar{a}\bar{b}\bar{c}\}$, qui se trouve être aussi l'équilibre de Nash faible et fort. De plus, ce sont ici des stratégies strictement dominées. L'ordre d'élimination de ces stratégies n'affecte donc pas le résultat.

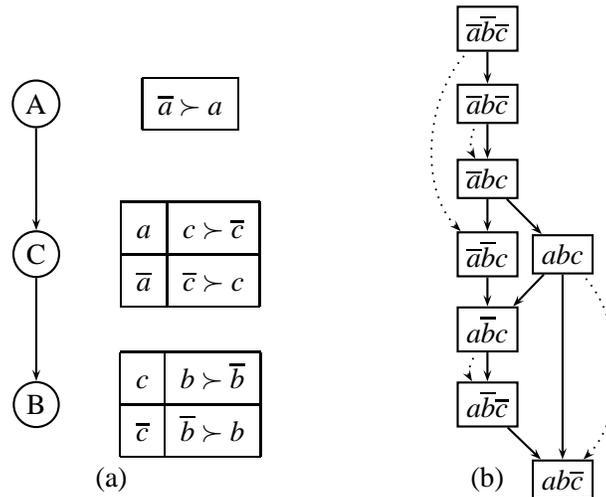


Figure 3.5 — CP-net des préférences et le pré-ordre associé du joueur 3

Etudions à présent un exemple avec des contraintes, en choisissant la méthode introduite dans [Boutilier *et al.*, 2004b] : nous ne représenterons que les préférences respectant les contraintes données.

Exemple 3.6. Soit le jeu $G = (N, V, \Gamma, \pi, \Phi)$ suivant :

- * $N = \{1, 2\}$
- * $V = \{a, b, c\}$, avec $D(a) = \{a, \bar{a}\}$, $D(b) = \{b, \bar{b}\}$ et $D(c) = \{c, \bar{c}\}$.
- * $\gamma_1 = \{a \leftrightarrow b\}$, $\gamma_2 = \top$,
- * $\pi_1 = \{a, b\}$, $\pi_2 = \{c\}$,
- * Le but du joueur 1 est représenté par le CP-net et la relation de préférence induite figure 3.6,
- * Le but du joueur 2 est représenté par le CP-net et la relation de préférence induite figure 3.7 (page suivante),

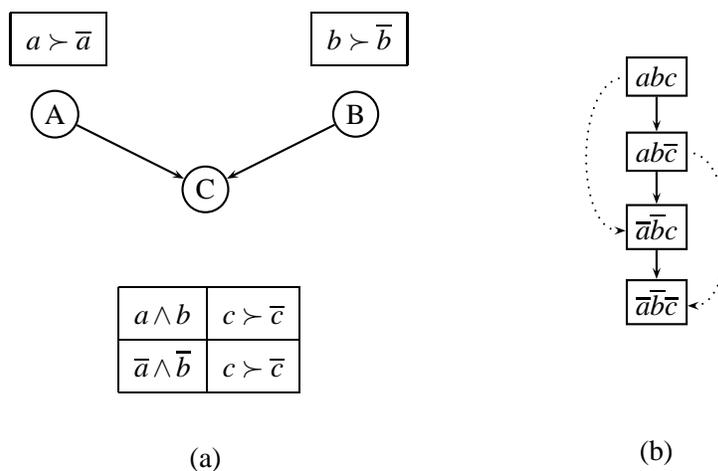


Figure 3.6 — CP-net des préférences et le pré-ordre associé du joueur 1

On remarque que le joueur 1 n'exprime pas ses préférences sur c s'il choisit $\bar{a}b$ ou $a\bar{b}$, ces deux états étant inaccessibles. Il est cependant possible de construire la relation de préférence associée en

utilisant la relation de transitivité (lemme 3.1 (page 78)) : nous savons ici que les états contenant ab seront toujours préférés aux états contenant $\bar{a}\bar{b}$.

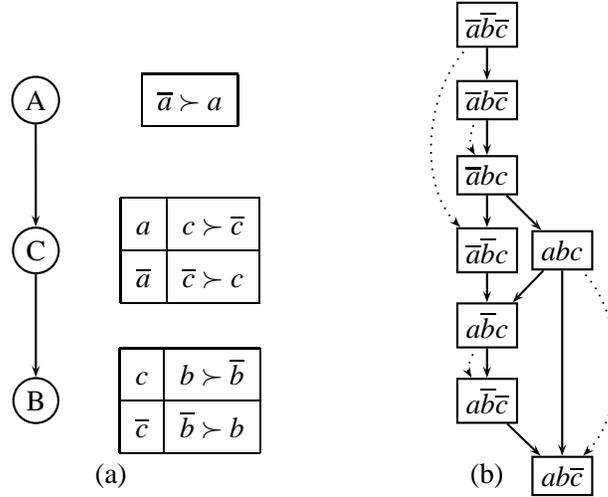


Figure 3.7 — CP-net des préférences et le pré-ordre associé du joueur 2

Pour calculer les équilibres de Nash de ce jeu, on ne s'occupe que des profils de stratégies présents dans les relations de préférence des deux joueurs : par exemple, le profil de stratégies $\bar{a}\bar{b}\bar{c}$ viole les contraintes du joueur 1 et ne peut donc pas être un PNE. Le calcul des équilibres de Nash faibles et forts nous donne ainsi ici comme résultat :

$$NE_{faible} = NE_{fort} = \{abc\}$$

L'élimination des stratégies dominées fournit le même résultat.

3.2.3 Propriétés des CP-jeux booléens

3.2.3.1 Stratégies dominées

Il faut tout d'abord remarquer que, dans ce cadre de travail, il ne peut y avoir "égalité" entre deux profils de stratégies, les CP-nets ne permettant pas de modéliser l'indifférence⁶. On sait donc alors que les stratégies faiblement et strictement dominées sont identiques. Plusieurs propriétés découlent de cette constatation.

Propriété 3.2. Soit G un CP-jeu booléen tel que le graphe G_i pour le joueur i est acyclique. i a une stratégie dominée si et seulement si i contrôle une variable indépendante (pour \succeq_i) de toutes les autres variables, c'est-à-dire si et seulement si il existe une variable $v \in \pi_i$ telle que $Pa(v) = \emptyset$.

Dans ce cas, s_i domine s'_i si et seulement si $(s_i)_v \succ_i (s'_i)_v$.

Preuve :

\Rightarrow Supposons que $\forall v \in \pi_i, Pa(v) \neq \emptyset$. Comme le graphe G_i est acyclique, nous savons qu'il existe $v \in \pi_i$ telle que $Pa(v) \subseteq \pi_{-i}$. Il existe donc une instantiation

⁶Hypothèse que nous relâcherons dans la section 3.2.4 (page 96)

u_v de tous les parents de v telle que $v \succ_{i,u_v} \bar{v}$; et il existe une autre instantiation u'_v de tous les parents de v telle que $\bar{v} \succ_{i,u'_v} v$ (sinon le graphe pourrait être simplifié de façon à avoir $Pa(v) = \emptyset$). Pour tout $s_i \in S_i$, il existe $s'_i \in S_i$ et il existe $s_{-i} \in S_{-i}$ tels que soit $(s_i, s_{-i}) \prec_{i,u_v} (s'_i, s_{-i})$, soit $(s_i, s_{-i}) \prec_{i,u'_v} (s'_i, s_{-i})$. i n'a donc aucune stratégie dominée.

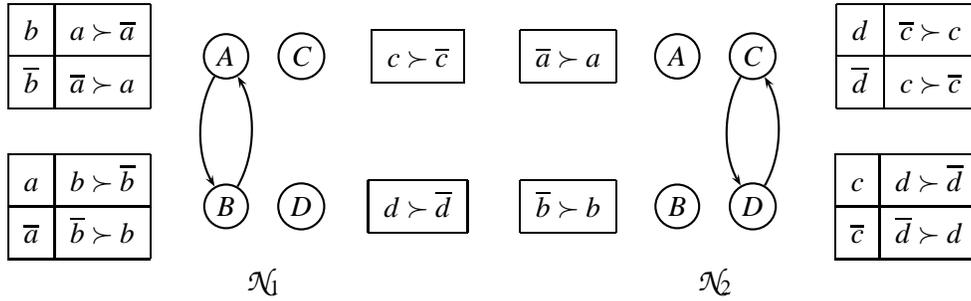
\Leftarrow Supposons que i est un joueur, et qu'il existe $v \in \pi_i$ tel que $Pa(v) = \emptyset$.

On a donc soit $v \succ_i \bar{v}$, soit $\bar{v} \succ_i v$. La stratégie s_i , définie en assignant la meilleure valeur à v (*i.e.* v dans le premier cas), est préférée par i à tous les autres s'_i définies en assignant l'autre valeur à v (*i.e.* \bar{v} dans le premier cas). Comme $Pa(v) = \emptyset$, nous savons que $\forall s_{-i} \in S_{-i}$, $(s_i, s_{-i}) \succ_i (s'_i, s_{-i})$.

■

Si \mathcal{G}_i n'est pas acyclique, i peut avoir une stratégie dominée même si $\nexists v \in \pi_i$ tel que $Pa(v) = \emptyset$.

Exemple 3.7. Soit $G = (N, V, \pi, \Gamma, \Phi)$, avec $N = \{1, 2\}$, $V = \{a, b, c, d\}$, $\pi_1 = \{a, b\}$, $\pi_2 = \{c, d\}$, $\gamma_1 = \gamma_2 = \top$, \mathcal{N}_1 et \mathcal{N}_2 sont représentés sur la figure suivante :



La stratégie ab domine toutes les autres stratégies du joueur 1, même si $Pa(a) = b$ et $Pa(b) = a$.

Dans \mathcal{N}_1 , la configuration de a et b , contrôlées par le joueur 1, est la même que la configuration de c et d dans \mathcal{N}_2 , qui sont contrôlés par le joueur 2. Pourtant, 2 n'a aucune stratégie dominée.

3.2.3.2 Equilibres de Nash en stratégies pures

Lemme 3.3. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen tel que les graphes \mathcal{G}_i sont tous acycliques, et soit $i \in N$. Pour tout $s_{-i} \in S_{-i}$ il existe une meilleure réponse pour i , c'est-à-dire, une stratégie $s_i^* \in S_i$ telle que pour tout $s'_i \neq s_i^*$, $(s_i^*, s_{-i}) \succ_i (s'_i, s_{-i})$.

Preuve du lemme : Supposons que s_i^* n'existe pas. Dans ce cas, $\exists s_{-i} \in S_{-i}$ tel que $\forall s_i \in S_i$, $\exists s'_i \neq s_i$ tel que $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$. Puis, il existe un $s''_i \neq s'_i$ tel que $(s''_i, s_{-i}) \succ_i (s'_i, s_{-i})$, et ainsi de suite. Comme chaque joueur a un nombre fini de stratégies, et que \succ_i est transitive, $\exists s_i, s'_i \in S_i$ tels que $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$, $(s''_i, s_{-i}) \succ_i (s'_i, s_{-i})$ et $(s_i, s_{-i}) \succ_i (s''_i, s_{-i})$, ce qui est en contradiction avec le fait que \mathcal{G}_i est acyclique.

■

Comme s_i^* ne dépend que de s_{-i} , la meilleure réponse de i à s_{-i} peut être notée par $r_i(s_{-i})$. Alors :

Lemme 3.4. s est un équilibre de Nash fort en stratégies pures si et seulement si pour tout i , $s_i = r_i(s_{-i})$.

Preuve du lemme :

\Rightarrow Soit $s = (s_1, \dots, s_n)$ un équilibre de Nash fort en stratégies pures : $\forall i \in N$, $\forall s'_i \in S_i$, $(s_i, s_{-i}) \succeq_i (s'_i, s_{-i})$. Mais, par définition d'un CP-net, nous ne pouvons pas avoir d'égalité entre deux profils de stratégies. Nous savons donc que $(s_i, s_{-i}) \neq_i (s'_i, s_{-i})$ si $s_i \neq s'_i$. Donc, $\forall i \in N$, $\forall s'_i \neq s_i \in S_i$, $(s_i, s_{-i}) \succ_i (s'_i, s_{-i})$, et alors $\forall i \in N$, $s_i = r_i(s_{-i})$.

\Leftarrow $\forall i \in N$, soit $s_i = r_i(s_{-i})$. Par définition, nous avons $\forall s'_i \neq s_i \in S_i$, $(s_i, s_{-i}) \succ_i (s'_i, s_{-i})$. Alors, $\forall i \in N$, $\forall s'_i \in S_i$, $(s_i, s_{-i}) \succeq_i (s'_i, s_{-i})$: s est un équilibre de Nash fort en stratégies pures. ■

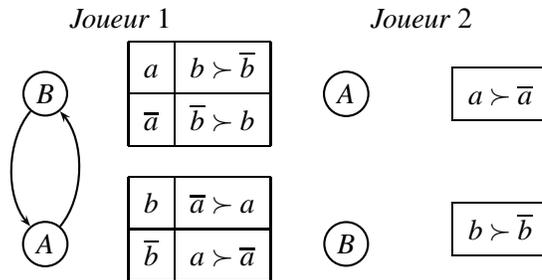
Propriété 3.3. Dans un CP-jeu booléen acyclique, les équilibres de Nash faibles et forts sont identiques.

Preuve : Il est évident que $NE_{fort} \subseteq NE_{faible}$. Vérifions que $NE_{faible} \subseteq NE_{fort}$.

Supposons que s est un équilibre de Nash faible, mais pas un équilibre de Nash fort. D'après le Lemme 3.4, il existe alors un $i \in N$ tel que $s_i \neq r_i(s_{-i})$. Mais, $(r_i(s_{-i}), s_{-i}) \succ_i (s'_i, s_{-i})$ pour tout s'_i , et en particulier pour $s'_i = s_i$. s ne peut donc pas être un équilibre de Nash faible en stratégies pures. ■

Si les graphes ne sont pas tous acycliques, cette propriété n'est plus garantie :

Exemple 3.8. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen, avec $N = \{1, 2\}$, $V = \{a, b\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\gamma_1 = \gamma_2 = \top$. Les CP-nets des joueurs sont représentés sur la figure suivante :



Les équilibres de Nash sont les suivants : $NE_{fort} = \emptyset$, $NE_{faible} = \{ab, \bar{a}b\}$.

En conséquence, on parlera dans la suite d'équilibres de Nash en stratégies pures (au lieu d'équilibres forts ou faibles) aussitôt que le graphe associé à chaque joueur est acyclique.

3.2.3.3 CP-jeux booléens avec graphe acyclique commun à tous les joueurs

Examinons à présent ce qui se passe lorsque tous les joueurs ont le même graphe, ce dernier étant acyclique.

Exemple 3.9. Soit le jeu $G = (N, V, \pi, \Gamma, \varphi)$ suivant :

- * $N = \{1, 2, 3\}$
- * $V = \{a, b, c, d, e\}$, avec $D(a) = \{a, \bar{a}\}$, $D(b) = \{b, \bar{b}\}$, $D(c) = \{c, \bar{c}\}$, $D(d) = \{d, \bar{d}\}$, $D(e) = \{e, \bar{e}\}$
- * $\gamma_1 = \gamma_2 = \gamma_3 = \top$,
- * $\pi_1 = \{a, d\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c, e\}$,
- * Les buts des trois joueurs sont représentés par un même graphe acyclique. Par contre, les joueurs n'ont pas les mêmes tables de préférences conditionnelles (représentées figure 3.8 pour le joueur 1, figure 3.9 pour le joueur 2 et figure 3.10 (page suivante) pour le joueur 3).

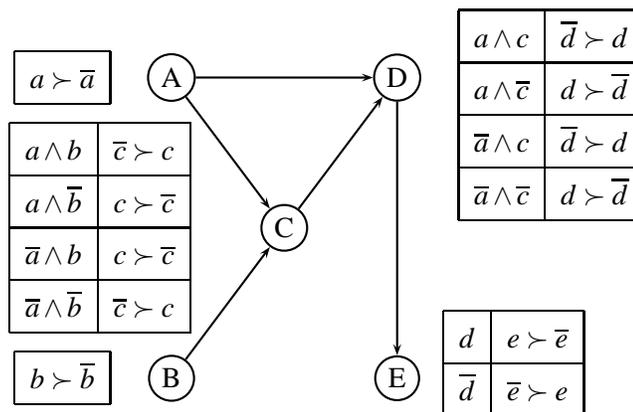


Figure 3.8 — CP-net des préférences du joueur 1

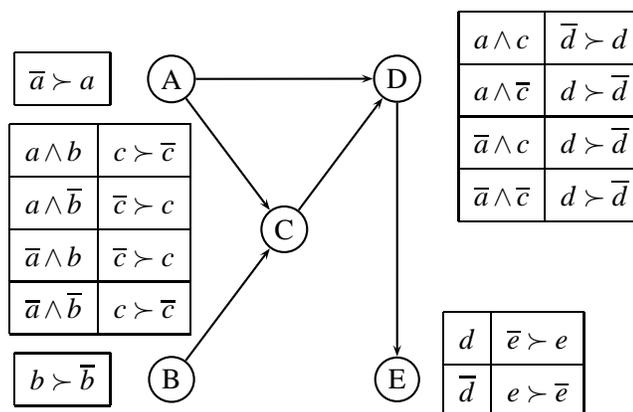


Figure 3.9 — CP-net des préférences du joueur 2

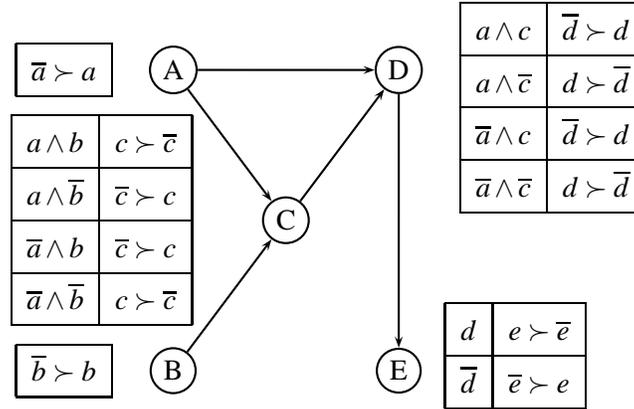


Figure 3.10 — CP-net des préférences du joueur 3

En étudiant ce jeu, on constate tout d'abord que le joueur 1 va instancier la variable a , qui est indépendante préférentiellement, de façon à satisfaire au mieux ses préférences. a sera donc instanciée à vrai. De même, le joueur 2 instanciera b à vrai.

Le joueur 3 sait que les joueurs 1 et 2 ont instancié leurs variables a et b de cette façon : il connaît leurs préférences et sait que ce sont des joueurs rationnels. La variable c devient donc indépendante préférentiellement selon l'instanciation de a et b . Il va donc instancier la variable c à vrai.

L'instanciation de la variable c par le joueur 3 permet à 1 d'instancier d à faux, ce qui permet enfin à 3 d'instancier e à faux.

On obtient donc le profil de stratégies $s = abc\bar{d}\bar{e}$. s est le seul équilibre de Nash de ce jeu.

En suivant le raisonnement donné par cet exemple, nous obtenons ainsi la propriété suivante :

Propriété 3.4. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen tel que les graphes \mathcal{G}_i sont tous identiques ($\forall i, j, \mathcal{G}_i = \mathcal{G}_j$) et acycliques.

Ce jeu G ainsi défini aura alors un et un seul équilibre de Nash en stratégie pure.

La preuve de cette propriété utilise la notion de *recherche du résultat optimal* (Outcome optimization queries) introduite par [Boutilier et al., 1999, 2004a] (voir définition 3.12 (page 78)), qui consiste intuitivement à parcourir le graphe des parents aux descendants, en instanciant chaque variable à sa valeur préférée selon l'instanciation de ses parents. Cette procédure est appelée *recherche avant* (forward sweep).

C'est exactement cette procédure que nous appliquons pour prouver la propriété précédente, et qui nous permet de déterminer l'unique équilibre de Nash d'un jeu dont les CP-nets représentant les buts des joueurs forment tous le même graphe acyclique.

Preuve : Nous allons faire cette démonstration en deux temps : tout d'abord nous démontrerons l'existence de cet équilibre de Nash, puis nous en démontrerons l'unicité.

* **Existence :**

Soit (x_1, \dots, x_k) un ordre topologique sur V respectant \mathcal{G}_i (pour tout i puisque les graphes sont identiques). On définit un équilibre de Nash en stratégies

pures s comme suit.

Soit a le joueur contrôlant la variable x_1 . Par construction, nous avons $Pa(x_1) = \emptyset$, et soit $x_1 \succ_a \bar{x}_1$, soit $\bar{x}_1 \succ_a x_1$. Soit la stratégie s_a assignant la meilleure valeur à x_1 (*i.e.* x_1 dans le premier cas, \bar{x}_1 dans le second). Il est possible de procéder de même pour toutes les variables selon l'ordre topologique, puisque par définition lorsque l'on doit instancier la variable x_i , les valeurs de tous ses parents ont été auparavant fixées. On obtient donc un profil de stratégies s .

Nous allons à présent montrer que ce profil de stratégies $s = s_1 \dots s_n$ est un équilibre de Nash en stratégies pures. Supposons qu'il n'en est pas un. Dans ce cas, soit un joueur i et une stratégie s'_i tels que $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$. Soit x la première variable dans π_i , en respectant l'ordre topologique défini ci-dessus, telle que $s_i[x] \neq s'_i[x]$. Alors, par construction de s_i , nous avons $s_i[x] \succ_{i, u_x} s'_i[x]$, où u_x est l'instanciation de tous les parents de x . Nous savons que cette instanciation est la même dans (s_i, s_{-i}) et (s'_i, s_{-i}) car nous avons choisi la première variable qui est différente dans s'_i et s_i selon l'ordre topologique. Donc, si l'on remplace $s'_i[x]$ par $s_i[x]$ dans s'_i , nous obtiendrons une meilleure stratégie pour i . Le même raisonnement montre que remplacer toutes les $s'_i[x]$ par $s_i[x]$ dans s'_i permet d'obtenir une meilleure stratégie, s_i . On a donc $(s_i, s_{-i}) \succ_i (s'_i, s_{-i})$, et $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$, ce qui est en contradiction avec le fait que tous les CP-nets acycliques sont satisfiables.

* **Unicité :**

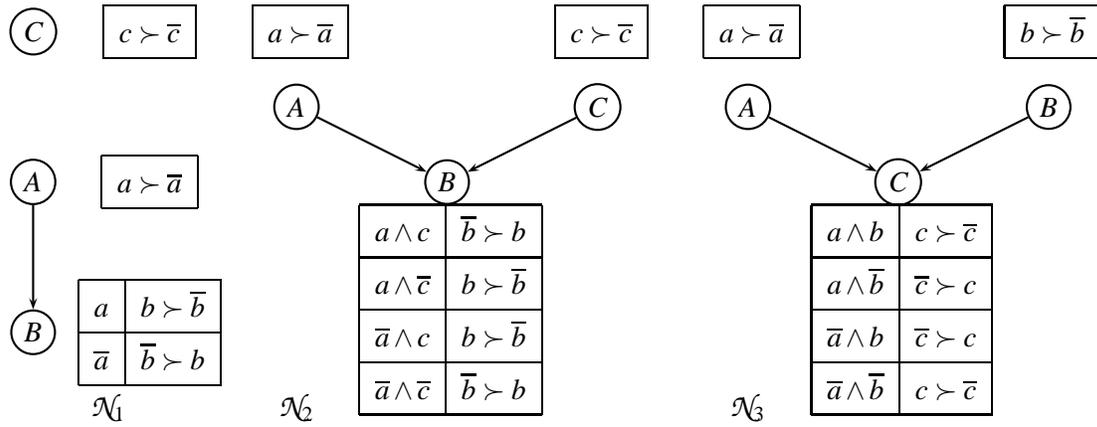
L'unicité peut se déduire facilement grâce à la construction ci-dessus. Supposons que G a deux équilibres de Nash différents, et que (x_1, \dots, x_k) est l'ordre topologique sur V respectant tous les \mathcal{G}_i . Une fois que cet ordre est fixé, la construction décrite ci-dessus est déterministe, et on obtient un unique s_c . Il existe donc au moins un équilibre de Nash s qui n'est pas construit selon cette méthode.

Soit x la première variable, selon l'ordre topologique, qui n'a pas la même valeur dans s et s_c . Soit i le joueur qui contrôle x , et soit u l'instanciation de toutes les variables précédant x selon l'ordre topologique. Par construction, u est identique dans s et s_c , et donne l'instanciation de toutes les variables de $Pa(x)$. Supposons que $\bar{x} \succ_{i, u} x$ (le cas symétrique est identique). Alors, par construction, s_c assigne \bar{x} à x , et donc s lui assigne x . Donc, si l'on remplace x par \bar{x} dans s , on obtient une stratégie (strictement) meilleure que s , ce qui est en contradiction avec le fait que s est un équilibre de Nash en stratégies pures pour G .

■

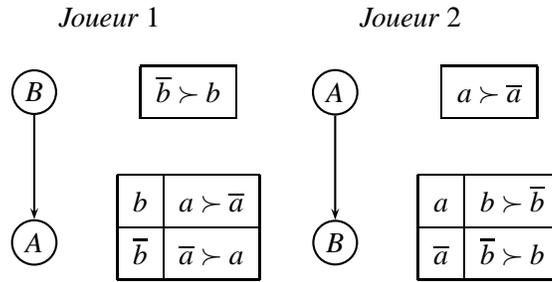
Si les graphes sont acycliques, mais ne sont pas identiques, ni l'existence ni l'unicité ne sont garanties :

Exemple 3.10. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen, avec $N = \{1, 2, 3\}$, $V = \{a, b, c\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\gamma_1 = \gamma_2 = \gamma_3 = \top$. Les CP-nets des joueurs sont représentés sur la figure suivante :



G n'a aucun équilibre de Nash en stratégies pures.

Exemple 3.11. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen, avec $N = \{1, 2\}$, $V = \{a, b\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\gamma_1 = \gamma_2 = \top$. Les CP-nets des joueurs sont représentés sur la figure suivante :



Ce jeu a 2 équilibres de Nash en stratégies pures : $NE = \{ab, \bar{a}\bar{b}\}$

En général, les graphes \mathcal{G}_i pour $i \in \{1, \dots, n\}$ ne sont pas identiques. Pourtant il est possible de les rendre identiques une fois que l'on a remarqué qu'un CP-net $\langle \mathcal{G}, \mathcal{T} \rangle$ peut être exprimé par un CP-net $\langle \mathcal{G}', \mathcal{T}' \rangle$ si l'ensemble des arcs dans \mathcal{G} est contenu dans l'ensemble des arcs dans \mathcal{G}' . On peut alors prendre comme graphe commun \mathcal{G} à tous les joueurs le graphe dont l'ensemble des arcs est l'union des ensembles d'arcs de $\mathcal{G}_1, \dots, \mathcal{G}_n$. Il se peut alors que le graphe résultant ne soit pas acyclique, auquel cas la propriété 3.4 (page 86) n'est pas applicable. Formellement :

Définition 3.15. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen tel que $\forall i \in N$, les graphes \mathcal{G}_i sont tous acycliques.

Pour chaque joueur i , $\mathcal{G}_i = \langle V, \text{Arc}_i \rangle$, V étant l'ensemble des nœuds du graphe⁷, et Arc_i représentant l'ensemble des arcs orientés du graphe représentant le CP-net de i .

L'union des graphes de G sera le graphe $\mathcal{G} = \langle V, \text{Arc}_1 \cup \dots \cup \text{Arc}_n \rangle$.

Définition 3.16. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen tel que $\forall i \in N$, les graphes \mathcal{G}_i sont tous acycliques.

On appelle $G^* = (N, V, \pi, \Gamma, \Phi^*)$ le **jeu équivalent par réécriture de G** dans lequel on a remplacé le graphe issu du CP-net représentant le but de chacun des joueurs par l'union de tous les graphes des joueurs.

⁷ici identique pour chacun des joueurs puisque c'est l'ensemble des variables du jeu.

Les tables de préférences conditionnelles sont modifiées de façon à correspondre avec le nouveau graphe, tout en donnant les mêmes préférences qu'avant : si l'arc (X, Y) est ajouté au graphe, la table de préférence conditionnelle de la variable Y sera la même que précédemment pour chaque instantiation $x \in D(X)$. Plus formellement, si on note \succ_i^y la relation associée à la table de préférence conditionnelle $CPT_i(Y)$ pour le CP-net du joueur i dans le jeu G , on a pour le jeu G^* : $\forall x \in D(X), \succ_{i,x}^y = \succ_{i,\bar{x}}^y = \succ_i^y$.

Propriété 3.5. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen tel que $\forall i \in N$, les graphes G_i sont tous acycliques. Soit $G^* = (N, V, \pi, \Gamma, \Phi^*)$ le jeu équivalent par réécriture de G .

Le jeu G^* permet d'obtenir les mêmes pré-ordres sur les profils de stratégies que le jeu G . On dit que G et G^* sont équivalents.

Preuve : Soit, pour chaque joueur $i \in N$, \succeq_i la relation de préférences qui satisfait le CP-net du joueur i pour le jeu G . Montrons que \succeq_i satisfait le CP-net du joueur i pour le jeu G^* .

$\forall X \in V$, deux cas se présentent :

1. X a les mêmes parents dans les CP-nets de G^* et de G . Dans ce cas là, la relation \succeq_i^x est la même $\forall x \in D(X)$.
2. X n'a pas les mêmes parents dans le CP-net du jeu G^* que dans le CP-net du jeu G . Un arc a été ajouté. Notons $\langle Y, X \rangle$ cet arc. On a alors $\forall y \in D(Y), \succeq_{i,y}^x = \succeq_i^x$. On a donc bien la même relation.

La relation \succeq_i satisfait donc bien le CP-net du joueur i pour le jeu G^* .

■

Remarque : Si le graphe du jeu G^* est acyclique, la propriété 3.4 (page 86) peut s'appliquer, et donc G^* a un et un seul équilibre de Nash en stratégies pures. Comme G et G^* définissent les mêmes relations de préférences, ce dernier est également le seul équilibre de Nash en stratégies pures de G (par contre, si le graphe de G^* est cyclique, ni l'existence ni l'unicité des équilibres de Nash ne sont garanties comme on peut le voir sur les exemples 3.10 (page 87) et 3.11 (page précédente)).

Exemple 3.12. Soit le jeu $G = (N, V, \pi, \Gamma, \Phi)$ suivant :

- * $N = \{1, 2\}$
- * $V = \{a, b, c\}$, avec $D(a) = \{a, \bar{a}\}$, $D(b) = \{b, \bar{b}\}$ et $D(c) = \{c, \bar{c}\}$.
- * $\gamma_1 = \gamma_2 = \top$,
- * $\pi_1 = \{a, b\}$, $\pi_2 = \{c\}$,
- * Le but du joueur 1 est représenté sur la figure 3.11 (page suivante),
- * Le but du joueur 2 est représenté sur la figure 3.12 (page suivante).

On calcule à présent l'union des graphes issus des CP-nets représentant les buts de ces deux joueurs, et l'on crée ainsi le jeu G^* équivalent par réécriture au jeu G . Les buts des deux joueurs sont alors représentés par les CP-nets représentés figure 3.13 (page suivante) et 3.14 (page 91).

On constate sur la figure 3.13 (page suivante) que pour calculer les équilibres de Nash de ce jeu, il faudrait comparer les profils de stratégies abc , $a\bar{b}c$, $\bar{a}bc$ et $\bar{a}\bar{b}c$ pour le joueur 1, pourtant $a\bar{b}c$ et $\bar{a}\bar{b}c$ sont incomparables. Cette "incomparabilité" est représentée sur la figure par une ligne avec des pointillés plus larges.

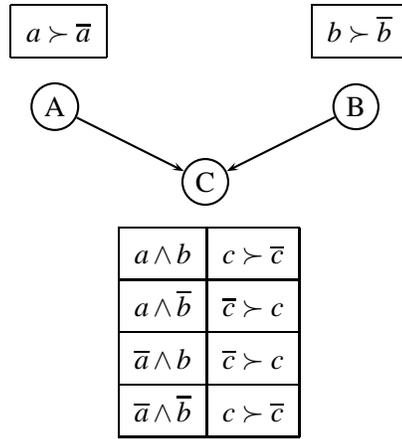


Figure 3.11 — CP-net des préférences du joueur 1

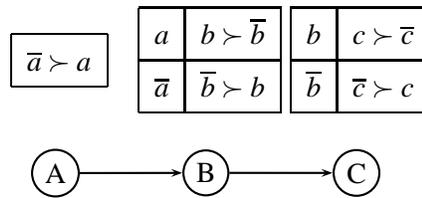


Figure 3.12 — CP-net des préférences du joueur 2

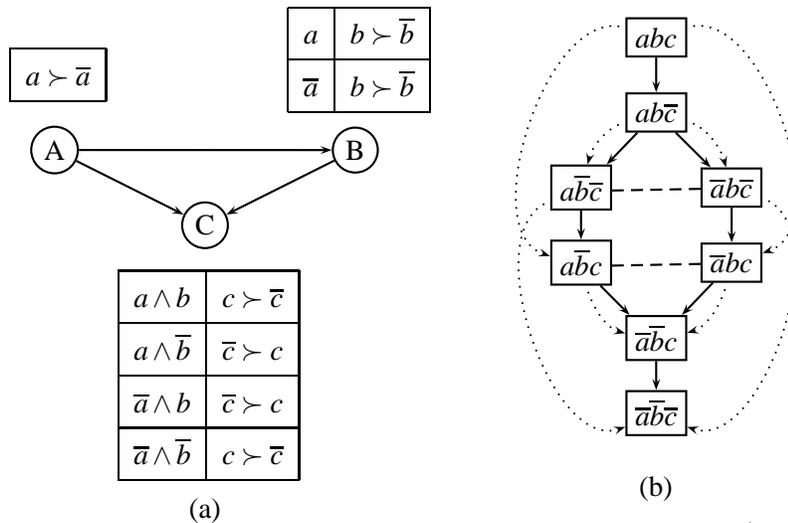


Figure 3.13 — CP-net des préférences et le pré-ordre associé du joueur 1 pour le jeu G^* équivalent par réécriture de G

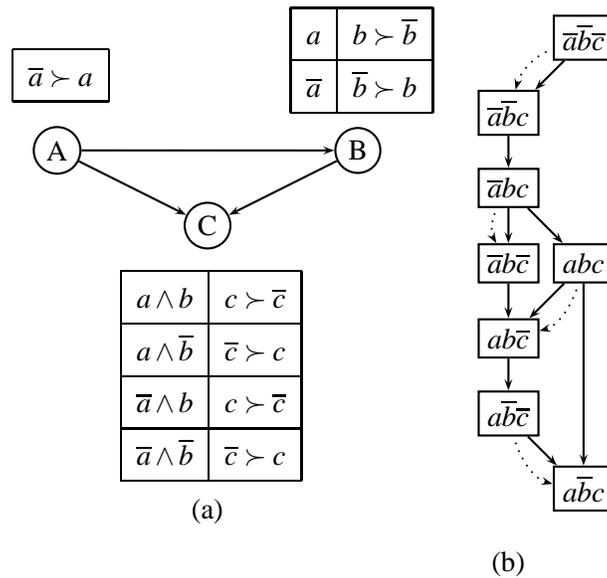


Figure 3.14 — CP-net des préférences et le pré-ordre associé du joueur 2 pour le jeu G^* équivalent par réécriture de G

L'union des graphes donne ici un graphe acyclique. On peut donc appliquer la propriété 3.4 (page 86) : le jeu G a un et un seul équilibre de Nash : $NE = \{abc\}$

Une dernière condition (moins intéressante en pratique car assez forte) garantit l'existence et l'unicité d'un équilibre de Nash en stratégies pures. Cette condition établit que si toutes les variables contrôlées par un joueur sont indépendantes des variables contrôlées par les autres joueurs (en d'autres termes, les variables parents des variables contrôlées par le joueur i sont contrôlées par i), et les CP-nets des joueurs sont acycliques, alors chaque joueur est capable d'instancier ses variables de façon optimale.

Propriété 3.6. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen tel que tous les graphes sont acycliques, et pour chaque joueur $i \in N$, pour tout $v \in \pi_i$, nous avons $Pa(v) \in \pi_i$. G possède alors un et un seul équilibre de Nash en stratégies pures.

La preuve de cette propriété est également donnée par la procédure de recherche en avant (forward sweep procedure), et suit le même raisonnement que la preuve de la propriété 3.4 (page 86).

3.2.3.4 CP-net global

Dans le cadre des CP-jeux booléens, les préférences de chaque agent sont données par un CP-net. Pourtant, pour calculer les équilibres de Nash nous nous intéressons aux variables contrôlées par un joueur donné sur le CP-net représentant ses préférences. L'idée ici est d'introduire un nouveau CP-net, que l'on appellera *CP-net global de G* , unique pour l'ensemble des joueurs, rassemblant pour chaque joueur les variables qu'il contrôle. Comme l'ensemble $\{\pi_1, \dots, \pi_n\}$ forme une partition sur V , chaque variable sera présente une et une seule fois dans ce CP-net.

Définition 3.17. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen.

On appelle **CP-net global de G** le CP-net $\mathcal{N}^+ = \langle \mathcal{G}^+, \mathcal{T}^+ \rangle$ défini par :

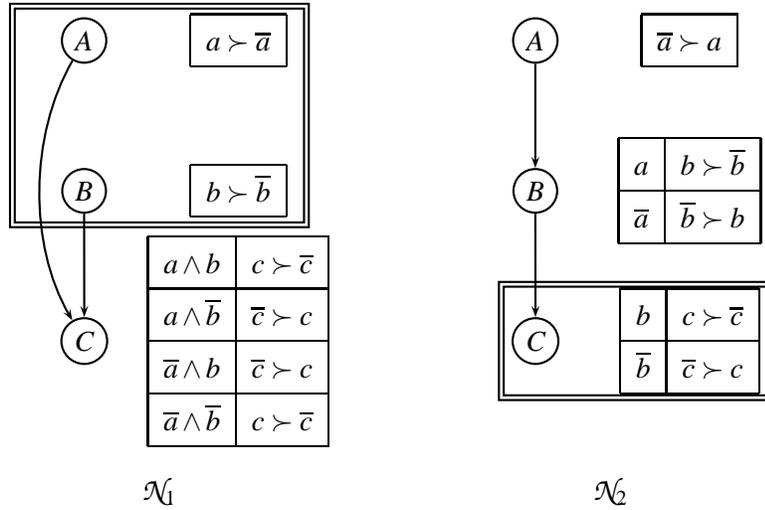
$$* \quad \forall i \in N, \forall v \in \pi_i, CPT_i(v) \in \mathcal{T}^+.$$

* $G^+ = \langle V, \text{Arc}_1^+ \cup \dots \cup \text{Arc}_n^+ \rangle$, avec chaque Arc_i^+ défini par

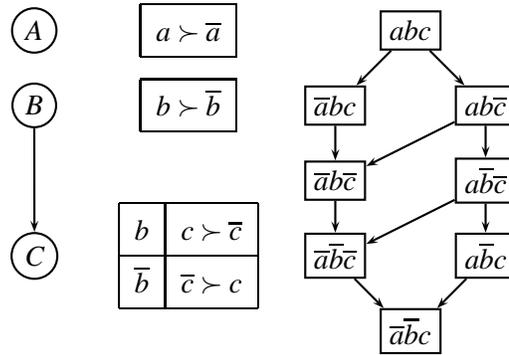
$$\text{Arc}_i^+ = \{(w, v) \in \text{Arc}_i \mid v \in \pi_i\}$$

où Arc_i est l'ensemble des arcs dans G_i .

Exemple 3.13. Considérons le CP-jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$ avec $N = \{1, 2\}$, $V = \{a, b, c\}$, $\pi_1 = \{a, b\}$, $\pi_2 = \{c\}$, $\gamma_1 = \gamma_2 = \top$. \mathcal{N}_1 et \mathcal{N}_2 sont représentés sur la figure suivante. Sur cette figure, chaque boîte contient les variables contrôlées par le joueur concerné, et donc contient les CPTs du CP-net global de G .



Le CP-net global \mathcal{N}^+ de G est représenté sur la figure suivante (on conserve les boîtes, les arcs entrants dans chaque boîte, et les arcs internes à chaque boîte).



Puisque calculer le CP-net global d'un jeu permet d'avoir un CP-net unique, calculer le résultat optimal de ce CP-net, s'il est acyclique, peut donner des résultats intéressants.

Propriété 3.7. ⁸ Soit G un CP-jeu booléen tel que tous les graphes G_i sont acycliques, et \mathcal{N}^+ le CP-net global de G . Soit s un profil de stratégies. Si \mathcal{N}^+ est un CP-net acyclique, on a l'équivalence suivante :

s est un équilibre de Nash si et seulement si S est une solution optimale de \mathcal{N}^+ .

⁸Merci à Nic Wilson de nous avoir donné cette idée.

Cette propriété permet de calculer plus simplement l'équilibre de Nash d'un CP-jeu booléen en utilisant la procédure de recherche en avant (forward sweep procedure).

Pour montrer cette propriété, nous avons besoin du lemme suivant :

Lemme 3.5. *Soit G un CP-jeu booléen, \mathcal{N} un CP-net acyclique de G , et s un profil de stratégies.*

S'il existe $s' : s' \succ s$, alors il existe $s'' : s'' \succ s$ tel que s et s'' ne diffèrent que d'une variable.

Preuve du lemme :

Soit s et s' deux profils de stratégies tels que $s' \succ s$. Supposons qu'il n'existe pas de $s'' \succ s$ tel que s et s'' ne diffèrent que d'une variable. Dans ce cas, comme l'indifférence est impossible, nous avons $s \succ s''$ pour tous s'' de ce type.

Soit $s = v_1 v_2 \dots v_k \bar{v}_{k+1} \dots \bar{v}_p$, avec $v_1, \dots, v_p \in V$, et $\forall v_j \in V$, soit u_j les affectations des parents de v_j . Considérons alors tous les s'' possibles :

$$\begin{aligned}
 s \succ \bar{v}_1 v_2 \dots v_k \bar{v}_{k+1} \dots \bar{v}_p &\Rightarrow v_1 \succ_{u_1} \bar{v}_1 \\
 s \succ v_1 \bar{v}_2 \dots v_k \bar{v}_{k+1} \dots \bar{v}_p &\Rightarrow v_2 \succ_{u_2} \bar{v}_2 \\
 &\vdots \\
 s \succ v_1 v_2 \dots \bar{v}_k \bar{v}_{k+1} \dots \bar{v}_p &\Rightarrow v_k \succ_{u_k} \bar{v}_k \\
 s \succ v_1 v_2 \dots v_k v_{k+1} \dots \bar{v}_p &\Rightarrow \bar{v}_{k+1} \succ_{u_{k+1}} v_{k+1} \\
 &\vdots \\
 s \succ v_1 v_2 \dots v_k \bar{v}_{k+1} \dots v_p &\Rightarrow \bar{v}_p \succ_{u_p} v_p
 \end{aligned}$$

Chaque variable est donc instanciée à sa meilleure valeur en fonction de l'instanciation de ses parents. Cette instanciation correspond à la procédure de recherche en avant. Selon le Lemme 3.2 (page 78), s est donc un résultat optimal pour \mathcal{N} , ce qui est en contradiction avec l'hypothèse : $\exists s'$ tel que $s' \succ s$. Donc, $\exists s''$ qui ne diffère de s que d'une seule variable et tel que $s'' \succ s$. ■

Revenons à présent à la propriété 3.7 (page ci-contre).

Preuve :

\Rightarrow Soit s un équilibre de Nash en stratégies pures. Supposons que s n'est pas une solution optimale de \mathcal{N}^+ .

Alors, d'après la définition 3.12 (page 78), $\exists s' \in S$ tel que $\mathcal{N}^+ \not\models s \succ^+ s'$.

Il existe donc une relation de préférence telle que $s' \succ^+ s$ qui satisfait \mathcal{N}^+ .

D'après le lemme 3.5, $\exists s'' \succ^+ s$ tel que s et s'' ne diffèrent que d'une seule variable : il existe un seul $v \in V$ tel que $v_{s''} \succ_u^+ v_s$, u étant l'instanciation dans s et s' de $Pa(v)$. Soit i le joueur tel que $v \in \pi_i$. On a alors $(s''_i, s_{-i}) \succ^+ (s_i, s_{-i})$.

$s'' \succ^+ s$ satisfait $CPT^+(v)$ pour tout $v \in V$. Or, par construction de \mathcal{N}^+ à partir

de G , $\forall i \in N$, $\forall v \in \pi_i$, $CPT_i(v) = CPT^+(v)$. Donc, pour le joueur i dans G , on a $v_{s''} \succ_i v_s$, et $\forall w \in V \setminus \{v\}$, $\forall j \in N$ tel que $w \in \pi_j$, $w_{s''} =_j w_s$. Alors : $(s''_i, s_{-i}) \succ_i (s_i, s_{-i})$.

s n'est donc pas un équilibre de Nash en stratégie pure.

\Leftarrow Soit s une solution optimale de \mathcal{N}^+ . Supposons que s n'est pas un équilibre de Nash en stratégie pure. Alors, $\exists i \in \{1, \dots, n\}$, $\exists s'_i \in S_i$ tel que $(s'_i, s_{-i}) \succ_i (s_i, s_{-i})$. Donc, $\exists W \subseteq \pi_i$ tel que $\forall w \in W$, $\forall p \in 2^{Pa(w)}$, $w_{s'} \succ_{i,p} w_s$, et $\forall x \in \pi_i \setminus W$, $\forall q \in 2^{Pa(x)}$, $x_{s'} =_i x_s$.

Par construction de \mathcal{N}^+ à partir de G , $\forall i \in N$, $\forall v \in \pi_i$, $CPT_i(v) = CPT^+(v)$. On aura donc toujours $\forall w \in W$, $\forall p \in 2^{Pa(w)}$, $w_{s'} \succ_p^+ w_s$, et $\forall x \in \pi_i \setminus W$, $\forall q \in 2^{Pa(x)}$, $x_{s'} =_q^+ x_s$. On a alors $(s'_i, s_{-i}) \succ^+ (s_i, s_{-i})$. Donc $s' \succ^+ s$, soit $\mathcal{N}^+ \not\models s \succ^+ s'$.

s n'est donc pas une solution optimale de \mathcal{N}^+ .

■

3.2.3.5 Complexité

Les résultats de cette section ont été obtenus avec Bruno Zanuttini.

Les lemmes 3.3 (page 83) et 3.4 (page 84) nous permettent de montrer la propriété suivante :

Propriété 3.8. *Soit $G = (N, V, \pi, \Gamma, \Phi)$ un CP-jeu booléen tel que les graphes G_i sont tous acycliques. Décider si un profil de stratégies s donné est un équilibre de Nash fort en stratégies pures est polynomial.*

Preuve : D'après le Lemme 3.4 (page 84), nous savons que s est un équilibre de Nash fort en stratégies pures si et seulement si s_i est une meilleure réponse pour chaque joueur i . On sait également d'après le Lemme 3.3 (page 83) que cette vérification est polynomiale.

■

Propriété 3.9. *Décider si un CP-jeu booléen acyclique $G = (N, V, \pi, \Gamma, \Phi)$ (i.e., dans lequel tous les CP-nets $\mathcal{N}_i \in \Phi$ sont acycliques) a un équilibre de Nash en stratégies pures est un problème NP-complet. Le problème est NP-difficile même si le nombre de joueurs est restreint à 2.*

Preuve : Grâce à la propriété 3.8, nous savons que ce problème est dans NP. Pour la difficulté, nous donnons une réduction à partir du problème de la satisfiabilité des formules en CNF (conjonction normal form) dans lesquelles chaque clause contient au plus trois littéraux et chaque variable apparaît au plus dans trois clauses [Garey et Johnson, 1979, comments for problem LO1].

L'intuition derrière cette réduction est la suivante : étant donné une formule en CNF ϕ définie comme décrit ci-dessus, un joueur 1 essaiera de satisfaire ϕ en instanciant ses variables. Plus précisément, il sera content seulement si toutes les clauses sont satisfaites, et si ce n'est pas le cas, il préférera changer la valeur d'au moins une

de ses variables dans l'instanciation courante. Un autre joueur 2 associera à chaque variable x_i un littéral $(\neg)p_i$ disant si x_i est positive ou non dans la stratégie du joueur 1.

La construction exacte est la suivante : soit une formule $\varphi = \bigwedge_{i=1}^k C_i$ dans laquelle chaque clause contient au plus trois littéraux et chaque variable apparaît dans au plus trois clauses, et soit deux joueurs 1,2. Soit $\{x_1, \dots, x_n\}$ l'ensemble des variables de φ . Pour chaque clause C_i de φ , soit δ_i une nouvelle variable (signifiant intuitivement que C_i est satisfaite). Enfin, pour chaque variable x_i apparaissant dans φ , soit p_i une nouvelle variable. On construit un CP-jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$ avec $N = \{1, 2\}$, $V = \{x_1, \dots, x_n\} \cup \{\delta_1, \dots, \delta_k\} \cup \{p_1, \dots, p_n\}$, $\pi_1 = \{x_1, \dots, x_n\} \cup \{\delta_1, \dots, \delta_k\}$, $\pi_2 = \{p_1, \dots, p_n\}$, et $\gamma_1 = \gamma_2 = \top$; le CP-net de chaque joueur est défini de la manière suivante : pour chaque variable x_i de φ , si x_i apparaît (positif ou négatif) dans trois clauses $C_{i_1}, C_{i_2}, C_{i_3}$ alors le CPT suivant est ajouté au CP-net du joueur 1.

$\delta_{i_1} \delta_{i_2} \delta_{i_3} p_i$	$x_i \succ \bar{x}_i$
$\delta_{i_1} \delta_{i_2} \delta_{i_3} \bar{p}_i$	$\bar{x}_i \succ x_i$
$\bar{\delta}_{i_1} \delta_{i_2} \delta_{i_3} p_i$	$\bar{x}_i \succ x_i$
$\bar{\delta}_{i_1} \delta_{i_2} \delta_{i_3} \bar{p}_i$	$x_i \succ \bar{x}_i$
$\delta_{i_1} \bar{\delta}_{i_2} \delta_{i_3} p_i$	$\bar{x}_i \succ x_i$
$\delta_{i_1} \bar{\delta}_{i_2} \delta_{i_3} \bar{p}_i$	$x_i \succ \bar{x}_i$
...	
$\bar{\delta}_{i_1} \bar{\delta}_{i_2} \bar{\delta}_{i_3} p_i$	$\bar{x}_i \succ x_i$
$\bar{\delta}_{i_1} \bar{\delta}_{i_2} \bar{\delta}_{i_3} \bar{p}_i$	$x_i \succ \bar{x}_i$

Si x_i apparaît dans une ou deux clauses seulement, un CPT similaire est ajouté, avec seulement une ou deux δ -variable(s) apparaissant dans la table. En d'autres mots, 1 préfère changer la valeur de x_i (et donc ce que 2 rapporte à propos de cette valeur) aussitôt qu'au moins une clause contenant x_i n'est pas satisfaite.

À présent, on ajoute le CPT suivant à chaque C_i dans φ , par exemple $C_i = (x_1 \vee \neg x_2 \vee \neg x_3)$.

$p_1 \bar{p}_2 \bar{p}_3$	$\delta_i \succ \bar{\delta}_i$
$p_1 p_2 \bar{p}_3$	$\delta_i \succ \bar{\delta}_i$
...	
$\bar{p}_1 p_2 \bar{p}_3$	$\delta_i \succ \bar{\delta}_i$
$\bar{p}_1 p_2 p_3$	$\bar{\delta}_i \succ \delta_i$

En d'autres mots, 1 préfère δ_i si C_i est satisfaite, et $\bar{\delta}_i$ sinon. On a donc :

- (1) dans tout équilibre de Nash en stratégies pures de G , δ_i est vrai si et seulement si la clause C_i est satisfaite.

(sinon le joueur 1 aurait une meilleure stratégie, puisqu'il contrôle δ_i).

Le CP-net du joueur 1 est alors défini de façon à être l'ensemble de toutes ces tables, avec des préférences quelconques sur les valeurs des variables p_i (qui n'ont pas de parent). La construction n'est pas ambiguë puisque une seule table est construite pour x_i ou δ_i , et le graphe est acyclique puisque les seules dépendances partent de

p_j vers δ_i , et de δ_k et p_j vers x_i .

Enfin, le CP-net de 2 est construit comme suit. Pour chaque variable x_i , le CPT suivant est ajouté :

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

Le CP-net du joueur 2 est acyclique, et

(2) dans tout équilibre de Nash en stratégies pures de G , pour tout i , p_i est vrai si et seulement si x_i est vraie.

(sinon le joueur 2 aurait une meilleure stratégie, puisqu'il contrôle p_i).

Nous avons à présent :

(3) dans tout équilibre de Nash en stratégies pures de G , si pour un i , C_i est faux, alors il existe un j tel que $x_j \equiv \neg p_j$.

Supposons que C_i est faux. Alors, d'après (1), δ_i est faux. Supposons à présent que x_j est une variable de C_i . Le CPT de 1 attaché à x_j contient deux préférences conditionnelles de la forme $\bar{\delta}_i(\dots)p_j : \bar{x}_j \succ x_j$ et $\bar{\delta}_i(\dots)\bar{p}_j : x_j \succ \bar{x}_j$. Puisque δ_i est faux et que 1 contrôle x_j , x_j aura la valeur opposée à celle de p_j dans un équilibre de Nash en stratégies pures de G .

A partir de (2) et (3), on sait que dans un équilibre de Nash en stratégies pures de G , chaque clause C_i est satisfaite. Et donc, si G a un équilibre de Nash en stratégies pures, alors φ est satisfiable.

Réciproquement, supposons que φ est satisfiable, et posons $\vec{x} = (x_1, \dots, x_n) \models \varphi$. Considérons le profil de stratégies défini en instanciant chaque x_i à la même valeur que dans \vec{x} , chaque p_i comme x_i , et chaque δ_i à vrai. On vérifie facilement que ce profil de stratégies est un équilibre de Nash en stratégies pures de G . Et donc, si φ est satisfiable, alors G a un équilibre de Nash en stratégies pures.

■

On peut observer que si l'union des graphes du CP-jeu booléen construit dans la preuve est cyclique, le joueur 1 peut alors obtenir des informations sur ses propres variables à partir de 2, et donc être influencé sur "comment jouer". Par exemple, il peut décider de jouer x_3 (\bar{x}_3) si x_3 est faux (vrai), et qu'une clause qui le contient n'est pas satisfaite. Ses décisions sur la variable x_3 sont donc en partie basées sur la valeur de x_3 elle-même, ce qui est évité grâce à l'acyclicité.

Nous n'avons malheureusement pas réussi à calculer la complexité du problème consistant à décider si un CP-jeu booléen quelconque a un équilibre de Nash en stratégies pures.

3.2.4 Introduction d'une relation d'indifférence

Nous supposons à présent que les relations d'ordre qui décrivent les préférences des joueurs dans les tables de préférences conditionnelles ne sont plus strictes. Il peut y avoir des relations d'indifférence entre deux valeurs pour une variable donnée.

On redéfinit alors dans ce cadre les tables de préférences conditionnelles et les CP-nets.

Définition 3.18. La table de préférence conditionnelle avec indifférence (notée $ICPT_i(X)$) décrit les préférences du joueur i sur les valeurs de la variable X , étant données toutes les combinaisons des valeurs des variables parents.

Pour chaque instantiation p de $Pa_i(X)$, $ICPT_i(X)$ spécifie un **préordre total** tel que l'on ait soit $x \succ_{i,p} \bar{x}$, soit $\bar{x} \succ_{i,p} x$, soit $x \sim_{i,p} \bar{x}$.

Définition 3.19. Un **ICP-net** pour le joueur i sur les variables $V = \{X_1, \dots, X_n\}$ est un graphe orienté G sur X_1, \dots, X_n . Chaque nœud X_j est annoté avec la table de préférence conditionnelle avec indifférence $ICPT_i(X_j)$ correspondante. Chaque table de préférence conditionnelle avec indifférence $ICPT_i(X_j)$ est associée à un **préordre total** $\succ_{i,p}^j$, selon chaque instantiation $p \in D(Pa(X_j))$.

Les ICP-jeux booléens sont donc des jeux booléens dont les buts sont donnés par des ICP-nets.

Exemple 3.14. Soit le ICP-jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$ suivant :

- * $N = \{1, 2\}$
- * $V = \{a, b, c\}$, avec $D(a) = \{a, \bar{a}\}$, $D(b) = \{b, \bar{b}\}$ et $D(c) = \{c, \bar{c}\}$.
- * $\pi_1 = \{a, b\}$, $\pi_2 = \{c\}$,
- * $\gamma_1 = \gamma_2 = \top$,
- * Le but du joueur 1, et le pré-ordre associé sur les profils de stratégies, sont représentés par la figure 3.15,
- * Le but du joueur 2, et le pré-ordre associé sur les profils de stratégies, sont représentés par la figure 3.16 (page suivante),

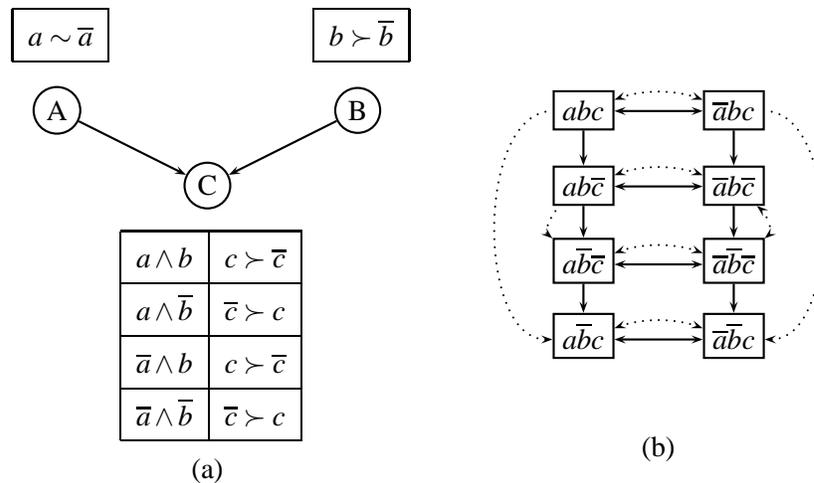


Figure 3.15 — ICP-net des préférences et le pré-ordre associé du joueur 1

Le calcul des équilibres de Nash faibles et forts nous donne ici comme résultat :

$$NE_{faible} = NE_{fort} = \{abc, ab\bar{c}, \bar{a}bc, \bar{a}b\bar{c}\}$$

Etudions maintenant sur un exemple ce qu'il se passe lorsque tous les joueurs ont un graphe commun acyclique.

Exemple 3.15. Soit le ICP-jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$ suivant :

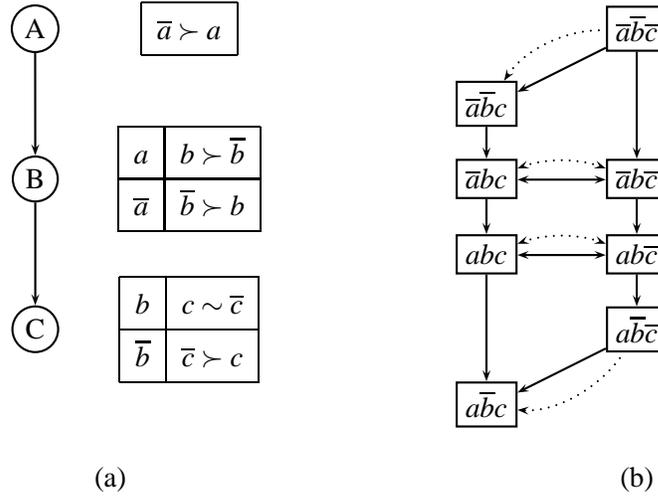


Figure 3.16 — ICP-net des préférences et le pré-ordre associé du joueur 2

- * $N = \{1, 2, 3\}$
- * $V = \{a, b, c\}$, avec $D(a) = \{a, \bar{a}\}$, $D(b) = \{b, \bar{b}\}$ et $D(c) = \{c, \bar{c}\}$.
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$,
- * $\gamma_1 = \gamma_2 = \gamma_3 = \top$,
- * Le but du joueur 1, et le pré-ordre associé sur les profils de stratégies, sont représentés par la figure 3.17,
- * Le but du joueur 2, et le pré-ordre associé sur les profils de stratégies, sont représentés par la figure 3.18 (page ci-contre),
- * Le but du joueur 3, et le pré-ordre associé sur les profils de stratégies, sont représentés par la figure 3.19 (page suivante).

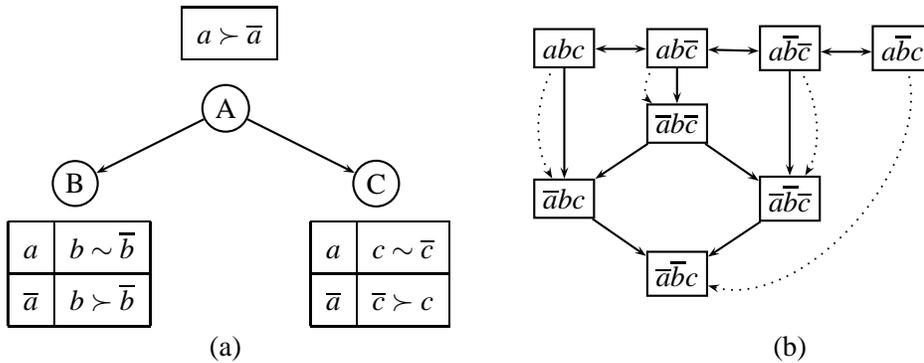


Figure 3.17 — ICP-net des préférences et le pré-ordre associé du joueur 1

Le calcul des équilibres de Nash faibles et forts nous donne ici comme résultat :

$$NE_{faible} = NE_{fort} = \{abc, ab\bar{c}\}$$

Cet exemple nous montre que l'unicité des équilibres de Nash faibles et forts n'est plus assurée. Par contre, l'existence l'est toujours.

Propriété 3.10. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un ICP-jeu booléen tel que les graphes G_i sont tous identiques ($\forall i, j, G_i = G_j$) et acycliques, avec des préférences conditionnelles.

Ce jeu G ainsi défini aura alors au moins un équilibre de Nash fort en stratégie pure.

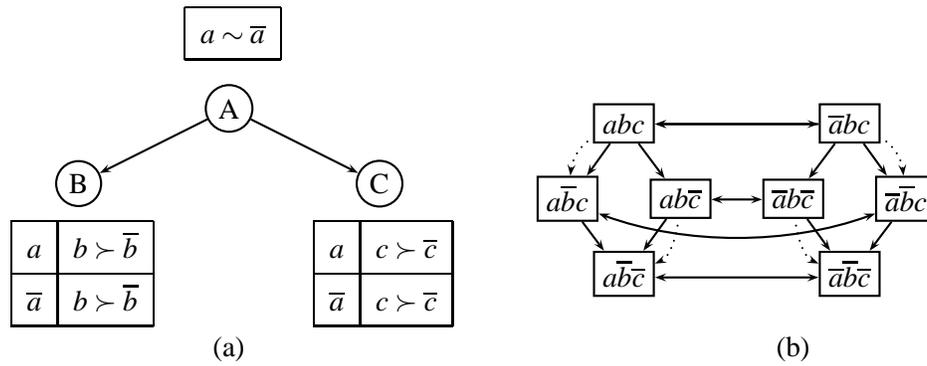


Figure 3.18 — ICP-net des préférences et le pré-ordre associé du joueur 2

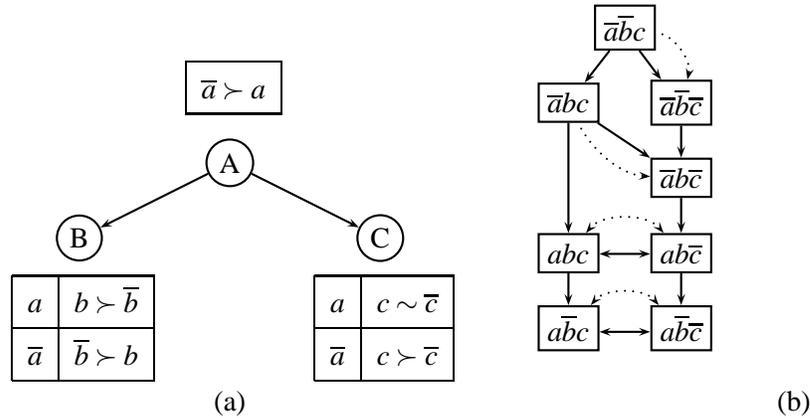


Figure 3.19 — ICP-net des préférences et le pré-ordre associé du joueur 3

Preuve : La démonstration de cette propriété est exactement la même que celle de la propriété 3.4 (page 86). En effet, cette partie de la démonstration n'utilise pas le fait que les préférences conditionnelles sont totales.

■

Remarque : Nous pouvons ici comme dans la section 3.2.3 (page 82) calculer le jeu G^* équivalent par réécriture de G en faisant l'union des graphes.

Si l'union des graphes des joueurs d'un jeu G permet d'obtenir un graphe acyclique, la propriété 3.10 (page précédente) est applicable sur le jeu G^* équivalent par réécriture à G . Comme ces deux jeux sont équivalents, le jeu G aura au moins un équilibre de Nash en stratégie pure.

3.3 Buts à priorité

Les CP-nets ne permettent pas de représenter toutes les préférences, ce langage n'est pas totalement expressif. Il est par exemple impossible de dire avec un CP-net "Mes préférences sont, par ordre décroissant, des vacances à la mer en été, puis des vacances à la montagne en hiver, des vacances à la montagne en été, et enfin des vacances à la mer en hiver". Cette relation de préférences est en effet du type $xy \succ \bar{x}\bar{y} \succ x\bar{y} \succ \bar{x}y$, c'est-à-dire qu'elle compare deux à deux des états n'étant pas identiques "toutes choses étant égales par ailleurs".

Pour cette raison, nous choisissons ici d'étudier un autre langage de représentation de préférences : les buts à priorité.

3.3.1 Etat de l'art

Dans des buts à priorité, les préférences d'un joueur sont exprimées par un ensemble de buts ordonnés selon une relation de priorité :

Définition 3.20. Une **base de buts à priorité** Σ est une collection $\langle \Sigma^1 ; \dots ; \Sigma^p \rangle$ d'ensembles de formules propositionnelles.

Σ^j représente l'ensemble des buts de priorité j , avec la convention suivante : plus j est petit, plus les formules dans Σ^j sont prioritaires.

Tout au long de cette section, nous allons utiliser un exemple unique représenté par une base propositionnelle stratifiée $(\Sigma, <)$. Pour illustrer les différentes définitions, nous allons étudier deux stratifications classiques :

- * La stratification numéro 1 de la base de préférences (une seule formule par strate : stratification stricte) est représentée figure 3.20.

Σ_1	$\rightarrow P$	je préfère le poisson
Σ_2	$P \rightarrow L$	ceux qui préfèrent le poisson préfèrent les légumes
Σ_3	$P \rightarrow \neg V$	ceux qui préfèrent le poisson ne préfèrent pas la viande
Σ_4	$L \rightarrow V$	ceux qui préfèrent les légumes préfèrent la viande
Σ_5	$L \rightarrow A$	ceux qui préfèrent les légumes préfèrent les artichauds
Σ_6	$A \rightarrow V$	ceux qui préfèrent les artichauds préfèrent la viande

Figure 3.20 — Stratification numéro 1 de la base de préférences

- * La stratification numéro 2 de la base de préférences (3 strates, 2 formules par strate) est représentée figure 3.21.

Σ_1	$\rightarrow P$	je préfère le poisson
	$P \rightarrow L$	ceux qui préfèrent le poisson préfèrent les légumes
Σ_2	$P \rightarrow \neg V$	ceux qui préfèrent le poisson ne préfèrent pas la viande
	$L \rightarrow V$	ceux qui préfèrent les légumes préfèrent la viande
Σ_3	$L \rightarrow A$	ceux qui préfèrent les légumes préfèrent les artichauds
	$A \rightarrow V$	ceux qui préfèrent les artichauds préfèrent la viande

Figure 3.21 — Stratification numéro 2 de la base de préférences

Dans le contexte des jeux booléens, nous aurons à comparer des profils de stratégies (donc des interprétations) et non plus des sous-bases consistantes, chaque joueur devant pouvoir choisir la stratégie, et donc les profils de stratégies, lui permettant d'obtenir le "meilleur résultat possible".

Soit $s \in \mathcal{S}$, on pose

$$\text{Sat}(s, \Sigma^j) = \{\varphi \in \Sigma^j \mid s \models \varphi\}$$

3.3.1.1 Ordre Discrimin

Définition 3.21. Les sous-bases consistantes A d'une base stratifiée Σ sont les sous-ensembles de Σ consistants au sens de la logique classique.

Une sous-base A est une **sous-base consistante maximale** pour l'inclusion de Σ si et seulement si :

- * A est une sous-base consistante,
- * il n'existe pas de sous-base consistante de Σ contenant strictement A .

Les sous-bases de Σ consistantes maximales sont aussi appelées **thèses** de Σ .

Les thèses correspondant à l'exemple présenté page 99 sont identiques, quelle que soit la stratification choisie. Elles sont représentées figure 3.22.

Thèse 1	Thèse 2	Thèse 3	Thèse 4	Thèse 5
$\rightarrow P$	$\rightarrow P$	$\rightarrow P$	$\rightarrow P$	$P \rightarrow L$
$P \rightarrow L$	$P \rightarrow L$	$P \rightarrow L$	$P \rightarrow \neg V$	$P \rightarrow \neg V$
$P \rightarrow \neg V$	$P \rightarrow \neg V$	$L \rightarrow V$	$L \rightarrow V$	$L \rightarrow V$
$L \rightarrow A$	$A \rightarrow V$	$L \rightarrow A$	$L \rightarrow A$	$L \rightarrow A$
		$A \rightarrow V$	$A \rightarrow V$	$A \rightarrow V$

Figure 3.22 — Thèses des bases stratifiées

Le calcul de sous-théories préférées a été introduit par [Brewka, 1989a,b].

Définition 3.22. Soit Σ une base stratifiée. Une **sous-théorie préférée** est un ensemble $A = A_1 \cup \dots \cup A_n$ tel que $\forall k \in [1, n], A_1 \cup \dots \cup A_k$ est un sous-ensemble maximal consistant de $\Sigma_1 \cup \dots \cup \Sigma_k$.

Par définition, les sous-théories préférées de Brewka sont des thèses de Σ .

- * Soit Σ la base stratifiée présentée figure 3.20 (page ci-contre). En appliquant la méthode de Brewka on obtient une seule sous-théorie préférée présentée figure 3.23.
- * Soit Σ la base stratifiée présentée figure 3.21 (page précédente). En appliquant le méthode de Brewka on obtient 3 sous-théories préférées présentées figure 3.24.

Une relation de préférence appelée "*discrimin*", découle de cette définition [Brewka, 1989b; Dubois *et al.*, 1992; Geffner, 1992; Benferhat *et al.*, 1993].

Définition 3.23. $s \succ^{disc} s'$ si et seulement si $\exists k \in \{1, \dots, p\}$ tel que :

$$\left| \begin{array}{l} \rightarrow P \\ P \rightarrow L \\ P \rightarrow \neg V \\ L \rightarrow A \end{array} \right.$$

Figure 3.23 — Sous-théorie préférée pour la stratification n°1

$$\left| \begin{array}{l} \rightarrow P \\ P \rightarrow L \\ P \rightarrow \neg V \\ L \rightarrow A \end{array} \right| \left| \begin{array}{l} \rightarrow P \\ P \rightarrow L \\ P \rightarrow \neg V \\ A \rightarrow V \end{array} \right| \left| \begin{array}{l} \rightarrow P \\ P \rightarrow L \\ L \rightarrow V \\ L \rightarrow A \\ A \rightarrow V \end{array} \right.$$

Figure 3.24 — Sous-théories préférées pour la stratification n°2

1. $Sat(s, \Sigma^k) \supset Sat(s', \Sigma^k)$, et
2. $\forall j < k, Sat(s, \Sigma^j) = Sat(s', \Sigma^j)$

Ordonnons les thèses trouvées pour chaque stratification selon le critère discrimin :

- * Soit Σ la base stratifiée présentée figure 3.20 (page 100). En appliquant l'ordre discrimin, on obtient l'ordonnancement suivant des thèses données en figure 3.22 (page précédente) :

$$\text{Thèse 1} \succ^{discr} \text{Thèse 2} \succ^{discr} \text{Thèse 3} \succ^{discr} \text{Thèse 4} \succ^{discr} \text{Thèse 5}$$

- * Soit Σ la base stratifiée présentée figure 3.21 (page 100). En appliquant l'ordre discrimin, on obtient l'ordonnancement suivant des thèses données en figure 3.22 (page précédente) :

$$\text{Thèse 1} \succ^{discr} \text{Thèse 4}$$

$$\text{Thèse 1} \succ^{discr} \text{Thèse 5}$$

$$\text{Thèse 2} \succ^{discr} \text{Thèse 4}$$

$$\text{Thèse 2} \succ^{discr} \text{Thèse 5}$$

$$\text{Thèse 3} \succ^{discr} \text{Thèse 4}$$

$$\text{Thèse 3} \succ^{discr} \text{Thèse 5}$$

Les thèses 1, 2 et 3 sont incomparables, tout comme les thèses 4 et 5.

D'autres travaux ont conduit à des définitions proches (l'ordre démocratique de [Cayrol *et al.*, 1993], et les sous-bases fortement maximales consistantes de [Dubois *et al.*, 1992]). Dans le cas d'une base stratifiée, tous ces travaux se rejoignent. Nous ne présenterons pas en détail ces autres méthodes ici.

3.3.1.2 Ordre Leximin

Le critère “leximin” consiste à sélectionner des sous-bases consistantes maximales pour la cardinalité en respectant la stratification. La définition de base (ordre basé sur la cardinalité) correspondant au cas d’une base de croyances non ordonnée est donnée dans [Dubois *et al.*, 1992; Benferhat *et al.*, 1993; Lehmann, 1995] :

Définition 3.24. $s \succ^{lex} s'$ si et seulement si $\exists k \in \{1, \dots, p\}$ tel que :

1. $|Sat(s, \Sigma^k)| > |Sat(s', \Sigma^k)|$, et
2. $\forall j < k, |Sat(s, \Sigma^j)| = |Sat(s', \Sigma^j)|$.

Ce critère consiste donc à comparer deux sous-bases consistantes en identifiant d’abord le plus haut niveau de priorité pour lequel elles ne satisfont pas le même nombre de buts, puis à préférer celle qui satisfait le plus grand nombre de buts à ce niveau.

Notation : On introduit le symbole suivant : \equiv^{lex} qui représente la relation d’équivalence issue de l’ordre \succeq^{lex} : soit A et B deux sous-bases d’une base Σ stratifiée. $A \equiv^{lex} B$ si et seulement si $\forall i = 1 \dots n, |A_i| = |B_i|$, avec $n =$ le nombre de strates de Σ .

Le pré-ordre ainsi obtenu est un pré-ordre total.

Ordonnons les thèses trouvées pour chaque stratification selon le critère leximin :

- * Soit Σ la base stratifiée correspondant à la stratification numéro 1 présentée figure 3.20 (page 100). En appliquant le critère leximin, on obtient l’ordonnement suivant des thèses de la figure 3.22 (page 101) :

$$\text{Thèse 5} \preceq^{lex} \text{Thèse 4} \preceq^{lex} \text{Thèse 3} \preceq^{lex} \text{Thèse 2} \preceq^{lex} \text{Thèse 1}$$

- * Soit Σ la base stratifiée correspondant à la stratification numéro 2 présentée figure 3.21 (page 100). En appliquant le critère leximin, on obtient l’ordonnement suivant des thèses de la figure 3.22 (page 101) :

$$\text{Thèse 5} \equiv^{lex} \text{Thèse 4} \preceq^{lex} \text{Thèse 2} \equiv^{lex} \text{Thèse 1} \preceq^{lex} \text{Thèse 3}$$

3.3.1.3 Ordre Best-out

L’idée est ici de repérer la strate la plus prioritaire amenant une inconsistance, et de ne prendre alors en compte que les formules de priorité supérieure à cette strate fatidique.

L’ordre “Best Out” décrit dans [Benferhat *et al.*, 1993] et [Dubois *et al.*, 1992] est défini à partir des sous bases consistantes et non à partir des thèses.

Définition 3.25. Soit $a(s) = \min\{j \text{ tel que } \exists \varphi \in \Sigma^j, s \not\models \varphi\}$, avec la convention $\min(\emptyset) = +\infty$.

Alors, $s \succeq^{bo} s'$ si et seulement si $a(s) \geq a(s')$.

Nous obtenons également ici un pré-ordre total. De plus, les éléments maximaux de l’ordre Best Out ne sont pas forcément des éléments maximaux pour l’inclusion ensembliste.

Pour l'exemple précédent nous avons donc $2^6 - 2$ possibilités à explorer (on ne tient pas compte de la base vide, ni de la base initiale supposée inconsistante), ce qui fait au plus 62 sous-bases consistantes qu'il faudrait comparer 2 à 2 pour pouvoir les ordonner. Une fois cette comparaison effectuée, nous trouvons les éléments maximaux suivants :

- * stratification 1 : les éléments maximaux sont toutes les sous-bases contenant

$$\{P, P \rightarrow L, P \rightarrow \neg V\}$$

- * stratification 2 : les éléments maximaux sont toutes les sous-bases contenant

$$\{P, P \rightarrow L\}$$

On note que \succeq^{bo} et \succeq^{lex} sont des relations de préférence totales, tandis que \succeq^{disc} est en général une relation de préférence partielle.

Comme cela a déjà été prouvé par [Benferhat *et al.*, 1993], les critères discrimin, leximin et best-out ne sont pas totalement indépendants. La propriété suivante est toujours valable dans le cadre des jeux booléens (donc quand on étudie des interprétations et non pas des sous-bases) :

Propriété 3.11.

$$(s \succ^{bo} s') \Rightarrow (s \succ^{disc} s') \Rightarrow (s \succ^{lex} s') \quad (3.3)$$

$$(s \succeq^{disc} s') \Rightarrow (s \succeq^{lex} s') \Rightarrow (s \succeq^{bo} s') \quad (3.4)$$

Preuve :

$$* (s \succ^{bo} s') \Rightarrow (s \succ^{disc} s') \Rightarrow (s \succ^{lex} s')$$

$$\begin{aligned} (s \succ^{bo} s') &\Leftrightarrow a(s) > a(s') \\ &\Leftrightarrow \underbrace{\min\{j \mid \exists \varphi_i \in \Sigma^j, s \not\models \varphi_i\}}_{\text{noté } x} > \underbrace{\min\{j \mid \exists \varphi_i \in \Sigma^j, s' \not\models \varphi_i\}}_{\text{noté } y} \end{aligned}$$

On sait donc que :

- * $\forall j < y, \forall \varphi_i \in \Sigma^j, s \models \varphi_i, s' \models \varphi_i$
- * pour la strate $y, \forall \varphi_i \in \Sigma^y, s \models \varphi_i, s' \not\models \varphi_i$
- * pour la strate $x, \forall \varphi_i \in \Sigma^x, s \not\models \varphi_i$

On en déduit que :

- $Sat(s, \Sigma^y) \supset Sat(s', \Sigma^y)$
- $\forall j < y, Sat(s, \Sigma^j) = Sat(s', \Sigma^j)$.

On a donc : $s \succ^{disc} s'$

Et donc :

- * $|Sat(s, \Sigma^y)| > |Sat(s', \Sigma^y)|$
- * $\forall j < y, |Sat(s, \Sigma^j)| = |Sat(s', \Sigma^j)|$.

Donc : $s \succ^{lex} s'$

$$\begin{aligned}
* \quad (s \succeq^{discr} s') &\Rightarrow (s \succeq^{lex} s') \Rightarrow (s \succeq^{bo} s') \\
(s \succeq^{discr} s') &\Leftrightarrow s \succ^{discr} s' \text{ ou } \forall k \in \{1, \dots, p\}, Sat(s, \Sigma^k) = Sat(s', \Sigma^k) \\
&\Rightarrow s \succ^{lex} s' \text{ ou } \forall k \in \{1, \dots, p\}, |Sat(s, \Sigma^k)| = |Sat(s', \Sigma^k)| \\
&\Rightarrow (s \succeq^{lex} s') \\
&\Leftrightarrow \exists k \text{ tel que } |Sat(s, \Sigma^k)| \geq |Sat(s', \Sigma^k)| \\
&\quad \text{et } \forall j < k, |Sat(s, \Sigma^j)| = |Sat(s', \Sigma^j)| \\
&\Rightarrow a(s) \geq a(s') \\
&\Leftrightarrow (s \succeq^{bo} s')
\end{aligned}$$

■

3.3.2 Utilisation des buts à priorité dans les jeux booléens

Nous pouvons à présent définir les BP-jeux booléens.

Définition 3.26. *Un BP-jeu booléen est un 5-uple $G = (N, V, \pi, \Gamma, \Phi)$, où $\Phi = (\Sigma_1, \dots, \Sigma_n)$ est une collection de bases de buts à priorité.*

On note $\Sigma_i = \langle \Sigma_i^1; \dots; \Sigma_i^p \rangle$, Σ_i^j représentant la strate j de Σ_i (l'ensemble de buts de priorité j pour le joueur i).

L'hypothèse que le nombre de niveaux de priorité est le même (p) pour chaque joueur n'entraîne pas de perte de généralité. En effet, ajouter des strates vides à une base de buts à priorité ne modifie pas la relation de préférence induite.

Nous utiliserons les notations suivantes :

- * Si G est un BP-jeu booléen et que $c \in \{disc, lex, bo\}$, alors $Pref_G^c = \langle \succeq_1^c, \dots, \succeq_n^c \rangle$.
- * $NE_{faible}^{disc}(G)$ (resp. $NE_{fort}^{disc}(G)$) représente l'ensemble des équilibres de Nash faibles (resp. forts) pour $Pref_G^{disc}$. Comme \succeq^{bo} et \succeq^{lex} sont des relations de préférence totales, la différence entre équilibres de Nash faible et fort n'est pas pertinente.

Exemple 3.16. *Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen, avec :*

- * $N = \{1, 2\}$,
- * $V = \{a, b, c\}$,
- * $\pi_1 = \{a, c\}$, $\pi_2 = \{b\}$,
- * $\gamma_1 = \gamma_2 = \top$,
- * $\Sigma_1 = \langle a; (\neg b, c) \rangle$,
- * $\Sigma_2 = \langle (\neg b, \neg c); \neg a \rangle$

Nous allons maintenant appliquer chacun des trois critères présentés plus haut afin d'obtenir un pré-ordre sur les profils de stratégies.

Discrimin *Pour chacun des joueurs, appliquons le critère discrimin (cf. définition 3.23 (page 101)).*

Nous trouvons les relations de préférences suivantes pour le joueur 1 :

$$\bar{a}\bar{b}c \succ_1^{disc} \bar{a}\bar{b}\bar{c} \succ_1^{disc} a\bar{b}\bar{c} \succ_1^{disc} \bar{a}bc \succ_1^{disc} \bar{a}\bar{b}\bar{c} \succ_1^{disc} \bar{a}\bar{b}c$$

$$\begin{aligned} \bar{a}\bar{b}c &\succ_1^{disc} abc \succ_1^{disc} a\bar{b}\bar{c} \succ_1^{disc} \bar{a}\bar{b}c \succ_1^{disc} \bar{a}bc \succ_1^{disc} \bar{a}\bar{b}\bar{c} \\ \bar{a}\bar{b}c &\succ_1^{disc} abc \succ_1^{disc} a\bar{b}\bar{c} \succ_1^{disc} \bar{a}\bar{b}c \succ_1^{disc} \bar{a}bc \succ_1^{disc} \bar{a}\bar{b}\bar{c} \\ \bar{a}\bar{b}c &\succ_1^{disc} \bar{a}\bar{b}\bar{c} \succ_1^{disc} a\bar{b}\bar{c} \succ_1^{disc} \bar{a}\bar{b}c \succ_1^{disc} \bar{a}bc \succ_1^{disc} \bar{a}\bar{b}\bar{c} \end{aligned}$$

Afin de pouvoir exploiter plus facilement ce résultat, nous le traduisons sous forme graphique. Ainsi, nous trouvons pour chacun des joueurs les relations partielles⁹ représentées figure 3.25.

Les flèches vont du profil de stratégies le plus préféré vers le profil de stratégies le moins préféré.

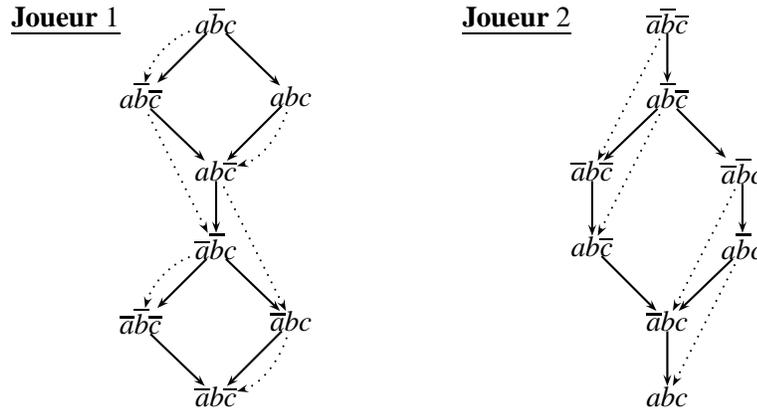


Figure 3.25 — Relations partielles sur les profils de stratégies calculées à partir du critère Discrimin

Nous pouvons à présent calculer les équilibres de Nash faibles et forts de ce jeu pour le critère discrimin :

$$\begin{aligned} NE_{faible}^{disc} &= \{\bar{a}\bar{b}c\} \\ NE_{fort}^{disc} &= \{\bar{a}\bar{b}c\} \end{aligned}$$

Etudions à présent les stratégies dominées de ce jeu :

- * Joueur 1 : Il est possible d'éliminer itérativement 3 des 4 stratégies de ce joueur. En effet :
 - * La stratégie $\bar{a}\bar{b}c$ est strictement dominée par la stratégie $\bar{a}c$. On l'élimine.
 - * Après cette élimination, la stratégie $\bar{a}c$ est strictement dominée par la stratégie $\bar{a}\bar{b}c$. On l'élimine.
 - * De nouveau, après cette élimination, la stratégie $\bar{a}\bar{b}c$ est strictement dominée par la stratégie $\bar{a}c$. Il ne nous reste donc plus que la stratégie $\bar{a}c$.
- * Joueur 2 : Après élimination des 3 stratégies du joueur 1, la stratégie b est strictement dominée par la stratégie \bar{b} . On ne conserve donc que cette dernière.

En éliminant itérativement les stratégies strictement dominées, on obtient donc un profil de stratégies résultat : $\{\bar{a}\bar{b}c\}$, qui se trouve être aussi l'équilibre de Nash faible et fort. De plus, ce sont ici des stratégies strictement dominées. L'ordre d'élimination de ces stratégies n'affecte donc pas le résultat.

Leximin On applique ici de la même manière que précédemment le critère leximin (cf. définition 3.24 (page 103)) à chacun des joueurs. Nous trouvons ainsi les relations totales suivantes données en figure 3.26, et nous pouvons calculer les équilibres de Nash faibles et forts de jeu pour le critère leximin.

⁹Pour calculer facilement les équilibres de Nash de ce jeu, nous mettons en exergue les relations qui nous intéressent pour chaque joueur. Ces relations apparaissent sous la forme de flèches en pointillé sur la figure.

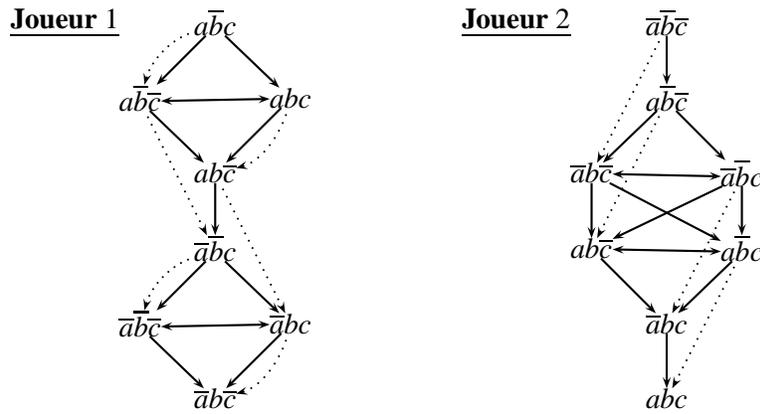


Figure 3.26 — Relations totales sur les profils de stratégies calculées à partir du critère Leximin

$$NE^{lex} = \{\bar{a}\bar{b}c\}$$

L'élimination des stratégies dominées se déroule selon le même schéma que précédemment, et nous fournit le même unique résultat : $\{\bar{a}\bar{b}c\}$.

Best out Si l'on applique l'ordre Best out (cf. définition 3.25) à chacun des joueurs, nous trouvons les relations totales décrites en figure 3.27¹⁰ à partir desquelles nous pouvons calculer les équilibres de Nash faibles et forts de ce jeu pour le critère best out :

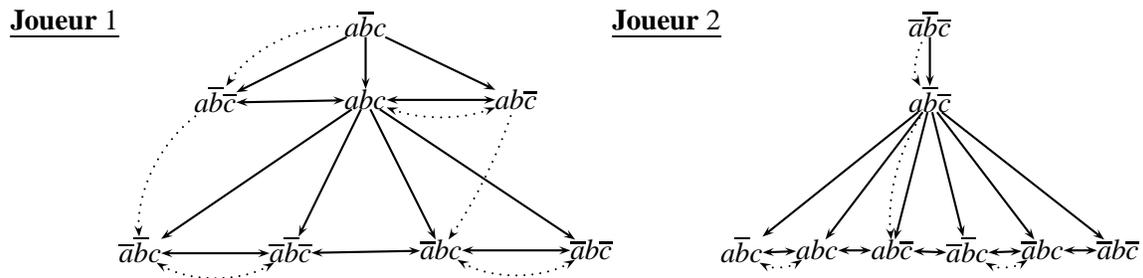


Figure 3.27 — Relations totales sur les stratégies calculées à partir de l'ordre Best out

$$NE^{bo} = \{abc, \bar{a}\bar{b}c\}$$

L'élimination des stratégies dominées donne ici un résultat différent :

- * **Joueur 1** : Il est possible d'éliminer itérativement 3 des 4 stratégies de ce joueur. En effet :
 - * La stratégie $\bar{a}\bar{b}c$ est strictement dominée par la stratégie $a\bar{c}$. On l'élimine.
 - * La stratégie $\bar{a}bc$ est strictement dominée par la stratégie $a\bar{c}$. On l'élimine.
 - * La stratégie $a\bar{c}$ est partiellement dominée par la stratégie ac , on l'élimine donc également.
- * **Joueur 2** : Après élimination des 3 stratégies du joueur 1, aucune des deux stratégies b ou \bar{b} ne domine partiellement, faiblement ou strictement l'autre.

¹⁰Pour le joueur 2, les nœuds du bas sont tous reliés les uns aux autres, mais nous n'indiquons pas toutes les flèches afin de ne pas alourdir le dessin. Comme cette relation de préférence est transitive, le résultat n'en est pas affecté. Nous faisons de même pour le joueur 1.

En éliminant itérativement les stratégies dominées, on obtient donc deux profils de stratégies résultat : $\{abc\}$ et $\{\bar{a}\bar{b}c\}$. On obtient donc le même résultat qu'avec les équilibres de Nash faibles et forts.

Comme ici deux stratégies sur 3 sont strictement dominées, l'ordre d'élimination choisi n'affecte pas les résultats obtenus.

Reprenons l'exemple 2.2 (page 34) du jeu du prisonnier, mais cette fois-ci dans sa forme classique.

Exemple 3.17. Dans cette version du dilemme du prisonnier, 2 détenus sont emprisonnés dans des cellules séparées. La police fait à chacun d'eux le même marché :

"Tu as le choix entre couvrir ton complice en te taisant (noté T_i , $i = 1, 2$), ou le trahir en le dénonçant (noté \bar{T}_i , $i = 1, 2$). Si tu le dénonces, et qu'il ne te dénonce pas, tu seras remis en liberté et lui écoperà de 10 ans de prison. Si vous vous dénoncez mutuellement, vous écopererez tous les deux de 5 ans de prison. Si aucun de vous ne dénonce l'autre, vous aurez tous les deux 6 mois de prison."

Voilà la forme normale de ce jeu, avec 2 prisonniers :

	2	T_2	\bar{T}_2
1		T_1	\bar{T}_1
	T_1	$(-1/2, -1/2)$	$(-10, 0)$
	\bar{T}_1	$(0, -10)$	$(-5, -5)$

Si l'on calcule l'équilibre de Nash en stratégies pures de ce jeu à partir de cette forme normale, on trouve :

$$NE = \{\bar{T}_1\bar{T}_2\}$$

Nous pouvons traduire ce jeu par le BP-jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$ suivant :

- * $N = \{1, 2\}$,
- * $V = \{T_1, T_2\}$,
- * $\pi_1 = \{T_1\}$, $\pi_2 = \{T_2\}$,
- * $\gamma_1 = \gamma_2 = \top$,
- * $\Sigma_1 = \langle T_2; \bar{T}_1 \rangle$,
- * $\Sigma_2 = \langle T_1; \bar{T}_2 \rangle$

Discrimin Si l'on applique le critère discrimin à chacun des joueurs, nous trouvons les relations totales décrites en figure 3.28 (page ci-contre).

On constate que ces relations totales, même si elles ne sont qu'ordinales, respectent bien l'ordre donné par la forme normale du jeu.

Le calcul des équilibres de Nash faibles et forts nous donne les résultats suivants :

$$NE_{faible}^{disc} = NE_{fort}^{disc} = \{\bar{T}_1\bar{T}_2\}$$

Ce jeu n'a pas de stratégie dominée, on ne peut donc pas en éliminer.

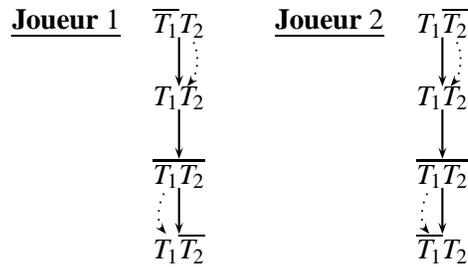


Figure 3.28 — Relations partielles calculées à partir du critère Discrimin

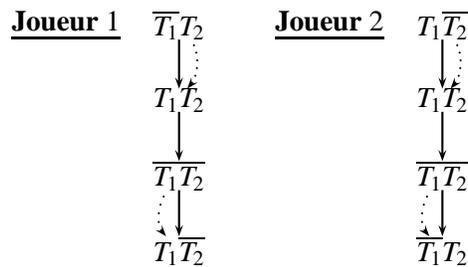


Figure 3.29 — Relations partielles calculées à partir du critère Leximin

Leximin Si l'on applique le critère leximin à chacun des joueurs, nous trouvons les relations totales décrites en figure 3.29. On constate que ce sont exactement les mêmes que celles trouvées en figure 3.28.

Ce jeu n'a toujours pas de stratégie dominée, et on obtient le même équilibre de Nash faible et fort :

$$NE^{lex} = \{\overline{T_1 T_2}\}$$

Best out Si l'on applique l'ordre best out à chacun des joueurs, nous trouvons les relations totales décrites sur la figure 3.30 et partir desquelles nous pouvons calculer les équilibres de Nash faibles et forts de ce jeu.

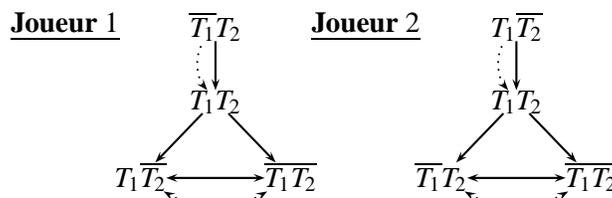


Figure 3.30 — Relations totales calculées à partir de l'ordre best out

On trouve :

$$NE^{bo} = \{T_1 \overline{T_2}, \overline{T_1} T_2, \overline{T_1} \overline{T_2}\}$$

Etudions à présent les stratégies dominées de ce jeu :

- * Joueur 1 : la stratégie $\overline{T_1}$ domine faiblement la stratégie T_1 . On élimine donc cette dernière.
- * Joueur 2 : Une fois la stratégie T_1 éliminée, le joueur 2 n'a plus de stratégie dominée.

En éliminant les stratégies dominées, on obtient donc deux profils de stratégies résultats : $\{\overline{T}_1 T_2, \overline{T}_1 \overline{T}_2\}$, ce qui raffine les résultats trouvés avec les équilibres de Nash.

Les stratégies ne sont pas ici strictement dominées, l'ordre d'élimination affecte donc les résultats. Etudions donc ce qui se passe si on élimine d'abord les stratégies dominées du joueur 2 :

- * Joueur 2 : la stratégie \overline{T}_2 domine faiblement la stratégie T_2 . On élimine donc cette dernière.
- * Joueur 1 : Une fois la stratégie T_2 éliminée, le joueur 1 n'a plus de stratégie dominée.

En éliminant les stratégies dominées, on obtient deux autres profils de stratégies résultats : $\{T_1 \overline{T}_2, \overline{T}_1 \overline{T}_2\}$, ce qui raffine les résultats trouvés avec les équilibres de Nash, mais de manière différente que celle obtenue avec l'ordre d'élimination précédent.

3.3.3 Quelques propriétés

Tout d'abord, il est possible de noter que les BP-jeux booléens avec le critère leximin et une seule strate par joueur correspondent aux jeux d'évaluation distribués introduits par [Harrenstein, 2004a].

Lemme 3.6. Soit $\succeq = \langle \succeq_1, \dots, \succeq_n \rangle$ et $\succeq' = \langle \succeq'_1, \dots, \succeq'_n \rangle$ deux collections de relations de préférences, et soit s un profil de stratégies.

1. Si \succeq est contenu dans \succeq' et si s est un équilibre de Nash fort en stratégies pures pour \succeq , alors s est un équilibre de Nash fort en stratégies pures pour \succeq' .
2. Si \succ est contenu dans \succ' et si s est un équilibre de Nash faible en stratégies pures pour \succ' , alors s est un équilibre de Nash faible en stratégies pures pour \succ .

Preuve du lemme :

1. Soit $s = (s_1, \dots, s_n)$ un équilibre de Nash fort en stratégies pures pour $\succeq = \langle \succeq_1, \dots, \succeq_n \rangle$. On a alors :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \preceq_i (s_i, s_{-i}) \quad (3.5)$$

Or, nous savons que \succeq est contenu dans \succeq' , i.e. $\forall i = 1 \dots n$, \succeq_i est contenu dans \succeq'_i . On a donc, d'après l'équation 3.5,

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \preceq'_i (s_i, s_{-i})$$

Le profil de stratégies s est donc bien un équilibre de Nash fort pour $\succeq' = \langle \succeq'_1, \dots, \succeq'_n \rangle$.

2. Soit $s = (s_1, \dots, s_n)$ un équilibre de Nash faible en stratégies pures pour $\succ' = \langle \succ'_1, \dots, \succ'_n \rangle$. On a alors :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \not\succeq'_i (s_i, s_{-i}) \quad (3.6)$$

Or, nous savons que \succ est contenu dans \succ' , donc $\not\succeq'$ est contenu dans $\not\succeq$, i.e. $\forall i = 1 \dots n$, $\not\succeq'_i$ est contenu dans $\not\succeq_i$. On a donc, d'après l'équation 3.6,

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \not\succeq_i (s_i, s_{-i})$$

Le profil de stratégies s est donc bien un équilibre de Nash faible pour $\succeq' = \langle \succeq'_1, \dots, \succeq'_n \rangle$.

■

De la propriété 3.11 (page 104) et du lemme 3.6 (page ci-contre) on déduit la propriété suivante :

Propriété 3.12. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un BP-jeu booléen et $Pref_G^c = \langle \succeq_1^c, \dots, \succeq_n^c \rangle$ l'ensemble des relations de préférences sur G à partir du critère $c \in \{disc, lex, bo\}$.

1. $NE_{fort}^{disc}(G) \subseteq NE^{lex}(G) \subseteq NE^{bo}(G)$
2. $NE^{lex}(G) \subseteq NE_{faible}^{disc}(G) \subseteq NE^{bo}(G)$

Preuve :

1. La suite d'implications " $s \in NE_{fort}^{disc} \Rightarrow s \in NE^{lex} \Rightarrow s \in NE^{bo}$ " est une conséquence directe de l'équation 3.4 (page 104) : $\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i$:

$$(s_i, s_{-i}) \succeq_i^{disc} (s'_i, s_{-i}) \Rightarrow (s_i, s_{-i}) \succeq_i^{lex} (s'_i, s_{-i}) \quad (3.7)$$

$$(s_i, s_{-i}) \succeq_i^{lex} (s'_i, s_{-i}) \Rightarrow (s_i, s_{-i}) \succeq_i^{bo} (s'_i, s_{-i}) \quad (3.8)$$

L'équation 3.7 nous montre que \succeq_i^{disc} est contenu dans \succeq_i^{lex} . Donc, d'après le lemme 3.6 (page précédente), si $s \in NE_{fort}^{disc}$, alors $s \in NE^{lex}$.

L'équation 3.8 nous montre que \succeq_i^{lex} est contenu dans \succeq_i^{bo} . Donc, d'après le lemme 3.6 (page précédente), si $s \in NE^{lex}$, alors $s \in NE^{bo}$.

2. La suite d'implications " $s \in NE^{lex} \Rightarrow s \in NE_{faible}^{disc} \Rightarrow s \in NE^{bo}$ " est une conséquence directe de l'équation 3.3 (page 104) : $\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i$:

$$(s_i, s_{-i}) \succ_i^{bo} (s'_i, s_{-i}) \Rightarrow (s_i, s_{-i}) \succ_i^{disc} (s'_i, s_{-i}) \quad (3.9)$$

$$(s_i, s_{-i}) \succ_i^{disc} (s'_i, s_{-i}) \Rightarrow (s_i, s_{-i}) \succ_i^{lex} (s'_i, s_{-i}) \quad (3.10)$$

L'équation 3.9 nous montre que \succ_i^{bo} est contenu dans \succ_i^{disc} . Donc, d'après le lemme 3.6 (page ci-contre), \succ_i^{disc} est contenu dans \succ_i^{bo} , et donc si $s \in NE_{faible}^{disc}$, alors $s \in NE^{bo}$.

L'équation 3.10 nous montre que \succ_i^{disc} est contenu dans \succ_i^{lex} . Donc, d'après le lemme 3.6 (page ci-contre), \succ_i^{lex} est contenu dans \succ_i^{disc} , et donc si $s \in NE^{lex}$, alors $s \in NE_{faible}^{disc}$.

■

Nous pouvons nous demander si un BP-jeu booléen peut être *approximé* en ne considérant que les k -premières strates de chaque joueur. L'objectif est double : obtenir un jeu plus simple (pour le calcul des équilibres de Nash en stratégies pures), et augmenter la possibilité de trouver un équilibre de Nash en stratégies pures significatif qui tient compte des strates les plus prioritaires.

Définition 3.27. Soit $G = (N = \{1, \dots, n\}, V, \pi, \Gamma, \Phi)$ un BP-jeu booléen. Soit $k \in \{1, p\}$.

$G^{[1 \rightarrow k]} = (N, V, \pi, \Gamma, \Phi^{[1 \rightarrow k]})$ représente le **jeu booléen k -réduit** de G dans lequel les buts de chaque joueur sont réduits à leurs k premières strates : $\Phi^{[1 \rightarrow k]} = \langle \Sigma_1^{[1 \rightarrow k]}, \dots, \Sigma_n^{[1 \rightarrow k]} \rangle$.

Lemme 3.7. Soit G un BP-jeu booléen. $\forall k \leq p$, $\forall c \in \{discr, lex, bo\}$ et $\forall i \in N$, on a :

$$s \succeq_i^{c, [1 \rightarrow k]} s' \Rightarrow s \succeq_i^{c, [1 \rightarrow k-1]} s' \quad (3.11)$$

$$s \not\succeq_i^{c, [1 \rightarrow k]} s' \Rightarrow s \not\succeq_i^{c, [1 \rightarrow k-1]} s' \quad (3.12)$$

Preuve du lemme :

1. Démontrons les implications $s \succeq_i^{c, [1 \rightarrow k]} s' \Rightarrow s \succeq_i^{c, [1 \rightarrow k-1]} s'$ une par une.

(a) Discrimin

$$s \succeq_i^{discr, [1 \rightarrow k]} s' \Leftrightarrow \exists l \in \{1, \dots, k\} \text{ tel que :}$$

$$Sat(s, \Sigma_i^l) \supseteq Sat(s', \Sigma_i^l) \text{ et } \forall j < l, Sat(s, \Sigma_i^j) = Sat(s', \Sigma_i^j)$$

2 cas sont maintenant possibles.

i. $l = k$.

Donc, $\forall j, j < k$, et donc $\forall j \in \{1, \dots, k-1\}$, $Sat(s, \Sigma_i^j) = Sat(s', \Sigma_i^j)$.

Ce qui nous donne bien $s \succeq_i^{discr, [1 \rightarrow k-1]} s'$.

ii. $l < k$, donc $l \leq k-1$.

Dans ce cas, on a $l \in \{1, \dots, k-1\}$. Et on a donc bien $s \succeq_i^{discr, [1 \rightarrow k-1]} s'$.

(b) Leximin

$$s \succeq_i^{lex, [1 \rightarrow k]} s' \Leftrightarrow \exists l \in \{1, \dots, k\} \text{ tel que}$$

$$|Sat(s, \Sigma_i^l)| \geq |Sat(s', \Sigma_i^l)| \text{ et } \forall j < l, |Sat(s, \Sigma_i^j)| = |Sat(s', \Sigma_i^j)|$$

2 cas sont maintenant possibles.

i. $l = k$.

Donc, $\forall j, j < k$, et donc $\forall j \leq k-1$, $|Sat(s, \Sigma_i^j)| = |Sat(s', \Sigma_i^j)|$.

Ce qui nous donne bien $s \succeq_i^{lex, [1 \rightarrow k-1]} s'$.

ii. $l < k$, donc $l \leq k-1$.

Dans ce cas, on a $l \in \{1, \dots, k-1\}$. Et on a donc bien $s \succeq_i^{lex, [1 \rightarrow k-1]} s'$.

(c) Best-out

$$\begin{aligned} s \succeq_i^{bo, [1 \rightarrow k]} s' &\Leftrightarrow a_i^{[1 \rightarrow k]}(s) \geq a_i^{[1 \rightarrow k]}(s') \\ &\quad \underbrace{\min(\{j \in [1, k] \text{ tel que } \exists \varphi_l \in \Sigma_i^j : s \not\models \varphi_l\})}_{\text{noté } A} \\ &\Leftrightarrow \geq \\ &\quad \underbrace{\min(\{j \in [1, k] \text{ tel que } \exists \varphi_l \in \Sigma_i^j : s' \not\models \varphi_l\})}_{\text{noté } B} \end{aligned}$$

3 cas sont maintenant possibles.

i. $A < k$, et $B \leq A < k$.

Alors, on a : $A \leq k-1$, et $B \leq A \leq k-1$.

Et donc, $a_i^{[1 \rightarrow k-1]}(s) \geq a_i^{[1 \rightarrow k-1]}(s')$.

ii. $A = k$, et $B \leq A = k$.

Dans ce cas, $\min(\{j \in [1, k-1] \text{ tel que } \exists \varphi_l \in \Sigma_i^j : s \not\models \varphi_l\}) = \infty$.

Donc $B \leq A = \infty$.

On a encore bien $a_i^{[1 \rightarrow k-1]}(s) \geq a_i^{[1 \rightarrow k-1]}(s')$.

iii. $A = \infty$.

On aura alors toujours $\min(\{j \in [1, k-1] \text{ tel que } \exists \varphi_l \in \Sigma_i^j : s \not\models \varphi_l\}) = \infty$.

Et donc $B \leq A = \infty$, et $a_i^{[1 \rightarrow k-1]}(s) \geq a_i^{[1 \rightarrow k-1]}(s')$.

Pour chacun de ces cas possibles, on a bien $s \succeq_i^{bo, [1 \rightarrow k-1]} s'$.

2. Démontrons les implications $s \not\prec_i^{c, [1 \rightarrow k]} s' \Rightarrow s \not\prec_i^{c, [1 \rightarrow k-1]} s'$ une par une. Pour cela, on remarque que :

$$(s \not\prec_i^{c, [1 \rightarrow k]} s' \Rightarrow s \not\prec_i^{c, [1 \rightarrow k-1]} s') \Leftrightarrow (s \succ_i^{c, [1 \rightarrow k-1]} s' \Rightarrow s \succ_i^{c, [1 \rightarrow k]} s')$$

(a) Discrimin

$$s \succ_i^{discr, [1 \rightarrow k-1]} s' \Leftrightarrow \exists l \in \{1, \dots, k-1\} \text{ tel que}$$

$$Sat(s, \Sigma_i^l) \supset Sat(s', \Sigma_i^l) \text{ et } \forall j < l, Sat(s, \Sigma_i^j) = Sat(s', \Sigma_i^j)$$

On a donc bien :

$$\exists l \in \{1, \dots, k\} \text{ tel que } Sat(s, \Sigma_i^l) \supset Sat(s', \Sigma_i^l) \text{ et } \forall j < l, Sat(s, \Sigma_i^j) = Sat(s', \Sigma_i^j)$$

$$\text{Donc, } s \succ_i^{discr, [1 \rightarrow k]} s'$$

(b) Leximin

$$s \succ_i^{lex, [1 \rightarrow k-1]} s' \Leftrightarrow \exists l \in \{1, \dots, k-1\} \text{ tel que}$$

$$|Sat(s, \Sigma_i^l)| > |Sat(s', \Sigma_i^l)| \text{ et } \forall j < l, |Sat(s, \Sigma_i^j)| = |Sat(s', \Sigma_i^j)|$$

On a donc bien : $\exists l \in \{1, \dots, k\} \text{ tel que}$

$$\exists l \in \{1, \dots, k\} : |Sat(s, \Sigma_i^l)| > |Sat(s', \Sigma_i^l)| \text{ et } \forall j < l, |Sat(s, \Sigma_i^j)| = |Sat(s', \Sigma_i^j)|$$

$$\text{Donc, } s \succ_i^{lex, [1 \rightarrow k]} s'$$

(c) Best-out

$$s \succ_i^{bo, [1 \rightarrow k-1]} s' \Leftrightarrow a_i^{[1 \rightarrow k-1]}(s) > a_i^{[1 \rightarrow k-1]}(s') \Leftrightarrow A > B$$

avec

$$* A = \min(\{j \in [1, k-1] \text{ tel que } \exists \varphi_l \in \Sigma_i^j : s \not\models \varphi_l\}),$$

$$* B = \min(\{j \in [1, k-1] \text{ tel que } \exists \varphi_l \in \Sigma_i^j : s' \not\models \varphi_l\}).$$

Ces equations sont évidemment toujours vraies pour $j \in [1, k]$.

On a donc bien $s \succ_i^{bo, [1 \rightarrow k]} s'$.

■

Propriété 3.13. Soit G un BP-jeu booléen, et soit $c \in \{discr, lex, bo\}$.

Si s est un équilibre de Nash fort (resp. faible) en stratégies pures pour $Pref_G^c_{[1 \rightarrow k]}$, alors s est aussi un équilibre de Nash fort (resp. faible) pour $Pref_G^c_{[1 \rightarrow k-1]}$.

Preuve :

1. Equilibres de Nash forts en stratégies pures :

Soit $s = (s_1, \dots, s_n)$ un équilibre de Nash fort en stratégies pures pour $Pref_G^c_{[1 \rightarrow k]}$. Alors :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \preceq_i^{[1 \rightarrow k]} (s_i, s_{-i})$$

Or, d'après le lemme 3.7, quel que soit le critère utilisé, $s \succeq_i^{[1 \rightarrow k]} s' \Rightarrow s \succeq_i^{[1 \rightarrow k-1]} s'$.

On a donc :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \preceq_i^{[1 \rightarrow k-1]} (s_i, s_{-i})$$

s est donc un équilibre de Nash fort en stratégies pures pour $Pref_G^c_{[1 \rightarrow k]}$.

2. Equilibres de Nash faibles en stratégies pures :

Soit $s = (s_1, \dots, s_n)$ un équilibre de Nash faible en stratégies pures pour $Pref_G^c_{[1 \rightarrow k]}$. Alors :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \not\preceq_i^{[1 \rightarrow k]} (s_i, s_{-i})$$

Or, d'après le lemme 3.7, quel que soit le critère utilisé, $s \not\preceq_i^{[1 \rightarrow k]} s' \Rightarrow s \not\preceq_i^{[1 \rightarrow k-1]} s'$.

On a donc :

$$\forall i \in \{1, \dots, n\}, \forall s'_i \in S_i, (s'_i, s_{-i}) \not\preceq_i^{[1 \rightarrow k-1]} (s_i, s_{-i})$$

s est donc un équilibre de Nash faible en stratégies pures pour $Pref_G^c_{[1 \rightarrow k]}$.

■

Lemme 3.8. Soit G un BP-jeu booléen, et soit $G^{[1]}$ le jeu booléen 1-réduit associé à G .

Si, quel que soit le critère utilisé (discrimin, leximin ou best-out), le jeu $G^{[1]}$ n'a pas d'équilibre de Nash en stratégies pures, alors le jeu G n'en a pas non plus.

Preuve du lemme : Supposons que le jeu G a un équilibre de Nash. Sachant que $G = G^{[1 \rightarrow p]}$, d'après la propriété 3.13, le jeu $G^{[1 \rightarrow p-1]}$ aura aussi un équilibre de Nash. Puis, en appliquant la même propriété, il en sera de même pour les jeux $G^{[1 \rightarrow p-2]}$,

$G^{[1 \rightarrow p-3]}$, jusqu'au jeu $G^{[1]}$.
 Donc, si G a un équilibre de Nash, alors $G^{[1]}$ a un équilibre de Nash.
 Et donc, si $G^{[1]}$ n'a pas d'équilibre de Nash, alors G n'aura pas d'équilibre de Nash.

■

La réciproque est fautive : le jeu $G^{[1]}$ peut avoir un équilibre de Nash, alors que le jeu G n'en a pas. Voici un contre-exemple :

Exemple 3.18. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen, avec :

- * $N = \{1, 2\}$,
- * $V = \{a, b\}$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$,
- * $\Sigma_1 = \langle a \rightarrow b; b \rightarrow a \rangle$,
- * $\gamma_1 = \gamma_2 = \top$,
- * $\Sigma_2 = \langle a \leftrightarrow \neg b; \neg b \rangle$

Si l'on applique l'ordre best out à chacun des joueurs, nous trouvons les relations totales données dans la figure 3.31.

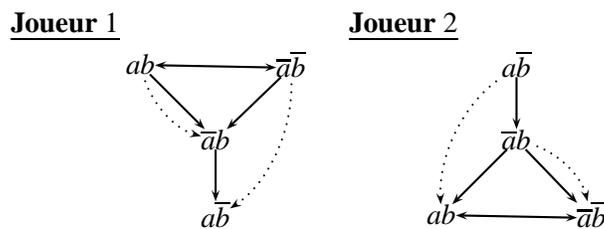


Figure 3.31 — Relations totales calculées à partir de l'ordre best out

Ce jeu n'a pas d'équilibre de Nash.

Si on étudie à présent le jeu booléen $G^{[1]}$ associé dans lequel les buts des joueurs sont réduits à la première strate de leur but dans G , on obtient le jeu suivant : $G^{[1]} = \{N, V, \pi, \Gamma, \Phi^{[1]}\}$ un jeu booléen, avec :

- * $N = \{1, 2\}$,
- * $V = \{a, b\}$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$,
- * $\gamma_1 = \gamma_2 = \top$,
- * $\Sigma_1^{[1]} = \{a \rightarrow b\}$,
- * $\Sigma_2^{[1]} = \{a \leftrightarrow \neg b\}$

On peut à présent construire la forme normale de $G^{[1]}$:

		2	\bar{b}	b
1				
	\bar{a}	(1, 0)	(1, 1)	
	a	(0, 1)	(1, 0)	

Ce jeu a un équilibre de Nash : $\bar{a}b$.

Cet exemple prouve que la propriété 3.13 peut être utilisée pour trouver le bon niveau d'approximation pour un BP-jeu booléen. Par exemple, nous pouvons nous focaliser sur le plus grand k tel que $G^{[1 \rightarrow k]}$ a un équilibre de Nash fort en stratégies pures, et faire de même pour les équilibres de Nash faibles.

D'autre part, cette propriété pourrait être utilisée pour identifier un équilibre de Nash "approché", en maximisant le nombre de strates pour lesquelles un équilibre de Nash existe. Par exemple, soit G un BP-jeu booléen. Nous avons déjà vu que si $G^{[1]}$ n'a aucun équilibre de Nash faible ou fort, G non plus. Mais si $G^{[1]}$ en a un, alors nous pouvons vérifier s'il existe toujours pour $G^{[1 \rightarrow 2]}$. Si ce n'est pas le cas, nous savons que G n'a aucun équilibre de Nash ; mais dans le cas contraire, il est possible de vérifier $G^{[1 \rightarrow 3]}$, et ainsi de suite. Si le nombre de strates est important, continuer cette vérification jusqu'à la strate p permet d'avoir ainsi une approximation de l'équilibre de Nash de G .

3.4 Préférences non dichotomiques et graphe de dépendance

Nous pouvons à présent généraliser le graphe de dépendance entre joueurs d'un jeu booléen à un L -jeu booléen, pour un langage L quelconque. Rappelons que, dans la section 2.3 (page 37), un joueur i était dépendant d'un joueur j si son but propositionnel φ_i était dépendant d'une variable contrôlée par j . Nous devons donc commencer par généraliser la dépendance entre une formule et une variable par une notion de dépendance entre une relation de préférence (ou une entrée syntactique dans un langage de représentation compacte de préférences à partir de laquelle il est possible de déduire cette relation de préférence) et une variable. Plusieurs définitions ont été étudiées dans [Besnard *et al.*, 2006] dans le cas général où les relations de préférences sont des pré-ordres partiels. Dans le cas spécifique où les relations de préférences sont des pré-ordres complets, il semble n'y avoir qu'une définition appropriée : une relation de préférence \succeq est indépendante d'une variable propositionnelle x si pour tous les états l'agent est indifférent entre cet état, et l'état obtenu en changeant la valeur de x (voir [Besnard *et al.*, 2006]).

Définition 3.28. Une relation de préférence \succeq sur S est **indépendante** de la variable propositionnelle $x \in V$ si et seulement si pour tout $s \in S$, $\text{switch}(s, x) \sim s$.

Le but Φ est indépendant de $x \in V$ si et seulement si la relation de préférence induite par Φ est indépendante de x .

On peut alors introduire le lemme suivant :

Lemme 3.9. Soit G un L -jeu booléen.

Le but Φ_i est indépendant d'une variable propositionnelle $x \in V$ si et seulement si il existe Φ'_i tel que x n'apparaît pas dans Φ'_i , et que Φ_i et Φ'_i induisent la même relation de préférence.

Preuve du lemme : On introduit la notation S^{-x} , qui est définie par : pour tout $j \in N$, $S_j^{-x} = \{s_j \in 2^{V \setminus \{x\}} \mid s_j \models \gamma_j\}$, et $S^{-x} = S_1^{-x} \times \dots \times S_n^{-x}$.
 \Rightarrow Φ_i est indépendant de x , donc, si on note \succsim_i la relation de préférence induite par Φ_i , on a pour tout $s \in S$, $\text{switch}(s, x) \sim_i s$. Cette relation de préférence est donc identique à celle définie sur S^{-x} . Il existe donc Φ'_i défini sur $V \setminus \{x\}$ qui induit cette relation de préférence.
 \Leftarrow Soit Φ'_i défini sur $V \setminus \{x\}$. La relation de préférence induite par Φ'_i est donc

définie sur S^{-x} . Si cette relation de préférence est équivalente à celle induite par Φ_i , on a pour tout $s \in S$, $\text{switch}(s, x) \sim_i s$. Φ_i est donc indépendant de x .

■

Il est à présent possible de définir l'ensemble des variables et joueurs utiles d'un L -jeu booléen :

Définition 3.29. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un L -jeu booléen. L'ensemble des **variables utiles** pour un joueur i , noté RV_i , est l'ensemble des variables $v \in V$ telles que Φ_i n'est pas indépendant de v .

L'ensemble des **joueurs utiles** pour un joueur i , noté RP_i , est l'ensemble des joueurs $j \in N$ tels que j contrôle au moins une variable utile de i : $RP_i = \bigcup_{v \in RV_i} \pi^{-1}(v)$.

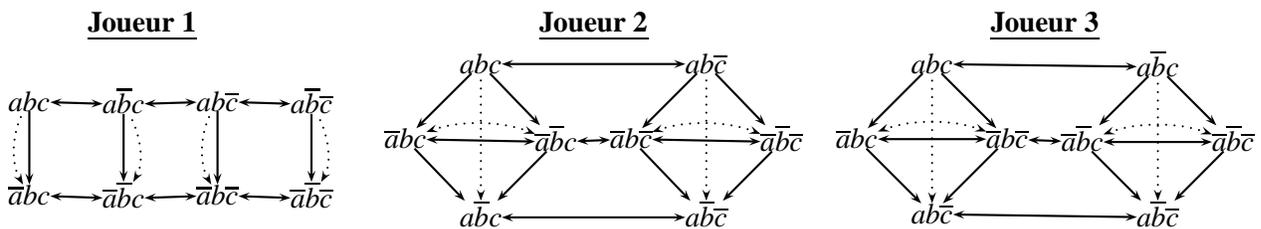
La définition du graphe de dépendance d'un L -jeu booléen est exactement celle donnée définition 2.9 (page 38).

Ces définitions ne dépendent pas du langage de représentation de préférences choisi. Pour illustrer ces concepts, nous donnons un exemple dans lequel les préférences sont représentées par des buts à priorité, en utilisant le critère leximin.

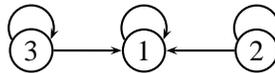
Exemple 3.19. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un BP-jeu booléen, avec

- * $N = \{1, 2, 3\}$,
- * $V = \{a, b, c\}$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$,
- * $\gamma_1 = \gamma_2 = \gamma_3 = \top$,
- * $\Sigma_1 = \langle a \rangle$,
- * $\Sigma_2 = \langle (b \vee \neg a); a \rangle$, et
- * $\Sigma_3 = \langle (c \vee \neg a); a \rangle$.

Les relations de préférences $\text{Pref}_G^{\text{lex}} = \langle \succeq_1^{\text{lex}}, \succeq_2^{\text{lex}}, \succeq_3^{\text{lex}} \rangle$ sont décrites ci-dessous :



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



La définition de la projection d'un L -jeu booléen sur un ensemble stable est également la même que la définition 2.11 (page 40).

Il est possible de généraliser quelques propriétés qui étaient valables pour les préférences dichotomiques. Par exemple, la propriété 2.3 (page 40) est généralisable dans ce cadre :

Propriété 3.14. Si B est un ensemble stable, $G_B = (B, V_B, \pi_B, \Gamma_B, \Phi'_B)$, avec Φ'_B étant équivalent à Φ_B et ne contenant que des variables de V_B , est un L -jeu booléen.

Preuve : La seule différence entre cette preuve et celle de la propriété 2.3 (page 40) est l'étape consistant à vérifier que toutes les variables utilisées par les joueurs dans B appartiennent à V_B . Nous devons vérifier que toutes les variables utilisées par les joueurs dans B appartiennent à V_B .

Supposons tout d'abord qu'il existe un $i \in B$ tel que Φ_i n'est pas indépendant de $V_{-B} = V \setminus V_B$. Donc, il existe $x \in V_{-B}$, il existe un $s \in S$ tel que $\text{switch}(s, x) \sim_i s$. Dans ce cas, d'après la définition 3.29 (page précédente), $x \in RV_i$. Comme $x \in V_{-B}$, et que B est stable, pour tout $j \in B$, $x \notin \pi_j$. Soit $k \in N \setminus B$ tel que $x \in \pi_k$. Comme $x \in RV_i$, on a par définition $k \in RP_i$. Donc, $k \in R(i)$, mais $k \notin B$: c'est en contradiction avec le fait que B est stable.

Φ_i est donc indépendant de V_{-B} . D'après le lemme 3.9 (page 116), il existe Φ'_i qui induit la même relation de préférences que Φ_i et qui ne contient aucune variable de V_{-B} .

■

Il en est de même pour la première assertion de la propriété 2.9 (page 46) :

Propriété 3.15. Soit G un L -jeu booléen tel que la partie irréflexive du graphe de dépendance \mathcal{P} de G soit acyclique. Alors G a un équilibre de Nash fort en stratégies pures.

Preuve :

On peut supposer sans perte de généralité que les joueurs sont numérotés de sorte que pour tout $i \in N$, si i dépend de j , alors $j < i$. Soit $s = (s_1, \dots, s_N)$ défini inductivement comme suit : pour $i = 1, \dots, N$, on prend s_i tel que pour tout s'_i , $(s_1, \dots, s_i) \succeq_i (s_1, \dots, s'_i)$. Un tel s_i existe car i ne dépend pas de k pour tout $k > i$. On a donc construit un profil de stratégies s tel que pour tout i , pour tout s'_i , $(s_i, s_{-i}) \succeq_i (s'_i, s_{-i})$. s est donc un équilibre de Nash fort en stratégies pures.

■

La propriété 2.10 (page 47) est également généralisable dans ce cadre :

Propriété 3.16. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un L -jeu booléen, $B \subseteq N$ un ensemble stable pour R , et $G_B = (B, V_B, \pi_B, \Gamma_B, \Phi'_B)$ la projection de G sur B .

Si s est un équilibre de Nash fort pour G , alors s_B est un équilibre de Nash fort pour G_B .

Preuve : Soit s un équilibre de Nash fort pour G : $\forall i \in N$, $\forall s'_i \in S_i$, $(s'_i, s_{-i}) \preceq_i (s_i, s_{-i})$.

Nous savons que $s = (s_B, s_{-B})$, et nous voulons vérifier que s_B est un équilibre de Nash fort pour G_B .

Soit $i \in B$. Supposons que s_B n'est pas un équilibre de Nash fort pour G_B , i.e. $\exists s'_i \in S_i$ tel que $(s'_i, s_{B-i}) \not\preceq_i (s_i, s_{B-i})$.

Comme $i \in B$, et que B est stable, nous savons que les seuls joueurs qui ont une

influence sur i sont dans B , et que donc s_{-B} n'a aucune influence sur les préférences de i . On a donc $(s'_i, s_{B-i}, s_{-B}) \not\preceq_i (s_i, s_{B-i}, s_{-B})$, ce qui est en contradiction avec le fait que s est un équilibre de Nash fort pour G . ■

Reprenons à présent la propriété 2.11 (page 48) :

Propriété 3.17. *Soit B et C deux ensembles stables de joueurs, et soit G_B et G_C deux L -jeux booléens associés.*

Si s_B est un équilibre de Nash fort pour G_B et s_C un équilibre de Nash fort pour G_C tels que $\forall i \in B \cap C, s_{B,i} = s_{C,i}$ ¹¹, alors $s_{B \cup C}$ est un équilibre de Nash fort pour le jeu $G_{B \cup C}$.

Preuve : B et C sont stables, donc d'après la propriété 2.2 (page 39), $B \cup C$ est un ensemble stable, et d'après la propriété 3.14 (page précédente), $G_{B \cup C}$ est un jeu booléen.

Soit $i \in B \cup C$. 3 cas sont possibles (2 étant symétriques) :

- * $i \in B \setminus C$ (ou $i \in C \setminus B$). s_B est un équilibre de Nash fort pour G_B , donc on a $\forall s'_i \in S_i, (s'_i, s_{B-i}) \preceq_i (s_i, s_{B-i})$. Puisque $i \notin C$, nous pouvons écrire $\forall s'_i \in S_i, (s'_i, s_{B-i}, s_C) \preceq_i (s_i, s_{B-i}, s_C)$, ce qui revient à $\forall s'_i \in S_i, (s'_i, s_{B \cup C - i}) \preceq_i (s_i, s_{B \cup C - i})$.
- * $i \in B \cap C$. On a alors $\forall k \in B \cap C, s_{B,k} = s_{C,k}$. s_B est un équilibre de Nash fort pour G_B , donc on a $\forall s'_i \in S_i, (s'_i, s_{B-i}) \preceq_i (s_i, s_{B-i})$; s_C est un équilibre de Nash fort pour G_C , donc on a $\forall s'_i \in S_i, (s'_i, s_{C-i}) \preceq_i (s_i, s_{C-i})$. Puisque $\forall k \in B \cap C, s_{B,k} = s_{C,k}$, nous avons $\forall s'_i \in S_i, (s'_i, s_{B-i}, s_{C-i}) \preceq_i (s_i, s_{B-i}, s_{C-i})$, c'est-à-dire $\forall s'_i \in S_i, (s'_i, s_{B \cup C - i}) \preceq_i (s_i, s_{B \cup C - i})$.

$s_{B \cup C}$ est donc bien un équilibre de Nash fort pour $G_{B \cup C}$. ■

On peut alors généraliser la propriété 2.12 (page 50) exactement de la même façon que dans le chapitre 2 (page 31) :

Propriété 3.18. *Soit $G = (N, V, \pi, \Gamma, \Phi)$ un L -jeu booléen, et soit $B_1 \dots B_p$ p ensembles stables de joueurs, tels que $B_1 \cup \dots \cup B_p = N$. Soit G_{B_1}, \dots, G_{B_p} les p L -jeux booléens associés.*

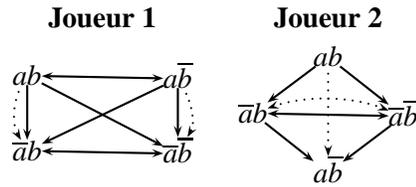
Si $\exists s_{B_1} \dots s_{B_p}$ des équilibres de Nash fort pour G_{B_1}, \dots, G_{B_p} tels que $\forall i, j \in \{1, \dots, p\}, \forall k \in B_i \cap B_j, s_{B_i, k} = s_{B_j, k}$, alors $s = (s_{B_1}, \dots, s_{B_p})$ est un équilibre de Nash fort pour G .

Exemple 3.19, suite : *Les ensembles de joueurs $B = \{1, 2\}$ et $C = \{1, 3\}$ sont stables. Nous obtenons ainsi deux nouveaux BP-jeux booléens :*

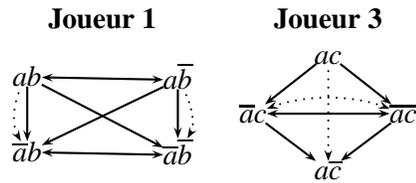
- * $G_B = (B, V_B, \Gamma_B, \pi_B, \Phi_B)$, avec $B = \{1, 2\}$, $V_B = \{a, b\}$, $\pi_1 = a$, $\pi_2 = b$, $\gamma_1 = \gamma_3 = \top$, $\Sigma_1 = \langle a \rangle$, et $\Sigma_2 = \langle (b \vee \neg a); a \rangle$. Les relations de préférences $Pre f_G^{lex} = \langle \succeq_1^{lex}, \succeq_2^{lex} \rangle$ sont représentées sur la figure suivante :

G_B a un équilibre de Nash (faible et fort) : $\{ab\}$ (noté $s_B = (s_{B,1}, s_{B,2})$).

¹¹ $s_{B,i}$ représente la stratégie du joueur i pour le jeu G_B .



* $G_C = (C, V_C, \pi_C, \Gamma_C, \Phi_C)$, avec $C = \{1, 3\}$, $V_C = \{a, c\}$, $\pi_1 = a$, $\pi_3 = c$, $\gamma_1 = \gamma_3 = \top$, $\Sigma_1 = \langle a \rangle$ et $\Sigma_3 = \langle (c \vee \neg a); a \rangle$. Les relations de préférences $Pre f_G^{lex} = \langle \succeq_1^{lex}, \succeq_3^{lex} \rangle$ sont représentées sur la figure suivante :



G_C a un équilibre de Nash (faible et fort) : $\{ac\}$ (noté $s_C = (s_{C,1}, s_{C,3})$).

$B \cap C = \{1\}$. Comme nous avons $s_{B,1} = s_{C,1} = a$: $G_{B \cup C}$ a un équilibre de Nash en stratégies pures : $\{abc\}$.

On remarque ici que les deux jeux que nous venons d'étudier ici, G_B et G_C , sont beaucoup plus petits, et donc faciles à étudier, que le jeu original présenté sur la figure 3.19 (page 117).

4 Jeux, logique propositionnelle et représentation compacte

Nous ne sommes pas les premiers à étudier des concepts tels que les équilibres de Nash et les stratégies dominées d'un point de vue logique. Mis à part les jeux booléens [Harrenstein *et al.*, 2001; Harrenstein, 2004a; Dunne et van der Hoek, 2004], il existe un certain nombre d'autres travaux sur l'étude des jeux statiques dans une perspective logique et IA, qui permettent de représenter de tels jeux en utilisant des langages de représentation compacte de préférences.

4.1 Jeux et programmation logique

Quelques lignes récentes de travail permettent d'exprimer des jeux avec des préférences *ordinales* dans des cadres bien connus en intelligence artificielle. Une façon, parmi d'autres, de représenter ces préférences est d'utiliser des *programmes logiques*, comme cela est fait dans [De Vos et Vermeir, 1999; Foo *et al.*, 2004].

Nous allons tout d'abord présenter ces programmes logiques, puis nous introduirons les travaux de [De Vos et Vermeir, 1999] et de [Foo *et al.*, 2004].

4.1.1 Programmes logiques

Nous allons ici donner rapidement le vocabulaire et les définitions utiles pour les programmes logiques.

On considère ici des programmes logiques étendus propositionnels avec deux sortes de négation, la négation classique \neg et la négation de défaut *not* [Gelfond et Lifschitz, 1991]. Intuitivement, *not* a est vrai quand il n'y a aucune raison de croire a , tandis que $\neg a$ est vrai lorsqu'il y a une preuve que a est faux. Un *programme logique étendu* P est une collection finie de règles de la forme :

$$c \leftarrow a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_k$$

dans laquelle c , a_j et b_l sont des littéraux.

Le littéral c forme la *tête* de la règle r , que l'on pourra noter $tete(r)$. Le *corps* de la règle r , noté $corps(r)$, représente les littéraux $a_1 \dots a_m, \text{not } b_1, \dots, \text{not } b_k$. a_1, \dots, a_m constituent les *prérequis* de la règle r et sont notés $pre(r)$

\leftarrow est oublié quand $m = 0$ et $k = 0$, et dans ce cas cette règle sera appelée un *fait*.

Définition 4.1. Une règle r est **entravée par un littéral** d s'il existe un $i \in \{1, \dots, k\}$ tel que $d = b_i$. r est **entravée par un ensemble de littéraux** X si X contient un littéral qui entrave r .

Une règle r est **applicable** dans X , noté $X \models \text{corps}(r)$, quand elle n'est pas entravée par X et que ses prérequis appartiennent à X .

Une règle r est **satisfiable** par X , noté $X \models r$, si $\text{tete}(r) \in X$ ou si $X \not\models \text{corps}(r)$.

Un ensemble de littéraux X est **consistant** si, pour tous les atomes a , si $a \in X$ alors $\neg a \notin X$.

Il est à présent possible de définir l'ensemble des réponses d'un programme logique.

Définition 4.2. Un **ensemble de réponses** d'un programme logique P est un ensemble de littéraux X satisfaisant deux conditions :

1. Si $r \in P$ est une règle applicable alors elle est appliquée, et $\text{tete}(r) \in X$,
2. Les seules règles utilisées sont celles qui ne sont pas entravées par X .

4.1.2 Programme logique de choix

Dans [De Vos et Vermeir, 1999], un jeu est représenté par un *programme logique de choix* (choice logic program), dans lequel un ensemble de règles exprime que le joueur a choisi la "meilleure réponse" étant donné les choix des autres joueurs.

Un programme logique de choix est un programme logique *semi-négatif* dans la mesure où il n'utilise pas de négation dans le corps des règles. L'opérateur \oplus est interprété comme un ou-exclusif. $p \oplus q$ signifie donc "p ou q, mais pas les deux".

Définition 4.3 ([De Vos et Vermeir, 1999]). Un **programme logique de choix** est un ensemble fini de règles de la forme

$$A \leftarrow B$$

où A et B sont des ensembles finis d'atomes.

Intuitivement, les atomes de A sont supposés être séparés par \oplus , tandis que B est une conjonction d'atomes (séparés par ",").

Un jeu sous forme normale est alors représenté par un programme logique de choix de la façon suivante :

Définition 4.4 ([De Vos et Vermeir, 1999]). Soit $G = (N, S, (\geq_i)_{i \in N})$ un jeu stratégique. Le **programme logique de choix** P_G associé à G contient les règles suivantes :

- * Pour chaque joueur i , P_G contient la règle $S_i \leftarrow$, qui assure que chaque joueur choisit exactement une stratégie de S_i .
- * Pour chaque joueur i et pour tout $s_{-i} \in S_{-i}$, P_G contient une règle $B_i(s_{-i}) \leftarrow s_{-i}$, où $B_i(s_{-i})$ représente la meilleure réponse du joueur i si les autres joueurs jouent s_{-i} . Cette règle modélise le fait que chaque joueur choisira la meilleure réponse possible étant donné les choix des autres joueurs.

Exemple 4.1. De nouveau, les stratégies du joueur 1 (resp. 2) sont dénotées A_1, T_1 (resp. A_2, T_2). Le **dilemme du prisonnier** peut être représenté par le programme logique de choix suivant :

$$A_1 \oplus T_1 \leftarrow \quad (4.1)$$

$$A_2 \oplus T_2 \leftarrow \quad (4.2)$$

$$A_1 \leftarrow T_2 \quad (4.3)$$

$$A_1 \leftarrow A_2 \quad (4.4)$$

$$A_2 \leftarrow T_1 \quad (4.5)$$

$$A_2 \leftarrow A_1 \quad (4.6)$$

L'équation 4.1 signifie que le joueur 1 a le choix entre jouer A_1 et T_1 . L'équation 4.3 signifie que si le joueur 2 joue T_2 , le joueur 1 préfère A_1 , etc.

Définition 4.5 ([De Vos et Vermeir, 1999]). Soit P un programme logique de choix. La **base de Herbrand** de P , dénotée \mathcal{B}_P , est l'ensemble de tous les atomes apparaissant dans les règles de P . Tout sous-ensemble de \mathcal{B}_P est une **interprétation**.

Une interprétation I est un **modèle** de P si pour toute règle $A \leftarrow B$, si $B \subseteq I$ alors $I \cap A$ est un singleton. Un modèle de P minimal pour l'inclusion est dit **stable**.

De Vos et Vermeir ont alors montré qu'il existe pour chaque jeu un programme logique de choix tel que l'ensemble des modèles stables du programme coïncide avec l'ensemble des équilibres de Nash du jeu.

Exemple 4.1 (page précédente) – suite *Etudions les différentes interprétations de ce jeu. On sait tout d'abord que les interprétations de \mathcal{B}_P comprenant $\{A_1, T_1\}$ ou $\{A_2, T_2\}$ ne peuvent pas être des modèles de P , car les deux premières règles seraient alors violées.*

Les interprétations composées d'un singleton ne sont pas non plus des modèles. Etudions par exemple l'interprétation $\{A_1\}$. On a pour la règle 4.6 $B \subseteq \{A_1\}$, et $\{A_1\} \cap A$ qui n'est pas un singleton.

Etudions à présent $\{A_1, A_2\}$, $\{A_1, T_2\}$, $\{T_1, A_2\}$ et $\{T_1, T_2\}$:

- * $\{A_1, A_2\}$ est un modèle, $\{A_1, A_2\} \cap A$ est un singleton pour toutes les règles telle que $B \subseteq \{A_1, A_2\}$.
- * $\{A_1, T_2\}$ n'est pas un modèle : on a pour la règle 4.6 $B \subseteq \{A_1, T_2\}$, et $\{A_1, T_2\} \cap A$ qui n'est pas un singleton.
- * $\{T_1, A_2\}$ n'est pas un modèle : on a pour la règle 4.4 $B \subseteq \{T_1, A_2\}$, et $\{T_1, A_2\} \cap A$ qui n'est pas un singleton.
- * $\{T_1, T_2\}$ n'est pas un modèle : on a pour la règle 4.3 $B \subseteq \{T_1, T_2\}$, et $\{T_1, T_2\} \cap A$ qui n'est pas un singleton.

Ce programme logique de choix a donc un seul modèle : $\{A_1, A_2\}$, qui est donc un modèle stable et qui est bien l'équilibre de Nash de ce jeu.

Cette propriété fournit une méthode systématique permettant de calculer les équilibres de Nash dans les jeux finis. Cette traduction a une limitation : sa taille. Un programme logique de choix a la même taille que la forme normale d'un jeu.

4.1.3 Programme logique avec des disjonctions ordonnées

4.1.3.1 Présentation et définitions

Les *programmes logiques avec des disjonctions ordonnées* (logic program with ordered disjunction - LPOD) ont été introduits dans [Brewka, 2002]. Ces programmes logiques sont des extensions de programmes logiques avec deux sortes de négation, et utilisent le connecteur \times introduit dans la logique propositionnelle qualitative (Qualitative Choice Logic - QCL) par [Brewka *et al.*, 2002]. Intuitivement, $a \times b$ signifie “si a est possible, alors je préfère a , sinon, je veux b ”.

Définition 4.6 ([Brewka, 2002; Gelfond et Lifschitz, 1991]). *Un LPOD est constitué de règles r de la forme :*

$$c_1 \times \dots \times c_n \leftarrow a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_k$$

dans laquelle c_i , a_j et b_l sont des littéraux. Intuitivement, la tête d’une règle de ce type se lit “si c est possible je veux avoir c_1 , si je ne peux pas avoir c_1 je préfère c_2 , ..., si je ne peux avoir aucun des c_i , $i \in \{1, \dots, n-1\}$, alors je veux c_n ”.

Les littéraux c_1, \dots, c_n sont appelés les **choix** de la règle r .

Exemple 4.2 ([Brewka, 2002]). *Marjolaine a un après-midi de libre. Elle a le choix entre aller à la plage ou au cinéma. Elle préfère habituellement aller au cinéma qu’à la plage, excepté lorsqu’il fait chaud, auquel cas elle préfère aller à la plage. Il fait normalement chaud en été, mais il peut y avoir des exceptions, et s’il pleut elle ne veut absolument pas aller à la plage. Ces informations peuvent être représentées par le LPOD suivant :*

1. $\text{cinema} \times \text{plage} \leftarrow \text{not } \text{chaud}$
2. $\text{plage} \times \text{cinema} \leftarrow \text{chaud}$
3. $\text{chaud} \leftarrow \text{ete}, \text{not } \neg \text{chaud}$
4. $\neg \text{plage} \leftarrow \text{pluie}$

Quelques définitions supplémentaires sont nécessaires pour définir les ensembles de réponses (answer sets) de ces programmes logiques particuliers.

Définition 4.7 ([Brewka, 2002]). *Soit $r = c_1 \times \dots \times c_n \leftarrow \text{corps}$ une règle. Pour $k \leq n$, on définit la k^{eme} option de r par*

$$r^k = c_k \leftarrow \text{corps}, \text{not } c_1, \dots, \text{not } c_{k-1}$$

Définition 4.8. *Soit P un LPOD. P' est une partie de programme de P (split program) s’il est obtenu en remplaçant chaque règle de P par une de ses options.*

Exemple 4.3 ([Brewka, 2002]). *Soit P un LPOD formé par les règles*

1. $a \times b \leftarrow \text{not } c$
2. $b \times c \leftarrow \text{not } d$

On obtient 4 parties de programmes :

$$\begin{array}{ll} P_1 & a \leftarrow \text{not } c \\ & b \leftarrow \text{not } d \\ P_2 & a \leftarrow \text{not } c \\ & c \leftarrow \text{not } d, \text{not } b \\ P_3 & b \leftarrow \text{not } c, \text{not } a \\ & b \leftarrow \text{not } d \\ P_4 & b \leftarrow \text{not } c, \text{not } a \\ & c \leftarrow \text{not } d, \text{not } b \end{array}$$

Définition 4.9. Soit P un LPOD. Un ensemble de littéraux R est un **ensemble de réponses** de P si R est un ensemble de réponses consistant d'une partie de programme P' de P .

Exemple 4.3 (page ci-contre) – suite Les ensembles de réponses pour chacune des parties de P sont : $P_1 : \{a, b\}$, $P_2 : \{c\}$, $P_3 : \{b\}$, et $P_4 : \emptyset$.

Le LPOD de cet exemple a donc 4 ensembles de réponses : $\{a, b\}$, $\{b\}$, $\{c\}$ et \emptyset .

Tous ces ensembles de réponses ne nous donnent pas la même satisfaction. $\{a, b\}$ donne la meilleure réponse pour chacune des deux règles, ce qui n'est pas le cas pour les deux autres règles. Pour faire la distinction entre les ensembles de réponses plus ou moins satisfaisants, [Brewka, 2002] introduit un degré de satisfaction d'une règle dans un ensemble de réponses, qui permet ensuite de trouver les ensembles de réponses préférés.

Définition 4.10 ([Brewka, 2002]). Soit R un ensemble de réponses d'un LPOD P . R satisfait la règle

$$c_1 \times \dots \times c_n \leftarrow a_1, \dots, a_m, \text{not } b_1, \dots, \text{not } b_k$$

- * avec le degré 1 s'il existe i tel que $a_i \notin R$, ou s'il existe l tel que $b_l \in R$;
- * avec le degré j , ($1 \leq j \leq n$), si pour tout i , $a_i \in R$, pour tout l , $b_l \notin R$, et $j = \min\{p | c_p \in R\}$.

Définition 4.11 ([Brewka, 2002]). Soit un ensemble de littéraux R . $R^i(P)$ représente l'ensemble de règles de P satisfaites par R au degré i .

Soit R_1 et R_2 deux ensembles de réponses d'un LPOD P . R_1 est **préférée à R_2** , noté $R_1 > R_2$, si et seulement si il existe un i tel que $R_2^i(P) \subset R_1^i(P)$ et que pour tout $j < i$, $R_2^j(P) = R_1^j(P)$.

Définition 4.12 ([Brewka, 2002]). Un ensemble de littéraux R est un **ensemble de réponses préféré** d'un LPOD P si et seulement si R est un ensemble de réponses de P et qu'il n'existe pas d'ensemble de réponses R' de P tel que $R' > R$.

Exemple 4.3 (page précédente) – suite Comme nous l'avons vu, ce LPOD a 3 ensembles de réponses : $R_1 = \{a, b\}$, $R_2 = \{b\}$ et $R_3 = \{c\}$.

R_1 satisfait les règles 1. et 2. avec un degré 1 ; R_2 satisfait 1. avec un degré 1, mais 2. avec un degré 2 ; et R_3 satisfait 1. avec un degré 2, et 2. avec un degré 1.

P a donc un seul ensemble de réponses préféré, R_1 .

Nous allons à présent présenter le lien, tel qu'il a été étudié par [Foo et al., 2004], entre LPOD et équilibre de Nash.

4.1.3.2 LPOD et équilibres de Nash

Dans [Foo et al., 2004], un jeu en forme normale est traduit par un LPOD dans lequel un ensemble de clauses représente les préférences de chaque joueur sur ses actions possibles, étant donnés tous les profils de stratégies des autres joueurs.

Exemple 4.4 ([Foo et al., 2004]). Si les stratégies du joueur 1 (resp. 2) sont dénotées A_1 , T_1 (resp. A_2 , T_2), le **dilemme du prisonnier** peut être représenté par l'ensemble de clauses suivant :

Clauses du joueur 1 :

$$A_1 \times T_1 \leftarrow T_2 \quad (4.7)$$

$$A_1 \times T_1 \leftarrow A_2 \quad (4.8)$$

L'équation 4.7 signifie que si le joueur 2 choisit de se taire (T_2), le joueur 1 préfère, s'il a le choix, avouer (A_1) que se taire (T_1). Ses préférences sont les mêmes si le joueur 2 avoue (cf. équation 4.8).

Clauses du joueur 2 :

$$A_2 \times T_2 \leftarrow T_1 \quad (4.9)$$

$$A_2 \times T_2 \leftarrow A_1 \quad (4.10)$$

Clauses de mouvement :

$$A_1 \vee T_1 \quad (4.11)$$

$$A_2 \vee T_2 \quad (4.12)$$

Les clauses de mouvement représentent les choix possibles de chacun des joueurs.

Il est alors démontré que les équilibres de Nash en stratégies pures correspondent exactement aux ensembles des réponses préférés.

Exemple 4.4 (page précédente) – suite *On cherche à présent les degré de satisfaction de chacun des ensembles de réponses de ce jeu pour chacun des joueurs. Le degré de satisfaction de l'ensemble de réponses (A_1, A_2) est¹ $(1, 1)$; le degré de satisfaction de (A_1, T_2) est $(1, 2)$; le degré de satisfaction de (T_1, A_2) est $(2, 1)$; et le degré de satisfaction de (T_1, T_2) est $(2, 2)$.*

Ce programme a donc un ensemble de réponses préférées, (A_1, A_2) , qui correspond bien à l'équilibre de Nash du dilemme du prisonnier.

Comme pour les programmes logiques de choix, cette traduction a une limitation : sa taille. Les LPODs ont la même taille que les formes normales du jeu (chaque joueur a besoin d'un nombre de clauses égal au nombre de profils de stratégies des autres joueurs). Cependant, cette limitation vient de la façon dont les LPODs sont construits à partir des jeux, et peut être corrigée en permettant aux joueurs d'exprimer leurs préférences par des LPODs (et des buts à priorité) quelconques, ce qui permettrait d'avoir une représentation beaucoup plus compacte.

Une différence importante avec notre cadre de travail, qui est valable aussi bien avec Foo et al [Foo *et al.*, 2004] qu'avec De Vos et Vermeir [De Vos et Vermeir, 1999], est que dans un jeu booléen les joueurs peuvent exprimer des préférences binaires arbitraires, jusqu'au cas extrême où la satisfaction du but d'un joueur dépend uniquement de variables contrôlées par d'autres joueurs.

¹(joueur 1, joueur 2)

4.2 Jeux et représentation graphique

4.2.1 CP-nets et équilibres de Nash

Les CP-nets, que nous avons présentés dans le chapitre 3 (page 69), sont un autre langage de représentation compacte, et sont utilisés dans [Apt *et al.*, 2005], pour représenter des jeux : les CP-nets sont vus comme des jeux en forme normale et vice versa². Chaque joueur i est associé à une variable X_i du CP-net, dont le domaine est l'ensemble des actions possibles du joueur. Les préférences sur les actions d'un joueur étant données les stratégies des autres joueurs sont exprimées dans une table de préférences conditionnelles.

Afin de pouvoir exprimer les préférences d'un joueur dans des tables de préférences conditionnelles, les relations de préférences dans un jeu stratégiques sont paramétrées par les stratégies des autres joueurs.

Définition 4.13 ([Apt *et al.*, 2005]). *Un jeu stratégique avec préférences paramétrées est un tuple $G = (N, S_1, \dots, S_n, \succ(s_{-1}), \dots, \succ(s_{-n}))$, où chaque $s_{-i} \in S_{-i}$.*

Dans ce contexte, un profil de stratégies s est un équilibre de Nash d'un jeu stratégique avec préférences paramétrées G si et seulement si pour tout $i \in \{1, \dots, n\}$, pour tout $s'_i \in S_i$, $s_i \succeq(s_{-i})s'_i$.

Exemple 4.5 ([Apt *et al.*, 2005]). *Si les stratégies du joueur 1 (resp. 2) sont dénotées H_1, R_1 (resp. H_2, R_2), la bataille des sexes (que nous avons présentée dans l'exemple 1.2 (page 11)) peut être représentée par les préférences paramétrées suivantes :*

$$\begin{aligned} \succ(H_2) &: H_1 \succ R_1 \\ \succ(R_2) &: R_1 \succ H_1 \\ \succ(H_1) &: H_2 \succ R_2 \\ \succ(R_1) &: R_2 \succ H_2 \end{aligned}$$

Il est montré dans [Apt *et al.*, 2005] que tous les CP-nets peuvent être traduits par un jeu stratégique ; et que tout jeu stratégique peut être traduit par un CP-net. Nous allons présenter ici la traduction d'un jeu vers un CP-net, la traduction réciproque étant quasiment symétrique.

Soit un jeu stratégique avec préférences paramétrées $G = (N, S_1, \dots, S_n, \succ(s_{-1}), \dots, \succ(s_{-n}))$. Ce jeu peut être associé à un CP-net $\mathcal{N}(G)$ de la façon suivante :

- * chaque joueur i correspond à une variable X_i ,
- * les éléments du domaine $D(X_i)$ de la variable X_i consistent en les stratégies du joueur i .
- * $Pa(X_i) = \{X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n\}$ l'ensemble des parents de la variable X_i .
- * pour chaque s_{-i} , la préférence conditionnelle $X_{-i} = s_{-i} : \succ(s_{-i})$ est ajoutée, où $\succ(s_{-i})$ est la relation de préférence sur l'ensemble des stratégies du joueur i si ses opposants jouent s_{-i} .

Exemple 4.5 – suite *La bataille des sexes avec préférences paramétrées peut être ainsi représentée par un CP-net à deux variables, X_1 et X_2 , correspondant aux joueurs 1 et 2, et telles que $D(X_1) = \{H_1, R_1\}$ et $D(X_2) = \{H_2, R_2\}$. Ce CP-net est représenté sur la figure 4.1 (page suivante).*

²Il ne s'agit donc pas ici d'utiliser les CP-nets pour exprimer les préférences de chaque joueur comme nous l'avons fait dans la section 3.2.2 (page 75) du chapitre 3.

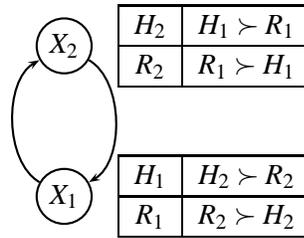


Figure 4.1 — CP-net de la bataille des sexes

Ce CP-net a deux résultats optimaux : $\{H_1, H_2\}$ et $\{R_1, R_2\}$, qui sont bien les équilibres de Nash du jeu.

Il est montré dans [Apt *et al.*, 2005] qu'un profil de stratégies est un équilibre de Nash du jeu G si et seulement si c'est un résultat optimal du CP-net $\mathcal{N}(G)$. De même, dans la traduction inverse, il est montré qu'un profil de stratégies est un résultat optimal d'un CP-net si et seulement si c'est un équilibre de Nash du jeu associé.

Exprimer un jeu avec des CP-nets peut être parfois plus compact que la forme normale explicite du jeu, pourvu que les préférences de certains joueurs ne dépendent pas de tous les autres joueurs.

Une différence importante avec notre cadre de travail est que nous permettons aux joueurs de contrôler un ensemble arbitraire de variables : nous ne voyons donc pas les joueurs comme étant des variables. La seule façon d'exprimer dans un CP-net qu'un joueur contrôle plusieurs variables consisterait à introduire une nouvelle variable dont le domaine serait l'ensemble de toutes les combinaisons de valeurs pour ces variables. La taille du CP-net serait alors exponentielle en fonction du nombre de variables.

Une autre différence (moins technique et plus fondamentale) entre ces deux lignes de travail, qui explique en fait les différences avec [Apt *et al.*, 2005], mais aussi avec [Foo *et al.*, 2004] et [De Vos et Vermeir, 1999], est que nous ne *traduisons* pas des jeux en forme normale en autre chose, mais que nous *exprimons* des jeux en utilisant un langage logique.

4.2.2 Forme normale graphique

La plupart des travaux permettant de représenter des jeux de façon compacte utilisent la notion de dépendance entre les joueurs et/ou les actions. Certains d'entre eux, dont [Koller et Milch, 2003; Kearns *et al.*, 2001; La Mura, 2000], partagent le mode de représentation des utilités des joueurs suivant : l'utilité du joueur i est décrite par une table associant une valeur numérique à chaque combinaison de valeurs de chacun des ensembles de variables utiles à i . La représentation des jeux avec de telles tables d'utilité est appelée *forme normale graphique* (graphical normal form, GNF) dans [Gottlob *et al.*, 2005]. Les dépendances entre joueurs et variables entraînent naturellement une relation de dépendance entre joueurs, identique à notre notion de dépendance : i dépend de j si la table d'utilité de i fait référence à une variable contrôlée par le joueur j . Dans [Koller et Milch, 2003; Gottlob *et al.*, 2005] les nœuds d'un graphe représentent les agents du jeu, et il y a un arc de i à j si i dépend de j . Cette représentation est plus compacte que la forme normale du jeu si le graphe n'est

pas une clique³.

La notion de dépendance entre joueurs et variables dans les jeux graphiques est utilisée pour exactement la même raison que notre graphe de dépendance, à savoir fractionner un jeu en un ensemble de plus petits jeux en interaction, et pouvant être résolus plus ou moins indépendamment.

4.2.2.1 Diagrammes d'influence multi-agents

[Koller et Milch, 2003] ont développé une représentation graphique pour les jeux dynamiques à connaissance incomplète : *multi-agent influence diagrams* (MAIDs). Ces MAIDs permettent de représenter des jeux de façon compacte, et ainsi d'étudier le calcul des équilibres de Nash en stratégies mixtes.

Les MAIDs distinguent trois types de variables : les variables de chance (permettant de prendre en compte les connaissances incomplètes sur l'état de monde), les variables de décision et les variables d'utilité, permettant de représenter l'utilité de chaque joueur.

Mis à part le fait que les MAIDs permettent de représenter la dynamique et les connaissances incomplètes, la principale différence entre les MAIDs et les jeux booléens est la façon de représenter les préférences. Comme décrit précédemment, les préférences dans les MAIDs sont représentées par des *tables d'utilité* : l'utilité d'un joueur est la somme des fonctions d'utilité locales, dont la taille est exponentielle dans le nombre de variables (de chance ou de décision) dont elles dépendent. Les tables d'utilité ne peuvent pas exprimer en espace polynomial les informations contenues dans une formule propositionnelle, ce qui implique que même la simple représentation propositionnelle que nous avons présentée dans le chapitre 2 (page 31) peut être parfois exponentiellement plus compacte que la représentation par tables d'utilité⁴.

Par exemple, la représentation logique de la version à n joueurs du dilemme du prisonnier, que nous avons étudiée dans l'exemple 2.2 (page 34), est exponentiellement plus compacte que sa représentation utilisant un MAID. En effet, les fonctions d'utilité de chaque joueur dépendent de toutes les variables du jeu, comme nous le montrons dans l'exemple suivant :

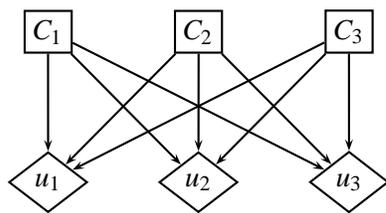
Exemple 4.6. Soit le jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$, avec

- * $N = \{1, 2, 3\}$,
- * $V = \{C_1, C_2, C_3\}$,
- * $\pi_1 = \{C_1\}$, $\pi_2 = \{C_2\}$, $\pi_3 = \{C_3\}$,
- * $\gamma_1 = \gamma_2 = \gamma_3 = \top$,
- * $\phi_1 = (C_1 \wedge C_2 \wedge C_3) \vee \neg C_1$,
- * $\phi_2 = (C_1 \wedge C_2 \wedge C_3) \vee \neg C_2$ et
- * $\phi_3 = (C_1 \wedge C_2 \wedge C_3) \vee \neg C_3$.

Ce jeu peut également être représenté par le MAID suivant, qui nécessite 3 variables de décision, aucune variable de chance et 3 tables d'utilités :

³Un graphe est une clique si tous les sommets sont deux-à-deux adjacents.

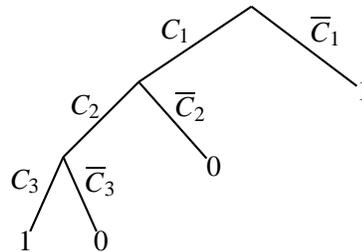
⁴Plus précisément, la représentation des fonctions d'utilité par la somme des utilités locales, chacune d'entre elles étant représentée par une table d'utilité, est polynomiale si et seulement si il existe une constante K limitant le nombre de variables apparaissant dans chaque table, c'est-à-dire si et seulement si les utilités des joueurs sont K -additives. Les fonctions d'utilité que l'on peut représenter par une formule propositionnelle sont K -additives si et seulement si la taille des formules utilisées dans la représentation sont limitées à K (voir par exemple [Chevalerey *et al.*, 2006]).



S	u_1	u_2	u_3	S	u_1	u_2	u_3
$C_1C_2C_3$	1	1	1	$\bar{C}_1C_2C_3$	1	0	0
$C_1C_2\bar{C}_3$	0	0	1	$\bar{C}_1C_2\bar{C}_3$	1	0	1
$C_1\bar{C}_2C_3$	0	1	0	$\bar{C}_1\bar{C}_2C_3$	1	1	0
$C_1\bar{C}_2\bar{C}_3$	0	1	1	$\bar{C}_1\bar{C}_2\bar{C}_3$	1	1	1

Quand le nombre n de joueurs varie, cette représentation grossit exponentiellement en n , alors qu'il reste polynomial pour un jeu booléen.

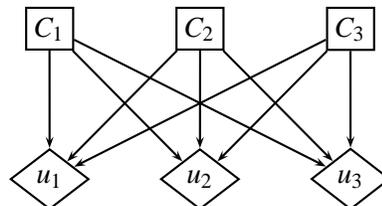
On peut remarquer que ces tables d'utilités peuvent être représentées en taille polynomiale par un arbre de décision, comme le montre l'arbre du joueur 1 :



Ce n'est cependant plus le cas dans l'exemple suivant :

Exemple 4.7. Soit $G = (N, V, \pi, \Gamma, \Phi)$, un jeu booléen avec $N = \{1, 2, \dots, n\}$, $V = \{C_1, \dots, C_n\}$, $\forall i \in N, \pi_i = \{C_i\}$, $\forall i \in N, \gamma_i = \top$, $\forall i \in N, \phi_i = C_i \leftrightarrow (\bigwedge_{j < i} \neg C_j \wedge \bigwedge_{k > i} C_k)$.

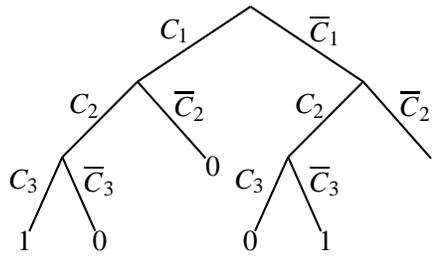
Ce jeu peut être représenté par le MAID suivant :



Les tables d'utilités sont les suivantes :

Profils de stratégies	u_1	u_2	u_3
$C_1C_2C_3$	1	0	0
$C_1C_2\bar{C}_3$	0	0	1
$C_1\bar{C}_2C_3$	0	1	0
$C_1\bar{C}_2\bar{C}_3$	0	0	1
$\bar{C}_1C_2C_3$	0	1	0
$\bar{C}_1C_2\bar{C}_3$	1	0	1
$\bar{C}_1\bar{C}_2C_3$	1	0	1
$\bar{C}_1\bar{C}_2\bar{C}_3$	1	1	0

Cette représentation est exponentielle en nombre d'agents, tout comme la représentation par un arbre de décision. Nous représentons ici celui du joueur 1 :



Cette représentation graphique des jeux est utilisée dans [Vickrey et Koller, 2002] pour trouver des algorithmes permettant de donner des équilibres approximatifs⁵ de ces jeux.

Les jeux à n joueurs et deux-actions⁶ sont représentés de la même façon dans [Kearns *et al.*, 2001], c'est-à-dire par un graphe et un ensemble de matrices permettant de représenter les utilités des joueurs. Un algorithme permet alors de calculer les équilibres de Nash (mixtes) à partir de ces jeux. Une faille de cet algorithme a été trouvée, et corrigée, dans [Elkind *et al.*, 2006].

4.2.2.2 Formes normales graphiques et restrictions

Les jeux sous forme normale graphique sont également étudiés dans [Gottlob *et al.*, 2005], qui ont utilisé la notion de voisins pour les définir : les voisins d'un joueur i dans un jeu graphique sont les joueurs pouvant influencer sur l'utilité de i .

Définition 4.14 ([Gottlob *et al.*, 2005]). *L'ensemble des voisins d'un joueur i , noté $\mathcal{V}(i)$, est l'ensemble des joueurs qui ont une influence sur i : si $j \notin \mathcal{V}(i)$, alors la fonction d'utilité de i ne dépend pas directement des actions de j .*

L'ensemble des voisins des joueurs permet de définir le graphe de dépendance de la même façon que celui défini par [Koller et Milch, 2003]. [Gottlob *et al.*, 2005] introduit également un hypergraphe de dépendance d'un jeu de la façon suivante :

Définition 4.15 ([Gottlob *et al.*, 2005]). *Soit un jeu stratégique G , et N l'ensemble des joueurs de ce jeu.*

*Le **graphe de dépendance** de G est un graphe non orienté $\mathcal{G}(G)$ dont l'ensemble des nœuds est N , et l'ensemble des arcs est défini par :*

$$\{\{i, j\} | j \in N \wedge i \in \mathcal{V}(j)\}$$

*L'**hypergraphe de dépendance** de G est un graphe non orienté $H(G)$ dont l'ensemble des nœuds est N , et l'ensemble des hyperarcs est défini par :*

$$\{\{i\} \cup \mathcal{V}(i) | i \in N\}$$

⁵Un équilibre est dit approximatif si la possibilité que chaque agent a de dévier est plus petit qu'un ϵ donné.

⁶Une généralisation à des jeux avec p actions est donnée dans [Kearns *et al.*, 2001].

Comme dans les MAIDs, la **forme normale graphique (GNF)** d'un jeu est définie en associant une table d'utilités à chaque joueur.

L'exemple que nous allons présenter maintenant est une généralisation de la bataille des sexes, que nous avons présentée dans l'exemple 1.2 (page 11), avec 4 joueurs.

Exemple 4.8 ([Gottlob *et al.*, 2005]). *4 amis, Alexia, Sylvain, Nicolas et Filou, doivent décider comment ils souhaitent passer leur soirée. Ils ont le choix entre regarder une comédie romantique ou un film d'horreur. Alexia a envie de passer la soirée avec Sylvain et Nicolas, peu importe le film. Sylvain a envie de voir un film d'horreur avec Alexia, mais sans Nicolas. Ce dernier veut voir une comédie romantique, avec Alexia mais sans Sylvain. Quant à Filou, il préférerait voir une comédie romantique avec Nicolas.*

Soit G un jeu stratégique tel que $N = \{1, 2, 3, 4\}$, Alexia étant le joueur 1, Sylvain le joueur 2, Nicolas le joueur 3 et Filou le joueur 4. $V = \{h_1, h_2, h_3, h_4\}$, h_1 (resp. h_2 , h_3 et h_4) signifiant que le joueur 1 (resp. 2, 3 et 4) va voir un film d'horreur, et \bar{h}_1 signifiant qu'il va voir une comédie romantique.

On a $\mathcal{V}(1) = \{2, 3\}$, $\mathcal{V}(2) = \{1, 3\}$, $\mathcal{V}(3) = \{1, 2\}$ et $\mathcal{V}(4) = \{3\}$. Le graphe et l'hypergraphe de dépendance de ce jeu sont représentés sur la figure 4.2.

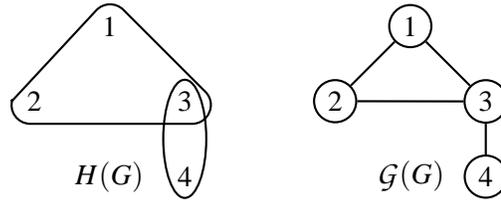


Figure 4.2 — Hypergraphe et graphe de dépendance du jeu G

Les fonctions d'utilité⁷ de ce jeu sont représentées sur la figure 4.3.

1	h_2h_3	$h_2\bar{h}_3$	\bar{h}_2h_3	$\bar{h}_2\bar{h}_3$
h_1	2	1	1	0
\bar{h}_1	0	1	1	2

2	h_1h_3	$h_1\bar{h}_3$	\bar{h}_1h_3	$\bar{h}_1\bar{h}_3$
h_2	2	3	1	2
\bar{h}_2	1	2	0	1

3	h_1h_2	$h_1\bar{h}_2$	\bar{h}_1h_2	$\bar{h}_1\bar{h}_2$
h_3	1	2	0	1
\bar{h}_3	2	3	1	2

4	h_3	\bar{h}_3
h_4	1	0
\bar{h}_4	1	2

Figure 4.3 — Fonctions d'utilité pour la forme normale graphique de G

[Gottlob *et al.*, 2005] ont étudié des restrictions spécifiques sur ces jeux graphiques, en limitant le nombre de voisins de chacun des joueurs, ou en imposant à la relation de dépendance d'être acyclique. Ils étudient l'impact de telles restrictions sur la complexité du problème consistant à vérifier l'existence d'un équilibre de Nash, et/ou sur le calcul de ces équilibres. Ils ont ainsi montré que le problème de l'existence d'un équilibre de Nash en stratégies pures dans un GNF qui a un voisinage limité (c'est-à-dire que pour tous les joueurs i , $|\mathcal{V}(i)| \leq k$), est NP-complet. La complexité est la même pour les jeux dont le graphe ou l'hypergraphe de dépendance est acyclique.

⁷Par exemple, $u_2(h_1h_2h_3) = 2$, car Sylvain est content de voir un film d'horreur (+1), et de le voir avec Alexia (+1). $u_2(h_1h_2\bar{h}_3) = 3$, car Sylvain est content de voir un film d'horreur (+1), de le voir avec Alexia (+1), et que Nicolas aille voir un autre film (+1).

La notion de voisin introduite dans [Gottlob *et al.*, 2005] est similaire à notre notion de joueur utile. La principale différence entre le graphe de dépendance de [Gottlob *et al.*, 2005] et le nôtre est que ce dernier est orienté, ce qui n'est pas le cas dans les GNFs. De plus, comme pour les MAIDs, la représentation des jeux par tables d'utilité est parfois beaucoup moins compacte que celle utilisant un jeu booléen.

4.2.2.3 G nets

Les G nets, introduits par [La Mura, 2000], sont une autre représentation compacte des jeux qui prend en compte la notion de causalité. Un G net est représenté par un graphe, dont les nœuds correspondent chacun aux variables du jeu, chaque nœud étant contrôlé par un et un seul agent (ou la nature⁸). Deux quantités sont associées à chaque nœud : sa probabilité, et l'utilité potentielle de ce nœud en fonction de ses voisins dans le graphe. Les arcs sont de deux types : des arcs orientés représentent les dépendances causales (c'est-à-dire les dépendances issues des probabilités), et les dépendances préférentielles sont représentés par des arcs non-orientés (c'est-à-dire les influences des utilités d'un nœud sur l'autre).

Comme les MAIDs, cette représentation est compacte si les utilités sont indépendantes, mais ne l'est plus si ce n'est pas le cas. Dans ce cas, la représentation par un jeu booléen peut être exponentiellement plus compacte que celle par un G net.

4.2.3 Symétries dans les fonctions d'utilité

Une autre approche pour représenter des jeux de façon compacte est de s'intéresser à l'indépendance des fonctions d'utilité, c'est-à-dire aux possibilités qu'un agent a d'affecter les autres agents, selon les actions dont il dispose, et donc souvent aussi selon les symétries dans les fonctions d'utilité des agents [Rosenthal, 1973; Monderer et Shapley, 1996; Kearns et Mansour, 2002; Roughgarden et Tardos, 2001]. Cette approche est très différente de celle utilisée dans les jeux booléens, nous la présentons néanmoins rapidement ici.

4.2.3.1 Jeux de congestion

Les jeux de congestion (*congestion games*) [Rosenthal, 1973], sont des jeux dans lesquels les actions disponibles consistent en un ensemble de ressources, et l'utilité de chaque joueur dépend du nombre de joueurs ayant sélectionné les mêmes ressources (c'est-à-dire ayant joué la même action).

Définition 4.16. *Un jeu de congestion est constitué d'un ensemble N de joueurs, d'un ensemble fini R de ressources, d'un ensemble d'actions S_i pour chaque joueur i tel que $S_i \subseteq 2^R$, et une fonction de coût $d : R \times N \rightarrow \mathbf{N}$ telle que $d_r(j)$ n'est pas décroissant en j .*

Soit $s = (s_1, \dots, s_n)$ un profil de stratégies, $r \in R$ une ressource, et $n_s(r)$ le nombre de joueurs ayant choisi la ressource r : $n_s(r) = |\{i | r \in s_i\}|$. On a alors

$$c_i(s) = -u_i(s) = \sum_{r \in s_i} d_r(n_s(r))$$

⁸Les choix pris par la nature dénotent ceux qui relèvent du hasard.

Intuitivement, chaque joueur choisit un ensemble de ressources (à partir des ensembles qui lui sont disponibles). Pour calculer les coûts que ces ressources demandent au joueur i (c'est-à-dire l'inverse de son utilité), on ajoute le coût de chaque ressource utilisée par i , en sachant que le coût d'une ressource r dépend du nombre de joueurs utilisant cette ressource ($n_s(r)$).

Définition 4.17. Dans un réseau d'un jeu de congestion, les familles d'ensembles S_i sont implicitement présentées comme des chemins dans un graphe. Soit un graphe orienté (V, R) , deux nœuds $d_i, a_i \in V$ pour chaque joueur i . Les arcs du graphe représentent les ressources du jeu, et une fonction de coût est associée à ces arcs. Le sous ensemble de R représentant les actions disponibles pour i est l'ensemble de tous les chemins allant de d_i à a_i .

Exemple 4.9 ([Fabrikant et al., 2004]). Soit le jeu de congestion suivant : trois joueurs, $\{1, 2, 3\}$ doivent se déplacer du point S au point T , représentés sur la figure 4.4.

Les arcs de ce graphe représentent les chemins possibles pour chacun des joueurs. Ces arcs sont étiquetés par la fonction coût qui dépend du nombre de joueurs empruntant cet arc : le premier chiffre correspond au coût de cet arc si un seul joueur l'emprunte, le second si deux joueurs l'empruntent, et le troisième si les trois joueurs y passent.

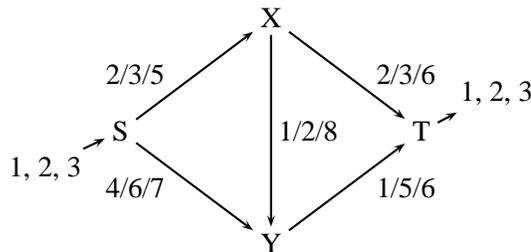


Figure 4.4 — Le réseau d'un jeu de congestion : trois joueurs doivent aller de S à T . Chaque arc est étiqueté avec les coûts pour chaque joueur suivant si le chemin est parcouru par 1, 2 ou 3 joueurs

Par exemple, si le joueur 1 emprunte le chemin $SXYT$, le joueur 2 le chemin SXT et le joueur 3 le chemin SYT , on aura $c_1(s) = 9$, $c_2(s) = 5$ et $c_3(s) = 9$.

Rosenthal a prouvé dans [Rosenthal, 1973] que tous les jeux de congestion ont un équilibre de Nash en stratégies pures. D'autre part, il est montré dans [Fabrikant et al., 2004] qu'il existe un algorithme polynomial pour trouver un équilibre de Nash en stratégies pures dans le cas où le réseau du jeu de congestion est symétrique.

L'utilité d'un joueur dans les jeux de congestion est liée au nombre de joueurs utilisant les mêmes ressources, et donc effectuant les mêmes actions que lui. La principale différence entre jeux booléens et jeux de congestion est que dans les jeux booléens, deux joueurs, qui contrôlent par définition deux ensembles de variables propositionnelles disjoints, ne peuvent donc pas effectuer la même action. La notion de coût d'une ressource ne peut donc être introduite dans un jeu booléen.

4.2.3.2 Jeux à effets locaux

Les jeux à effets locaux (*local-effect games*, LEGs) [Leyton-Brown et Tennenholtz, 2003] permettent de modéliser des situations dans lesquelles les effets des agents les uns sur les autres peuvent être asymétriques.

Définition 4.18 ([Leyton-Brown et Tennenholtz, 2003]). Soit $G = \langle \mathcal{A}, \mathcal{F}, n \rangle$ un **jeu à effets locaux** à n joueurs. \mathcal{A} est l'ensemble des actions disponibles pour les joueurs. Soit D la distribution des joueurs sur les actions. $D(a)$ représente le nombre de joueurs ayant choisi l'action $a \in \mathcal{A}$.

Pour chaque couple d'actions $a, a' \in \mathcal{A}$, $\mathcal{F}_{a,a'} : \mathbb{Z}^+ \rightarrow \mathbb{R}^+$ est la fonction de coût, strictement monotone, qui donne le coût des effets locaux de l'action a à l'action a' qui dépend du nombre de joueurs qui choisissent l'action a . À partir de ces définitions, il est possible de construire la fonction c d'un agent i qui choisit une action $a \in \mathcal{A}$:

$$c(i) = \mathcal{F}_{a,a}(D(a)) + \sum_{b \in \mathcal{A}, b \neq a} \mathcal{F}_{b,a}(D(b))$$

Un LEG peut être représenté par un graphe dont les nœuds représentent les actions des joueurs, et les arcs représentent les dépendances d'utilité spécifiques au contexte entre les actions : il existe un nœud pour chaque action, et un arc du nœud a au nœud b si $\forall x, \mathcal{F}_{a,b}(x) \neq 0$. Les fonctions de la forme $\mathcal{F}_{a,a}(x)$ seront appelées *fonctions de nœuds*, tandis que les fonctions de la forme $\mathcal{F}_{a,b}(x)$ seront appelées *fonctions d'arcs*.

Soit $\mathcal{V}(a)$ l'ensemble des nœuds tel qu'il y ait un arc partant de a et arrivant sur ce nœud. L'hypothèse suivante est faite :

$$\forall a, b \in \mathcal{A}, b \in \mathcal{V}(a) \rightarrow a \in \mathcal{V}(b)$$

Ces préconditions nécessaires aux LEGs (tous les agents contrôlent le même ensemble d'actions ; l'utilité de chacun se décompose en la somme des effets locaux pour les actions individuelles ; et l'indépendance des utilités est toujours bidirectionnelle) montrent que les LEGs ne permettent pas de représenter tous les jeux.

Si tous les jeux à effets locaux n'ont pas d'équilibres de Nash en stratégies pures, de nombreux cas particuliers, possédant des PNEs car satisfaisant certaines caractéristiques particulières, ont été identifiés dans [Leyton-Brown et Tennenholtz, 2003].

Comme pour les jeux de congestion, la fonction de coût existant dans un LEG, qui dépend du nombre de joueurs ayant choisi une action, ne peut être introduite dans un jeu booléen. D'autre part, l'indépendance des fonctions d'utilité peut ne pas être bidirectionnelle dans un jeu booléen.

4.2.3.3 Jeux graphiques d'actions

Les jeux graphiques d'action (*action graph games*, AGGs), introduits dans [Bhat et Leyton-Brown, 2004], peuvent être vus comme des LEGs non restreints. Ils peuvent représenter tous les jeux, notamment des jeux graphiques comme ceux de [Koller et Milch, 2003]. Comme les LEGs, un AGG est un graphe tel qu'il y a un nœud pour chaque action du jeu, et un arc entre deux nœuds s et s' si et seulement si s est dépendant de s' dans ce jeu, c'est-à-dire si les utilités des joueurs choisissant s sont dépendantes des joueurs choisissant s' .

Définition 4.19 ([Bhat et Leyton-Brown, 2004]). Un **jeu graphique d'actions** est un tuple $\langle N, S, \mathcal{V}, u \rangle$, où $N = \{1, \dots, n\}$ est l'ensemble des joueurs, S est l'ensemble des profils de stratégies, S_i étant les actions disponibles pour un joueur $i \in N$. Les agents peuvent avoir des actions communes, on note donc $\mathcal{A} = \bigcup_{i \in N} S_i$ l'ensemble des actions du jeu.

Δ représente l'ensemble des distributions possibles des agents sur les actions, où une distribution consiste en le nombre d'agents choisissant une action donnée. Pour une distribution donnée $D \in \Delta$,

$D(a)$ est le nombre d'agents ayant choisi l'action a . $D: S \rightarrow \mathbb{N}^{|\mathcal{A}|}$ est une fonction qui associe à chaque profil de stratégies s une distribution d'agents D .

Soit \mathcal{G} le graphe d'actions. \mathcal{G} contient un nœud pour chaque action $a \in \mathcal{A}$. La relation de voisinage est donnée par $\mathcal{V}: \mathcal{A} \rightarrow 2^{\mathcal{A}}$. Il y a alors un arc de a à a' dans \mathcal{G} si et seulement si $a' \in \mathcal{V}(a)$. Il est possible d'avoir $a \in \mathcal{V}(a)$.

La fonction d'utilité est la même pour tous les joueurs. Elle associe une valeur numérique à une action a et une distribution d'agents D : $u: \mathcal{A} \times \Delta \rightarrow \mathbb{R}$.

Le problème de l'existence d'un équilibre de Nash en stratégies pures a été étudié dans [Jiang et Leyton-Brown, 2007]. Il a été ainsi montré que dans le cas général, le problème consistant à décider s'il existe un équilibre de Nash en stratégies pures dans un AGG est NP-complet. Pourtant, il existe des classes de ce jeu pour lesquelles ce problème est polynomial, par exemple si les AGGs ayant un nombre d'actions, et donc de nœuds, limité à k .

Comme dans les jeux de congestion et les LEGs, l'utilité d'un joueur dans un AGG dépend du nombre de joueurs ayant choisi chaque action, et donc ne peut être introduite dans un jeu booléen.

5

Coalitions efficaces dans les jeux booléens

Nous nous situons ici de nouveau dans le cadre de préférences dichotomiques. Comme nous l'avons vu dans le chapitre 2 (page 31), les jeux booléens permettent de représenter des jeux stratégiques de manière compacte, en utilisant le pouvoir d'expression et la concision de la logique propositionnelle. Les joueurs d'un jeu booléen ayant des préférences dichotomiques, la notion suivante émerge naturellement : une coalition dans un jeu booléen est efficace si elle peut garantir à tous ses membres que leurs buts sont satisfaits. Notre objectif dans ce chapitre est de définir et de caractériser ces coalitions efficaces, et de voir comment elles sont liées à la notion de noyau dans un jeu coalitionnel (voir la définition 1.11 (page 26)).

Après avoir présenté les fonctions d'effectivité en section 5.1, nous allons étudier les propriétés de ces fonctions d'effectivité associées aux jeux booléens en section 5.2. Dans la section 5.3, nous étudierons en détail la notion de coalition efficace : nous donnerons une caractérisation exacte des ensembles de coalitions correspondant à des ensembles de coalitions efficaces associées à un jeu booléen, et nous ferons le lien entre coalition efficace et noyau¹.

5.1 Fonctions d'effectivité

Les fonctions d'effectivité ont été développées en choix social afin de modéliser le pouvoir des coalitions [Moulin, 1983; Abdou et Keiding, 1991]. On rappelle qu'une **coalition** C est un sous-ensemble quelconque de l'ensemble des agents N . N est appelée la **grande coalition**. Etant donné un ensemble d'alternatives S parmi lesquelles un ensemble d'individus N doit choisir, une *fonction d'effectivité* $\text{Eff}: 2^N \rightarrow 2^{2^S}$ associe un sous-ensemble de S à chaque coalition. $X \in \text{Eff}(C)$ est interprété par "la coalition C est effective² pour X ".

Définition 5.1. Une **fonction d'effectivité coalitionnelle** est une fonction $\text{Eff}: 2^N \rightarrow 2^{2^S}$ monotone : pour toute coalition $C \subseteq N$, $X \in \text{Eff}(C)$ implique $Y \in \text{Eff}(C)$ quand $X \subseteq Y \subseteq S$.

La fonction Eff associe à chaque groupe de joueurs l'ensemble des issues³ du jeu que ce groupe a

¹Les résultats donnés dans ce chapitre ont fait l'objet de plusieurs publications : [Bonzon *et al.*, 2008, 2007b]

²La définition "coalition efficace" pourrait mieux convenir à la traduction de "effective coalition", mais nous utiliserons ce terme plus tard (voir définition 5.4 (page 143)), et donc nous conservons ici le terme "effectif".

³Dans un jeu booléen, une issue du jeu correspond à un profil de stratégies. Mais, dans un cadre plus général, une issue du jeu peut correspondre à un plan d'actions, ou à l'utilité obtenue par chaque joueur.

le pouvoir d'atteindre. On interprète habituellement $X \in \text{Eff}(C)$ comme “les joueurs dans C ont une stratégie commune leur permettant d'obtenir une issue dans X ”.

Les fonctions d'effectivité peuvent satisfaire quelques propriétés additionnelles ([Pauly, 2001]) :

Coalition-monotone : Eff est *coalition-monotone* si pour tout $C \subseteq C' \subseteq N$, $\text{Eff}(C) \subseteq \text{Eff}(C')$.

Régularité : Eff est *C-régulière* si pour tout X , si $X \in \text{Eff}(C)$, alors $\bar{X} \notin \text{Eff}(\bar{C})$ ⁴. Eff est *régulière* si Eff est *C-régulière* pour toute coalition C .

Maximalité : Eff est *C-maximale* si pour tout X , si $\bar{X} \notin \text{Eff}(\bar{C})$, alors $X \in \text{Eff}(C)$. Eff est *maximale* si Eff est *C-maximale* pour toute coalition C .

Superadditivité : Eff est *superadditive* si pour tout $C, C' \subseteq N$ et $X, Y \subseteq S$, si $X \in \text{Eff}(C)$ et $Y \in \text{Eff}(C')$ alors $X \cap Y \in \text{Eff}(C \cup C')$.

Pauly (2001) a précisé le sens du mot “effectif” dans le cadre des jeux stratégiques en définissant l' α -effectivité : une coalition $C \subseteq N$ est dite **α -effective** pour $X \subseteq S$ si et seulement si les joueurs de cette coalition ont une stratégie commune leur permettant d'obtenir une issue dans X , *quelles que soient les stratégies des autres joueurs*.

Définition 5.2. Pour un jeu stratégique G non vide, une **fonction d' α -effectivité coalitionnelle** est une fonction $\text{Eff}_G^\alpha : 2^N \rightarrow 2^{2^S}$ définie comme suit : $X \in \text{Eff}_G^\alpha(C)$ si et seulement si $\exists s_C \forall s_{-C}, (s_C, s_{-C}) \in X$.

5.2 Coalitions et fonctions d'effectivité dans les jeux booléens

Nous voulons à présent définir les fonctions d'effectivité correspondant au cas particulier des jeux booléens. Une des particularités des jeux booléens est la définition des stratégies individuelles comme étant l'attribution de valeurs de vérité sur un ensemble donné de variables propositionnelles. On peut se demander à quel point cette spécificité est restrictive. Clairement, la définition de S_i comme étant $\text{Mod}_{\pi_i}(\gamma_i)$ entraîne quelques contraintes sur le pouvoir des joueurs. Notre but est de donner une caractérisation exacte des fonctions d'effectivité induites par les jeux booléens. Puisque dans un jeu booléen le pouvoir d'un agent i est indépendant de son but ϕ_i , il suffira de considérer les pré-jeux booléens⁵ pour étudier les fonctions d'effectivité.

À partir d'un pré-jeu booléen G , on obtient une fonction d' α -effectivité Eff_G^α , qui caractérise les coalitions composées de joueurs indépendants des joueurs n'appartenant pas à cette coalition pour obtenir une issue X quelconque.

Définition 5.3. Soit $G = (N, V, \pi, \Gamma)$ un pré-jeu booléen. La **fonction d' α -effectivité coalitionnelle induite par G** est la fonction $\text{Eff}_G^\alpha : 2^N \rightarrow 2^{2^S}$ définie par : pour tout $X \subseteq S$ et tout $C \subseteq N$, $X \in \text{Eff}_G^\alpha(C)$ s'il existe $s_C \in S_C$ tel que pour tout $s_{-C} \in S_{-C}$, $(s_C, s_{-C}) \in X$.

On peut remarquer que les fonctions d' α -effectivité induites par des pré-jeux booléens peuvent être exprimées de façon équivalente par des fonctions⁶ $\text{Eff}_G : 2^N \rightarrow 2^{L_V}$ associant à des coalitions des

⁴ \bar{X} représente $S \setminus X$, et \bar{C} représente $N \setminus C$.

⁵Rappelons qu'un pré-jeu booléen est un jeu booléen dans lequel on ne considère pas les buts des joueurs, soit le 4-uplet (N, V, π, Γ) (voir définition 2.2 (page 34)).

⁶Pour simplifier les notations, les fonctions d' α -effectivité induites par un pré-jeu booléen G seront notées Eff_G au lieu de Eff_G^α .

ensembles de formules logiques : $\varphi \in \text{Eff}_G(I)$ si $\text{Mod}_{\tau_I}(\varphi) \in \text{Eff}_G(I)$. Cette définition entraîne évidemment une indépendance de syntaxe, c'est-à-dire si $\varphi \equiv \psi$ alors $\varphi \in \text{Eff}_G(I)$ si et seulement si $\psi \in \text{Eff}_G(I)$.

Cette définition est un cas particulier de la fonction d' α -effectivité induite par un jeu stratégique (voir [Pauly, 2001], Chapitre 2), puisqu'un pré-jeu booléen est une forme stratégique spécifique. Par conséquent, cette fonction satisfait les propriétés suivantes (cf. [Pauly, 2001], Théorème 2.27) :

1. $\forall C \subseteq N, \emptyset \notin \text{Eff}_G(C)$;
2. $\forall C \subseteq N, S \in \text{Eff}_G(C)$;
3. pour tout $X \subseteq S$, si $\bar{X} \notin \text{Eff}_G(\emptyset)$ alors $X \in \text{Eff}_G(N)$;
4. Eff_G est superadditive.

Une fonction d' α -effectivité qui satisfait ces quatre propriétés est dite **fortement jouable**. On peut noter que la propriété "fortement jouable" implique les propriétés de régularité et de coalition-monotonie ([Pauly, 2001], Lemme 2.26).

Pourtant, nous aimerions avoir une caractérisation encore plus précise de ces fonctions d' α -effectivité correspondant à un pré-jeu booléen. Pour cela, nous devons tout d'abord définir deux propriétés de plus, en utilisant la notation suivante : $At(C)$ dénotera les **ensembles minimaux de** $\text{Eff}(C)$, c'est-à-dire que $At(C) = \{X \in \text{Eff}(C) \mid \text{il n'existe pas de } Y \in \text{Eff}(C) \text{ tel que } Y \subseteq X\}$.

Atomicité : Eff satisfait l'*atomicité* si pour tout $C \subseteq N$, $At(C)$ forme une partition de S .

Décomposabilité : Eff satisfait la *décomposabilité* si pour tout $I, J \subseteq N$ et pour tout $X \subseteq S$, $X \in \text{Eff}(I \cup J)$ si et seulement s'il existe $Y \in \text{Eff}(I)$ et $Z \in \text{Eff}(J)$ tels que $X = Y \cap Z$.

On peut noter que la décomposabilité est une propriété forte qui entraîne la superadditivité.

Propriété 5.1. *Une fonction d' α -effectivité coalitionnelle Eff satisfait*

1. *la jouabilité forte,*
2. *l'atomicité,*
3. *la décomposabilité, et*
4. $\text{Eff}(N) = 2^S \setminus \emptyset$.

si et seulement s'il existe un pré-jeu booléen $G = (N, V, \pi, \Gamma)$ et une fonction injective $\mu : S \rightarrow 2^V$ tels que pour tout $C \subseteq N$: $\text{Eff}_G(C) = \{\mu(X) \mid X \in \text{Eff}(C)\}$.

La preuve de cette propriété est assez longue. Elle repose sur les lemmes 5.1 à 5.5 (page 142).

Si G est un pré-jeu booléen, l'ensemble des ensembles minimaux (appelés atomes) pour la fonction d'effectivité Eff_G sera noté par At_G .

Commençons par montrer la direction (\Leftarrow) de la propriété 5.1.

Lemme 5.1. *Pour tout pré-jeu booléen G , Eff_G est fortement jouable, et satisfait l'atomicité, la décomposabilité et $\text{Eff}_G(N) = 2^S \setminus \emptyset$.*

Preuve du lemme : Eff_G est une fonction (spécifique) d' α -effectivité. Le Theorème 2.27 dans [Pauly, 2001] prouve donc que Eff_G est fortement jouable (et, a fortiori, superadditive).

Pour l'atomicité, remarquons tout d'abord que $X \in \text{At}_G(C)$ si et seulement si X est l'ensemble de toutes les π_C -interprétations satisfaisant $\gamma_C = \bigwedge_i \gamma_i$, ce qui implique clairement que tous les ensembles distincts de $\text{At}_G(C)$ sont disjoints. Remarquons ensuite que $\bigcup_{s_C \in S_C} \{s \mid s \supseteq s_C\} = S$. Donc, $\text{At}_G(C)$ forme une partition de S .

Pour la décomposabilité, de gauche à droite : Soit $X \in \text{Eff}_G(I \cup J)$. Alors il existe une stratégie commune $s_{I \cup J}$ telle que si $W = \{s \in S \mid s \supseteq s_{I \cup J}\}$, alors $W \subseteq X$. Considérons à présent $Y = \{s \in S \mid s \supseteq s_I\}$ et $Z = \{s \in S \mid s \supseteq s_J\}$. Nous avons $Y \in \text{Eff}_G(I)$, $Z \in \text{Eff}_G(J)$ et $X = Y \cap Z$. De droite à gauche : soit $Y \in \text{Eff}_G(I)$ et $Z \in \text{Eff}_G(J)$, alors grâce à la superadditivité, $Y \cap Z \in \text{Eff}_G(I \cup J)$.

Enfin, on peut facilement vérifier que $\text{Eff}_G(N) = 2^S \setminus \emptyset$.

■

Lemme 5.2. *S'il existe un pré-jeu booléen $G = (N, V, \pi, \Gamma)$ et une fonction injective $\mu : S \rightarrow 2^V$ telle que pour tout $C \subseteq N : \text{Eff}_G(C) = \{\mu(X) \mid X \in \text{Eff}(C)\}$, alors Eff est fortement jouable, et satisfait l'atomicité, la décomposabilité et $\text{Eff}(N) = 2^S \setminus \emptyset$.*

Preuve du lemme : Comme Eff_G satisfait ces propriétés, et que μ est une bijection entre S et $\mu(S)$, ces propriétés sont transférées à Eff .

■

Prouvons à présent la direction (\Rightarrow) de la propriété 5.1 (page précédente).

Lemme 5.3. *Soit G un pré-jeu booléen et T_i un sous-ensemble minimal de $\text{Eff}_G(i)$. Alors, $T_i = \{s \mid s \supseteq s_i\}$ pour tout $s_i \in S_i$.*

Preuve du lemme : $X \in \text{Eff}_G(i)$ si X contient $S_1 \times \dots \times S_{i-1} \times S_i^* \times S_{i+1} \times \dots \times S_n$ pour certains $S_i^* \subseteq S_i$. Donc, les sous-ensembles minimaux de $\text{Eff}_G(i)$ sont exactement ceux de la forme $S_1 \times \dots \times S_{i-1} \times \{s_i\} \times S_{i+1} \times \dots \times S_n$, c'est-à-dire de la forme $\{s \mid s \supseteq s_i\}$.

■

À partir de maintenant, soit Eff une fonction d'effectivité coalitionnelle, étant fortement jouable, et satisfaisant l'atomicité, la décomposabilité et $\text{Eff}(N) = 2^S \setminus \emptyset$.

Remarquons tout d'abord que la décomposabilité implique que Eff est entièrement déterminée par $\{\text{At}(i), i \in N\}$.

Lemme 5.4. *Pour chaque $s \in S$ il existe un unique (Z_1, \dots, Z_n) tel que pour tout i , $Z_i \in \text{At}(i)$ et $Z_1 \cap \dots \cap Z_n = \{s\}$.*

Preuve du lemme : Soit $s \in S$. Comme $\text{Eff}(N) = 2^S \setminus \{\emptyset\}$, on a $\{s\} \in \text{Eff}(N)$, et par décomposabilité, il existe (T_1, \dots, T_n) tel que pour tout i , $T_i \in \text{Eff}(i)$ et $T_1 \cap \dots \cap T_n =$

$\{s\}$.

Soit $i \in N$. Par définition de $At(i)$, il existe $Z_i \in At(i)$ tel que $s \in Z_i$ et $Z_i \subseteq T_i$. Supposons qu'il existe $Z'_i \in At(i)$ tel que $s \in Z'_i$ et $Z'_i \subseteq T_i$. $Z_i \cap Z'_i \neq \emptyset$, puisque s appartient à Z_i et à Z'_i . Donc, par atomicité, $Z_i = Z'_i$, et cela vaut pour tout i .

■

Le lemme 5.4 nous permet d'écrire que, pour chaque s et i , $Z_i(s)$ est le seul sous-ensemble de $At(i)$ contenant s . Pour toute coalition non vide C , nous écrivons $Z_C(s) = \bigcap_{i \in C} Z_i(s)$.

Ensuite, nous construisons G^* à partir de Eff comme suit :

- * Pour chaque i , on numérote $At(i)$: soit r_i une fonction bijective de $At(i)$ dans $\{0, 1, \dots, |At(i)| - 1\}$. Ensuite, on crée $p_i = \lceil \log_2 |At(i)| \rceil$ variables propositionnelles $x_i^1, \dots, x_i^{p_i}$. Enfin, on pose $V = \{x_i^j \mid i \in N, 1 \leq j \leq p_i\}$;
- * Pour chaque i : $\pi_i = \{x_i^1, \dots, x_i^{p_i}\}$;
- * Pour chaque i et chaque $j \leq p_i$, soit $\varepsilon_{i,j}$ le j ème chiffre dans la représentation binaire de i . On note que $\varepsilon_{i,p_i} = 1$ par définition de p_i . Si x est une variable propositionnelle, nous utiliserons alors la notation suivante : $0.x = \neg x$ et $1.x = x$. On définit ensuite

$$\gamma_i = \bigwedge_{j \in \{2, \dots, p_i\}, \varepsilon_{i,j} = 0} \left(\bigwedge_{1 \leq k \leq j-1} \varepsilon_{i,j}.x_i^k \rightarrow \neg x_i^j \right)$$

- * Enfin, pour chaque $s \in S$, soit $\mu(s) \in 2^V$ défini par : $x_i^j \in \mu(s)$ si et seulement si le j ème chiffre de la représentation binaire de $r_i(Z_i(s))$ est 1.

Pour chaque $i \in N$ et chaque $Z \in At(i)$, soit $k = r_i(Z)$ et $s_i(Z)$ la stratégie du joueur i dans G correspondant à la représentation binaire de k utilisant $\{x_i^1, \dots, x_i^{p_i}\}$, x_i^1 étant le bit le plus significatif. Par exemple, si $p_i = 3$ et $r_i(Z_i) = 6$ alors $s_i(Z) = (x_i^1, x_i^2, \neg x_i^3)$.

S_{G^*} dénote l'ensemble des états associés à G^* , eux-mêmes dénotés par s_{G^*} . L'ensemble des atomes de Eff_{G^*} est noté $At_{G^*}(i)$.

Remarque : Pour suivre cette construction, il peut être utile de voir comment elle fonctionne sur un exemple. Soit $N = \{1, 2, 3\}$, $S = \{1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C\}$, $At(1) = \{1234, 5678, 9ABC\}$, $At(2) = \{13579B, 2468AC\}$, $At(3) = \{12569C, 3478AB\}$ ⁷.

Par décomposabilité, on a $At(12) = \{13, 24, 57, 68, 9B, AC\}$, $At(13) = \{12, 34, 56, 78, 9C, AB\}$, et $At(23) = \{159, 37B, 26C, 48A\}$.

$|At(1)| = 3$, donc $p_1 = 2$. $|At(2)| = |At(3)| = 2$, donc $p_2 = p_3 = 1$. Alors, $V = \{x_1^1, x_1^2, x_2^1, x_3^1\}$.

Soit $At(1) = \{Z_0, Z_1, Z_2\}$, c'est-à-dire $r_1(1234) = 0$, $r_1(5678) = 1$ et $r_1(9ABC) = 2$. De même, $r_2(13579B) = 0$, $r_2(2468AC) = 1$, $r_3(12569C) = 0$ et $r_3(3478AB) = 1$.

⁷On omet les parenthèses pour les sous-ensembles de S : 1234 représente $\{1, 2, 3, 4\}$ et ainsi de suite.

Étudions $s = 6$. Nous avons $s = 5678 \cap 2468AC \cap 12569C$, et donc $s_{G^*} = \mu(s) = (\neg x_1^1, x_1^2, x_2^1, \neg x_3^1)$. Les contraintes sont $\gamma_1 = (x_1^1 \rightarrow \neg x_1^2)$, $\gamma_2 = \gamma_3 = \top$.

On aura donc $G^* = (N, V, \pi, \Gamma)$ défini par $N = \{1, 2, 3\}$, $V = \{x_1^1, x_1^2, x_2^1, x_3^1\}$, $\pi_1 = \{x_1^1, x_1^2\}$, $\pi_2 = \{x_2^1\}$, $\pi_3 = \{x_3^1\}$, $\gamma_1 = (x_1^1 \rightarrow \neg x_1^2)$ et $\gamma_2 = \gamma_3 = \top$.

Lemme 5.5. *Pour tout $i \in N$ et $Z \in At(i) : \mu(Z) = \{s_{G^*} \in S_{G^*} \mid s_{G^*} \supseteq s_i(Z)\}$*

Preuve du lemme : Soit $i \in N$ et $Z \in At(i)$. Soit $s_{G^*} \in \mu(Z)$; par définition de $\mu(Z)$, il existe un $s \in S$ tel que $\mu(s) = s_{G^*}$. Considérons la décomposition de s en atomes, c'est-à-dire $\{s\} = Z_1(s) \cap \dots \cap Z_n(s)$ (cf. Lemme 5.4). Par construction de μ , la projection de $\mu(s)$ sur $\{x_i^1, \dots, x_i^{p_i}\}$ correspond à la représentation binaire de $r_i(s)$. Donc $\mu(s) = s_{G^*}$ étend $s_i(Z)$.

Réciproquement, soit s_{G^*} tel que $s_{G^*} \supseteq s_i(Z)$. Pour tout $j \leq n$, soit k_j le nombre dont la représentation binaire dans $\{x_j^1, \dots, x_j^{p_j}\}$ est la projection de s_{G^*} sur $\{x_j^1, \dots, x_j^{p_j}\}$. Soit s défini par $\{s\} = Z_1(k_1) \cap \dots \cap Z_n(k_n)$. Par construction de μ , nous avons $\mu(s) = s_{G^*}$. De plus, $Z_i(k_i) = Z$ par atomicité, c'est-à-dire $s \in Z$. Donc $s_{G^*} \in \mu(Z)$. ■

Preuve : (Propriété 5.1, fin)

Nous devons encore prouver que pour tout $C \subseteq N$ et tout $X \subseteq S$, $X \in \text{Eff}(C)$ est vrai si et seulement si $\mu(X) \in \text{Eff}_{G^*}(C)$.

La décomposabilité de Eff et Eff_{G^*} implique qu'il suffit de montrer que pour tout i et tout $X \subseteq S$, $X \in \text{Eff}(i)$ si et seulement si $\mu(X) \in \text{Eff}_{G^*}(i)$. Puisque Eff et Eff_{G^*} satisfont la propriété de coalition-monotonie, il suffit de montrer que pour tout i , $Z_i \in At_{G^*}(i)$ implique $\mu(Z_i) \in \text{Eff}_{G^*}(i)$ et $T_i \in At_{G^*}(i)$ implique $\mu^{-1}(T_i) \in \text{Eff}(i)$.

Soit $Z_i \in At(i)$. Puisque $r(Z_i) \leq p_i$, nous avons $s_i(Z_i) \models \gamma_i$, et donc $s_i(Z_i) \in \text{Eff}_{G^*}(i)$. Le lemme 5.5, montre que $\mu(Z_i) = \{s_{G^*} \mid s_{G^*} \supseteq s_i(Z_i)\}$. Donc, $\mu(Z_i) \in \text{Eff}_{G^*}(i)$. Réciproquement, soit $T_i \in At_{G^*}(i)$. Le lemme 5.3 montre que $T_i = \{s \mid s \supseteq s_i\}$ pour tout $s_i \in S_i$. Soit $s_i = (\varepsilon_{i,1}.x_1^1, \dots, \varepsilon_{i,p_i}.x_i^{p_i})$ et $q(s_i) = \sum_{k=1}^{p_i} 2^{p_i-k}.\varepsilon_{i,k}$. Remarquons que $q(s_i) \leq p_i$, car $s_i \in S_i$ implique $s_i \models \gamma_i$. À présent, considérons $j = r_i^{-1}(q(s_i))$. Soit $Z_i^j \in At(i)$ tel que $r_i(Z_i^j) = j$. Nous avons $\mu(Z_i^j) = \{s \mid s \supseteq s_i\} = T_i$. Maintenant, $Z_i^j \in \text{Eff}(i)$, car $Z_i^j \in At(i)$. Donc, $\mu^{-1}(T_i) \in \text{Eff}(i)$.

Nous avons à présent prouvé que pour $C \subseteq N$ et tout $X \subseteq S$, $X \in \text{Eff}(C)$ est vrai si et seulement si $\mu(X) \in \text{Eff}_{G^*}(C)$. Donc, si Eff est fortement jouable, et satisfait l'atomicité, la décomposabilité et $\text{Eff}(N) = 2^S \setminus \emptyset$, alors il existe un jeu $G (= G^*)$ et une fonction injective $\mu : S \rightarrow 2^V$ telle que pour tout $C \subseteq N : \text{Eff}(C) = \{\mu(X) \mid X \in \text{Eff}(C)\}$. ■

Nous avons donc caractérisé les fonctions d' α -effectivité correspondant à un pré-jeu booléen. Nous voulons à présent définir et caractériser les coalitions efficaces dans un jeu booléen.

5.3 Coalitions efficaces

Nous considérons à présent les jeux booléens complets, et nous allons définir la notion de *coalition efficace*. Informellement, une coalition est efficace dans un jeu booléen si et seulement si tous les joueurs de cette coalition ont ensemble la possibilité de satisfaire tous leurs buts.

5.3.1 Définition et caractérisation

Définition 5.4. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen. Une coalition $C \subseteq N$ est **efficace** si et seulement s'il existe $s_C \in S_C$ tel que pour tout s_{-C} , $(s_C, s_{-C}) \models \bigwedge_{i \in C} \varphi_i$. On dénote par $EC(G)$ l'ensemble de toutes les coalitions efficaces d'un jeu G . On considère que la coalition vide \emptyset est efficace, car $\varphi_{\emptyset} = \bigwedge_{i \in \emptyset} \varphi_i \equiv \top$ est toujours satisfait.

Une coalition $C \subseteq N$ est **efficace minimale** s'il n'existe pas de coalition $B \subset C$ efficace.

On peut penser que dire que la coalition vide est efficace n'a pas grand sens. Quoi qu'il en soit, il serait possible de changer la définition d'une coalition efficace en excluant \emptyset sans qu'il soit nécessaire de changer les autres notions et résultats.

Exemple 5.1. Soit $G = (N, V, \Gamma, \pi, \Phi)$, avec $V = \{a, b, c\}$, $N = \{1, 2, 3\}$, $\gamma_i = \top$ pour tout i , $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\varphi_1 = (\neg a \wedge b)$, $\varphi_2 = (\neg a \vee \neg c)$ et $\varphi_3 = (\neg b \wedge \neg c)$.

Remarquons tout d'abord que $\varphi_1 \wedge \varphi_3$ est inconsistant. Aucune coalition contenant $\{1, 3\}$ ne peut donc être efficace. $\{1\}$ n'est pas efficace car φ_1 ne peut pas être satisfaite en fixant uniquement la valeur de a . Similairement, $\{2\}$ et $\{3\}$ ne sont pas efficaces non plus. $\{1, 2\}$ est efficace : les joueurs 1 et 2 ont une stratégie commune leur permettant de satisfaire leurs buts : $s_{\{1,2\}} = \bar{a}b \models \varphi_1 \wedge \varphi_2$. $\{2, 3\}$ est efficace car $s_{\{2,3\}} = \bar{b}\bar{c} \models \varphi_2 \wedge \varphi_3$. On a donc $EC(G) = \{\emptyset, \{1, 2\}, \{2, 3\}\}$.

On peut déjà voir à partir de cet exemple simple que EC n'est fermé "ni par le bas, ni par le haut" : en effet, si C est efficace, un sous-ensemble ou sur-ensemble de C peut ne pas être efficace. On peut également voir que EC n'est fermé ni par intersection, ni par union : $\{1, 2\}$ et $\{2, 3\}$ sont efficaces, mais ni $\{1, 2\} \cap \{2, 3\}$, ni $\{1, 2\} \cup \{2, 3\}$ ne le sont.

Exemple 5.2 (Echange de rein, d'après [Abraham et al., 2007]). Soit n couples d'individus, chacun d'entre eux étant constitué d'un receveur R_i , en attente d'un nouveau rein, et d'un donneur D_i , qui est prêt à donner un de ses reins pour sauver R_i . Comme le rein du donneur D_i n'est pas nécessairement compatible avec celui de R_i , une stratégie pour sauver le plus de gens possible consiste à considérer le graphe $\langle \{1, \dots, n\}, E \rangle$ qui contient un nœud $i \in 1, \dots, n$ pour chaque couple (D_i, R_i) , et un arc (i, j) si le rein de D_i est compatible avec celui de R_j . Une solution de ce problème consiste à identifier dans le graphe des cycles disjoints : D_i accepte de donner son rein si et seulement si R_i en reçoit un. Une solution optimale (qui sauve le plus de gens possible) de ce problème serait d'obtenir une couverture optimale du graphe par des cycles disjoints. Ce problème peut être vu comme le jeu booléen G suivant :

- * $N = \{1, \dots, n\}$;
- * $V = \{g_{ij} \mid i, j \in \{1, \dots, n\}\}$; si g_{ij} est mis à vrai, alors D_i donne un de ses reins à R_j .
- * $\pi_i = \{g_{ij} \mid 1 \leq j \leq n\}$;

- * pour tout i , $\gamma_i = \{\bigwedge_{j \neq k} \neg(g_{ij} \wedge g_{ik}) \wedge (\bigvee_j g_{ij}) \rightarrow (\bigvee_k g_{ki})\}$. La première condition exprime le fait qu'un donneur peut donner au plus un rein, tandis que la seconde exprime le fait qu'un donneur accepte de ne donner un rein que s'il en reçoit un.
- * pour tout i , $\varphi_i = \bigvee_{(j,i) \in E} g_{ji}$ exprime le fait que le but de i est de recevoir un rein compatible avec R_i .

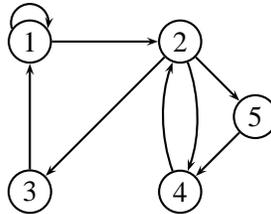
La seconde condition exprimée dans les contraintes de chacun des joueurs est inutile. En effet, le calcul des coalitions efficace permet de s'assurer que chaque membre d'une telle coalition satisfera son but, et donc recevra un rein compatible. Il est donc possible de simplifier la description de cette situation par le jeu booléen G suivant :

- * $N = \{1, \dots, n\}$;
- * $V = \{g_{ij} \mid i, j \in \{1, \dots, n\}\}$; si g_{ij} est mis à vrai, alors D_i donne un de ses reins à R_j .
- * $\pi_i = \{g_{ij}; 1 \leq j \leq n\}$;
- * pour tout i , $\gamma_i = \bigwedge_{j \neq k} \neg(g_{ij} \wedge g_{ik})$ exprime le fait qu'un donneur peut donner au plus un rein.
- * pour tout i , $\varphi_i = \bigvee_{(j,i) \in E} g_{ji}$ exprime le fait que le but de i est de recevoir un rein compatible avec R_i .

Par exemple, soit $n = 5$ et $E = \{(1,1), (1,2), (2,3), (2,4), (2,5), (3,1), (4,2), (5,4)\}$. Alors, on a : $G = (N, V, \Gamma, \pi, \Phi)$, avec

- * $N = \{1, 2, 3, 4, 5\}$
- * $V = \{g_{ij} \mid 1 \leq i, j \leq 5\}$;
- * $\forall i, \gamma_i = \bigwedge_{j \neq k} \neg(g_{ij} \wedge g_{ik})$
- * $\pi_1 = \{g_{11}, g_{12}, g_{13}, g_{14}, g_{15}\}$, et de même pour π_2 , etc.
- * $\varphi_1 = g_{11} \vee g_{31}$; $\varphi_2 = g_{12} \vee g_{42}$; $\varphi_3 = g_{23}$; $\varphi_4 = g_{24} \vee g_{54}$; $\varphi_5 = g_{25}$.

Le graphe correspondant est le suivant :



On voit clairement que les coalitions efficaces correspondent aux solutions de ce problème. Dans notre exemple, les coalitions efficaces sont \emptyset , $\{1\}$, $\{2,4\}$, $\{1,2,4\}$, $\{1,2,3\}$, $\{2,4,5\}$ et $\{1,2,4,5\}$.

Une caractérisation possible des coalitions efficaces dans un jeu booléen est la suivante :

Propriété 5.2. Soit $N = \{1, \dots, n\}$ un ensemble d'agents, et $SC \in 2^{2^N}$ un ensemble de coalitions. Il existe un jeu booléen G sur N tel que l'ensemble des coalitions efficaces de G est SC (i.e. $EC(G) = SC$) si et seulement si SC satisfait ces deux propriétés :

- (1) $\emptyset \in SC$.
- (2) pour tout $I, J \in SC$ tel que $I \cap J = \emptyset$ alors $I \cup J \in SC$.

Ainsi un ensemble de coalitions correspond à l'ensemble des coalitions efficaces d'un jeu booléen si et seulement si (a) il contient l'ensemble vide et (b) il est fermé par l'union pour les coalitions disjointes.

La preuve de cette propriété est également longue. Elle repose sur les lemmes 5.6 à 5.9 (page suivante).

Lemme 5.6. *Soit I et J deux coalitions d'un jeu booléen G . Si I et J sont efficaces et que $I \cap J = \emptyset$, alors $I \cup J$ est efficace.*

Preuve du lemme : Si I est efficace, alors nous savons que $\exists s_I \in S_I$ tel que $s_I \models \bigwedge_{i \in I} \varphi_i$. Il en est de même pour J : $\exists s_J \in S_J$ tel que $s_J \models \bigwedge_{j \in J} \varphi_j$. De plus, comme $I \cap J = \emptyset$, nous avons $(s_I, s_J) \models \bigwedge_{i \in I \cup J} \varphi_i$, donc $I \cup J$ est une coalition efficace. ■

Le lemme 5.6 prouve la direction (\Rightarrow) de la Propriété 5.2. La direction (\Leftarrow) de cette preuve est plus complexe, et nécessite de construire le jeu booléen G suivant pour chaque ensemble de coalitions \mathcal{SC} satisfaisant (1) et (2) :

- * $V = \{\text{connect}(i, j) \mid i, j \in N\}$ (toutes les connexions possibles entre joueurs) ;
- * $\forall i, \gamma_i = \top$;
- * $\pi_i = \{\text{connect}(i, j) \mid j \in N\}$ (toutes les connexions à partir du joueur i) ;
- * $\varphi_i = \bigvee_{I \in \mathcal{SC}, i \in I} F_I$, où

$$F_I = \left(\bigwedge_{j, k \in I} \text{connect}(j, k) \right) \wedge \left(\bigwedge_{j \in I, k \notin I} \neg \text{connect}(j, k) \right)$$

(le joueur i veut que tous les joueurs de sa coalition soient interconnectés, et qu'il n'y ait pas de connexion "sortant" de cette coalition).

Nous voulons montrer que l'ensemble $EC_G = \mathcal{SC}$ (EC_G étant l'ensemble des coalitions efficaces pour G).

Commençons par montrer que $\mathcal{SC} \subseteq EC_G$. Soit $I \in \mathcal{SC}$. Si chaque agent $i \in I$ joue $(\bigwedge_{j \in I} \text{connect}(i, j)) \wedge (\bigwedge_{k \notin I} \neg \text{connect}(i, k))$, alors φ_i est satisfait pour tout $i \in I$. I est alors une coalition efficace pour G , et \mathcal{SC} est inclus dans $EC(G)$.

Afin de prouver que $EC_G \subseteq \mathcal{SC}$, établissons les lemmes suivants :

Lemme 5.7. *Pour toute collection $\mathcal{SC} = \{C_i, i = 1, \dots, q\} \subseteq 2^N$, $\bigwedge_{1 \leq i \leq q} F_{C_i}$ est satisfiable si et seulement si pour tout $i, j \in \{1, \dots, q\}$, soit $C_i = C_j$, soit $C_i \cap C_j = \emptyset$.*

Preuve du lemme :

- (a) Si $C_1 = \dots = C_q = C$ alors $\bigwedge_{1 \leq i \leq q} F_{C_i} \equiv F_C$. Comme F_C est satisfait par toute interprétation assignant chaque $\text{connect}(i, j)$ tel que $i, j \in C$ à vrai, et chaque $\text{connect}(i, j)$ tel que $i \in C$ et $j \notin C$ à faux (l'instanciation des variables $\text{connect}(i, j)$ tel que $i, j \notin C$ est indifférente), F_C est consistant.
- (b) Supposons à présent que pour tout $i, j \in \{1, \dots, q\}$, soit $C_i = C_j$, soit $C_i \cap C_j = \emptyset$.

Alors, $\bigwedge_{1 \leq i \leq q} F_{C_i}$ est équivalent à

$$\begin{aligned} & (\bigwedge_{1 \leq i \leq q} \bigwedge_{j,k \in C_i} \text{connect}(j,k)) \quad \wedge \\ & (\bigwedge_{1 \leq i \leq q} \bigwedge_{j \in C_i, k \notin C_i} \neg \text{connect}(j,k)) \end{aligned}$$

et $\bigwedge_{1 \leq i \leq q} F_{C_i}$ est satisfait par toute interprétation assignant chaque $\text{connect}(j,k)$ tel que j,k appartiennent au même C_i à vrai, et chaque $\text{connect}(j,k)$ tel que $j \in C_i$ pour chaque i et $k \notin C_i$ à faux. $\bigwedge_{1 \leq i \leq q} F_{C_i}$ est donc satisfiable.

- (c) Supposons que pour tout $i, j \in \{1, \dots, q\}$, nous avons $C_i \cap C_j \neq \emptyset$ et $C_i \neq C_j$. Soit $k \in C_i \cap C_j$ et (sans perte de généralité) $l \in C_i \setminus C_j$. Alors, $F_{C_i} \models \text{connect}(k,l)$ et $F_{C_j} \models \neg \text{connect}(k,l)$, donc $F_{C_i} \wedge F_{C_j}$ n'est pas satisfiable, et a fortiori, $\bigwedge_{1 \leq i \leq q} F_{C_i}$ non plus. ■

On définit à présent la *couverture d'une coalition I par un sous-ensemble disjoint de SC* comme un tuple $\vec{C} = \langle C_i | i \in I \rangle$ de coalitions tel que : (i) pour chaque $k \in I$, $C_k \in \mathcal{SC}$; (ii) pour tout $C_j, C_k \in \vec{C}$, soit $C_j = C_k$, soit $C_j \cap C_k = \emptyset$; (iii) pour chaque $i \in I$, $i \in C_i$. Soit $\text{Cov}(\mathcal{SC}, I)$ l'ensemble de toutes les couvertures de I par un sous-ensemble disjoint de \mathcal{SC} .

Par exemple, si $\mathcal{SC} = \{\emptyset, 1, 24, 123, 124\}$ alors $\text{Cov}(\mathcal{SC}, 12) = \{\langle 1, 24 \rangle, \langle 123, 123 \rangle, \langle 124, 124 \rangle\}$ ⁸, $\text{Cov}(\mathcal{SC}, 123) = \{\langle 123, 123, 123 \rangle\}$, $\text{Cov}(\mathcal{SC}, 124) = \{\langle 1, 1, 24 \rangle, \langle 1, 24, 24 \rangle, \langle 124, 124, 124 \rangle\}$ et $\text{Cov}(\mathcal{SC}, 234) = \text{Cov}(\mathcal{SC}, 1234) = \emptyset$.

Lemme 5.8. Pour tout $I \neq \emptyset$, Φ_I est équivalent à $\bigvee_{\vec{C} \in \text{Cov}(\mathcal{SC}, I)} \bigwedge_{i \in I} F_{C_i}$.

Preuve du lemme :

$$\begin{aligned} \Phi_I & \equiv \bigwedge_{i \in I} \Phi_i \\ & \equiv \bigwedge_{i \in I} (\bigvee_{I \in \mathcal{SC}, i \in I} F_I) \\ & \equiv \bigvee_{\substack{\langle C_i, i \in I \rangle \text{ tel que } C_i \in \mathcal{SC} \\ \text{et } i \in C_i \text{ pour tout } i \in I}} (\bigwedge_{i \in I} F_{C_i}) \end{aligned}$$

Le lemme 5.7 montre que $\bigwedge_{i \in I} F_{C_i}$ est satisfiable si et seulement si pour tout $i, j \in I$, soit $C_i = C_j$, soit $C_i \cap C_j = \emptyset$. Donc, $\Phi_I \equiv \bigvee_{\vec{C} \in \text{Cov}(\mathcal{SC}, I)} \bigwedge_{i \in I} F_{C_i}$. ■

Par exemple, si $\mathcal{SC} = \{1, 24, 123, 124\}$ alors $\Phi_{12} \equiv (F_1 \wedge F_{24}) \vee F_{123} \vee F_{124}$; $\Phi_{123} \equiv F_{123}$; $\Phi_{124} \equiv (F_1 \wedge F_{24}) \vee F_{124}$; $\Phi_{234} \equiv \perp$.

Lemme 5.9. Soit $I \subseteq 2^A$ et sachant que $\forall I, J \in \mathcal{SC}, I \cap J = \emptyset \Rightarrow I \cup J \in \mathcal{SC}$, Φ_I est satisfiable si et seulement si il existe $J \in \mathcal{SC}$ tel que $I \subseteq J$.

Preuve du lemme : Le cas $I = \emptyset$ est simple : $\Phi \equiv \top$ est satisfiable, et $\emptyset \in \mathcal{SC}$ par hypothèse, donc il existe $J \in \mathcal{SC}$ ($J = \emptyset$) tel que $I \subseteq J$.

⁸Il y a 2 joueurs dans $I = \{1, 2\}$, donc chaque \vec{C} dans $\text{Cov}(\mathcal{SC}, 12)$ contient 2 coalitions, une pour chaque joueur, satisfaisant les conditions (i), (ii) et (iii).

Supposons à présent que $I \neq \emptyset$.

\Rightarrow Supposons que Φ_I est satisfiable. Le lemme 5.8 montre que si Φ_I est équivalent à $\bigvee_{\vec{C} \in \text{Cov}(\mathcal{SC}, I)} \bigwedge_{i \in I} F_{C_i}$, alors il existe un \vec{C} dans $\text{Cov}(\mathcal{SC}, I)$ tel que $\bigwedge_{i \in I} F_{C_i}$ est satisfiable, et donc $\text{Cov}(\mathcal{SC}, I)$ n'est pas vide. $\vec{C} \in \text{Cov}(\mathcal{SC}, I)$ implique que :

- (i) Pour tout $i \in I, C_i \in \mathcal{SC}$;
- (ii) pour tout $i, j \in I$, soit $C_i = C_j$, soit $C_i \cap C_j = \emptyset$.
- (iii) $I \subseteq \bigcup_{i \in I} C_i$

A partir de (i), (ii) et sachant que $\forall I, J \in \mathcal{SC}, I \cap J = \emptyset \Rightarrow I \cup J \in \mathcal{SC}$, on prouve que $\bigcup_{i \in I} C_i \in \mathcal{SC}$. (iii) permet alors de montrer qu'il existe un $J \in \mathcal{SC}$ ($J = \bigcup_{i \in I} C_i$) tel que $I \subseteq J$.

\Leftarrow Supposons maintenant qu'il existe un $J \in \mathcal{SC}$ tel que $I \subseteq J$. Alors $\Phi_J \models \Phi_I$, et Φ_J est consistant (en considérant l'interprétation assignant chaque $\text{connect}(i, j)$ tel que $i, j \in J$ à vrai).

■

Preuve : (Propriété 5.2, fin)

Il reste à prouver que $\text{EC}_G \subseteq \mathcal{SC}$. Soit I une coalition efficace telle que $I \notin \mathcal{SC}$ (ce qui implique $I \neq \emptyset$, à cause de l'hypothèse $\emptyset \in \mathcal{SC}$).

- * Si $I = N$ alors il n'existe pas de $J \in \mathcal{SC}$ tel que $I \subseteq J$ (car $I \notin \mathcal{SC}$), et alors le lemme 5.9 implique que Φ_I n'est pas satisfiable. I ne peut donc pas être efficace pour G .
- * Supposons à présent que $I \neq N$ et définissons la \bar{I} -stratégie suivante $s_{\bar{I}}$ ($\bar{I} = N \setminus I$) : pour chaque $i \in \bar{I}, s_i = \{\neg \text{connect}(i, j) \mid j \in I\}$ (peu importe l'instanciation des variables $\text{connect}(i, j)$ telles que $j \notin I$). Soit $\vec{C} = \langle C_i, i \in I \rangle \in \text{Cov}(\mathcal{SC}, I)$.

Nous affirmons tout d'abord qu'il existe un $i^* \in I$ tel que C_{i^*} n'est pas contenue dans I . En effet, supposons que pour tout $i \in I, C_i \subseteq I$. Puisque $i \in C_i$ est vrai pour tout i , nous avons $\bigcup_{i \in I} C_i = I$. À présent, $C_i \in \mathcal{SC}$ pour tout i , et toutes les C_i, C_j distinctes sont disjointes deux à deux. La propriété amène $I \in \mathcal{SC}$, ce qui est faux par hypothèse.

Soit $k \in C_{i^*} \setminus I$ (un tel k existe car C_{i^*} n'est pas inclus dans I). Il faut que $\text{connect}(k, i^*)$ soit vrai pour satisfaire F_{C_i} car i et k sont dans C_i . On a donc $s_k \models \neg F_{C_i}$, et a fortiori $s_{\bar{I}} \models \neg F_{C_i}$, ce qui entraîne $s_{\bar{I}} \models \neg \bigwedge_{i \in I} F_{C_i}$.

Ceci étant vrai pour tout $\vec{C} \in \text{Cov}(\mathcal{SC}, I)$, nous avons $s_{\bar{I}} \models \bigwedge_{\vec{C} \in \text{Cov}(\mathcal{SC}, I)} \neg \bigwedge_{i \in I} F_{C_i}$, c'est-à-dire $s_{\bar{I}} \models \neg \bigvee_{\vec{C} \in \text{Cov}(\mathcal{SC}, I)} \bigwedge_{i \in I} F_{C_i}$. Avec le lemme 5.8, ceci entraîne $s_{\bar{I}} \models \neg \Phi_I$. Donc I ne contrôle pas Φ_I et I ne peut pas être efficace pour G .

■

La notion de coalition efficace que nous introduisons est la même que celle de coalition

fructueuse (successful coalition) dans un jeu coalitionnel qualitatif (QCG⁹, notion introduite dans [Wooldridge et Dunne, 2004]). En effet, une coalition fructueuse dans un QCG est une coalition ayant une stratégie (ou une possibilité) de satisfaire tous les membres de cette coalition.

Il est naturel d'utiliser ici la notion de stratégie gagnante (similaire celle de coalition efficace, à la différence près qu'elle s'intéresse à des singletons de joueurs (voir définition 2.5 (page 37))), et d'étudier les propriétés de ces stratégies gagnantes pour un joueur ou pour une coalition de joueurs. Pour cela, nous utiliserons la notion d'impliquant premier (voir définition 2 (page 8)).

Propriété 5.3. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen.

1. $\forall i \in N$, i a une stratégie gagnante si et seulement si $\exists \alpha \in PI_{\pi_i}(\varphi_i)$ tel que $\alpha \models \gamma_i$
2. Une coalition de joueurs $I \subseteq N$ est efficace si et seulement si $\exists \alpha \in PI_{\bigcup_{i \in I} \pi_i}(\bigwedge_{i \in I} \varphi_i)$ tel que $\alpha \models \bigwedge_{i \in I} \gamma_i$

Intuitivement, si le but qu'un joueur (resp. un ensemble de joueurs noté I) cherche à satisfaire contient un terme (donc une conjonction de littéraux) α dont tous les littéraux sont contrôlés par le joueur, alors le joueur (resp. I) a une stratégie gagnante. En effet, ce terme α peut être satisfait, donc le but qui le contient est aussi satisfait, et le joueur (resp. l'ensemble des joueurs I) gagne.

Preuve :

1. Pour un joueur.

\Leftarrow D'après la définition 2 (page 8), si $\exists \alpha \in PI_{\pi_i}(\varphi_i)$, alors $\alpha \models \varphi_i$ et $Lit(\alpha) \subseteq \pi_i$.

Comme $\alpha \models \gamma_i$, i possède une stratégie $s_i \in S_i$ telle que : $\forall s_{-i} \in S_{-i}, (s_{-i}, s_i) \models \varphi_i$.

\Rightarrow D'après la définition 2.5 (page 37), i possède une stratégie gagnante si et seulement si :

$$\exists s_i \in S_i, \forall s_{-i} \in S_{-i} (s_{-i}, s_i) \models \varphi_i$$

Le joueur i peut donc satisfaire son but quels que soient les choix de ses adversaires. Il existe donc $\alpha \in S_i$ tel que : $\alpha \models \varphi_i$ et $Lit(\alpha) \subseteq \pi_i$. α est un π_i -impliquant de φ_i . Soit α est un π_i -impliquant premier de φ_i , et donc $\alpha \models \gamma_i$ (car $\alpha \in S_i$) ; soit il existe α' π_i -impliquant premier de φ_i tel que $\alpha \models \alpha'$ et $\alpha' \models \gamma_i$.

2. Pour une coalition de joueurs, la preuve est identique en remplaçant π_i par $\bigcup_{i \in I} \pi_i$, γ_i par $\bigwedge_{i \in I} \gamma_i$ et φ_i par $\bigwedge_{i \in I} \varphi_i$.

■

Une coalition de joueurs aura donc une stratégie gagnante (et sera donc efficace) si et seulement si il existe un impliquant premier de la conjonction de tous les buts des membres de la coalition, cet impliquant premier étant composé uniquement de variables contrôlées par les membres de la coalition.

Étudions ce problème sur un exemple simple.

Exemple 5.3. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen. On donne

⁹Nous introduirons plus longuement les QCGs dans la section 5.4.2 (page 161).

- * $V = \{a, b, c\}, N = \{1, 2, 3\},$
- * $\gamma_1 = \neg a, \gamma_2 = \gamma_3 = \top,$
- * $\pi_1 = \{a\}, \pi_2 = \{b\}, \pi_3 = \{c\},$
- * $\varphi_1 = (a \leftrightarrow (b \wedge c)),$
- * $\varphi_2 = (\neg a \vee \neg c)$ et
- * $\varphi_3 = (a \wedge \neg b).$

On constate tout d'abord qu'aucun joueur isolé n'a de stratégie gagnante.

On remarque ensuite que les trois joueurs ne peuvent pas gagner tous ensemble. En effet, $\varphi_1 \wedge \varphi_3$ est incohérent. Donc il ne peut pas exister de coalition entre 1 et 3, et les seules coalitions possibles sont $\{1, 2\}$ et $\{2, 3\}$. Etudions-les.

- * $\{1, 2\}$

$$(\varphi_1 \wedge \varphi_2) = (\neg a \wedge \neg b) \vee (\neg a \wedge \neg c)$$

Il existe un impliquant premier, $(\neg a \wedge \neg b)$, qui ne contient que des variables contrôlées par 1 et 2. et qui satisfait les contraintes des deux joueurs $((\neg a \wedge \neg b) \models \gamma_1 \wedge \gamma_2)$. Cette coalition a donc une stratégie gagnante. Comme c'est la plus petite coalition contenant 1 et 2, c'est une coalition efficace minimale.

- * $\{2, 3\}$

$$(\varphi_2 \wedge \varphi_3) = (a \wedge \neg b \wedge \neg c)$$

Cette coalition n'a pas de stratégie gagnante car la variable a , qui n'est contrôlée par aucun joueur de cette coalition, appartient au seul impliquant premier. Ce n'est donc pas une coalition efficace pour le jeu G .

Remarquons que de façon (presque) équivalente, la recherche d'une stratégie gagnante correspond à une résolution de $QBF_{2,\exists}$ [Stockmeyer, 1977]. En effet, comme on l'a vu dans la propriété 5.3 (page précédente), la recherche d'une stratégie gagnante correspond à la recherche d'un impliquant premier. Or, cette recherche est du type $\exists A \forall B \varphi(A, B)$: pour un but φ donné, qui peut être le but d'un joueur ou d'une coalition I , on cherche s'il existe un impliquant α de φ ne contenant que des variables contrôlées par i . Si α existe, on veut que pour tout autre impliquant, α soit le "plus petit".

Par ailleurs, l'existence d'une stratégie gagnante est une instance du problème de *contrôlabilité* en logique propositionnelle [Boutilier, 1994; Lang et Marquis, 1998b] (voir aussi [?] pour une extension de la contrôlabilité à un cadre multi-agents).

5.3.2 Coalition efficace et noyau

Nous allons à présent faire le lien entre la notion de coalition efficace et celle, bien connue, de noyau (ou cœur) dans un jeu coalitionnel. Dans des jeux coalitionnels avec des préférences ordinales, le noyau est habituellement défini comme suit (voir par exemple [Aumann, 1967; Owen, 1982; Myerson, 1991]) : un profil de stratégies s est dans le *noyau* d'un jeu coalitionnel si et seulement s'il n'existe pas de coalition C telle que tous les membres de cette coalition ont une stratégie commune s_C qui leur permet à tous d'obtenir une meilleure utilité qu'avec s .

Ici, nous considérerons également une notion de noyau plus forte que cette dernière : un profil de stratégies s est dans le *noyau fort* d'un jeu coalitionnel si et seulement s'il n'existe pas de coalition C telle que tous les membres de cette coalition ont une stratégie commune s_C qui leur permet à tous

d'obtenir une utilité au moins aussi bonne que celle obtenue avec s , et une meilleure utilité pour au moins un des membres de C .

Définition 5.5. Soit G un jeu booléen.

Le **noyau (faible)** de G , dénoté par $Noyau(G)$, est l'ensemble des profils de stratégies $s = (s_1, \dots, s_n)$ tels qu'il n'existe pas de $C \subset N$ et pas de $s_C \in S_C$ tels que pour tout $i \in C$ et tout $s_{-C} \in S_{-C}$, $(s_C, s_{-C}) \succ_i s$.

Le **noyau fort** de G , dénoté par $FNoyau(G)$, est l'ensemble des profils de stratégies $s = (s_1, \dots, s_n)$ tels qu'il n'existe pas de $C \subset N$ et pas de $s_C \in S_C$ tels que pour tout $i \in C$ et tout $s_{-C} \in S_{-C}$, $(s_C, s_{-C}) \succeq_i s$, et qu'il existe un $i \in C$ tel que pour tout $s_{-C} \in S_{-C}$, $(s_C, s_{-C}) \succ_i s$.

Ce concept de noyau faible est équivalent à la notion d'**équilibre de Nash fort** introduit par [Aumann, 1959]¹⁰ : dans un équilibre de Nash fort, les coalitions se forment afin de mettre en corrélation les stratégies de leurs membres. Cette notion implique, au moins implicitement, l'hypothèse que la coopération impose nécessairement que les joueurs soient capables de signer des "accords d'engagement" : les joueurs doivent suivre les stratégies sur lesquelles ils se sont engagés, même si certains d'entre eux tireraient profit à dévier de cette stratégie. Si les joueurs d'une coalition C s'entendent sur une stratégie s_C , au moins un de ces joueurs est satisfait par cette stratégie : $\exists i \in C$ tel que $s \models \varphi_i$.

Dans le cadre des jeux booléens, la relation entre le noyau (faible) et l'ensemble de coalitions efficaces est donné par le simple résultat suivant.

Propriété 5.4. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen. $s \in Noyau(G)$ si et seulement si s satisfait au moins un membre de chaque coalition efficace, c'est-à-dire, pour tout $C \in EC(G)$, $s \models \bigvee_{i \in C} \varphi_i$.

Preuve : $s = (s_1, \dots, s_n) \in Noyau(G)$ si et seulement si

$$\begin{aligned} & \nexists C \subset N \text{ tel que } \exists s_C \in S_C, \forall i \in C, \forall s_{-C} \in S_{-C}, (s_C, s_{-C}) \succ_i s \\ \Leftrightarrow & \nexists C \subset N \text{ tel que } \exists s_C \in S_C, \forall i \in C, \forall s_{-C} \in S_{-C}, ((s_C, s_{-C}) \models \varphi_i) \wedge (s \models \neg \varphi_i) \\ \Leftrightarrow & \nexists C \subset N \text{ tel que } \exists s_C \in S_C, \forall i \in C, (s_C \models \varphi_i) \wedge (s \models \neg \varphi_i) \\ \Leftrightarrow & \nexists C \subset N \text{ tel que } \exists s_C \in S_C, (s_C \models \bigwedge_{i \in C} \varphi_i) \wedge (s \models \bigwedge_{i \in C} \neg \varphi_i) \\ \Leftrightarrow & \nexists C \subset N \text{ tel que } (s \models \bigwedge_{i \in C} \neg \varphi_i) \wedge (\exists s_C \in S_C, s_C \models \bigwedge_{i \in C} \varphi_i) \\ \Leftrightarrow & \forall C \subset N, (s \models \bigvee_{i \in C} \varphi_i) \vee (\forall s_C \in S_C, s_C \models \bigvee_{i \in C} \neg \varphi_i) \\ \Leftrightarrow & \forall C \subset N, \text{ si } \exists s_C \in S_C : s_C \models \bigwedge_{i \in C} \varphi_i \text{ alors } s \models \bigvee_{i \in C} \varphi_i \end{aligned}$$

$\exists s_C \in S_C : s_C \models \bigwedge_{i \in C} \varphi_i$ signifie que la coalition C est efficace. On a donc $s = (s_1, \dots, s_n) \in Noyau(G)$ si et seulement si $\forall C \subset N$, si $C \in EC(G)$ alors $s \models \bigvee_{i \in C} \varphi_i$.

■

En particulier, quand aucune coalition d'un jeu booléen G n'est efficace, alors tous les profils de stratégies sont dans $Noyau(G)$. Cette propriété entraîne le résultat suivant :

Propriété 5.5. Décider si un profil de stratégies s est dans le noyau faible d'un jeu booléen G est Π_2^P -complet.

¹⁰Il est facile de montrer cette équivalence : c'est juste une ré-écriture de la définition donnée dans [Aumann, 1959].

Preuve : L'appartenance à Π_2^P est immédiate : le problème de décider si un profil de stratégies s n'est pas dans le noyau faible de G est dans Σ_2^P . En effet, nous devons tout d'abord trouver une coalition $C \subseteq N$, et ensuite deviner une stratégie $s'_C \in S_C$ pour cette coalition telle que $\forall i \in C, (s'_C, s_{-C}) \succ_i s$. Nous avons donc à vérifier que $(s'_C, s_{-C}) \models \varphi_i$ et $s \models \neg\varphi_i$, ce qui est dans P.

Pour montrer la difficulté, nous allons montrer que le problème complémentaire est Σ_2^P -complet. Cette difficulté est obtenue à partir d'une réduction du problème consistant à décider de la validité d'une instance de $\text{QBF}_{2,\exists}$. Soit $Q = \exists A, \forall B, \Phi$, où A et B sont deux ensembles disjoints de variables, et Φ est une formule de $L_{A \cup B}$. On définit un jeu booléen à deux joueurs par $\varphi_1 = x$, $\varphi_2 = \Phi \wedge y$, où x et y sont deux nouvelles variables ($x, y \notin A \cup B$), $\pi_1 = B \cup \{x\}$ et $\pi_2 = A \cup \{y\}$. Ce jeu peut être évidemment construit en temps polynomial étant donné Q .

Soit M_A une A -interprétation quelconque, M_B une B -interprétation quelconque, $s_1 = (x, M_B)$, et $s_2 = (\bar{y}, M_A)$. $s = (s_1, s_2)$.

- * Supposons que Q est valide. Il existe donc un M_A tel que $M_A \models \Phi$, et $s \models \neg\varphi_2$. Il existe donc $C = \{2\} \subset N$ telle que $PI_{\pi_2}(\varphi_2) \neq \emptyset$ (donc $\{2\}$ est une coalition efficace d'après la propriété 5.3 (page 148)), et pourtant telle que $s \models \neg\varphi_2$. D'après la propriété 5.4 (page précédente), $s \notin \text{Noyau}(G)$.
- * Réciproquement, si Q n'est pas valide, pour tout M_A , il existe un M_B tel que $M_A M_B \models \neg\Phi$. Dans ce cas, 2 n'est pas efficace (puisque 2 ne contrôle pas toutes les variables permettant d'obtenir Φ , cf. propriété 5.3 (page 148)). L'ensemble des coalitions efficaces de ce jeu est donc $\text{EC}(G) = \{\emptyset, \{1\}, \{1, 2\}\}$, et $\forall C \in \text{EC}(G)$, il existe $i \in C$ tel que $s \models \varphi_i$ (puisque $s \models \varphi_1$). Donc $s \in \text{Noyau}(G)$.

On a donc bien $s \notin \text{Noyau}(G)$ si et seulement si Q est valide, et donc $s \in \text{Noyau}(G)$ si et seulement si Q n'est pas valide, ou de façon équivalente ssi $\neg Q = \forall A, \exists B, \neg\Phi$ est valide. ■

On prouve également que le noyau faible d'un jeu booléen ne peut pas être vide.

Propriété 5.6. *Pour tout jeu booléen G , $\text{Noyau}(G) \neq \emptyset$.*

Preuve : On construit l'ensemble de coalitions E comme suit. Tout d'abord, on initialise E à \emptyset . Ensuite, tant qu'il existe une coalition C dans $\text{EC}(G)$ telle que $C \cap C' = \emptyset$ est vrai pour tout $C' \in E$, on ajoute cette coalition C dans E ($E := E \cup \{C\}$). Une fois cet algorithme achevé, E est un ensemble de coalitions efficaces $\{C_i, i \in I\}$ toutes disjointes entre elles. Grâce à la Propriété 2, on sait que $\cup_{i \in I} C_i$ est efficace.

Il existe donc $s_E \in S_E$ tel que $s_E \models \bigwedge_{i \in E} \varphi_i$, et E contient au moins un élément de chaque coalition efficace (si ce n'était pas le cas, il resterait une coalition efficace C telle que l'intersection de C et de tous les C_i serait vide, et dans ce cas l'algorithme aurait continué de tourner, et incorporé C dans E).

Soit s un profil de stratégies étendant s_C . s satisfait au moins un membre de chaque coalition efficace, et donc grâce à la Propriété 5.4, $\text{Noyau}(G) \neq \emptyset$.

■

Le noyau fort d'un jeu booléen est plus difficile à caractériser en termes de coalitions efficaces. Nous avons seulement l'implication suivante :

Propriété 5.7. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen, et s un profil de stratégies. Si $s \in FNoyau(G)$, alors pour tout $C \in EC(G)$ et tout $i \in C$, $s \models \varphi_i$.

Preuve : Nous allons prouver que s'il existe $C \in EC(G)$ et s'il existe $i \in C$ tel que $s \models \neg\varphi_i$, alors $s \notin FNoyau(G)$.

$$\begin{aligned} & \exists C \in EC(G), \exists i \in C \text{ tel que } s \models \neg\varphi_i \\ \Leftrightarrow & \exists C \subseteq N, \exists s_C \in S_C, \exists i \in C \text{ tel que } (s_C \models \bigwedge_{j \in C} \varphi_j) \wedge (s \models \neg\varphi_i) \\ \Rightarrow & \exists C \subseteq N, \exists s'_C \in S_C, \forall s_{-C} \in S_{-C}, \exists i \in C \text{ tel que} \\ & ((s_C, s_{-C}) \models \varphi_i) \wedge (s \models \neg\varphi_i) \wedge (s_C \models \bigwedge_{j \in C} \varphi_j) \\ \Rightarrow & \exists C \subseteq N, \exists s_C \in S_C, \forall s_{-C} \in S_{-C}, \exists i \in C \text{ tel que} \\ & ((s_C, s_{-C}) \succ_i S) \wedge (\forall i \in C (s_C, s_{-C}) \succeq_i s) \end{aligned}$$

On sait donc que si $\exists C \in EC(G)$ et $\exists i \in C$ tels que $s \models \neg\varphi_i$, alors $s \notin FNoyau(G)$.

■

Donc, une stratégie présente dans le noyau fort d'un jeu G satisfait les buts de tous les membres de toutes les coalitions efficaces. L'exemple suivant montre que la réciproque est fautive.

Exemple 5.4. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen avec

- * $V = \{a, b, c, d, e, f\}$, $N = \{1, 2, 3, 4, 5, 6\}$,
- * $\gamma_i = \top$ pour tout i ,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, $\pi_4 = \{d\}$, $\pi_5 = \{e\}$, $\pi_6 = \{f\}$,
- * $\varphi_1 = b \vee d$, $\varphi_2 = a \vee c$, $\varphi_3 = \neg b \vee d$, $\varphi_4 = e$, $\varphi_5 = \neg a \wedge \neg b \wedge \neg c$ et $\varphi_6 = \neg a \wedge \neg c \wedge \neg d$.

Ce jeu a deux coalitions efficaces : $\{1, 2\}$ et $\{2, 3\}$.

Soit $s = abcd\bar{e}f$. Nous avons $s \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \neg\varphi_4 \wedge \neg\varphi_5 \wedge \neg\varphi_6$. Donc, $\forall C \in EC(G)$, $\forall i \in C$, $s \models \varphi_i$.

Pourtant, $s \notin FNoyau(G)$: $\exists C' = \{1, 2, 3, 4, 5\} \subset N$ tel que $\exists s_C = abcde \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3 \wedge \varphi_4 \wedge \neg\varphi_5$. Donc, $\forall s_{-C}$, $(s_C, s_{-C}) \succeq_1 s$, $(s_C, s_{-C}) \succeq_2 s$, $(s_C, s_{-C}) \succeq_3 s$, $(s_C, s_{-C}) \succeq_5 s$, et $(s_C, s_{-C}) \succ_4 s$. $s \notin FNoyau(G)$.

Remarquons que le noyau fort d'un jeu booléen peut être vide : dans l'exemple 5.1, l'ensemble des coalitions efficaces est $\{\emptyset, \{1, 2\}, \{2, 3\}\}$. Il n'y a donc pas de $s \in S$ tel que pour tout $C \in EC(G)$, pour tout $i \in C$, $s \models \varphi_i$, et donc $FNoyau(G) = \emptyset$. Pourtant, nous pouvons montrer que la condition sous laquelle le noyau fort d'un jeu booléen est non vide est équivalente à la suivante sur les coalitions efficaces.

Propriété 5.8. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen.

$FNoyau(G) \neq \emptyset$ si et seulement si $\bigcup \{C \subseteq N \mid C \in EC(G)\} \in EC(G)$, c'est-à-dire si et seulement si l'union de toutes les coalitions efficaces est efficace.

Preuve : Soit $MEC(G) = \bigcup_{C \subseteq N} \{C \in EC(G)\}$, MEC signifiant *Coalitions Efficaces Maximales*.

\Leftarrow $FNoyau(G) \neq \emptyset$. Soit $s \in FNoyau(G)$. A partir de la Propriété 5.7, nous savons que $\forall C \in EC(G), \forall i \in C, s \models \varphi_i$. Donc $\forall i \in MEC(G), s \models \varphi_i$. Donc $MEC(G) \in EC(G)$.

\Rightarrow $MEC(G) \in EC(G)$. Soit $s_{MEC(G)} \in S_{MEC(G)}$ tel que $\forall s_{-MEC(G)}, s_{MEC(G)} \models \Phi_{MEC(G)}$. On cherche s tel que $s \in FNoyau(G)$.

Soit $s_{-MEC(G)} \in S_{-MEC(G)}$ tel que $MAX = \{i | s = (s_{MEC(G)}, s_{-MEC(G)}) \models \varphi_i\}$ soit maximal pour \subseteq . $s_{-MEC(G)}$ existe, dans le pire des cas $s \models \Phi_{MEC(G)}$. Comme MAX est maximal, on ne peut pas trouver un $C \subseteq N$ tel que $\exists s_C \in S_C$ tel que $\forall s_{-C} \in S_{-C}, \forall i \in C, (s_C, s_{-C}) \succeq_i s$, et $\exists i \in C, (s_C, s_{-C}) \succ_i s$. Si on suppose que C existe, alors $\forall i \in N$ tel que $s \models \varphi_i$, on a $s_C \models \varphi_i$, et $\exists i \in N$ tel que $s \not\models \varphi_i$ et $s_C \models \varphi_i$. Dans ce cas, MAX n'est pas maximal pour \subseteq .

■

Nous pouvons à présent identifier la complexité de quelques problèmes clés concernant les coalitions efficaces.

Propriété 5.9. *Décider s'il existe une coalition efficace non vide dans un jeu booléen est un problème Σ_2^p -complet, et est Σ_2^p -difficile même si $n = 2$.*

Preuve : L'appartenance à Σ_2^p est immédiate. Pour montrer que décider s'il existe une coalition efficace non vide dans un jeu booléen est un problème Σ_2^p -difficile (même avec 2 agents), considérons la réduction polynomiale à partir de $QBF_{2,\exists}$ suivante : à chaque instance $I = \exists a_1 \dots a_p \forall b_1 \dots b_q \varphi$ de $QBF_{2,\exists}$, considérons le jeu booléen $G_I = \langle N, V, \pi, \Gamma, \Phi \rangle$, avec

- * $N = \{1, 2\}, \gamma_1 = \gamma_2 = \top$,
- * $V = \{a_1 \dots a_p, b_1 \dots b_q, x\}$,
- * $\pi_1 = \{a_1, \dots, a_p, x\}, \pi_2 = \{b_1, \dots, b_q\}$,
- * $\varphi_1 = \varphi$ et $\varphi_2 = \neg \varphi \wedge x$.

Ni $\{2\}$, ni $\{1, 2\}$ ne peuvent être efficaces. La seule coalition efficace non vide possible est donc $\{1\}$. Or, $\{1\}$ est efficace si et seulement si I est une instance valide de $QBF_{2,\exists}$.

■

Propriété 5.10.

- * *décider si un agent i appartient à au moins une coalition efficace pour G est un problème Σ_2^p -complet, et est Σ_2^p -difficile même si $n = 2$.*
- * *décider si un agent i appartient à toute coalition efficace non vide pour G est un problème Π_2^p -complet, et est Π_2^p -difficile même si $n = 2$.*

Preuve : Pour les deux problèmes, l'appartenance est facile à obtenir.

Pour montrer que décider si un agent i appartient à au moins une coalition efficace

pour G est un problème Σ_2^P -difficile (même avec deux joueurs) considérons la réduction polynomiale de $\text{QBF}_{2,\exists}$ suivante : à chaque instance $I = \exists a_1 \dots a_p \forall b_1 \dots b_q \varphi$ de $\text{QBF}_{2,\exists}$, considérons le jeu booléen $G_I = \langle N, V, \pi, \Gamma, \Phi \rangle$, avec

- * $N = \{1, 2\}$, $\gamma_1 = \gamma_2 = \top$,
- * $V = \{a_1, \dots, a_p, b_1, \dots, b_q\}$,
- * $\pi_1 = \{a_1, \dots, a_p\}$, $\pi_2 = \{b_1, \dots, b_q\}$,
- * $\varphi_1 = \varphi$ et $\varphi_2 = \neg\varphi$.

$\{1\}$ est efficace si et seulement s'il existe une stratégie s_1 telle que $s_1 \models \varphi$, c'est-à-dire si et seulement si I est valide. $\{1, 2\}$ ne peut pas être efficace car $\varphi_1 \wedge \varphi_2 = \top$. Donc, 1 appartient à une coalition efficace si et seulement si $\{1\}$ est efficace, c'est-à-dire si et seulement si I est valide.

Pour montrer que décider si un agent i appartient à toute coalition efficace non vide pour G est un problème Π_2^P -difficile, même s'il n'y a que deux agents, considérons la réduction polynomiale de $\text{QBF}_{2,\forall}$ suivante : à chaque instance $I = \forall a_1 \dots a_p \exists b_1 \dots b_q \varphi$ de $\text{QBF}_{2,\forall}$, considérons le jeu booléen $G_I = \langle N, V, \pi, \Gamma, \Phi \rangle$, avec

- * $N = \{1, 2\}$, $\gamma_1 = \gamma_2 = \top$,
- * $V = \{a_1 \dots a_p, b_1 \dots b_q, x\}$,
- * $\pi_1 = \{a_1, \dots, a_p, x\}$, $\pi_2 = \{b_1, \dots, b_q\}$,
- * $\varphi_1 = \neg\varphi$ et $\varphi_2 = \varphi \wedge x$.

Ni $\{2\}$ ni $\{1, 2\}$ ne peuvent être efficaces. Puisque 2 ne peut appartenir à aucune coalition efficace, 2 appartient à toute coalition efficace non vide si et seulement si aucune coalition n'est efficace. Donc, 2 appartient à toute coalition efficace non vide si et seulement si $\{1\}$ n'est pas efficace, c'est-à-dire si $\exists a_1 \dots a_p \forall b_1 \dots b_q \neg\varphi$ n'est pas valide, ou, de façon équivalente, si $\forall a_1 \dots a_p \exists b_1 \dots b_q \varphi$ est valide.

■

5.3.3 Lien avec les graphes de dépendance

Les graphes de dépendance entre les joueurs, introduits dans le chapitre 2, définition 2.9 (page 38), nous permettent de trouver quelques propriétés supplémentaires.

Nous savons déjà que si deux coalitions disjointes I et J sont efficaces, alors leur union sera efficace. La réciproque est fautive dans le cas général : il peut y avoir deux ensembles I et J disjointes tels que $I \cup J$ est efficace, mais ni I ni J ne l'est. Pourtant, cette réciproque est vraie dans ce cas particulier :

Propriété 5.11. *Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen, et I et J deux coalitions telles que¹¹ :*

- * $I \cap J = \emptyset$,
- * $I \cup J$ est efficace,
- * $R(I) \cap J = \emptyset$, et
- * $R(J) \cap I = \emptyset$.

I et J sont alors toutes deux efficaces.

¹¹ $R(I) = \bigcup_{i \in I} R(i)$, avec R la relation de dépendance sur G donnée par la définition 2.9 (page 38).

Preuve : Nous savons que $I \cup J$ est efficace. Donc, il existe $s_{I \cup J} \in S_{I \cup J}$ tel que $s_{I \cup J} \models (\bigwedge_{i \in I \cup J} \varphi_i)$.

Comme $I \cap J = \emptyset$, nous pouvons écrire que :

$$\begin{aligned} & \exists s_I \in S_I, \exists s_J \in S_J : (s_I, s_J) \models (\bigwedge_{i \in I \cup J} \varphi_i) \\ \Leftrightarrow & \exists s_I \in S_I, \exists s_J \in S_J : (s_I, s_J) \models (\bigwedge_{i \in I} \varphi_i \wedge \bigwedge_{j \in J} \varphi_j) \\ \Leftrightarrow & \exists s_I \in S_I, \exists s_J \in S_J : ((s_I, s_J) \models (\bigwedge_{i \in I} \varphi_i)) \wedge ((s_I, s_J) \models (\bigwedge_{j \in J} \varphi_j)) \end{aligned}$$

On sait également que $\forall i \in I, j \in J, j \notin RP_i$ (resp. $i \notin RP_j$). Donc, $\forall i \in I, j \in J, \forall v \in \text{Var}(PI(\varphi_i)), v \notin \pi_j$ (resp. $\forall w \in \text{Var}(PI(\varphi_j)), w \notin \pi_i$). On sait donc qu'aucun joueur de J ne contrôle de variable apparaissant dans le but d'un joueur de I (et vice versa).

Comme nous avons $\exists s_I \in S_I, ((s_I, s_J) \models (\bigwedge_{i \in I} \varphi_i))$ et $\forall i \in I, j \in J, \forall v \in \text{Var}(PI(\varphi_i)), v \notin \pi_j$, nous avons : $s_I \models (\bigwedge_{i \in I} \varphi_i)$ (resp. $s_J \models (\bigwedge_{j \in J} \varphi_j)$).

I et J sont donc tous deux efficaces. ■

Propriété 5.12. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen. Si $B \subseteq N$ est un ensemble stable pour R , alors B est une coalition efficace de G ($B \subseteq \text{EC}(G)$) si et seulement si $\varphi_B = \bigwedge_{i \in B} \varphi_i$ est consistant.

Preuve : Soit B un ensemble stable pour R . Nous avons alors :

$$\begin{aligned} & \forall i \in B, \forall j \text{ tel que } j \in R(i), j \in B \\ \Leftrightarrow & \forall i \in B, RP_i \subseteq B \\ \Rightarrow & \forall i \in B, \exists s_B \in S_B \text{ tel que } s_B \models \varphi_i \\ \Leftrightarrow & \exists s_B \in S_B \text{ tel que } s_B \models \bigwedge_{i \in B} \varphi_i \text{ si et seulement si } \bigwedge_{i \in B} \varphi_i \not\models \perp \end{aligned}$$

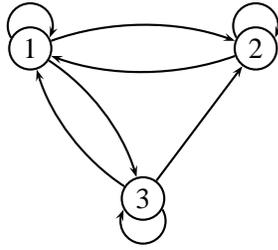
■

La réciproque n'est pas nécessairement vraie, comme on peut le voir sur l'exemple suivant :

Exemple 5.5. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen défini par :

- * $V = \{a, b, c\}$,
- * $N = \{1, 2, 3\}$,
- * $\pi_1 = \{a\}, \pi_2 = \{b\}, \pi_3 = \{c\}$,
- * $\varphi_1 = (a \vee c) \wedge \neg b, \varphi_2 = a \wedge \neg b$ et $\varphi_3 = a \wedge b \wedge c$.

Le graphe de dépendance \mathcal{P} de G est le suivant :



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

La coalition $\{1, 2\}$ est efficace, mais n'est pas stable pour R : $R(\{1, 2\}) = \{1, 2, 3\} \not\subseteq \{1, 2\}$.

Pourtant, cette réciproque peut être vraie sous la condition restrictive disant que la satisfaction des buts des joueurs dépend uniquement des actions d'un seul joueur, c'est-à-dire si RP_i est un singleton pour tout $i \in B$.

Propriété 5.13. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen. Si $B \subseteq N$ est une coalition efficace de G ($B \subseteq EC(G)$) telle que $\forall i \in B, |RP_i| = 1$, alors B est stable pour R .

Dans ce cas, une coalition B telle que $\varphi_B = \bigwedge_{i \in B} \varphi_i \neq \perp$ est efficace si et seulement si B est stable pour R .

Preuve : B est une coalition efficace, donc $\bigwedge_{i \in B} \varphi_i$ est consistant, et $\exists s_B \in S_B$ tel que $s_B \models \bigwedge_{i \in B} \varphi_i$.

On sait que $\forall i \in B, |RP_i| = 1$. Donc, $\exists j \in N$ tel que $RP_i = \{j\}$, i.e. $\forall v \in \text{Var}(PI(\varphi_i)), v \in \pi_j$. Comme nous avons $s_B \models \varphi_i$, avec $s_B \in S_B$, nous savons que B contrôle au moins une variable de φ_i . Donc $j \in B$, et donc B est stable pour R .

■

Dans ce cas particulier où RP_i est un singleton pour tout $i \in B$, nous avons cette caractérisation graphique intuitive des coalitions efficaces :

Propriété 5.14. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen tel que $\forall i \in N, |RP_i| = 1$. Pour toute coalition $C \subseteq N$, C forme un cycle dans le graphe de dépendance si et seulement si C est stable pour R et constitue une coalition efficace minimale.

Preuve :

\Rightarrow Comme $\forall i, |RP_i| = 1$, chaque joueur ne peut avoir qu'un arc sortant dans le graphe de dépendance. Donc, s'il y a un cycle entre p joueurs, et si l'on renomme ces joueurs en respectant l'ordre topologique, nous avons $RP_1 = \{2\}$, $RP_2 = \{3\}$, ..., $RP_{p-1} = \{p\}$, $RP_p = \{1\}$. Soit $C = \{1, \dots, p\}$. Comme nous avons évidemment $R(C) = C$ ($\forall i \in C, RP_i = \{(i+1) \text{ modulo } p\} \in C$), C est stable pour R .

De plus, nous savons que $\forall i, j \in C, RP_i \neq RP_j$, donc $\forall i, j \in C, \varphi_i \wedge \varphi_j \neq \perp$. D'après la propriété 5.12 (page précédente), C est efficace.

Supposons qu'il existe $I \subset C$ qui soit efficace. Il existe donc s_I telle que $s_I \models \bigwedge_{i \in I} \varphi_i$. Comme $|RP_i| = 1, \forall i \in I, RP_i \in I$. Et donc $I = C$. C est une coalition

efficace minimale.

⇐ Si C est stable pour R , alors $\forall i \in C, \exists j \in C$ tel que $RP_i = \{j\}$. Donc, si $C = \{1, \dots, p\}$, on peut renommer ces joueurs de façon à avoir $RP_1 = \{2\}$, $RP_2 = \{3\}$, \dots , $RP_{p-1} = \{p\}$, $RP_p = \{1\}$. C forme alors un cycle dans le graphe de dépendance.

■

Un autre problème intéressant est l'étude des coalitions efficaces dans les jeux booléens dont les buts ont des structures syntaxiques spécifiques. Ainsi, si les buts sont formés de clauses positives (resp. termes positifs), c'est-à-dire si les clauses (resp. termes) ne contiennent que des littéraux positifs, nous avons la caractérisation intuitive suivante :

Propriété 5.15. Soit $G = (N, V, \Gamma, \pi, \Phi)$ un jeu booléen.

1. si pour tout $i \in N$, φ_i est un terme positif, alors pour tout $B \subseteq N$, B est efficace si et seulement si B est stable pour R .
2. si pour tout $i \in N$, φ_i est une clause positive, alors pour tout $B \subseteq N$, B est efficace si et seulement si il existe un cycle dans le graphe de dépendance associé à G donc les nœuds sont exactement les membres de B .

Preuve :

1. ⇒ Soit B un ensemble stable pour R . Comme $\forall i, \varphi_i$ est un terme positif, nous savons que $\bigwedge_{i \in B} \varphi_i$ est consistant. Donc, d'après la propriété 5.12 (page 155), B est une coalition efficace.

⇐ Soit B une coalition efficace. Il existe donc $s_B \in S_B$ telle que $s_B \models \bigwedge_{i \in B} \varphi_i$. Comme $\forall i, \varphi_i = \bigwedge_{v \in Lit(\varphi_i)} v$, $s_B \models \bigwedge_{v \in \bigcup_{i \in B} Lit(\varphi_i)} v$. Donc, $\forall v \in \bigcup_{i \in B} Lit(\varphi_i)$, $v \in \pi_B$, et donc $\forall i \in B, RP_i \subseteq B$. B est stable pour R .

2. Soit $C = \{1, \dots, p\}$ un cycle entre p joueurs. Donc, $\forall i \in C, \exists j \in C$ tel que $j \in RP_i$. Comme $\forall i \in N, \varphi_i = \bigvee_{v \in Lit(\varphi)} v$, nous savons que $\forall i \in C, \exists j \in C, \exists s_j \in S_j$ telle que $s_j \models \varphi_j$.

Comme tous les φ_i sont des clauses positives, on sait également que $\bigwedge_{i \in N} \varphi_i \not\models \perp$. Alors, $\exists s_C \in S_C$ telle que $s_C \models \bigwedge_{i \in C} \varphi_i$.

■

5.4 Travaux connexes

5.4.1 Graphe de dépendance et coalitions admissibles

5.4.1.1 Graphe de dépendance

Dans [Sichman et Conte, 2002], Sichman et Conte ont introduit des graphes de dépendance permettant de représenter des dépendances et/ou sur les actions nécessaires à un agent pour réaliser son but, et sur les agents contrôlant ces actions.

Ces graphes de dépendance sont donc différents de ceux que nous utilisons dans ce manuscrit (définition 2.9 (page 38)) : l'ensemble des nœuds de ce graphe est formé des agents et des actions disponibles, et les arcs sont étiquetés selon les buts à exécuter :

Définition 5.6 ([Sichman et Conte, 2002]). *Un graphe de dépendance est un tuple $\langle N, Ac, \Phi, E \subseteq [N \times Ac \times \Phi] \cup [Ac \times N] \rangle$, où*

- * $N = \{1, \dots, n\}$ est l'ensemble des agents,
- * $Ac = \{a_1, \dots, a_p\}$ est l'ensemble des actions disponibles,
- * $\Phi = \{\phi_1, \dots, \phi_q\}$ est l'ensemble des buts, et
- * E est un ensemble d'arcs étiquetés tels que si $(i, a, \phi) \in E$, alors $(a, i) \notin E$.

Un arc $(i, a, \phi) \in E$ représente le fait que ϕ est un but de l'agent i , que l'action a est nécessaire à i pour atteindre ce but, et que i n'est pas capable d'effectuer cette action.

$(a, i) \in E$ signifie que l'agent i est capable d'effectuer l'action a .

Nous pouvons à présent définir la dépendance basique, puis les dépendances et/ou.

Définition 5.7 ([Sichman et Conte, 2002]). *Soit un graphe de dépendance $\langle N, Ac, \Phi, E \rangle$.*

- * Un agent i **dépend basiquement** d'un agent j pour effectuer une action a , nécessaire pour atteindre le but ϕ , dénoté par $dep_basique(i, j, \phi, a)$, si et seulement si $(i, a, \phi) \in E$ et $(a, j) \in E$.
- * Un agent i **ou-dépend** d'un groupe d'agents Q pour effectuer une action a , nécessaire pour atteindre le but ϕ , dénoté par $dep_OU(i, Q, \phi, a)$, si et seulement si
 1. $|Q| > 1$ et
 2. $Q = \{j \in N \mid dep_basique(i, j, \phi, a)\}$.
- * Un agent i **et-dépend** d'un groupe d'agents $Q \subseteq [2^N \setminus \emptyset]$ pour effectuer l'ensemble d'actions $A \subseteq Ac$, nécessaires pour atteindre le but ϕ , dénoté par $dep_AND(i, Q, \phi, A)$, si et seulement si
 1. $|A| > 1$,
 2. $A = \{a \in Ac \mid (i, a, \phi) \in E\}$ et
 3. $P \in Q$ si et seulement si il existe une action $a \in A$ telle que $P \subseteq \{i \in N \mid (a, i) \in E\}$.

Exemple 5.6 ([Sauro, 2006]). *La figure 5.1 (page suivante) représente les trois types de dépendance que nous venons de définir.*

Le graphe représenté sur la figure (a) représente la dépendance basique. L'agent 1 a besoin d'effectuer l'action a pour accomplir son but ϕ , et pour cela il a besoin de l'agent 2.

Le graphe représenté sur la figure (b) représente la dépendance OU. L'agent 1 dépend basiquement des agents 2 et 3 pour effectuer l'action a lui permettant d'obtenir son but ϕ .

Le graphe représenté sur la figure (c) représente la dépendance ET. L'agent 1 a besoin d'effectuer les actions a et b pour obtenir son but ϕ . Pour l'action a , il OU-dépend des agents 2 et 3 ; et pour l'action b il dépend basiquement de 4.

5.4.1.2 Coalitions admissibles

En suivant les idées de [Sichman et Conte, 2002], Boella, Sauro et van der Torre [Boella *et al.*, 2005b,a; Sauro, 2006; Boella *et al.*, 2006] représentent une coalition par un et-graphe de dépendance

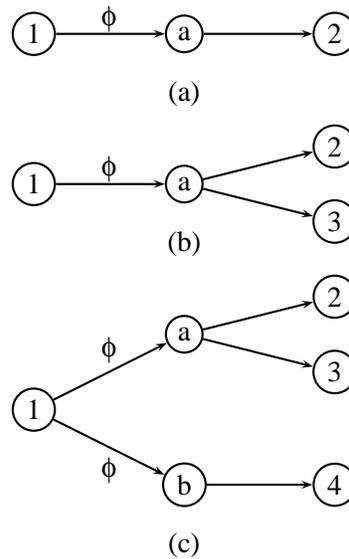


Figure 5.1 — Représentation graphique de la dépendance basique (a), de la dépendance OU (b) et de la dépendance ET (c)

entre agents. L'ensemble de nœuds ϑ de ce graphe est l'ensemble des agents de la coalition, et les arcs ξ sont étiquetés. Si on note Φ l'ensemble des buts de cette coalition, un et-arc étiqueté par $\phi \in \Phi$ d'un agent i vers un ensemble d'agents Q signifie que i désire le but ϕ qui peut être atteint par les agents de Q .

Pour représenter une coalition, le graphe construit de cette façon doit satisfaire deux conditions. La première est que seuls les agents permettant d'obtenir un ou plusieurs buts peuvent être représentés. La seconde est qu'il ne peut y avoir d'agents pouvant satisfaire leur but seuls.

Définition 5.8 ([Boella *et al.*, 2006]). Une **coalition** est représentée par un et-graphe $\mathcal{G} = \langle \vartheta, \xi \rangle$, où ϑ est un ensemble fini de nœuds et $\xi \subseteq \vartheta \times (2^\vartheta \setminus \emptyset) \times \Phi$ est un ensemble d'arcs étiquetés.

\mathcal{G} satisfait deux conditions :

1. pour tout nœud $i \in \vartheta$, il existe au moins un et-arc (j, Q, ϕ) tel que $i \in Q$; et
2. ξ ne contient aucun et-arc de la forme $(i, \{i\}, \phi)$.

Par abus de notation, nous écrivons $(Q, \phi) \in \mathcal{G}$ s'il existe un i tel que $(i, Q, \phi) \in \mathcal{G}$. (Q, ϕ) est un engagement de \mathcal{G} . Une sous-coalition \mathcal{G}' est un sous-graphe de \mathcal{G} dans lequel des engagements de \mathcal{G} sont supprimés : il existe $(Q, \phi) \in \mathcal{G}$ tel que $(Q, \phi) \notin \mathcal{G}'$.

Boella, Sauro et van der Torre ont introduit le critère *do-ut-des* permettant de former des coalitions en formalisant une notion de réciprocité pouvant être décrite informellement par "je donne quelque chose uniquement si je reçois quelque chose en retour", et en permettant d'assurer qu'aucune sous-coalition ne peut se former indépendamment.

Dans [Boella *et al.*, 2005b,a; Sauro, 2006] cette propriété est caractérisée au moyen d'une relation de préférences et d'une notion de dominance. Nous allons donner ici une définition basée sur les propriétés topologiques des chaînes d'échanges issue de [Boella *et al.*, 2006].

Une séquence finie de et-arcs $(i_1, Q_1, \phi_1) \dots (i_p, Q_p, \phi_p)$ est un *chemin* si et seulement si pour tout $2 \geq h \geq p$, $i_h \in Q_{h-1}$. Ces chemins formalisent les chaînes d'échange entre agents. On note $out(i)$

l'ensemble des et-arcs partant de i : $out(i) = \{(i, Q, \phi)\}$. $out^*(i)$ contient tous les chemins partant de i , et n'en contient pas d'autres.

Définition 5.9 ([Boella et al., 2006]). Une coalition $\mathcal{G} = \langle \emptyset, \xi \rangle$ satisfait la propriété **do-ut-des** si et seulement si

1. il n'existe pas deux engagements $(Q, \phi), (Q', \phi) \in \mathcal{G}$ tels que $Q \neq Q'$ (pas de sous-coalition indépendante), et
2. pour tout $(i, Q, \phi) \in \xi$ et pour tout $j \in Q$, $(i, Q, \phi) \in out^*(j)$ (réciprocité).

Exemple 5.7 ([Boella et al., 2006]). Les deux coalitions représentées sur la figure 5.2 (a) et (b) sont do-ut-des.

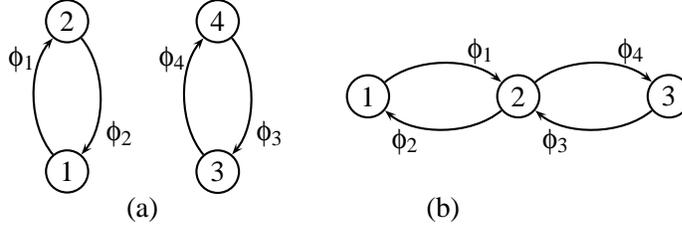


Figure 5.2 — Coalitions satisfaisant le critère do-ut-des

Ces définitions amènent la propriété suivante :

Propriété 5.16. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen, et soit C une coalition telle que $\bigwedge_{i \in C} \varphi_i$ est consistant.

Si la coalition $\mathcal{G} = \langle C, \xi \rangle$ est do-ut-des, avec ξ l'ensemble des arcs (i, RP_i, φ_i) pour tout $i \in C$, alors C est une coalition efficace.

Preuve :

Comme un et-arc dans \mathcal{G} relie un joueur à tous les joueurs qui lui sont utiles pour satisfaire son but, $\mathcal{G} = \langle C, \xi \rangle$ forme un graphe de dépendance sur $C \subseteq N$. (i, RP_i, φ_i) correspond donc à $R(i, j)$ pour tout $j \in RP_i$. Par définition, on a donc $out(i) = R(i)$ et $out^*(i) = R^*(i)$.

Soit $i \in C$. D'après la définition 5.8 (page précédente), nous savons qu'il existe un joueur $j \in C$ tel que $(j, RP_j, \varphi_j) \in \xi$, et $j \in RP_i$. D'après la seconde condition de la définition 5.9, nous savons que $(j, RP_j, \varphi_j) \in out^*(i) = R^*(i)$. Donc $(i, RP_i, \varphi_i) \in \xi$. On sait donc que pour tout $i \in C$, $(i, RP_i, \varphi_i) \in \xi$, et donc que pour tout $k \in RP_i$, $k \in C$. C est donc un ensemble stable, et d'après la propriété 5.12 (page 155) C est une coalition efficace. ■

Par contre, la réciproque est fautive, comme le montre l'exemple suivant :

Exemple 5.8. $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen tel que $N = \{1, 2, 3\}$, $V = \{a, b, c\}$, $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$, pour tout i , $\gamma_i = \top$, $\varphi_1 = a \wedge c$, $\varphi_2 = b \leftrightarrow c$, et $\varphi_3 = b$.

La coalition $C = \{1, 2, 3\}$ est efficace, et pourtant, d'après la définition 5.9, $\mathcal{G} = \langle C, \xi \rangle$, avec $\xi = \{(1, \{1, 3\}, \varphi_1), (2, \{2, 3\}, \varphi_2), (3, \{2\}, \varphi_3)\}$, n'est pas do-ut-des car la réciprocity n'est pas respectée : 1 a besoin de 3 pour satisfaire son but, mais n'aide personne à satisfaire le sien.

Un second critère est introduit dans [Boella *et al.*, 2006] : la propriété *it-dud*, qui consiste en trois conditions :

1. la propriété *do-ut-des* ;
2. pour tout agent $i \in C$, $out^*(i) = \xi$, afin d'empêcher une coalition de se séparer en deux sous-graphes, comme c'est le cas dans la figure 5.2 (page ci-contre)(a) ;
3. enfin, la troisième condition consiste à empêcher que deux sous-coalitions puissent se former plutôt que la grande, comme c'est le cas dans la figure 5.2 (page précédente)(b). Il ne doit donc pas exister un agent i et une bipartition O_1, O_2 de $out(i)$ telle que $O_1^* \cap O_2^*$ est vide.

Définition 5.10 ([Boella *et al.*, 2006]). *Une coalition $\mathcal{G} = \langle \vartheta, \xi \rangle$ satisfait la propriété **it-dud** si et seulement si pour tous les agents $i \in \vartheta$*

1. *il n'existe pas deux engagements $(Q, \phi), (Q', \phi) \in \mathcal{G}$ tels que $Q \neq Q'$,*
2. *$out^*(i) = \xi$, et*
3. *il n'existe pas de bipartition O_1, O_2 de $out(i)$ telle que $O_1^* \cap O_2^* = \emptyset$.*

Comme nous l'avons vu, les deux coalitions de l'exemple 5.2 (page ci-contre) ne satisfont pas cette propriété. Par contre, la coalition représentée sur la figure 5.3 est une coalition *it-dud*.

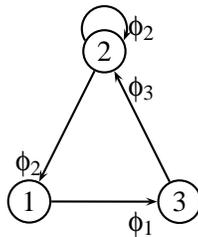


Figure 5.3 — Coalition satisfaisant le critère *it-dud*

Un algorithme permettant de trouver toutes les sous-coalitions d'une coalition C satisfaisant la propriété *it-dud* est introduit dans [Boella *et al.*, 2006; Sauro, 2006].

[Boella *et al.*, 2005b,a; Sauro, 2006; Boella *et al.*, 2006] utilisent les *et-graphes* de dépendance entre agents introduits par [Sichman et Conte, 2002] pour introduire deux critères d'admissibilité pour les coalitions : le critère *do-ut-des* et le critère *it-dud*. Nous avons montré que le critère *do-ut-des*, et donc par définition le critère *it-dud*, est plus restrictif que celui d'efficacité que nous avons introduit dans ce chapitre.

5.4.2 Jeux qualitatifs coalitionnels

Comme les jeux booléens, les jeux qualitatifs coalitionnels (QCG), introduits dans [Wooldridge et Dunne, 2004], sont des jeux dans lesquels les agents n'assignent pas des valeurs d'utilité à chaque issue du jeu, mais sont satisfaits si leurs buts sont atteints. Un QCG est défini de la façon suivante :

Définition 5.11 ([Wooldridge et Dunne, 2004]). *Un jeu coalitionnel qualitatif (QCG) à n joueurs est un $(n + 3)$ -uplet $G = (\Phi, N, \Phi_1, \dots, \Phi_n, W)$, où*

- * $\Phi = \{\phi_1, \dots, \phi_m\}$ est l'ensemble des buts possibles,

- * $N = \{1, \dots, n\}$ est l'ensemble des agents,
- * pour tout $i \in N$, $\Phi_i \subseteq \Phi$ est l'ensemble des buts de l'agent i . i est satisfait si un de ses buts l'est, mais peu importe lequel.
- * $W : 2^N \rightarrow 2^{2^\Phi}$ est une fonction caractéristique qui associe à chaque coalition $C \subseteq N$ un ensemble de choix : si $\Phi' \subseteq W(C)$, alors la coalition C peut obtenir tous les buts appartenant à Φ' simultanément.

Exemple 5.9. Soit un QCG défini par

- * $N = \{1, 2, 3\}$,
- * $\Phi = \{\phi_1, \phi_2, \phi_3\}$,
- * pour tout i , $\Phi_i = \{\phi_i\}$, et
- * la fonction caractéristique W définie pour chaque coalition C par

$$W(C) = \begin{cases} \{\{\phi_1, \phi_2\}, \{\phi_1, \phi_3\}, \{\phi_2, \phi_3\}\} & \text{si } |C| \geq 2 \\ \emptyset & \text{sinon.} \end{cases}$$

Ainsi définis, les QCGs permettent de modéliser des problèmes courants en théorie des jeux coalitionnels, et de calculer la complexité de problèmes associés. Le problème de l'existence d'une coalition fructueuse (c'est-à-dire une coalition ayant la possibilité d'atteindre les objectifs de tous ses membres), ou celui de la vacuité du noyau sont étudiés dans [Wooldridge et Dunne, 2004] en utilisant ainsi les QCGs.

Une première différence entre QCG et jeux booléens est qu'il n'y a aucune fonction d'affectation de contrôle dans les QCG : la fonction caractéristique permet de savoir quels buts peut atteindre une coalition donnée, mais ne donne pas les "capacités" de chaque joueur en particulier.

Une seconde est que chaque agent dans un QCG peut avoir un ensemble de buts, et être satisfait si au moins un de ses buts est satisfait, tandis que dans un jeu booléen classique chaque agent n'a qu'un seul but. Cette différence peut pourtant être contournée : si par exemple les buts d'un joueur i dans un QCG sont $\Phi_i = \{\phi_1, \phi_2\}$, ils peuvent être représentés par $\varphi_i = \phi_1 \vee \phi_2$ dans un jeu booléen. Comme dans un QCG, on aura alors que le joueur i est satisfait si le but ϕ_1 ou le but ϕ_2 est satisfait, et peu importe lequel.

Enfin, les fonctions caractéristiques des QCGs peuvent ne pas être monotones, contrairement aux fonctions d'utilité dans les jeux booléens. Il est cependant possible de représenter un QCG ayant une fonction caractéristique monotone avec un jeu booléen, comme nous le montrons sur l'exemple 5.9.

Exemple 5.9, suite : Ce QCG peut être représenté par un jeu booléen à trois joueurs, tels que les buts de ces joueurs sont les buts ϕ_1 , ϕ_2 et ϕ_3 . Il faut ensuite écrire ces buts sous forme de formules propositionnelles et faire en sorte qu'ils respectent la fonction caractéristique du QCG.

Ce QCG peut être donc représenté par exemple par le jeu booléen $(N, V, \pi, \Gamma, \Phi)$, avec

- * $V = \{a, b, c, d, e\}$,
- * $\pi_1 = \{a, b\}$, $\pi_2 = \{c, d\}$, $\pi_3 = \{e\}$,
- * pour tout i , $\gamma_i = \top$,
- * $\varphi_1 = \phi_1 = (a \leftrightarrow (b \leftrightarrow c)) \vee (c \leftrightarrow (d \leftrightarrow e)) \vee (a \leftrightarrow (b \leftrightarrow e))$,
- * $\varphi_2 = \phi_2 = (a \wedge b \wedge c) \vee (a \neg b \wedge c) \vee (c \wedge d \wedge e) \vee (c \neg d \wedge e) \vee (a \wedge b \wedge e) \vee (a \neg b \wedge e)$,

$$* \varphi_3 = \phi_3 = (a \wedge \neg b \wedge c) \vee (\neg a \wedge b \wedge \neg c) \vee (c \wedge \neg d \wedge e) \vee (\neg c \wedge d \wedge \neg e) \vee (a \wedge \neg b \wedge e) \vee (\neg a \wedge b \wedge \neg e).$$

Dans ce jeu, on voit qu'aucun joueur seul ne peut atteindre son but, mais que les coalitions $\{1,2\}$, $\{1,3\}$, $\{2,3\}$ et $\{1,2,3\}$ peuvent atteindre ensemble les buts $\phi_1 \wedge \phi_2$, $\phi_1 \wedge \phi_3$ et $\phi_2 \wedge \phi_3$. La fonction caractéristique W est donc bien respectée.

De même, un jeu booléen peut être représenté par un QCG :

Exemple 5.1 (page 143), suite : *Le jeu booléen G peut être représenté par un QCG à 3 joueurs. La fonction caractéristique de ce QCG doit associer à chaque groupe de joueurs les buts qu'ils peuvent atteindre.*

Ainsi, nous pouvons représenter cet exemple par la fonction caractéristique W suivante :

$$\begin{aligned} * W(\{2\}) &= \emptyset, \\ * W(\{1\}) &= W(\{3\}) = W(\{1,3\}) = \{\phi_2\}, \\ * W(\{1,2\}) &= \{\phi_1, \phi_2\}, \\ * W(\{2,3\}) &= \{\phi_2, \phi_3\}, \\ * W(\{1,2,3\}) &= \{\{\phi_1, \phi_2\}, \{\phi_2, \phi_3\}\}. \end{aligned}$$

Les QCGs ont été étendus de deux façons distinctes.

Dans [Dunne et Wooldridge, 2004], une relation de préférences est associée aux buts de chaque joueur, permettant ainsi de définir des *jeux coalitionnels avec préférences* (QCGPs). Ainsi, comme dans un QCG, un ensemble de buts est associé à chaque joueur, mais, dans un QCGP, ce joueur a des préférences sur ces buts. L'ajout de ces préférences permet de définir des concepts classiques de la théorie des jeux coalitionnels, comme la notion de noyau ou d'ensemble stable, et de trouver des propriétés permettant de caractériser ces concepts dans des QCGPs.

D'autre part, il est montré dans [Dunne *et al.*, 2007] qu'il est possible de raisonner sur les QCGs en utilisant la logique des coalitions de [Pauly, 2001]. Une relation de correspondance entre les QCGs et les interprétations de la logique des coalitions est introduite, permettant ainsi de définir sous quelles circonstances ces interprétations permettent de caractériser correctement un QCG.

Comme les jeux booléens, un QCG est un jeu dans lequel les joueurs sont satisfaits si leur but est atteint. Nous avons montré que si la fonction caractéristique d'un QCG est monotone, il est possible de traduire ce QCG en un jeu booléen, et vice-versa.

5.4.3 Logique des jeux coalitionnels

Dans [Agotnes *et al.*, 2006], Agotnes, van der Hoek et Wooldridge ont développé une logique pour représenter et raisonner sur les jeux coalitionnels sans utilités transférables. Contrairement à la logique des coalitions [Pauly, 2001], la logique des jeux coalitionnels (Coalitional Game Logic - CGL) introduite dans [Agotnes *et al.*, 2006] permet de représenter les préférences des joueurs, et établit un lien direct entre les formules de la logique et les propriétés des jeux coalitionnels.

Définition 5.12 ([Agotnes *et al.*, 2006]). *Un jeu coalitionnel sans utilités transférables à n joueurs est un $(n+3)$ -uplet $G = (N, \Omega, W, \succeq_1, \dots, \succeq_n)$, où*

$$* N = \{1, \dots, n\} \text{ est un ensemble non vide d'agents,}$$

- * $\Omega = \{\omega_1, \dots, \omega_m\}$ est un ensemble non vide d'issues du jeu,
- * $W : (2^N \setminus \emptyset) \rightarrow 2^\Omega$ est une fonction caractéristique qui associe à chaque coalition $C \subseteq N$ un ensemble de choix : si $\omega \subseteq W(C)$, alors la coalition C peut atteindre l'issue ω du jeu.
- * pour tout agent $i \in N$, $\succeq_i \subseteq \Omega \times \Omega$ est une relation de préférences complète, réflexive et transitive.

Agotnes et collègues caractérisent ensuite quelques concepts de solution de la théorie des jeux coalitionnels dans leur logique, comme le noyau.

Une des principales différences entre jeux coalitionnels et jeux booléens est, comme avec les QCGs, le fait que la fonction caractéristique permet de savoir quels buts peut atteindre une coalition donnée, mais ne donne pas les “capacités” de chaque joueur en particulier. On remarque aussi ici que la fonction caractéristique peut être non-monotone, contrairement aux fonctions d'utilité des jeux booléens.

En Section 2.2 (page 33), nous avons mentionné un lien avec le problème de contrôlabilité en logique propositionnelle étudié par Boutilier [Boutilier, 1994] et Lang et al [Lang et Marquis, 1998b]. Une récente ligne de travail dans ce domaine [van der Hoek et Wooldridge, 2005] étudie une logique de coopération dans laquelle on suppose que chaque agent contrôle un ensemble de variables propositionnelles. Alors que notre travail a consisté à enrichir les jeux booléens en se concentrant sur les préférences et les concepts de solution classiques, van der Hoek et Wooldridge [van der Hoek et Wooldridge, 2005] s'intéressent au pouvoir réel des agents : ils raisonnent sur l'ensemble des états qu'un groupe d'agents peut atteindre.

Conclusion

Dans un système multi-agents, plusieurs agents rationnels interagissent afin de satisfaire au mieux leurs préférences. Notre objectif au long de ce travail était de spécifier de façon concise et efficace les interactions entre agents rationnels dans de tels systèmes. Pour cela, nous avons choisi de nous appuyer sur la théorie des jeux, qui est un modèle formel abouti pour l'étude de ces interactions. Une des principales lacunes de cette théorie est la façon de représenter les utilités des joueurs, coûteuse en place mémoire et en temps d'exécution. Nous avons vu qu'une solution pour pallier ces problèmes est d'utiliser un langage de représentation compacte de préférences. Nous avons commencé par choisir de représenter les préférences des joueurs de façon ordinale, ce qui nous a mené à la logique propositionnelle. En effet, utiliser la logique propositionnelle pour représenter les préférences des joueurs permet non seulement de simplifier les jeux, mais aussi d'utiliser les propriétés et les outils bien connus de cette logique pour étudier les caractéristiques de tels jeux. Nous avons donc choisi d'étudier le cas où chaque agent contrôle un ensemble fini de variables binaires. Ces jeux, appelés *jeux booléens*, ont été introduits par [Harrenstein *et al.*, 2001; Harrenstein, 2004a; Dunne et van der Hoek, 2004].

Les jeux booléens permettent de modéliser de manière compacte les préférences des joueurs dans le cadre de jeux statiques. Toutefois, ces jeux booléens, tels qu'ils ont été introduits par [Harrenstein *et al.*, 2001; Harrenstein, 2004a; Dunne et van der Hoek, 2004], correspondent à une spécification très particulière (2 joueurs, jeux à somme nulle, utilités binaires). Nous avons donc étendu ces jeux sur plusieurs points :

Jeux booléens à n joueurs et somme non nulle : Nous avons tout d'abord généralisé les jeux booléens de manière à représenter des jeux avec un nombre arbitraire de joueurs et à somme non nulle, mais en gardant l'hypothèse que les préférences de chaque joueur sont représentées par une formule propositionnelle unique, ce qui ne permet de représenter que des utilités binaires. Nous avons ensuite vu que des outils simples issus de la logique propositionnelle permettent de caractériser équilibres de Nash en stratégies pures et stratégies dominées. Nous avons ainsi établi un lien entre les jeux booléens (étendus ou pas) et les notions d'impliquants et d'impliquants premiers (notions bien connues de tous ceux qui s'intéressent au problème de la satisfiabilité en logique classique). Nous avons également identifié la complexité algorithmique de quelques problèmes liés aux jeux booléens.

Bien entendu, les jeux booléens évoqués dans [Harrenstein *et al.*, 2001; Harrenstein, 2004a] peuvent être vus comme des cas particuliers de ces "jeux booléens étendus".

Introduction de préférences non dichotomiques : Ce choix d'utilités binaires (pour lequel les

agents peuvent seulement exprimer leur totale satisfaction ou leur total mécontentement, sans niveaux intermédiaires) étant une vraie perte de généralité, nous avons incorporé des préférences non dichotomiques dans les jeux booléens afin que chaque joueur puisse représenter ses préférences de manière plus souple. Nous avons vu qu'exprimer des préférences non binaires dans un cadre propositionnel est possible en utilisant un langage de représentation compacte de préférences. Nous avons choisi de nous restreindre aux préférences ordinales, et nous avons intégré deux de ces langages aux jeux booléens : les *CP-nets* et les *buts à priorité*. Nous avons ensuite calculé les propriétés des équilibres de Nash en stratégies pures et stratégies dominées dans ces nouveaux jeux, et identifié la complexité algorithmique de quelques problèmes.

Graphe de dépendances entre joueurs : Nous avons également montré que la notion intuitive de dépendance entre joueurs dans un jeu booléen permet de donner une caractérisation simple des équilibres de Nash en stratégies pures. De plus, nous avons montré que ces propriétés sont valables non seulement dans le cas des jeux booléens avec des préférences dichotomiques, mais aussi pour des jeux booléens généralisés, dans lesquels les préférences sont représentées par un langage de représentation compacte.

Coalitions efficaces : Nous avons prouvé que les jeux booléens peuvent être employés pour représenter compactement des jeux coalitionnels pour lesquels les joueurs ont des préférences dichotomiques. Cette spécificité nous a amené à définir la notion intéressante des coalitions efficaces. Nous avons donné une caractérisation exacte des ensembles de coalitions qui correspondent à l'ensemble de coalitions efficaces pour un jeu booléen, et nous avons donné plusieurs résultats sur le calcul de ces coalitions efficaces. Notons que certaines de nos notions et de nos résultats ne s'appuient pas explicitement sur l'utilisation de la logique propositionnelle. Par exemple, les coalitions efficaces peuvent être définies dans un cadre plus général dans lequel les buts sont simplement exprimés en tant qu'ensembles non vides d'états. Cependant, beaucoup de notions (en particulier, la fonction π d'assignement de contrôle) deviennent beaucoup moins claires lorsqu'on ne prend pas en compte la représentation propositionnelle.

Une limitation de nos résultats sur les coalitions est qu'ils s'appliquent uniquement à des préférences dichotomiques. Cependant, comme nous l'avons illustré sur l'exemple 5.2 (page 143), il existe des problèmes qui s'expriment avec des buts dichotomiques.

Ce travail, encore très préliminaire, ouvre de nombreuses perspectives, que nous allons présenter à présent.

Introduction de langages d'action et partage de variables

Les seules actions disponibles dans un jeu booléen consistent à affecter une valeur à un ensemble de variables données. C'est une perte de généralité : en effet on ne peut représenter des actions ayant des préconditions nécessaires pour pouvoir être accomplies (par exemple, pour pouvoir lire la nuit, il faut que la lumière soit allumée), ou encore des actions ayant un effet sur l'instanciation d'autres variables (par exemple si j'éteins la lumière, alors je ne peux plus lire la nuit). Il serait donc utile d'avoir un langage permettant de représenter des actions plus sophistiquées.

Une première idée pour cela est de modéliser des jeux à l'aide d'un langage d'actions déterministes. Dans ce cas, chaque joueur n'aura plus un ensemble de variables propositionnelles à mettre à vrai ou à faux, mais un ensemble d'actions déterministes à accomplir, qui modifieront l'instanciation des variables propositionnelles. Un langage d'action sert à décrire les effets qu'ont les actions sur les fluents¹². Il existe de nombreux langages d'action, comme par exemple le langage A , introduit dans [Gelfond et Lifschitz, 1993], ou le langage C , introduit dans [Giunchiglia et Lifschitz, 1998].

Pour pouvoir appliquer des actions dans un jeu booléen, l'état du monde (les valeurs que prennent les variables du jeu) a besoin d'être défini au début du jeu. On doit donc introduire l'état initial dans la définition d'un jeu booléen. De plus, les joueurs ne contrôleront plus les variables du jeu, mais les actions.

Définition 1. Soit un ensemble V de variables propositionnelles, $N = \{1, 2, \dots, n\}$ un ensemble de joueurs, Ac un ensemble d'actions, $\Gamma = \{\gamma_1, \dots, \gamma_n\}$ un ensemble de contraintes, une fonction d'assignement de contrôle $\pi : N \rightarrow Ac$, $\Phi = \{\phi_1, \dots, \phi_n\}$ l'ensemble des formules représentant les buts des joueurs, où chaque ϕ_i est une formule de L_V , et E_0 l'état initial, c'est-à-dire l'instanciation de chaque variable de V au début du jeu.

Le jeu booléen avec actions correspondant est alors représenté par le tuple $(N, V, Ac, \Gamma, \pi, \Phi, E_0)$.

Dans le contexte des jeux booléens, l'ensemble des fluents¹³ peut être défini comme étant l'ensemble des instantiations possibles des variables de V . Les jeux booléens étant statiques, les joueurs jouent simultanément. Il faut donc étudier l'effet des actions jouées au même instant.

Plusieurs cas sont alors possibles : les actions peuvent être déterministes sans problème d'interaction (auquel cas deux actions ne peuvent pas influencer sur la même variable) ou déterministes avec problèmes d'interaction (auquel cas deux actions (ou plus) contrôlées par différents joueurs peuvent influencer sur la même variable du jeu). Lorsque les actions sont déterministes sans problème d'interaction, il est possible de raisonner exactement comme pour un jeu booléen classique, sous réserve que l'on connaisse l'état initial du jeu.

Par contre, si les actions sont déterministes avec des problèmes d'interaction, une variable peut être modifiée par deux actions différentes. Supposons par exemple qu'une lampe éteinte soit contrôlée par deux interrupteurs. Si l'on active exactement au même instant ces deux interrupteurs, il est impossible de prévoir à l'avance quel sera le résultat de ces deux actions : la lampe sera-t-elle toujours éteinte, ou sera-t-elle allumée ?

Cette situation est identique à celle consistant à étudier le problème de la gestion de variables partagées, c'est-à-dire de variables contrôlées par plusieurs joueurs dans le cas où les actions ne sont pas nécessairement sophistiquées.

Exemple 1. *Cédric et Caroline sont invités à une soirée chez des amis avec leur fille Jade. Caroline aimerait bien que Cédric aille à cette soirée avec Jade afin de profiter d'une soirée tranquille chez elle. Cédric lui aimerait confier Jade à une baby-sitter afin de profiter calmement de cette soirée avec Caroline et leurs amis. Jade n'a qu'un an et n'a pas le choix, elle fait ce que ses parents décident pour elle.*

Cette situation peut être modélisée par le jeu booléen $G = (N, V, \pi, \Gamma, \Phi)$ suivant :

¹²Un fluent est "quelque chose" qui peut dépendre de la situation, comme par exemple l'endroit où se trouve un objet que l'on peut déplacer. Un fluent propositionnel est une assertion qui peut être vraie ou fausse selon la situation.

¹³Les fluents utilisés ici sont des fluents propositionnels.

- * $N = \{1, 2\}$, Caroline étant le joueur 1, et Cédric le joueur 2,
- * $V = \{a, b, c, d\}$, a signifiant que Caroline va à la fête, b que Cédric y va, c que Jade y accompagne au moins un de ses parents, et enfin d que Jade est confiée à une babysitter.
- * $\pi_1 = \{a, c, d\}$, $\pi_2 = \{b, c, d\}$, (chacun des parents décide pour lui-même, mais ils peuvent tous deux décider pour leur fille)
- * pour tout i , $\gamma_i = \{c \rightarrow (a \vee b), (a \wedge b \wedge \neg c) \rightarrow d\}$ (Jade ne peut aller à la fête que si un de ses parents y va, et est confiée à une baby-sitter si ses deux parents y vont sans elle.),
- * $\varphi_1 = \neg a \wedge b \wedge c \rightarrow d$ et $\varphi_2 = a \wedge b \wedge \neg c \wedge d$.

Comment savoir quelle sera l'instanciation des variables c et d , c'est-à-dire si Jade ira à la fête ou si elle sera confiée à une baby-sitter ?

Le jeu est statique, les joueurs instancient leurs variables en même temps. Comment savoir alors lequel a instancié la variable ? Cette question est similaire à celle posée lors de l'introduction d'actions déterministes avec problèmes d'interaction.

Il serait donc intéressant d'étudier quels langages d'actions peuvent être utilisés dans un jeu booléen, et comment les introduire dans ces jeux de façon formelle. Ensuite, il serait intéressant d'étudier comment raisonner sur des jeux booléens ayant des actions déterministes avec problèmes d'interaction, et sur des jeux ayant des variables partagées.

Pour cela, une idée à exploiter serait de considérer que l'on ne connaît pas la valeur des variables partagées (ou de celles modifiées par plusieurs actions concurrentes), et de voir les résultats que l'on obtient ainsi. Une seconde serait d'introduire un joueur pour chaque variable partagée (modifiée par des actions concurrentes) qui contrôlerait cette variable, et dont le but serait la conjonction des buts des joueurs contrôlant cette variable si ces buts sont consistants, et leur disjonction sinon.

Relations d'inférences et jeux booléens

Une autre piste de travail consiste à voir un jeu booléen comme étant une base de connaissances (en assimilant connaissances et profils de stratégies), et d'essayer de trouver de nouvelles connaissances - ou les connaissances les plus pertinentes de ce jeu - à partir d'inférences non monotones. Pour cela, il faut définir des relations d'inférences non monotones à partir de ces jeux booléens étendus, sur le modèle de celles définies à partir des jeux booléens classiques par [Harrenstein, 2004b] .

Si $G = (N, V, \pi, \Gamma, \Phi)$ est un jeu booléen, et si PNE est l'ensemble des équilibres de Nash en stratégies pures de G , il est possible de définir les relations d'inférence non monotone suivantes :

UNI : $G \sim_{UNI} \varphi$ si et seulement si $\forall s \in PNE, s \models \varphi$.

EXI : $G \sim_{EXI} \varphi$ si et seulement si $\exists s \in PNE, s \models \varphi$.

ARG : $G \sim_{ARG} \varphi$ si et seulement si $\exists s \in PNE, s \models \varphi$ et $\forall s \in PNE, s \not\models \neg \varphi$.

Exemple 2. Soit $G = (N, V, \pi, \Gamma, \Phi)$ un jeu booléen tel que :

- * $V = \{a, b, c\}$, $N = \{1, 2, 3\}$,
- * $\pi_1 = \{a\}$, $\pi_2 = \{b\}$, $\pi_3 = \{c\}$,
- * pour tout i , $\gamma_i = \top$,
- * $\varphi_1 = \neg a \vee (a \wedge \neg c)$, $\varphi_2 = a \leftrightarrow (b \leftrightarrow c)$ et $\varphi_3 = (a \wedge \neg b \wedge \neg c) \vee (\neg a \wedge b \wedge c)$.

stratégie de 3 : c			stratégie de 3 : \bar{c}		
1 \ 2	b	\bar{b}	1 \ 2	b	\bar{b}
a	(0, 1, 0)	(0, 0, 0)	a	(1, 0, 0)	(1, 1, 1)
\bar{a}	(1, 0, 1)	(1, 1, 0)	\bar{a}	(1, 1, 0)	(1, 0, 0)

Ce jeu a deux équilibres de Nash en stratégies pures : $\{\bar{a}\bar{b}\bar{c}, \bar{a}bc\}$.

On a donc :

- * $G \sim_{UNI} \neg b$,
- * $G \sim_{EXI} a, G \sim_{EXI} \neg a, G \sim_{EXI} \neg b, G \sim_{EXI} \neg c, G \sim_{EXI} c$,
- * $G \sim_{ARG} \neg b$.

Il serait intéressant d'étudier les propriétés de déduction [Kraus *et al.*, 1990; Gärdenfors et Markinson, 1994] de ces relations d'inférences non monotones, afin de voir si elles permettent d'obtenir des résultats intéressants. Par exemple, ces relations de conséquences permettent-elles de sélectionner un "meilleur" équilibre de Nash s'il en existe plusieurs ? Ou une "meilleure" stratégie dominée ?

Argumentation

Une autre piste de recherche consiste à utiliser les jeux booléens pour résoudre des problèmes d'argumentation, et vice versa. La première idée serait de concevoir les arguments avancés par les individus comme étant les "coups" d'un joueur dans un jeu stratégique. Cette idée a été introduite par [Dung, 1995], et a été étudiée par [Rubinstein, 2000] et par [Sireyjol, 2004].

Cette démarche se base sur l'aspect dynamique du jeu. Or, nous avons étudié jusqu'à présent quasiment exclusivement des jeux statiques. Dans notre cadre de travail, il pourrait donc être intéressant de raisonner de manière statique sur le graphe des arguments d'un système d'argumentation avec des outils de la théorie des jeux, et notamment ceux spécifiques aux jeux booléens. Pour cela, il faut commencer par traduire un système d'argumentation en un jeu booléen.

Commençons par rappeler comment se compose un système d'argumentation [Simari et Loui, 1992; Cayrol, 1995; Dung, 1995]. Une des définitions est la suivante¹⁴ :

Définition 2. *Un système d'argumentation SA est défini par :*

- * K : ensemble de connaissances
- * Ass : ensemble d'hypothèses (littéraux)
- * \mathcal{A} : ensemble d'arguments, de la forme $\{a_1, \dots, a_n\}$, avec $\forall a_i, a_i = \langle \Xi, \varphi \rangle$, où Ξ est le support de a_i et φ sa conclusion, tel que :
 - (1) $\Xi \subseteq Ass$,
 - (2) $K \cup \Xi \not\models \perp$,
 - (3) $K \cup \Xi \models \varphi$, et

¹⁴Il existe de nombreuses autres manières de définir un système d'argumentation.

Ξ minimal pour l'inclusion parmi les ensembles vérifiant (1), (2) et (3)

* $\mathcal{R}: \mathcal{A} \rightarrow \mathcal{A}$ une relation d'attaque telle que :

* soit $\mathcal{R} = \text{undercut}$:

$\langle \Xi_1, \varphi_1 \rangle \text{undercut} \langle \Xi_2, \varphi_2 \rangle$ si et seulement si $\exists \varphi \in \Xi_2$ et $\varphi_1 \equiv \neg \varphi$.

* soit $\mathcal{R} = \text{rebut}$:

$\langle \Xi_1, \varphi_1 \rangle \text{rebut} \langle \Xi_2, \varphi_2 \rangle$ si et seulement si $\varphi_1 \equiv \neg \varphi_2$ (rebut est une relation symétrique).

Une idée pour traduire un système d'argumentation en jeu booléen est de considérer chaque argument comme étant un joueur ayant deux buts : le but principal est la conclusion de l'argument, et le but secondaire est le support de l'argument. Il va donc falloir utiliser des buts à priorité. Il faut également prendre en compte l'ensemble K des connaissances du SA. Pour cela on introduit un ensemble de contraintes globales C dans un jeu booléen, telles que :

- * Un profil de stratégies s est cohérent si et seulement si $s \cup C \not\models \perp$,
- * Un profil de stratégies s ne peut être un équilibre de Nash fort ou faible en stratégies pures que si s est cohérent,
- * Un profil de stratégies s satisfait une formule φ si et seulement si $s \cup C \models \varphi$,

Un système d'argumentation SA se traduit alors en un jeu booléen $G = (N, V, \pi, \Gamma, C, \Phi)$ de la manière suivante :

- * $N = \{1, \dots, n\}$ avec $n = |\mathcal{A}|$ (un joueur dans un jeu booléen pour chaque argument dans un système d'argumentation) ;
- * $C = K$;
- * $V = \{v \mid \text{Var}(v) \in \text{Ass}\}$ (l'ensemble des variables du jeu est l'ensemble des variables contenues dans l'ensemble des hypothèses du SA (Ass contenant des littéraux)) ;
- * pour tout i , $\pi_i = \{v \mid \text{Var}(v) \in \Xi_i\}$;
- * pour tout i , $\gamma_i = \top$ et
- * $\Sigma_i = \langle \varphi_i; \Xi_i \rangle$

S'il existe plusieurs arguments (a_1, \dots, a_p) tels qu'il existe une variable $v \in \Xi_1 \cap \dots \cap \Xi_p$, il faut réécrire le SA avant de le traduire afin d'éviter qu'une variable ne soit contrôlée par plusieurs joueurs. Cette réécriture peut être faite de la manière suivante :

- * pour chaque a_i , v est renommée en v_i ,
- * v est remplacée par $v_1 \wedge \dots \wedge v_p$ dans tous les φ_i dans lesquels v apparaît,
- * les contraintes suivantes sont ajoutées à K : $(v_1 \wedge \dots \wedge v_p) \vee (\neg v_1 \wedge \dots \wedge \neg v_p)$ et $v \leftrightarrow (v_1 \wedge \dots \wedge v_p)$.

Cette transformation permet de conserver le fait que chaque variable est contrôlée par un et un seul joueur.

Exemple 3. Soit le système d'argumentation SA suivant :

- * $K = \{(\neg \text{ACK} \wedge \text{IMP}) \rightarrow \text{PUB}, \text{PRIV} \rightarrow \text{ACK}\}$,
- * $\text{Ass} = \{\neg \text{ACK}, \text{IMP}, \text{PRIV}\}$,
- * $\mathcal{A} = \{a_1, a_2\}$, avec $a_1 = \langle \{\neg \text{ACK}, \text{IMP}\}, \text{PUB} \rangle$ et $a_2 = \langle \{\text{PRIV}\}, \text{ACK} \rangle$.

Ce SA peut être traduit dans le jeu booléen $G = (N, V, \pi, \Gamma, C, \Phi)$ suivant :

- * $N = \{1, 2\}$,
- * $C = \{(\neg ACK \wedge IMP) \rightarrow PUB, PRIV \rightarrow ACK\}$,
- * $V = \{ACK, IMP, PRIV\}$,
- * $\pi_1 = \{ACK, IMP\}$,
- * $\pi_2 = \{PRIV\}$,
- * *pour tout i , $\gamma_i = \top$,*
- * $\Sigma_1 = \langle PUB; \neg ACK \wedge IMP \rangle$, *et*
- * $\Sigma_2 = \langle ACK; PRIV \rangle$

Il serait intéressant de voir par exemple si cette traduction permet de retrouver les attaques d'un système d'argumentation à partir d'un jeu booléen, ou encore de retrouver les extensions d'arguments (cf. [Dung, 1995]) à partir d'un jeu booléen, et de faire des raisonnements sur la traduction inverse (d'un jeu booléen à un système d'argumentation).

Bibliographie

- Joseph ABDOU et Hans KEIDING : *Effectivity functions in social choice*. Kluwer, 1991.
- David ABRAHAM, Avrim BLUM et Tuomas SANDHOLM : Clearing Algorithms for Barter Exchange Markets : Enabling Nationwide Kidney Exchange. *Dans Proceedings of Conference on Electronic Commerce (EC'07)*, 2007.
- Thomas AGOTNES, Wiebe van der HOEK et Michael WOOLDRIDGE : On the Logic of Coalitional Games. *Dans Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'06)*, pages 153–160. ACM Press, 2006.
- Krzysztof R. APT, Fancesca ROSSI et Kristen Brent VENABLE : CP-nets and Nash equilibria. *Dans Proceedings of Third International Conference on Computational Intelligence, Robotics and Autonomous Systems (CIRAS'05)*, Singapore, December 13-16 2005. Elsevier.
- Robert J. AUMANN : Acceptable Points in General n-person Games. *Dans A.W. TUCKER et R.D. LUCE, éditeurs : Contributions to the Theory of Games IV*. Princeton University Press, 1959.
- Robert J. AUMANN : A Survey of Cooperative games without Side payments. *Dans M. SHUBIK, éditeur : Essays in Mathematical Economics in Honour of Oskar Morgenstern*. Princeton University Press, 1967.
- Salem BENFERHAT, Claudette CAYROL, Didier DUBOIS, Jérôme LANG et Henri PRADE : Inconsistency management and prioritized syntax-based entailment. *Dans BAJCSY, éditeur : Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI'93)*, pages 640–645, Chambéry, France, 1993.
- Salem BENFERHAT, Didier DUBOIS, Souhila KACI et Henri PRADE : Possibilistic representation of preference : Relating prioritized goals and satisfaction levels expressions. *Dans Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR'02)*, pages 158–169, 2002.
- Salem BENFERHAT, Didier DUBOIS et Henri PRADE : Towards a possibilistic logic handling of preferences. *Applied Intelligence*, 14:303–317, 2001.
- Philippe BESNARD, Jérôme LANG et Pierre MARQUIS : Variable Forgetting in Preference Relations over Propositional Domains. *Dans Proceedings of the 17th European Conference on Artificial Intelligence (ECAI'06)*, pages 763–764. Springer-Verlag, 2006.

- Navin BHAT et Kevin LEYTON-BROWN : Computing Nash equilibria of Action-Graph Games. *Dans Conference on Uncertainty in Artificial Intelligence (UAI'04)*, pages 35–42, 2004.
- Guido BOELLA, Luigi SAURO et Leendert van der TORRE : Admissible agreements among goal-directed agents. *Dans Proceedings of the International Conference on Intelligent Agent Technology (IAT'05)*, pages 543–554. IEEE, 2005a.
- Guido BOELLA, Luigi SAURO et Leendert van der TORRE : Reducing Coalition Structure via Agreement Specification. *Dans Proceedings of the Fourth International Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)*, 2005b. Poster.
- Guido BOELLA, Luigi SAURO et Leendert van der TORRE : Strengthening Admissible Coalitions. *Dans 17th European Conference on Artificial Intelligence (ECAI'06)*, 2006.
- Elise BONZON, Marie Christine LAGASQUIE-SCHIEX et Jérôme LANG : Jeux booléens statiques et représentation compacte de préférences : rapport préliminaire. *Dans* Andreas HERZIG, Yves LESPÉRANCE et Abdel-Ilhah MOUADDIB, éditeurs : *Modèles Formels de l'Interaction (MFI'05)*, pages 55–64. Editions Cépaduès, 25-27 mai 2005.
- Elise BONZON, Marie-Christine LAGASQUIE-SCHIEX et Jérôme LANG : Compact preference representation for Boolean games. *Dans Proceedings of the 9th Pacific Rim International Conference on Artificial Intelligence (PRICAI'06)*, volume 4099, pages 41–50. Springer-Verlag, 2006a.
- Elise BONZON, Marie-Christine LAGASQUIE-SCHIEX et Jérôme LANG : Dependencies between players in Boolean games. *Dans Proceedings of European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'07)*, pages 743–754. Springer, 2007a.
- Elise BONZON, Marie-Christine LAGASQUIE-SCHIEX et Jérôme LANG : Efficient coalitions in Boolean games. Research report IRIT/RR–2007-, IRIT, 2007b.
- Elise BONZON, Marie-Christine LAGASQUIE-SCHIEX et Jérôme LANG : Efficient coalitions in Boolean games. *Dans* Krzysztof APT et Robert van ROOIJ, éditeurs : *Texts in Logic and Games*, volume 5, chapitre New Perspectives on Games and Interaction, pages 283–297. Amsterdam University Press, 2008.
- Elise BONZON, Marie-Christine LAGASQUIE-SCHIEX, Jérôme LANG et Bruno ZANUTTINI : Boolean games revisited. *Dans Proceedings of the 17th European Conference on Artificial Intelligence (ECAI'06)*, pages 265–269. Springer-Verlag, 2006b.
- Craig BOUTILIER : Toward a logic for qualitative decision theory. *Dans Proceedings of International Conference on Principles of Knowledge Representation and Reasoning (KR'94)*, pages 75–86, 1994.
- Craig BOUTILIER, Ronen I. BRAFMAN, Carmel DOMSHLAK, Holger H. HOOS et David POOLE : CP-nets : A Tool for Representing and Reasoning with Conditional *Ceteris Paribus* Preference Statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004a.
- Craig BOUTILIER, Ronen I. BRAFMAN, Carmel DOMSHLAK, Holger H. HOOS et David POOLE : Preference-Based Constrained Optimization with CP-nets. *Computational Intelligence*, 20(2):137–157, 2004b.

- Craig BOUTILIER, Ronen I. BRAFMAN, Holger H. HOOS et David POOLE : Reasoning with Conditional Ceteris Paribus Preference Statements. *Dans Proceedings of Uncertainty in Artificial Intelligence (UAI'99)*, pages 71–80, 1999.
- Felix BRANDT, Felix FISCHER et Markus HOLZER : Symmetries and the Complexity of Pure Nash Equilibrium. *Dans Proceedings of the 24th International Symposium on Theoretical Aspects of Computer Science (STACS'07)*, volume 4393, pages 212–223. Springer-Verlag, 2007.
- Gerhard BREWKA : Nonmonotonic logics - a brief overview. *Artificial Intelligence Communications*, 2(2):88–97, 1989a.
- Gerhard BREWKA : Preferred subtheories : An extended logical framework for default reasoning. *Dans SRIDHARAN, éditeur : Proceedings of the 11th International Joint Conference on Artificial Intelligence (IJCAI'89)*, pages 1043–1048, Detroit, 1989b.
- Gerhard BREWKA : Logic Programming with Ordered Disjunction. *Dans Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI'02)*, 2002.
- Gerhard BREWKA, Salem BENFERHAT et Daniel LE BERRE : Qualitative Choice Logic. *Dans Proceedings of the 8th International Conference on Knowledge Representation and Reasoning (KR'02)*, pages 158–169, Toulouse, France, 2002.
- Claudette CAYROL : On the Relation between Argumentation and Non-monotonic Coherence-Based Entailment. *Dans Chris S. MELLISH, éditeur : Proc. of the 14th IJCAI*, pages 1443–1448, Montréal, Canada, 1995. Morgan Kaufmann.
- Claudette CAYROL, Véronique ROYER et Claire SAUREL : Management of preferences in assumption-based reasoning. *Dans Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems (IPMU'92)*, pages 13–22, Berlin, janvier 1993. Springer Verlag.
- Yann CHEVALEYRE, Ulle ENDRISS et Jérôme LANG : Expressive Power of Weighted Propositional Formulas for Cardinal Preference Modelling. *Dans P. DOHERTY, J. MYLOPOULOS et C. WELTY, éditeurs : Proceedings of the 10th International Conference on Principles of Knowledge Representation and Reasoning (KR'06)*, pages 145–152. AAAI Press, June 2006.
- Vincent CONITZER et Tuomas SANDHOLM : Complexity Results about Nash Equilibrium. *Dans International Joint Conferences on Artificial Intelligence (IJCAI'03)*, pages 765–771, 2003.
- Sylvie COSTE-MARQUIS, Jérôme LANG, Paolo LIBERATORE et Pierre MARQUIS : Expressive power and succinctness of propositional languages for preference representation. *Dans Proceedings of the 9th International Conference on Principles of Knowledge Representation and Reasoning (KR'04)*, pages 203–212, 2004.
- Antoine COURNOT : *Recherches sur les Principes Mathématiques de la Théorie des Richesses*. 1838.
- Marina DE VOS et Dirk VERMEIR : Choice Logic Programs and Nash Equilibria in Strategic Games. *Dans Jorg FLUM et Mario RODRIGUEZ-ARTALEJO, éditeurs : Computer Science Logic (CSL'99)*, volume 1683, pages 266–276, 1999.

- Xiaotie DENG, Christos PAPANIMITRIOU et Muli SAFRA : Complexity results about Nash equilibria. *Dans Proceedings of the ACM Symposium on Theory of Computing (STOC'02)*, pages 67–71, 2002.
- Carmel DOMSHLAK : *Modeling And Reasoning About Preferences With CP-nets*. Thèse de doctorat, Ben-Gurion University of the Negev, Negev, Israel, 2002.
- Jon DOYLE, Yoav SHOHAM et Michael P. WELLMAN : A Logic of Relative Desire (Preliminary Report). *Dans 6th International Symposium for Methodologies for Intelligent Systems (ISMIS'91)*, pages 16–31, 1991.
- Jon DOYLE et Michael P. WELLMAN : Preferential Semantics for Goals. *Dans Proceedings of the National Conference on Artificial Intelligence (AAAI'91)*, pages 698–703, 1991.
- Didier DUBOIS, Jérôme LANG et Henri PRADE : Inconsistency in possibilistic knowledge bases : To live with it or not live with it. *Dans L.A. ZADEH et J. KACPRZYK, éditeurs : Fuzzy Logic for the Management of Uncertainty*, pages 335–351. Wiley, New York, 1992.
- Phan Minh DUNG : On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-persons games. *Artificial Intelligence*, 77:321–357, 1995.
- Paul E. DUNNE et Wiebe van der HOEK : Representation and Complexity in Boolean Games. *Dans Proceedings of the 9th European Conference on Logics in Artificial Intelligence (JELIA'04)*, volume LNCS 3229, pages 347–359. José Júlio Alferes et João Alexandre Leite (eds), 2004.
- Paul E. DUNNE, Wiebe van der HOEK et Michael WOOLDRIDGE : A Logical Characterisation of Qualitative Coalitional Games. *Journal of Applied Non-Classical logics*, 17(4):477–509, 2007.
- Paul E. DUNNE et Michael WOOLDRIDGE : Preferences in Qualitative Coalitional Games. *Dans Proceedings of the 6th Workshop on Game Theoretic and Decision Theoretic Agents (GTDT'04)*, pages 29–38. New York, July 2004.
- Florence Dupin de SAINT CYR, Jérôme LANG et Thomas SCHIEX : Penalty logic and its link with Dempster-Shafer theory. *Dans Uncertainty in Artificial Intelligence (UAI'95)*, pages 204–211, 1992.
- Francis Ysidro EDGEWORTH : La Teoria pura del monopolio. *Giornale degli Economisti*, pages 13–31, 1897.
- Edith ELKIND, Leslie Ann GOLDBERG et Paul W. GOLDBERG : Nash Equilibria in Graphical Games on Trees Revisited. *Dans Proceedings of Conference on Electronic Commerce (EC'06)*, 2006.
- Alex FABRIKANT, Christos PAPANIMITRIOU et Kunal TALWAR : The Complexity of Pure Nash Equilibria. *Dans Proceedings of the 36th ACM Symposium on Theory of Computing (STOC'04)*, pages 604–612, 2004.
- Timothy W. FININ, Richard FRITZSON, Don MCKAY et Robin MCENTIRE : KQML As An Agent Communication Language. *Dans Proceedings of the Third International Conference on Information and Knowledge Management (CIKM'94)*, pages 456–463, 1994.

- Norman FOO, Thomas MEYER et Gerhard BREWKA : LPOD Answer Sets and Nash Equilibria. *Dans* M. MAHER, éditeur : *Proceedings of the 9th Asian Computer Science Conference (ASIAN'04)*, pages 343–351. Chiang Mai, Thailand, Springer LNCS 3321, 2004.
- Peter GÄRDENFORS et David MARKINSON : Nonmonotonic inference based on expectations. *Artificial Intelligence*, 65:197–245, 1994.
- Michael R. GAREY et David S. JOHNSON : *Computers and Intractability : A Guide to the Theory of NP-completeness*. W.H. Freeman and Company, New York, 1979.
- Hector GEFFNER : *Default reasoning : causal and conditional theories*. MIT Press, 1992.
- Michael GELFOND et Vladimir LIFSCHITZ : Classical negation in logic programs and disjunctive databases. *New Generation Computing*, 9:365–385, 1991.
- Michael GELFOND et Vladimir LIFSCHITZ : Representing Actions and Change by Logic Programs. *Journal of Logic Programming*, 17(2, 3, 4):301–323, 1993.
- Itzhak GILBOA et Eitan ZEMEL : Nash and correlated equilibria : Some complexity considerations. *Games and Economic Behavior*, 1:80–93, 1989.
- Enrico GIUNCHIGLIA et Vladimir LIFSCHITZ : An Action Language Based on Causal Explanation : Preliminary Report. *Dans Proceedings of the National Conference on Artificial Intelligence (AAAI'98)*, pages 623–630, 1998.
- Georg GOTTLÖB, Gianluigi GRECO et Francesco SCARCELLO : Pure Nash Equilibria : Hard and Easy Games. *Journal of Artificial Intelligence Research*, 24:357–406, 2005.
- Peter HADDAWY et Steve HANKS : Representations for Decision-Theoretic Planning : Utility Functions for Deadline Goals. *Dans* J.A. ALLEN, R. FIKES et E. SANDEWALL, éditeurs : *International Conference on Knowledge Representation and Reasoning, (KR'92)*, pages 71–82. Cambridge, Morgan Kaufmann, 1992.
- Sven Ove HANSSON : A New Semantical Approach to the Logic of Preference. *Erkenntnis*, 31:1–42, 1989.
- Sven Ove HANSSON : An overview of decision theory. Rapport technique 41, SKN, 1991.
- Sven Ove HANSSON : *The Structure of Values and Norms*. Cambridge University Press, 2001.
- Paul HARRENSTEIN : *Logic in Conflict*. Thèse de doctorat, Utrecht University, 2004a.
- Paul HARRENSTEIN : Logical Consequence and the Theory of Games. *Philosophia Scientiae*, 8(2): 179–193, 2004b. Special issue on logic and games.
- Paul HARRENSTEIN, Wiebe van der HOEK, John-Jules MEYER et Cees WITTEVEEN : Boolean Games. *Dans* J. van BENTHEM, éditeur : *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.
- John HILLAS et Elon KOHLBERG : Foundations of strategic equilibrium. *Dans* R. AUMANN et S. HART, éditeurs : *Handbook of Game Theory*, volume 3, pages 1598–1663. North-Holland, 2002.

- Albert Xin JIANG et Kevin LEYTON-BROWN : A polynomial-time algorithm for Action-Graph Games. *Dans American Association for Artificial Intelligence (AAAI'06)*, pages 679–684, 2006.
- Albert Xin JIANG et Kevin LEYTON-BROWN : Computing Pure Nash Equilibria in Symmetric Action-Graph Games. *Dans Association for the Advancement of Artificial Intelligence (AAAI)*, 2007. To Appear.
- Hirofumi KATSUNO et Alberto O. MENDELZON : On the Difference between Updating a Knowledge Base and Revising It. *Dans International Conference on Knowledge Representation and Reasoning, (KR'92)*, pages 387–394, 1991.
- Hirofumi KATSUNO et Alberto O. MENDELZON : Propositional Knowledge Base Revision and Minimal Change. *Artificial Intelligence*, 52:263–294, 1992.
- Michael KEARNS, Michael L. LITTMAN et Satinder SINGH : Graphical Models for Game Theory. *Dans Proceedings of Uncertainty in Artificial Intelligence (UAI'01)*, pages 253–260, 2001.
- Michael J. KEARNS et Yishay MANSOUR : Efficient Nash Computation in Large Population Games with Bounded Influence. *Dans Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence (UAI'02)*, pages 259–266, 2002.
- Daphne KOLLER, Nimrod MEGGIDO et Bernhard von STENGE : Efficient computation of equilibria for extensive two-person games. *Games and Economic Behavior*, 14:73–92, 1996.
- Daphne KOLLER et Brian MILCH : Multi-Agent Influence Diagrams for Representing and Solving Games. *Games and Economic Behavior*, 45(1):181–221, 2003. Full version of paper in IJCAI '01.
- Sarit KRAUS, Daniel J. LEHMANN et Menachem MAGIDOR : Nonmonotonic Reasoning, Preferential Models and Cumulative Logics. *Artificial Intelligence*, 44:167–207, 1990.
- Pierfrancesco LA MURA : Game Networks. *Dans Proceedings of Uncertainty in Artificial Intelligence (UAI'00)*, pages 335–342, 2000.
- Yannis LABROU et Timothy W. FININ : Semantics and Conversations for an Agent Communication Language. *Dans International Joint Conferences on Artificial Intelligence (IJCAI'97)*, pages 584–591, 1997.
- Céline LAFAGE et Jérôme LANG : Logical representation of preference for group decision making. *Dans Proceedings of the Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR'00)*, pages 457–468, 2000.
- Céline LAFAGE et Jérôme LANG : Propositional distances and preference representation. *Dans Proceedings of European Conferences on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU'01)*, pages 48–59, 2001.
- Jérôme LANG : Conditional desires and utilities – An alternative logical framework for qualitative decision theory. *Dans European Conference on Artificial Intelligence (ECAI'96)*, pages 318–322, 1996.
- Jérôme LANG : Logical preference representation and combinatorial vote. *Dans Annals of Mathematics and Artificial Intelligence*, volume 42, pages 37–71. Kluwer, 2004.

- Jérôme LANG : Représentation logique de préférences. *Dans* Denis BOUYSSOU, Didier DUBOIS, Marc PIRLOT et Henri PRADE, éditeurs : *Concepts et méthodes pour l'aide à la décision*, volume 1. Lavoisier, 2006.
- Jérôme LANG, Paolo LIBERATORE et Pierre MARQUIS : Conditional independence in propositional logic. *Artificial Intelligence*, 141 (1-2):79–121, 2002a.
- Jérôme LANG, Paolo LIBERATORE et Pierre MARQUIS : Propositional Independence - Formula-Variable Independence and Forgetting. *Journal of Artificial Intelligence Research*, 18:391–443, 2003.
- Jérôme LANG et Pierre MARQUIS : Complexity Results for Independence and Definability in Propositional Logic. *Dans International Conference on Knowledge Representation and Reasoning, (KR'98)*, pages 356–367, 1998a.
- Jérôme LANG et Pierre MARQUIS : Two forms of dependence in propositional logic : controllability and definability. *Dans Proceedings of the National Conference on Artificial Intelligence (AAAI'98)*, pages 268–273, 1998b.
- Jérôme LANG, Leendert van der TORRE et Emil WEYDERT : Utilitarian desires. *International Journal of Autonomous Agents and Multi-Agent Systems*, 5:329–363, 2002b.
- Daniel J. LEHMANN : Another Perspective on Default Reasoning. *Annals of Mathematics and Artificial Intelligence*, 15:61–82, 1995.
- David LEWIS : *Counterfactuals*. Basil Blackwell, 1973.
- Kevin LEYTON-BROWN et Moshe TENNENHOLTZ : Local-Effect Games. *Dans International Joint Conferences on Artificial Intelligence (IJCAI'03)*, pages 772–777, 2003.
- Fangzhen LIN et Raymond REITER : Forget it. *Dans Proceedings of the National Conference on Artificial Intelligence (AAAI'94)*, pages 154–159, 1994.
- R. Duncan LUCE et Howard RAIFFA : *Games and Decisions : Introduction and Critical Survey*. Wiley & Sons, 1957.
- Pierre MARQUIS : *Consequence Finding Algorithms*, in *Handbook on Defeasible Reasoning and Uncertainty Management Systems*, volume 5 : Algorithms for Defeasible and Uncertain Reasoning, S. Moral and J. Kohlas, chapitre 2, pages 41–145. Kluwer Academic Publisher, 2000.
- Richard D. MCKELVEY et Andrew MCLENNAN : *Computation of equilibria in finite games*, volume 1, chapitre Handbook of Computational Economics. Elsevier, 1996.
- Nimrod MEGGIDO et Christos PAPADIMITRIOU : On total functions, existence theorems, and computational complexity. *Theoretical Computer Science*, 81(2):317–324, 1991.
- Dov MONDERER et Lloyd S. SHAPLEY : Potential games. *Games and Economic Behaviour*, 14:124–143, 1996.
- Hervé MOULIN : *The strategy of social choice*. North-Holland, 1983.

- Roger B. MYERSON : *Game Theory : Analysis of Conflict*. Harvard University Press, 1991.
- John NASH : Equilibrium points in n -person games. *Proceedings of the National Academy of Sciences*, 36:48–49, 1950.
- Martin J. OSBORNE et Aryel RUBINSTEIN : *A course in game theory*. MIT Press, 1994.
- Guillermo OWEN : *Game Theory*. Academic Press, 1982.
- Christos PAPADIMITRIOU : *Computational Complexity*. Addison-Wesley, 1994a.
- Christos PAPADIMITRIOU : On the complexity of the parity argument and other inefficient proofs of existence. *Journal of Computer and System Sciences*, 48(3):498–532, 1994b.
- Christos PAPADIMITRIOU : Algorithms, games, and the internet. *Dans Proceedings of the ACM Symposium on Theory of Computing (STOC'04)*, pages 749–753, 2001.
- Marc PAULY : *Logic for Social Software*. Thèse de doctorat, Institute for Logic, Language and Computation, Universiteit van Amsterdam, Amsterdam, 2001.
- Gadi PINKAS : Propositional nonmonotonic reasoning and inconsistency in symmetric neutral networks. *Dans Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'91)*, pages 252–530, 1991.
- Steve PRESTWICH, Francesca ROSSI, Kristen Brent VENABLE et Toby WALSH : Constrained CP-nets. *Dans Italian Conference on Computational Logic*, 2004.
- Anand S. RAO et Michael P. GEORGEFF : Modeling rational agents within a BDI-architecture. *Dans James ALLEN, Richard FIKES et Erik SANDEWALL, éditeurs : Proceedings of the 2nd International Conference on Principles of Knowledge Representation and Reasoning (KR'91)*, pages 473–484. Morgan Kaufmann publishers, 1991.
- Robert W. ROSENTHAL : A class of games possessing pure-strategy Nash equilibria. *International Journal of Game Theory*, 2:65–67, 1973.
- Tim ROUGHGARDEN et Eva TARDOS : Bounding the Inefficiency of Equilibria in Nonatomic Congestion Games. Rapport technique TR2002-1866, Cornell, 2001.
- Aryel RUBINSTEIN : *Economics and Language*. Cambridge University Press, 2000.
- Luigi SAURO : *Formalizing Admissibility Criteria in Coalition Formation among Goal Directed Agents*. Thèse de doctorat, Università Degli Studi di Torino Dipartimento di Informatica, 2006.
- John SEARLE : *Speech Acts*. Cambridge University Press, 1969.
- Reinhard SELTEN : Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft*, 121:301–324, 667–689, 1965.
- Jaime Simão SICHMAN et Rosaria CONTE : Multi-Agent Dependence by Dependence Graphs. *Dans Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS'02)*, pages 483 – 490, 2002.

- Guillermo R. SIMARI et Ronald P. LOUI : A mathematical treatment of defeasible reasoning and its implementation. *Artificial Intelligence*, 53:125–157, 1992.
- Denis SIREYJOL : Parallèles entre théorie de l'argumentation et théorie des jeux. Mémoire de D.E.A., DEA RCFR, Université Toulouse III, 2004.
- Larry STOCKMEYER : The polynomial-time hierarchy. *Theoretical Computer Science*, 3:1–22, 1977.
- Wiebe van der HOEK et Michael WOOLDRIDGE : On the Logic of Cooperation and Propositional Control. *Artificial Intelligence*, 164(1-2):81–119, 2005.
- David VICKREY et Daphne KOLLER : Multi-Agent Algorithms for Solving Graphical Games. *Dans Proceedings of the Eighteenth National Conference on Artificial Intelligence (AAAI'02)*, pages 345–351, 2002.
- John von NEWMANN et Oskar MORGENSTERN : *Theory of Games and Economic Behaviour*. Princeton University Press, 1944.
- Georg Henrik von WRIGHT : *The logic of preference*. Edinburgh University Press, 1963.
- Michael WOOLDRIDGE et Paul E. DUNNE : On the computational complexity of qualitative coalitional games. *Artificial Intelligence*, 158/1:27–73, 2004.
- Bruno ZANUTTINI : New polynomial classes for logic-based abduction. *Journal of Artificial Intelligence Research*, 19:1–10, 2003.

Index

Atomicité	139	Forme extensive	16
Base de buts à priorité	100	Forme normale	17
Best Out	103	Forme normale graphique	128
BP-jeu booléen	105	Fortement jouable	139
Ceteris Paribus	75	Forward sweep procedure	78
Clause	7	Grande coalition	13, 137
Coalition	12, 35, 137	Graphe de dépendance	38
Coalition efficace	143	Impliquant, impliquant premier	7
Coalition efficace minimale	143	Indépendance	37, 116
Coalition gagnante	14	Indépendance préférentielle	75
Coalition perdante	14	Indépendance préférentielle conditionnelle	75
Coalition-monotone	138	Jeu à somme nulle	14
CP-jeu booléen	79	Jeu booléen à n joueurs	34
CP-net	76	Jeu coopératif	12
CP-net global	91	Jeu dynamique	11
Cube	7	Jeu en information imparfaite	12
Discrimin	101	Jeu en information parfaite	11
Duopole de Stackelberg	60	Jeu non coopératif	14
Décomposabilité	139	Jeu statique	10
Ensemble stable	39	Joueurs utiles	37, 117
Equilibre de Nash en stratégies mixtes	20	L-impliquant, L-impliquant premier	8
Equilibre de Nash en stratégies pures	19, 41	L-jeu booléen	71
Equilibre de Nash faible	72	Langage de représentation de préférences	70
Equilibre de Nash fort	72	Leximin	103
Equilibre parfait de Selten	27	Maximalité	138
Fonction caractéristique	13	Meneur	60
Fonction d' α -effectivité	138	Noyau, Noyau fort	25, 26, 150
Fonction d'affectation de contrôle	34	Oubli total, oubli partiel	8, 43
Fonction d'effectivité	137		
Fonction d'utilité	10, 36		

Profil de stratégies	10, 35
Programme logique	121
Projection d'un jeu	40
Projection d'une formule	8, 43
Pré-jeu booléen	34
Résultat optimal	78
Relation de préférence	69, 77
Régularité	138
Stratégie	10, 35
Stratégie faiblement dominée	22, 51, 72
Stratégie gagnante	37
Stratégie mixte	20
Stratégie partiellement dominée	73
Stratégie strictement dominée	22, 51, 72
Suiveur	60
Supperaditivité	138
Table de préférence conditionnelle	76
Terme	7
Union des graphes	88
Utilité non transférable	13
Utilité transférable	13
Variables utiles	37, 117

Liste des symboles

Σ^j : Ensembles des buts de priorité j	100	CPT : Table de préférence conditionnelle	76
Γ : Ensemble des contraintes	34	$D(X_i)$: Domaine de la variable X_i	75
Φ : Ensembles des buts	34	DNF : Disjonction de termes	7
Σ_2^p : Classe des langages reconnaissables en temps polynomial par une machine de Turing non-déterministe munie d'oracles NP	56	$EC(G)$: Ensemble des coalitions efficaces du jeu G	143
Σ : Base de buts à priorité	100	Eff : Fonction d'effectivité	137
$\exists x : \varphi$: Oubli partiel de x dans φ	8, 43	$FNoyau(G)$: Noyau fort du jeu G	150
$\forall x : \varphi$: Oubli complet de x dans φ	8, 43	$G_B = (B, V_B, \pi_B, \Gamma_B, \Phi_B)$: Projection du jeu G sur B	40
γ_i : Ensembles des contraintes du joueur i	34, 35	L_V : Langage propositionnel construit à partir de V	7
π : Fonction d'affectation de contrôle	34	$Lit(\varphi)$: Ensemble des littéraux formant la formule φ	7
π_i : Ensembles des variables contrôlées par le joueur i	34	$Mod_X(\varphi)$: Ensemble des X -interprétations satisfaisant φ	7
π_{-i} : Ensemble des variables contrôlées par tous les joueurs sauf i	36	N : Ensembles des joueurs	10, 34
\succ : Relation de préférences stricte	69	$Noyau(G)$: Noyau (faible) du jeu G	150
\succ^{bo} : Relation de préférence respectant le critère best out	103	NE_{faible} : Equilibre de Nash faible	72
\succ^{disc} : Relation de préférence respectant le critère discrimin	101	NE_{fort} : Equilibre de Nash fort	72
\succ^{lex} : Relation de préférence respectant le critère leximin	103	$\mathcal{P} = \langle N, R \rangle$: Graphe de dépendance	38
$\succ_{\mathcal{N}}$: Relation de préférence induite par le CP-net \mathcal{N}	77	$Pa(X)$: Ensemble des variables parents de la variable X	76
\succeq : Relation de préférence	69	$PI(\psi)$: Ensemble des impliquants premiers de ψ	7
$(\varphi)_{M_X}$: Interprétation partielle de φ par M_X	7	$PI_L\psi$: Ensemble des L-impliquants premiers de ψ	8
φ_i : But du joueur i	34	PNE : Équilibres de Nash en stratégies pures	19, 41
2^V : Ensemble des interprétations pour V	7		
$At(C)$: Ensembles minimaux de $Eff(C)$	139		
CNF : Conjonction de termes	58		

$R(i)$: Ensemble des joueurs nécessaires à i
pour satisfaire son but 38

$R^*(i)$: Ensemble des joueurs ayant une in-
fluence directe ou indirecte sur i . . 38

RP_i : Ensemble des joueurs utiles pour le
joueur i 37

RV_i : Ensemble des variables utiles pour le
joueur i 37

S : Ensemble des profils de stratégies . . 10, 35

S_i : Ensemble des stratégies du joueur i 10, 35

s : Profil de stratégies 10, 35

s_i : Stratégie du joueur i 10, 35

s_{-i} : Profil de stratégies s privé de la stratégie
du joueur i 10, 36

$u_i(s)$: Utilité obtenue par le joueur i lorsque le
profil de stratégies s est joué . . 10, 36

V : Ensemble de variables propositionnelles 7,
34

$Var(\varphi)$: Ensemble des variables présentes
dans la formule φ 7