Boolean Games with Norms

Xin Sun

Faculty of Science, Technology and Communication, University of Luxembourg
xin.sun@uni.lu

Abstract. In the present paper we overlay boolean game with norms. Norms distinguish illegal strategies from legal strategies. Two types of legal strategy and legal Nash equilibrium are defined. These two equilibrium are viewed as solution concepts for law abiding agents in norm augmented boolean games. Our formal model is a combination of boolean games and so called input/output logic. We study various complexity issues related to legal strategy and legal Nash equilibrium.

Key words: boolean game, norm, input/output logic

1 Introduction

The study of the interplay of games and norms can be divided into two main branches: the first, mostly originating from economics and game theory [9, 19, 20], treats norms as mechanisms that enforce desirable properties of social interactions; the second, that has its roots in social sciences and evolutionary game theory [31, 10] views norms as (Nash or correlated) equilibrium that result from the interaction of rational agents. A survey of the interaction between games and norms can be found in Grossi *et al* [15]. This paper belongs to the first branch.

In this paper we study the combination of boolean games and norms. Boolean game is a class of games based on propositional logic. It was firstly introduced by Harrenstein et al. [17] and further developed by several researchers [16, 23, 11, 8, 6, 25]. In a boolean game, each agent i is assumed to have a goal, represented by a propositional formula ϕ_i over some set of propositional variables $\mathbb{P}$. Each agent i is associated with some subset $\mathbb{P}_i$ of the variables, which are under the unique control of agent i. The choices, or strategies, available to i correspond to all the possible assignment of truth or falsity to the variables in $\mathbb{P}_i$. An agent will try to choose an assignment so as to satisfy his goal ϕ_i. Strategic concerns arise because whether i's goal is in fact satisfied will depend on the choices made by other agents.

Norms regulate agents' behaviors in boolean games. Shoham and Tennenholtz's early work on behavior change under norms [27, 28] has considered only a relatively simple view of norms, where some actions or states are designated as violations. Alechina *et al* [1] studies how conditional norms regulate agents' behaviors, but permissive norms plays no role in their framework. In this paper we study how agents' behavior are changed by permissive and obligatory conditional norms. Norms distinguish illegal strategies from legal strategies. By designing norms appropriately, non-optimal equilibrium might be avoided. To represent norms in boolean games we need a logic of norms, which has been extensively studied in the deontic logic community.

Various deontic logic have been developed since von Wright's first paper [32] in this area. In the first volume of the handbook of deontic logic [12], input/output logic [21, 22] appears as one of the new achievements in deontic logic in this century. Input/output logic takes its origin in the study of conditional norms. The basic idea is: norms are conceived as a deductive machine, like a black box which produces normative statements as output, when we feed it factual statements as input.

In this paper we use input/output logic as the logic of norms. Given a normative multi-agent system, which contains a boolean game, a set of norms and certain environment. Every strategy of every agent is classified as legal or illegal. Notions like legal Nash equilibrium are then naturally defined.

The structure of this paper is the following: We present some background knowledge, including boolean game, input/output logic and complexity theory in Section 2. Normative multi-agent system are introduced and its complexity issues are discussed in Section 3. We conclude this paper in Section 4.

2 Background

2.1 Propositional logic

Let $\mathbb{P} = \{p_0, p_1, \ldots\}$ be a finite set of propositional variables and let $L_{\mathbb{P}}$ be the propositional language built from $\mathbb{P}$ and boolean constants $\top$ (true) and $\bot$ (false) with the usual connectives $\neg, \vee, \wedge, \rightarrow$ and $\leftrightarrow$. Formulas of $L_{\mathbb{P}}$ are denoted by ϕ, ψ etc. A literal is a variable $p \in \mathbb{P}$ or its negation. $2^{\mathbb{P}}$ is the set of the valuations for $\mathbb{P}$, with the usual convention that for $V \in 2^{\mathbb{P}}$ and $p \in V$, V gives the value true to p if $p \in V$ and false otherwise. $\vDash$ denotes the classical logical consequence relation.

Let $X \subseteq \mathbb{P}$, 2^X is the set of X-valuations. A partial valuation (for $\mathbb{P}$) is an X-valuation for some $X \subseteq \mathbb{P}$. Partial valuations are denoted by listing all variables of X, with a " $+$ " symbol when the variable is set to be true and a " $-$ " symbol when the variable is set to be false: for instance, let $X = \{p, q, r\}$, then the X-valuation $V = \{p, r\}$ is denoted $\{+p, -q, +r\}$. If $\{\mathbb{P}_1, \ldots, \mathbb{P}_n\}$ is a partition of $\mathbb{P}$ and $V_1, \ldots, V_n$ are partial valuations, where $V_i \in 2^{\mathbb{P}_i}$, $(V_1, \ldots, V_n)$ denotes the valuation $V_1 \cup \ldots \cup V_n$.

2.2 Boolean game

Boolean games introduced by Harrenstein *et al* [17] are zero-sum games with two players, where the strategies available to each player consist in assigning a truth value to each variable in a given subset of $\mathbb{P}$. Bonzon *et al* [7] give a more general definition of a boolean game with any number of players and not necessarily zero-sum. In this paper we further generalizes boolean games such that the utility of each agent is not necessarily in $\{0, 1\}$. Such generalization is reached by representing the goals of each agent as a set of weighted formulas. The idea of using weighted formulas to define utility can be found in many work among which we mention satisfiability games [4] and weighted boolean formula games [23].

Definition 1 (boolean game). *A boolean game is a 4-tuple* $(Agent, \mathbb{P}, \pi, Goal)$*, where*

1. $Agent = \{1, \ldots, n\}$ *is a set of agents.*
2. $\mathbb{P}$ *is a finite set of propositional variables.*
3. $\pi : Agent \mapsto 2^{\mathbb{P}}$ *is a control assignment function such that* $\{\pi(1), \ldots, \pi(n)\}$ *forms a partition of* $\mathbb{P}$*. For each agent i,* $2^{\pi(i)}$ *is the strategy space of i.*
4. $Goal = \{Goal_1, \ldots, Goal_n\}$ *is a set of weighted formulas of* $L_{\mathbb{P}}$*. That is, each* $Goal_i$ *is a finite set* $\{\langle\phi_1, m_1\rangle, \ldots, \langle\phi_k, m_k\rangle\}$ *where* $\phi_j \in L_{\mathbb{P}}$ *and* m_j *is a real number.*

A strategy for agent i is a partial valuation for all the variables i controls. Note that since $\{\pi(1), \ldots, \pi(n)\}$ forms a partition of $\mathbb{P}$, a strategy profile S is a valuation for $\mathbb{P}$. In the rest of the paper we make use of the following notation, which is standard in game theory. Let $G = (Agent, \mathbb{P}, \pi, Goal)$ be a boolean game with $Agent = \{1, \ldots, n\}$, $S = (s_1, \ldots, s_n)$ be a strategy profile. s_{-i} denotes the projection of S on $Agent - \{i\}$: $s_{-i} = (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$.

Agents' utilities in boolean games are induced by their goals. For every agent i and every strategy profiles S, $u_i(S) = \Sigma\{m_j : \langle\phi_j, m_j\rangle \in Goal_i, S \vDash \phi_j\}$. Dominating strategies and pure-strategy Nash equilibria are defined as usual in game theory [24].

Example 1 *Let* $G = (Agent, \mathbb{P}, \pi, Goal)$ *where* $Agent = \{1, 2\}$*,* $\mathbb{P} = \{p, q, s\}$*,* $\pi(1) = \{p\}$*,* $\pi(2) = \{q, s\}$*,* $Goal_1 = \{\langle p \leftrightarrow q, 1\rangle, \langle s, 2\rangle\}$*,* $Goal_2 = \{\langle p \wedge q, 2\rangle, \langle\neg s, 1\rangle\}$*. This boolean game is depicted as follows:*

	$+q, +s$	$+q, -s$	$-q, +s$	$-q, -s$
$+p$	$(3, 2)$	$(1, 3)$	$(2, 0)$	$(0, 1)$
$-p$	$(2, 0)$	$(0, 1)$	$(3, 0)$	$(1, 1)$

2.3 Input/output logic

In input/output logic, a norm is an ordered pair of formulas $(\phi, \psi) \in L_{\mathbb{P}} \times L_{\mathbb{P}}$. There are two types of norms which are used in input/output logic, obligatory norms and permissive norms. Let $N = O \cup P$ be a set of obligatory and permissive norms. A pair $(\phi, \psi) \in O$, call it an obligatory norm, is read as "given ϕ, it is obligatory to be ψ". A pair $(\phi, \psi) \in P$, call it a permissive norm, is read as "given ϕ, it is permitted to be ψ".

Obligatory norms O can be viewed as a function from $2^{L_{\mathbb{P}}}$ to $2^{L_{\mathbb{P}}}$ such that for a set Φ of formulas, $O(\Phi) = \{\psi \in L_{\mathbb{P}} : (\phi, \psi) \in O \text{ for some } \phi \in \Phi\}$.

Definition 2 (Semantics of input/output logic [21]). *Given a finite set of obligatory norms O and a finite set of formulas* Φ*,* $out(O, \Phi) = Cn(O(Cn(\Phi)))$*, where* Cn *is the consequence relation of propositional logic.*[1]

[1] In Makinson and van der Torre [21], this logic is called simple-minded input/output logic. Different input/output logics are developed in Makinson and van der Torre [21] as well. A technical introduction of input/output logic can be found in Sun [30].

Intuitively, the procedure of the semantics is as following: We first have in hand a set of formulas Φ (call it the input) as a description of the current state. We then close it by logical consequence $Cn(\Phi)$. The set of norms, like a deductive machine, accepts this logically closed set and produces a set of formulas $O(Cn(\Phi))$. We finally get the output $Cn(O(Cn(\Phi)))$ by applying the logical closure again. $\psi \in out(O, \phi)$ is understood as "ψ is obligatory given facts Φ and norms O".

Example 2 *Let p, q, r are propositional variables. Let $O = \{(p, q), (p \vee q, r), (r, p)\}$. Then $out(O, \{p\}) = Cn(O(Cn(\{p\}))) = Cn(\{q, r\})$.* □

Input/output logic is given a proof theoretic characterization. We say that an ordered pair of formulas is derivable from a set O iff (a, x) is in the least set that extends $O \cup \{(\top, \top)\}$ and is closed under a number of derivation rules. The following are the rules we need:

- SI (strengthening the input): from (ϕ, ψ) to (χ, ψ) whenever $\chi \vDash \phi$.
- WO (weakening the output): from (ϕ, ψ) to (ϕ, χ) whenever $\psi \vDash \chi$.
- AND (conjunction of output): from (ϕ, ψ) and (ϕ, χ) to $(\phi, \psi \wedge \chi)$.

The derivation system based on the rules SI, WO and AND is denoted as $deriv(O)$.

Example 3 *Let $O = \{(p \vee q, r), (q, r \to s)\}$, then $(q, s) \in deriv(O)$ because we have the following derivation*

1.	$(p \vee q, r)$	*Assumption*
2.	(q, r)	*1, SI*
3.	$(q, r \to s)$	*Assumption*
4.	$(q, r \wedge (r \to s))$	*2,3, AND*
5.	(q, s)	*4, WO*

In Makinson and van der Torre [21], the following soundness and completeness theorem is proved:

Theorem 1 ([21]). *Given a set of obligatory norms O,*

$$\psi \in out(O, \{\phi\}) \text{ iff } (\phi, \psi) \in deriv(O).$$

Permission in input/output logic Philosophically, it is common to distinguish between two kinds of permission: negative permission and positive permission. Negative permission is straightforward to describe: something is negatively permitted according to certain norms iff it is not prohibited by those norms. That is, iff there is no obligation to the contrary. Positive permission is more elusive. Makinson and van der Torre [22] distinguish two types of positive permission: static and dynamic permission. For the sake of simplicity, in this paper when discuss positive permission we only mean static permission.

Definition 3 (negative permission [22]). *Given a finite set of norms $N = O \cup P$ and a finite set of formulas Φ, $NegPerm(N, \Phi) = \{\psi \in L_{\mathbb{P}} : \neg\psi \notin out(O, \Phi)\}$.*

Intuitively, ϕ is negatively permitted iff ϕ is not forbidden. Since a formula is forbidden iff its negation is obligatory, ϕ is not forbidden is equivalent to $\neg\phi$ is not obligatory. Permissive norms plays no role in negative permission.

Definition 4 (positive permission [22]). *Given a finite set of formulas Φ, a finite set of norms $N = O \cup P$ where O is a set of obligatory norms and P is a set of permissive norms.*

- *If $P \neq \emptyset$, then $PosPerm(N, \Phi) = \{\psi \in L_{\mathbb{P}} : \psi \in out(O \cup \{(\phi', \psi')\}, \Phi)$, for some $(\phi', \psi') \in P\}\}$.*
- *If $P = \emptyset$, then $PosPerm(N, \Phi) = out(O, \Phi)$.*

Intuitively, permissive norms are treated like weak obligatory norms, the basic difference is that while the latter may be used jointly, the former may only be applied one by one. As an illustration of such difference, image a situation in which a man is permitted to date either one of two girls, but not both of them. Alternative definitions of positive permission can be found in Makinson and van der Torre [22], Stolpe [29] and Governatori [14].

2.4 Complexity theory

Complexity theory is the theory to investigate the time, memory, or other resources required for solving computational problems. In this subsection we briefly review those concepts and results from complexity theory which will be used in this paper. More comprehensive introduction of complexity theory can be found in [3]

We assume the readers are familiar with notions like Turing machine and the complexity class P, NP and coNP. Oracle Turing machine and two complexity classes related to oracle Turing machine will be used in this paper.

Definition 5 (oracle Turing machine [3]). *An oracle for a language L is a device that is capable of reporting whether any string w is a member of L. An oracle Truing machine M^L is a modified Turing machine that has the additional capability of querying an oracle. Whenever M^L writes a string on a special oracle tape it is informed whether that string is a member of L, in a single computation step.*

P^{NP} is the class of problems solvable by a deterministic polynomial time Turing machine with an NP oracle. NP^{NP} is the class of problems solvable by a non-deterministic polynomial time Turing machine with an NP oracle. Another name for the class NP^{NP} is Σ_2^p. Σ_{i+1}^p is the class of problems solvable by a non-deterministic polynomial time Turing machine with a Σ_i^p oracle. Π_i^p is the class of problems of which the complement is in Σ_i^p.

3 From boolean game to normative multi-agent system

In recent years, normative multi-agent system [5, 2] arises as a new interdisciplinary academic area bringing together researchers from multi-agent system [26, 34, 33], deontic logic [12] and normative system [13, 18, 1]. By combining boolean games and norms, we here develop a new approach for normative multi-agent system.

Definition 6 (normative multi-agent system). *A normative multi-agent system is a triple* (G, N, E) *where*

- $G = (Agent, \mathbb{P}, \pi, Goal)$ *is a boolean game.*
- $N = O \cup P \subseteq L_{\mathbb{P}} \times L_{\mathbb{P}}$ *is a finite set of obligatory and permissive norms.*
- $E \subseteq L_{\mathbb{P}}$ *is a finite set of formulas representing the environment.*

3.1 Legal strategy

In a normative multi-agent system, agent's strategies are classified as either legal or illegal. The basic idea is viewing strategies as formulas and using the mechanism of input/output logic to decide whether a formula is permitted.

Definition 7 (legal strategy). *Given a normative multi-agent system* (G, N, E) *where* $N = O \cup P$*, for each agent* i*, a strategy* $(+p_1, \ldots, +p_m, -q_1, \ldots, -q_n)$ *is negatively legal if*

$$p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_n \in NegPerm(N, E).$$

The strategy is positively legal if

$$p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_n \in PosPerm(N, E).$$

Example 4 *Consider the prisoner's dilemma augmented with norms, where the two prisoners are brothers who are morally required to protect each other. Let* (G, N, E) *be a normative multi-agent system as following:*

- $G = (Agent, \mathbb{P}, \pi, Goal)$ *is a boolean game with*
  - $Agent = \{1, 2\}$,
  - $\mathbb{P} = \{p, q\}$,
  - $\pi(1) = \{p\}$, $\pi(2) = \{q\}$,
  - $Goal_1 = \{\langle p, 2\rangle, \langle \neg q, 3\rangle\}$, $Goal_2 = \{\langle q, 2\rangle, \langle \neg p, 3\rangle\}$.
- $N = O \cup P$ *where* $O = \{(\top, \neg p)\}$, $P = \{(\top, \neg q)\}$.
- $E = \emptyset$.

	$+q$	$-q$
$+p$	$(2,2)$	$(5,0)$
$-p$	$(0,5)$	$(3,3)$

Then $out(O, E) = Cn(\{\neg p\})$, $\{\neg p, q, \neg q\} \subseteq NegPerm(N, E)$. *Therefore* $\{-p\}$, $\{+q\}$, $\{-q\}$ *are negatively legal while* $\{+p\}$ *is not. Moreover we have* $PosPerm(N, E) = out(O \cup P, E) = Cn(\{\neg p, \neg q\})$*, Therefore* $\{-p\}$ *and* $\{-q\}$ *are positively legal while neither* $\{+p\}$ *nor* $\{+q\}$ *is.* ⊣

Having defined notions of legal strategy. A natural question to ask is how complex is it to decide whether a strategy is legal. The following theorems give a first answer to this question.

Theorem 2. *Given a normative multi-agent system* (G, N, E) *and a strategy* $(+p_1, \ldots, +p_m, -q_1, \ldots, -q_n)$*, deciding whether this strategy is negatively legal is NP complete.*

Proof. Concerning the NP hardness, we prove by reducing the satisfiability problem of propositional logic to our problem: Let $\phi \in L_{\mathbb{P}}$ be a formula. Let $N = \{(\neg\phi, \neg p)\}$, $E = \emptyset$. Then $p \in NegPerm(N, E)$ iff $\neg p \notin out(N, E) = Cn(N(Cn(E))) = Cn(N(Cn(\top)))$ iff $\not\models \neg\phi$ iff ϕ is satisfiable.

Now we prove the NP membership. We provide the following non-deterministic Turing machine to solve our problem. Let $N = \{(\phi_1, \psi_1), \ldots, (\phi_n, \psi_n)\}$, E be a finite set of formulas and $p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k$ be a formula.

1. Guess a sequence of valuation $V_1, \ldots, V_n, V'$ on the propositional letters appears in $E \cup \{\phi_1, \ldots, \phi_n\} \cup \{\psi_1, \ldots, \psi_n\} \cup \{p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k\}$.
2. Let $N' \subseteq N$ be the set of obligatory norms which contains all (ϕ_i, ψ_i) such that $V_i(E) = 1$ and $V_i(\phi_i) = 0$.
3. Let $\Psi = \{\psi : (\phi, \psi) \in N - N'\}$.
4. If $V'(\Psi) = 1$ and $V'(p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k) = 0$. Then return "accept" on this branch. Otherwise return "reject" on this branch.

It can be verified that $p_1 \wedge \ldots \wedge p_m \wedge \neg q_1 \wedge \ldots \wedge \neg q_k \notin Cn(N(Cn(E)))$ iff the algorithm returns "accept" on some branch and the time complexity of the non-deterministic Turing machine is polynomial. $\dashv$

Theorem 3. *Given a normative multi-agent system* (G, N, E) *and a strategy* $(+p_1, \ldots, +p_m, -q_1, \ldots, -q_n)$*, deciding whether this strategy is positively legal is coNP complete.*

Proof. The coNP hardness can be proved by a reduction from the tautology problem of propositional logic. Here we omit the details.

Concerning the coNP membership, note that $PosPerm(N, E) = out(O \cup \{(\phi_1, \psi_1)\}, E) \cup \ldots \cup out(O \cup \{(\phi_m, \psi_m)\}, E)$, where $N = O \cup P$, $P = \{(\phi_1, \psi_1), \ldots, (\phi_m, \psi_m)\}$. The NP membership follows from the fact that the NP class is closed under union. $\dashv$

3.2 Legal Nash equilibrium

A (pure-strategy) legal Nash equilibrium is a strategy profile which contains only legal strategies and no agent can improve his utility by choosing another legal strategy, given others do not change their strategies.

Definition 8 (Legal Nash equilibrium). *Given a normative multi-agent system* (G, N, E)*, A strategy profile* $S = (s_1, \ldots, s_n)$ *is a negatively legal Nash equilibrium if*

- *for every agent* i*,* s_i *is a negatively legal strategy*
- *for every agent* i*, for every negatively legal strategy* $s'_i \in S_i$*,* $u_i(S) \geq u_i(s'_i, s_{-i})$*.*

Positively legal Nash equilibrium is defined analogously.

Example 5 *In the normative multi-agent system presented in Example 4, there is no negatively legal Nash equilibrium and $(-p, -q)$ is the unique positively legal Nash equilibrium.*

Example 6 *Let (G, N, E) be a normative system as following:*

- $G = (Agent, \mathbb{P}, \pi, Goal)$ *is a boolean game with*
  - $Agent = \{1, 2\}$,
  - $\mathbb{P} = \{p, q\}$,
  - $\pi(1) = \{p\}$, $\pi(2) = \{q\}$,
  - $Goal_1 = Goal_2 = \{\langle p \wedge q, 2\rangle, \langle \neg p \wedge \neg q, 3\rangle\}$.
- $N = O \cup P$ *where* $O = \{(\top, \neg p), (\top, \neg q)\}$, $P = \emptyset$.
- $E = \emptyset$.

	$+q$	$-q$
$+p$	$(2, 2)$	$(0, 0)$
$-p$	$(0, 0)$	$(3, 3)$

Without norms there are two Nash equilibria: $(+p, +q)$ and $(-p, -q)$. There is only one negatively/positively legal Nash equilibrium: $(-p, -q)$. From the perspective of social welfare, $(+p, +q)$ is not an optimal equilibrium because its social welfare is $2 + 2 = 4$, while the social welfare of $(-p, -q)$ is $3 + 3 = 6$. Therefore this example shows that by designing norms appropriately, non-optimal equilibrium might be avoided

Theorem 4. *Given a normative multi-agent system (G, N, E) and a strategy profile $S = (s_1, \ldots, s_n)$. Deciding whether S is a negatively legal Nash equilibrium is NP hard and in coNP^{NP}.*

Proof. (sketch) It is NP hard because deciding whether a single strategy is legal is already NP hard.

Concerning the coNP^{NP} membership, we prove by giving the following algorithm on a non-deterministic Turing machine with oracle SAT to solve the complement of this problem.

1. Test if S is a negatively legal strategy profile. If no, then return "accept". Otherwise continue.
2. Guess a strategy profile S'.
3. Test if S' is a legal strategy profile. If yes, continue. Otherwise return "reject" on this branch.
4. Test if $u_i(S) < u_i(S')$ for some i. If yes, return "accept" on this branch. Otherwise return "reject" on this branch.

It can be verified that S is NOT a negatively legal Nash equilibrium iff the non-deterministic Turing returns "accept" on some branches. Therefore deciding whether S is a negatively legal Nash equilibrium is in coNP^{NP}. ⊣

Theorem 5. *Given a normative multi-agent system* (G, N, E) *and a strategy profile* $S = (s_1, \ldots, s_n)$*. Deciding whether* S *is a positively legal Nash equilibrium of* G *is coNP hard and in coNP*^{NP}*.*

Proof. (sketch) Similar to the proof of Theorem 4. ⊣

Theorem 6. *Given a normative multi-agent system* (G, N, E)*. Deciding whether there is a negatively/positively legal Nash equilibrium of* G *is* Σ_2^P *hard and in* Σ_3^P*.*

Proof. (sketch) The lower bound follows from the fact that deciding Nash equilibrium for boolean games without norms is Σ_2^P complete [7]. Concerning the upper bound, recall that $\Sigma_3^P = NP^{\Sigma_2^P}$. The problem can be solved by a polynomial time non-deterministic Turing machine with an Σ_2^P oracle. ⊣

4 Conclusion

In the present paper we introduce boolean game with norms. Norms distinguish illegal strategies from legal strategies. Using ideas from input/output logic, two types of legal strategies are discussed, as well as two types of legal Nash equilibrium. After formally presenting the model, we use examples to show that non-optimal Nash equilibrium can be avoided by implementing norms. We study the complexity issues related to legal strategy and legal Nash equilibrium.

References

1. Natasha Alechina, Mehdi Dastani, and Brian Logan. Reasoning about normative update. In Francesca Rossi, editor, *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*. IJCAI/AAAI, 2013.
2. Giulia Andrighetto, Guido Governatori, Pablo Noriega, and Leendert W. N. van der Torre, editors. *Normative Multi-Agent Systems*, volume 4 of *Dagstuhl Follow-Ups*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.
3. Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, New York, USA, 2009.
4. Vittorio Bilò. On satisfiability games and the power of congestion games. In Ming-Yang Kao and Xiang-Yang Li, editors, *Algorithmic Aspects in Information and Management, Third International Conference, AAIM 2007, Portland, OR, USA, June 6-8, 2007, Proceedings*, volume 4508 of *Lecture Notes in Computer Science*, pages 231–240. Springer, 2007.
5. Guido Boella, Leendert van der Torre, and Harko Verhagen. Introduction to the special issue on normative multiagent systems. *Autonomous Agents and Multi-Agent Systems*, 17(1):1–10, 2008.
6. Elise Bonzon, Marie-Christine Lagasquie-Schiex, and Jérôme Lang. Dependencies between players in boolean games. *Int. J. Approx. Reasoning*, 50(6):899–914, 2009.

7. Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang, and Bruno Zanuttini. Boolean games revisited. In Gerhard Brewka, Silvia Coradeschi, Anna Perini, and Paolo Traverso, editors, *ECAI 2006, 17th European Conference on Artificial Intelligence, August 29 - September 1, 2006, Riva del Garda, Italy, Including Prestigious Applications of Intelligent Systems (PAIS 2006), Proceedings*, volume 141 of *Frontiers in Artificial Intelligence and Applications*, pages 265–269. IOS Press, 2006.
8. Elise Bonzon, Marie-Christine Lagasquie-Schiex, Jérôme Lang, and Bruno Zanuttini. Compact preference representation and boolean games. *Autonomous Agents and Multi-Agent Systems*, 18(1):1–35, 2009.
9. Ronald Coase. The problem of social cost. *Journal of Law and Economics,*, 1, 1960.
10. James Coleman. *Foundations of Social Theory*. Belnap Press, 1998.
11. Paul E. Dunne, Wiebe van der Hoek, Sarit Kraus, and Michael Wooldridge. Cooperative boolean games. In Lin Padgham, David C. Parkes, Jörg P. Müller, and Simon Parsons, editors, *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008), Estoril, Portugal, May 12-16, 2008, Volume 2*, pages 1015–1022. IFAAMAS, 2008.
12. Dov Gabbay, John Horty, Xavier Parent, Ron van der Meyden, and Leendert van der Torre, editors. *Handbook of Deontic Logic and Normative Systems*. College Publications, London, 2014.
13. Thomas Å gotnes, Wiebe van der Hoek, Juan A. Rodríguez-Aguilar, Carles Sierra, and Michael Wooldridge. On the logic of normative systems. In Manuela M. Veloso, editor, *IJCAI 2007, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12, 2007*, pages 1175–1180, 2007.
14. Guido Governatori, Francesco Olivieri, Antonino Rotolo, and Simone Scannapieco. Computing strong and weak permissions in defeasible logic. *J. Philosophical Logic*, 42(6):799–829, 2013.
15. Davide Grossi, Luca Tummolini, and Paolo Turrini. Norms in game theory. In *Handbook of Agreement Technologies*. 2012.
16. Paul Harrenstein. *Logic in conflict*. PhD thesis, Utrecht University, 2004.
17. Paul Harrenstein, Wiebe van der Hoek, John-Jules Meyer, and Cees Witteveen. Boolean games. In *Proceedings of the 8th Conference on Theoretical Aspects of Rationality and Knowledge*, TARK '01, pages 287–298, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.
18. Andreas Herzig, Emiliano Lorini, Frédéric Moisan, and Nicolas Troquard. A dynamic logic of normative systems. In Toby Walsh, editor, *IJCAI 2011, Proceedings of the 22nd International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, July 16-22, 2011*, pages 228–233. IJCAI/AAAI, 2011.
19. Leonid Hurwicz. Institutions as families of game forms. *Japanese Economic Review*, 47(2):113–132, 1996.
20. Leonid Hurwicz. But who will guard the guardians? *American Economic Review*, 98(3):577–585, 2008.
21. David Makinson and Leendert van der Torre. Input-output logics. *Journal of Philosophical Logic*, 29:383–408, 2000.
22. David Makinson and Leendert van der Torre. Permission from an input/output perspective. *Journal of Philosophical Logic*, 32:391–416, 2003.
23. Marios Mavronicolas, Burkhard Monien, and Klaus W. Wagner. Weighted boolean formula games. In Xiaotie Deng and Fan Chung Graham, editors, *Internet and Network Economics, Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007, Proceedings*, volume 4858 of *Lecture Notes in Computer Science*, pages 469–481. Springer, 2007.

24. Martin Osborne and Ariel Rubinstein. *A Course in Game Theory*. The MIT Press, Cambridge, Mass., 1994.
25. Luigi Sauro and Serena Villata. Dependency in cooperative boolean games. *J. Log. Comput.*, 23(2):425–444, 2013.
26. Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations*. Cambridge University Press, 2009.
27. Yoav Shoham and Moshe Tennenholtz. On the synthesis of useful social laws for artificial agent societies (preliminary report). In William R. Swartout, editor, *Proceedings of the 10th National Conference on Artificial Intelligence. San Jose, CA, July 12-16, 1992.*, pages 276–281. AAAI Press / The MIT Press, 1992.
28. Yoav Shoham and Moshe Tennenholtz. On social laws for artificial agent societies: Off-line design. *Artif. Intell.*, 73(1-2):231–252, 1996.
29. Audun Stolpe. A theory of permission based on the notion of derogation. *J. Applied Logic*, 8(1):97–113, 2010.
30. Xin Sun. How to build input/output logic. In Nils Bulling, Leendert W. N. van der Torre, Serena Villata, Wojtek Jamroga, and Wamberto Weber Vasconcelos, editors, *Computational Logic in Multi-Agent Systems - 15th International Workshop, CLIMA XV, Prague, Czech Republic, August 18-19, 2014. Proceedings*, volume 8624 of *Lecture Notes in Computer Science*, pages 123–137. Springer, 2014.
31. Edna Ulmann-Margalit. *The Emergence of Norms.* Clarendon Press,, Oxford, 1977.
32. Georg von Wright. Deontic logic. *Mind*, 60:1–15, 1952.
33. Gerhard Weiss, editor. *Multiagent systems*. MIT press, 2 edition, 2013.
34. Michael Wooldridge. *An Introduction to MultiAgent Systems (2. ed.)*. Wiley, 2009.