Reasoning about Extensive Games

Sieuwert van Otterloo
Department of Computer Science
University of Liverpool
United Kingdom

Abstract

Extensive games of perfect information can be used as a model for multi agent decision making. For instance auctions and voting can be modeled as extensive games. An extension of modal logic called GLP^- is defined that allows us to express whether each agent has the appropriate amount of influence in the decision making process. In this paper a complete axiom system for this logic is presented.

1 Introduction

The Internet is not only a source of information, it also allows people to interact with each other. Sometimes this happens informally, for instance with chat and messaging services. In other cases the interaction occurs according to a formal protocol, for instance in the case of electronic auctions, games such as chess and electronic opinion polls. These formal protocols can be modeled as games, in which several agents interact in order to reach a certain outcome. Typically one wants to establish that such a protocol is fair, in the sense that each agent has the appropriate amount of influence on the outcome. In this paper a logic is presented that can express these fairness properties. This logic is interpreted over extensive games of perfect information.

In figure 1 an example extensive game form is shown. In this game form agent A first decides whether a should hold or not. Then agent B can decide whether proposition b should hold or not. A possible story could be that a indicates that A dresses in black, and b indicates that B dresses in black. This protocol has four outcomes, which are represented by leaf nodes. Each arrow corresponds to an action that an agent can take. When B makes a decision, he knows what agent A has chosen. This property is called *perfect information.* In games such as poker this is not always the case, because a player does not know what cards

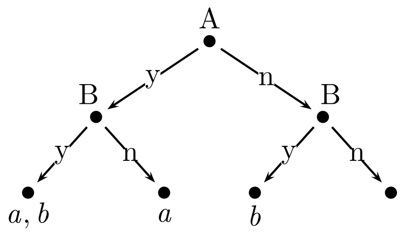

Figure 1: A simple game form G_1

the other players are holding. Poker is thus an imperfect information game. In this paper only perfect information games are discussed. One reason for this is that these games are easier to implement, because there is no need to hide any information from anyone.

This paper is structured as follows. The next section, section 2, contains basic definitions. Section 3 presents an axiom system for this logic. In section 4 related work is discussed and section 5 is the conclusion.

2 Definitions

Extensive games can be represented in different ways. The next definitions use sequences of actions to represent a game tree [7].

Definition 1. *A set of finite sequences H is* prefix-closed *if for any sequence h and action a it is the case that $ha \in H$ implies $h \in H$. For any set of sequences H and $h \in H$ we define the set of next actions $A(H,h) = \{a \mid ha \in H\}$ and the set of terminal sequences $Z(H) = \{h \in H \mid A(H,h) = \emptyset\}$.*

Sequences of actions can be used to denote specific plays of a game. Such sequences are also called histories or runs. $Z(H)$ denotes the set of all sequences that cannot be extended. These are called terminal histories or sequences, and correspond to outcomes. The set $A(H,h)$ consists of all actions that can be played in h. The set H implicitly defines a tree, since one can think of H as containing all paths in the tree that start from the root and go down the tree.

Definition 2. *An* interpreted extensive game form *F is a tuple $F = (\Sigma, H, \mathit{turn}, P, \pi)$, where Σ is a finite sets of agents, P is a finite set of atomic propositions, H is a non-empty, prefix-closed set of finite sequences, turn is a function $\mathit{turn} : H \setminus Z(H) \rightarrow \Sigma$ and $\pi : Z(H) \rightarrow 2^P$ returns the true atomic propositions of any terminal history.*

A game form is different from a game because it does not contain the preferences of players. The word 'interpreted' indicates that the structure contains atomic propositions. These atomic propositions allow us to evaluate propositional logic formulas in the outcome states. Strategies are often defined only for single agents. Here a slightly more general definition is used that allows for coalition strategies. Each strategy is intended for a set of agents. A strategy returns exactly one action. This is often called a pure strategy [7].

Definition 3. *Let $F = (\Sigma, H, \mathit{turn}, P, \pi)$ be a game form and $\Gamma \subset \Sigma$ a coalition of agents. A strategy σ_Γ for Γ is a function with domain $\{h \in H \mid \mathit{turn}(h) \in \Gamma\}$ such that $\sigma_\Gamma(h) \in A(H,h)$.*

Definition 4. *Let $F = (\Sigma, H, \mathit{turn}, P, \pi)$ be a game form and σ_Γ a strategy. The updated model $F' = Up(F, \sigma_\Gamma)$ is defined as $F' = (\Sigma, H', \mathit{turn}', P, \pi')$ where H' is the largest subset of H such that $ha \in H'$ implies $h \in H'$ and either $\mathit{turn}(h) \notin \Gamma$ or $a = \sigma_\Gamma(h)$.*

An example game form has already been introduced and displayed in figure 1. This game form can be described as $F_1 = (\Sigma, H, \mathit{turn}, P, \pi)$ with $\Sigma = \{A, B\}$, $P = \{a, b\}$, $H = \{\epsilon, y, n, yy, yn, ny, nn\}$. The function *turn* is defined such that $\mathit{turn}(\epsilon) = A$ and $\mathit{turn}(y) = \mathit{turn}(n) = B$. The interpretation function π satisfies the following: $\pi(yy) = \{a, b\}$, $\pi(yn) = \{a\}$, $\pi(ny) = \{b\}$ and $\pi(nn) = \emptyset$. Agent A has two strategies σ_A in this example, and agent B has four strategies.

Logical Language

In this subsection the language GLP^- is presented. This language depends on a set of agents Σ and a set of atomic propositions P, but these are omitted for readability. This language is a restriction of the language GLP, which stands for *Game Logic with Preferences* [9].

Definition 5. *Let $con(L) = \{\bot, \phi \to \psi \mid \phi, \psi \in L\}$ and let propositional logic $\mathcal{P}$ be defined as the smallest language L such that $L = P \cup con(L)$. Let $glp^-(L) = \{[\Gamma : \phi]\Box\phi, \Box\phi \mid \phi \in \mathcal{P}, \Gamma \subset \Sigma\}$. The language GLP^- is the smallest language L such that $L = glp^-(L) \cup con(L)$.*

In this definition we use a minimal set of logical connectives, consisting of implication and falsum. All other usual connectives can be defined in the following way.

$$\begin{aligned} \neg\phi &= \phi \to \bot & \phi \leftrightarrow \psi &= (\phi \to \psi) \wedge (\psi \to \phi) \\ \phi \vee \psi &= \neg\phi \to \psi & \phi \nabla \psi &= \neg(\phi \leftrightarrow \psi) \\ \phi \wedge \psi &= \neg(\phi \to \neg\psi) & \Diamond\phi &= \neg\Box\neg\phi \end{aligned}$$

The two languages propositional logic and GLP^- are distinct. The formulas $p \vee \neg p$ and $p \to q$ are propositional logic formulas, but are not themselves GLP^- formulas. Propositional logic is interpreted over a set $T \subset P$ of atomic propositions.

$T \models \bot$		never
$T \models p$	iff	$p \in T$
$T \models \phi \to \psi$	iff	not $T \models \phi$ or $T \models \psi$

The logic GLP^- is interpreted over an interpreted extensive game form $F = (\Sigma, H, turn, P, \pi)$.

$F \models \bot$		never
$F \models \phi \to \psi$	iff	not $F \models \phi$ or $F \models \psi$
$F \models \Box\phi$	iff	$\forall h \in Z(H) : \pi(h) \models \phi$
$F \models [\Gamma : \phi]\Box\phi$	iff	$\exists\sigma_\Gamma \forall h \in Z(H') : \pi'(h) \models \phi$ where $(\Sigma, H', turn', P, \pi') = Up(F, \sigma_\Gamma)$

Intuitively, the box $\Box\phi$ is a universal quantifier. It expresses that ϕ holds in every state. The construction $[\Gamma : \phi]\Box\phi$ expresses that Γ has a strategy so that if it uses this strategy, any reachable outcome satisfies ϕ. The next table lists properties that are true for the example F_1.

$$\begin{aligned} F_1 &\models [A : a]\Box a \wedge [A : \neg a]\Box\neg a \\ F_1 &\models [B : b]\Box b \wedge [B : \neg b]\Box\neg b \\ F_1 &\models [B : a \leftrightarrow b]\Box(a \leftrightarrow b) \wedge [B : a\nabla b]\Box(a\nabla b) \end{aligned}$$

One can conclude that agent B, because it goes second, can control more. If fairness demands that A and B have the same amount of influence, then this protocol is not fair.

3 Completeness

Using the interpretation of GLP^-, one can test whether a candidate protocol F has a desired property ϕ by checking whether $F \models \phi$. This is called model checking [4] and for many logics there are computer programs that can do this efficiently. It has been proven that for GLP, and thus for GLP^-, the model checking problem is tractable [9]. In this section we focus on the more difficult problem to determine whether there exists a protocol F that satisfies a given property ϕ. We define an axiom system that can be used to construct proofs. If a formula ϕ has no model, then a proof of $\neg\phi$ exists. If $\neg\phi$ cannot be proven then the completeness proof suggests a method for constructing a protocol F so that $F \models \phi$.

The notation $\vdash \phi$ is used to express that it has been proven that $F \models \phi$ holds for all models F. If ϕ is an instance of one of the axioms given below, then one can immediately conclude $\vdash \phi$. If one has established $\vdash \phi$ and $\vdash \phi \rightarrow \psi$, then the rule *modus ponens* can be used to conclude $\vdash \psi$. These are the only rules that we allow. This Hilbert style of proofs is common in modal logic [3].

The next table lists four axioms that can be written without the coalition operator $[\Gamma : \phi]$. For the greek letter τ one may substitute any instance of any propositional logic tautology that one can contain using uniform substitution. For instance $[\Sigma : p]\Box p \vee \neg[\Sigma : p]\Box p$ is an instance of $p \vee \neg p$. For all other greek letters one may substitute any propositional logic formula.

τ	TAUTOLOGY
$\Box\tau$	BOX-TAUTOLOGY
$\Box\phi \rightarrow \Diamond\phi$	SERIALITY
$\Box(\phi \rightarrow \psi) \rightarrow (\Box\phi \rightarrow \Box\psi)$	DISTRIBUTION

One can prove, but this is not done here, that these axioms are complete for the fragment of GLP^- in which the construction $[\Gamma : \phi]$ is not used. The axiom SERIALITY follows from the fact each game form must have a non-empty set of runs.

The remaining axioms are listed in the table below. One may substitute any formula for ϕ, ψ, as long as the result is in GLP^-.

$([\Gamma : \phi]\Box\phi \wedge \Box(\phi \rightarrow \psi)) \rightarrow [\Gamma : \psi]\Box\psi$	SPECIFICITY
$[\Gamma : \phi]\Box\phi \leftrightarrow \neg[\Sigma\backslash\Gamma : \neg\phi]\Box\neg\phi$	MINIMAX
$[\Gamma : \phi]\Box\phi \rightarrow [\Gamma \cup \Gamma_2 : \phi]\Box\phi$	MONOTONICITY
$[\emptyset : \phi]\Box\phi \leftrightarrow \Box\phi$	NOBODY

All axioms that are given here are sound, which means that for each axiom ϕ and game form F it is the case that $F \models \phi$. The proofs are not hard and have been omitted for space considerations. In the remainder of this section it is proven that this set of axioms is complete: that each valid formula ϕ has a proof.

Definition 6. *A formula ϕ is* valid *iff for any model F it is that case that $F \models \phi$. A formula ϕ is* satisfiable *if there is a model F so that $F \models \phi$. A formula is* consistent *if not $\vdash \neg\phi$.*

Definition 7. *A set of formulas $S \subset GLP^-$ is* maximally consistent *if the following conditions are all met: $\bot \notin S$, all axioms $\phi_A \in S$, if $\phi, \phi \rightarrow \psi \in S$ then $\psi \in S$, and for any formula ϕ either $\phi \in S$ or $\neg\phi \in S$.*

It is a well known fact that for each consistent formula ϕ one can find a consistent set S so that $\phi \in S$ [3]. Furthermore it can be proven that for any model F, the set $\{\phi \mid F \models \phi\}$ is maximally consistent.

Definition 8. *A formula of the form $\Box\phi$ is* basic*. A formula of the form $[\Gamma : \phi]\Box\phi$ is called* simple*. For any set S of formulas, a simple formula $[\Gamma : \psi]\Box\psi \in S$ is* specific *if there is no $[\Gamma : \chi]\Box\chi \in S$ so that $\Box(\chi \rightarrow \psi) \in S$ but not $\Box(\psi \rightarrow \chi) \in S$.*

The next lemma tells us that it is enough to check only formulas that are simple and specific to ensure that two maximally consistent sets are the same.

Lemma 1. *Suppose that S and T are maximally consistent sets. Let S' contain all basic and all specific formulas of S and T' all basic and all specific formulas of T. If $S' = T'$ then $S = T$.*

Proof. Suppose that S and T are maximally consistent sets. Let S' contain all basic and all specific formulas of S and T' all basic and all specific formulas of T. Suppose also that $S' = T'$. Let $\xi = [\Gamma : \psi]\Box\psi \in S$. We have to show that $\xi \in T$. If ξ is specific, then $\xi \in S'$, thus $\xi \in T'$ and $\xi \in T$. If not, then there is some 'more specific' formula $[\Gamma : \chi]\Box\chi \in S$ so that $\Box(\chi \to \psi) \in S$. This formula itself need not be specific, since there might be an even more specific formula that rules out $[\Gamma : \chi]\Box\chi$. However, since P is finite, there is only a finite number of possible assignments. This means there must be a specific formula $[\Gamma : \chi]\Box\chi \in S'$ and $\Box(\chi \to \psi) \in S$. Since $S' = T'$ we know that $[\Gamma : \chi]\Box\chi \in T'$ and thus $[\Gamma : \chi]\Box\chi \in T$. Since $\Box(\chi \to \psi) \in S$ is basic, we can conclude that $\Box(\chi \to \psi) \in S' = T' \subset T$. Using the axiom SPECIFITY and the fact that T is maximally consistent, we conclude that $[\Gamma : \psi]\Box\psi \in T$.

It is now proven that S and T contain the same simple formulas. Consider now a formula of the form $\neg[\Gamma : \psi]\Box\psi$. If $\neg[\Gamma : \psi]\Box\psi \in S$, axiom MINIMAX can be used to show that $[\sigma \setminus \Gamma : \neg\psi]\Box\neg\psi \notin S$. Since this is a simple formula, we conclude that $[\sigma \setminus \Gamma : \neg\psi]\Box\neg\psi \notin T$. Using MINIMAX again we obtain $\neg[\Gamma : \psi]\Box\psi \in T$.

A useful property of maximally consistent sets is that if $\phi \wedge \psi \in S$ then $\phi \in S$ and $\psi \in S$. Moreover if $\phi \vee \psi \in S$ then $\phi \in S$ or $\psi \in S$ (or both). For every formula $\phi \in S$ in conjunctive form we can conclude that $\phi \in T$. Since every propositional formula is equivalent to a formula in conjunctive normal form, we may conclude that for any formula ϕ it is the case that $\phi \in S \Leftrightarrow \phi \in T$. Therefore $S = T$. □

Theorem 1. *For all $\phi \in GLP^-$: If ϕ is consistent then ϕ is satisfiable.*

Proof. Let a consistent formula $\phi \in GLP^-$ be given. Let S be a maximally consistent set so that $\phi \in S$ and let S' contain all basic and specific formulas of S. Below a model F is constructed so that $\forall\psi \in S' : F \models \psi$. Lemma 1 can then be used to conclude that $F \models \phi$.

The model F we are searching for is defined recursively using a function $F(C, \mathcal{A}, r)$. The outcome of this function depends on a set of basic and simple formulas C, on a set of active agents $\mathcal{A}$ and on a representation function $r : \Sigma \to 2^\Sigma$. The set $r(X)$ contains the agents that are represented by agent X. The model F is defined as $F = F(S', \mathcal{A}_0, r_0)$. Initially, all agents are active agents: $\mathcal{A}_0 = \Sigma$. Each agent initially only represents itself: $r_0(X) = X$. The function r can also be applied to coalitions of agents. This is defined by $r(\Gamma) = \cup_{X \in \Gamma} r(X)$. The pair $\mathcal{A}, r$ can be used calculate a new set set of simple and basic formulas $S(C, \mathcal{A}, r)$ from a given subset C.

$$S(C, \mathcal{A}, r) = \{\Box\psi \mid \Box\psi \in C\} \cup \{[\Gamma : \psi]\Box\psi \mid \Gamma \subset \mathcal{A}, [r(\Gamma) : \psi]\Box\psi \in C\}$$

The game form $F(C, \mathcal{A}, r)$ is defined in the following way. If $\mathcal{A}$ contains exactly one active agent X, then we define a model $F(C, \mathcal{A}, r) = (\Sigma, H, turn, P, \pi)$ where $H = \{\epsilon, \psi \mid [X : \psi]\Box\psi \in C, [X : \psi]\Box\psi \text{ is specific}\}$. Define $turn(\epsilon) = X$ and let $\pi(\psi)$ be any set such that $\pi(\psi) \models \psi$. It is not hard to show that any formula $\chi \in C$ is satisfied by this model.

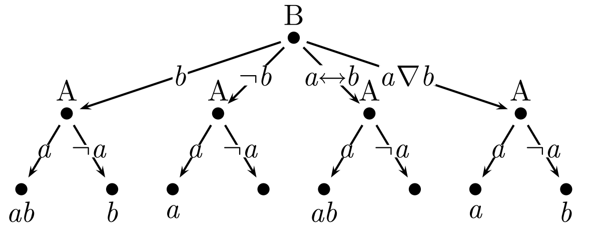

Figure 2: Alternative F_2

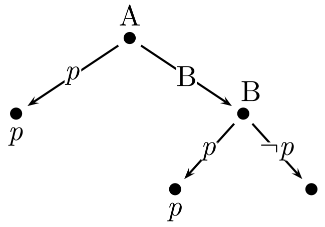

Figure 3: Game form F_3

If $\mathcal{A}$ has two or more members, define $F(C, \mathcal{A}, r) = (\Sigma, H, turn, P, \pi)$ as follows. Take any agent $X \in \mathcal{A}$. Define $turn(\epsilon) = X$, so that this becomes the acting agent of the current situation. The set of options $A(H, \epsilon)$ consists of two parts: $A(H, \epsilon) = E \cup J$. The set E consists of all specific choices of X: $E = \{\psi_e \mid [X : \psi_e]\Box\psi_e \in C \text{ is specific}\}$. These choices lead to the subgame $F(C', \mathcal{A}, r)$ where $C' = \{\Box\psi_e, \Box\psi, [\Gamma : \chi]\Box\chi \mid \Box\psi, [\Gamma : \chi]\Box\chi \in C, X \notin \Gamma\}$. The set J contains all other active agents: $J = \{Y \in \mathcal{A} \mid Y \neq X\}$. These choices Y lead to the subgames $F(C^Y, \mathcal{A} \setminus \{X\}, r')$ where $C^Y = S(C, \mathcal{A} \setminus \{X\}, r')$, and r' is such that $r'(Y) = \{Y, X\}$ and $r'(Z) = \{Z\}$ for $Z \neq Y$. Intuitively, choosing Y means that agent Y will now make all decisions for agent X.

We must show that $F(C, \mathcal{A}, r)$ satisfies all formulas in C. This is done using induction. The induction hypothesis is that all smaller models have this property. First consider formulas of the form $\Box\phi \in C$. These formulas are present in each set C' that is used to construct a subgame. Using the induction hypothesis we know that all outcomes of all choices satisfy ϕ, and thus $F(C, \mathcal{A}, r) \models \Box\phi$.

Consider now $[\Gamma : \psi]\Box\psi$ with $X \notin \Gamma$. This formula is also present in any set C' and by induction hypothesis we know that there is thus a strategy in each subgame for Γ to ensure ψ. We can combine these subgame strategies into a strategy σ_Γ for the whole game that guarantees ψ, and thus $F(C, \mathcal{A}, r) \models [\Gamma : \psi]\Box\psi$.

Finally consider $[\Gamma : \psi]\Box\psi$ with $X \in \Gamma$. If $\Gamma = \{X\}$ then there is some specific $\psi_e \in E$ so that $\Box(\psi_e \to \psi) \in C$. This choice leads to a submodel $F(C', \mathcal{A}, r)$. From $\psi_e \in C'$ and the induction hypothesis it follows that there is a strategy σ_X for this submodel that guarantees ψ. Agent X can now use a strategy σ'_X so that $\sigma_X(\epsilon) = \{\psi_e\}$ and within the subgame $F(C', \mathcal{A}, r)$, strategy σ'_X makes the same choices as σ_X. This strategy guarantees ψ and thus $F(C, \mathcal{A}, r) \models [\Gamma : \psi]\Box\psi$. If there is more than one agent in Γ, then X can join any of the other agents $Y \in \Gamma$. By induction the coalition $\Gamma \setminus \{X\}$ will have a strategy for guaranteeing ψ in the subgame $F(C', \mathcal{A} \setminus \{X\}, r')$, and thus $F(C, \mathcal{A}, r) \models [\Gamma : \psi]\Box\psi$.

The model $F = F(S', \mathcal{A}_0, r_0)$ thus satisfies all formulas $\psi \in S'$. From lemma 1 it follows that F satisfies all formulas in S and thus $F \models \phi$. □

Corollary 1. *For any protocol F there is an equivalent protocol F' in which each agent only moves once, and all agents move in a given order.*

This corollary can be illustrated for the example protocol of figure 1. According to a proof there should be an equivalent protocol in which agent B moves first. This is indeed the case, and the protocol is illustrated in figure 2. One can see that B in this case can choose from four options.

In the construction of the proof, each agent has a choice whether it want to use one of its abilities (set E) or whether it wants to join a specific agent (set J). In order to illustrate these two possibilities, consider the property $\phi_3 = [A : p]\Box p \wedge [B : p]\Box p \wedge [AB : \neg p]\Box\neg p$. There is only one atomic proposition in this example, so $P = \{p\}$. There are only four distinct formulas that one can express: $p, \neg p, \bot, p \rightarrow p$. Suppose that S is a maximally consistent set containing ϕ_3. The ability $[A : p]\Box p$ is more specific than $[A : p \rightarrow p]\Box(p \rightarrow p)$. Thus agent A has one specific ability p. In the game form F_3 that is constructed in the proof, agent A has two options. It can use this ability, or it can join agent B. The game form is depicted in figure 3. In this protocol agent A and B have exactly the same amount of influence on the outcome, thus one could call this protocol fair.

4 Related Work

Alternating time temporal logic is an extension of temporal logic that, like GLP^-, allows one to specify the powers of coalitions of agents [2]. Initially ATL was mostly used for model checking, using the model checker Mocha [1]. Only recently has a complete axiom system for ATL been discovered [6]. This system was inspired by Pauly's completeness proof for coalition logic [8]. The logic ATL is more expressive and thus complex than GLP^- because it extends temporal logic. Another difference between GLP^- and ATL and coalition logic is that GLP^- only interprets propositions at outcome states. In the other two logics outcome states are not treated differently from intermediate states. The most striking difference however is the fact that both ATL and coalition logic allow for concurrent moves. The class of game forms that they allow is thus wider. Once consequence is that the minimax axiom does not hold for ATL and coalition logic.

In future work I hope to compare complexity results for mechanism design using the presented axiom system with non-logical approaches for the automated design of protocols [5].

5 Conclusion

The logic GLP^- was introduced for verification of multi agent protocols. This paper contains a complete axiom system for this logic. This makes the logic suitable for automated design of protocols, and allows for more elegant verification. One of the corollaries of the proof is that the order in which agents move does not matter. For every ordering of agents and every consistent speficication one can find a protocol. This result, and the fact that this logic does not allow concurrent moves, sets this logic apart from other frameworks such as ATL and coalition logic.

Acknowledgement: Olivier Roy has provided many valuable comments and suggestions.

References

[1] R. Alur, L. de Alfaro, T. A. Henzinger, S. C. Krishnan, F. Y. C. Mang, S. Qadeer, S. K. Rajamani, and S. Taşiran. MOCHA user manual. University of Berkeley Report, 2000.

[2] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time temporal logic. In *Proceedings of the 38th IEEE Symposium on Foundations of Computer Science*, pages 100–109, Florida, October 1997.

[3] P. Blackburn, M. de Rijke, and Y. Venema. *Modal Logic.* Cambridge University Press: Cambridge, England, 2001.

[4] E. M. Clarke, O. Grumberg, and D. A. Peled. *Model Checking.* The MIT Press: Cambridge, MA, 2000.

[5] V. Conitzer and T. Sandholm. Complexity of mechanism design. In *Proceedings of the Uncertainty in Artificial Intelligence Conference (UAI), Edmonton, Canada.*, 2002.

[6] V. Goranko and G. van Drimmelen. Decidability and complete axiomatization of the alternating-time temporal logic. *Theoretical Computer Science*, pages 1–39, to appear.

[7] M. J. Osborne and A. Rubinstein. *A Course in Game Theory.* The MIT Press: Cambridge, MA, 1994.

[8] M. Pauly. A modal logic for coalitional power in games. *Journal of Logic and Computation*, 12:149–166, 2002.

[9] S. van Otterloo, W. van der Hoek, and M. Wooldridge. Preferences in game logics. In *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, New York, July 2004.