Journal of Logic, Language and Information (2006) 15: 425–452
DOI: 10.1007/s10849-006-9014-1

Knowledge Condition Games

SIEUWERT VAN OTTERLOO[1,*], WIEBE VAN DER HOEK[2]
and MICHAEL WOOLDRIDGE[2]
[1]*Eemstraat 35, 3522 SN Utrecht*
E-mail: sieuwert@bluering.nl
[2]*University of Liverpool, Department of Computer Science, Liverpool L69 7ZF*
E-mails: wiebe@csc.liv.ac.uk, mjw@csc.liv.ac.uk

(Received 1 June 2005; in final form 20 December 2005)

Abstract. Understanding the flow of knowledge in multi-agent protocols is essential when proving the correctness or security of such protocols. Current logical approaches, often based on model checking, are well suited for modeling knowledge in systems where agents do not act strategically. Things become more complicated in strategic settings. In this paper we show that such situations can be understood as a special type of game – a *knowledge condition game* – in which a coalition "wins" if it is able to bring about some epistemic condition. This paper summarizes some results relating to these games. Two proofs are presented for the computational complexity of deciding whether a coalition can win a knowledge condition game with and without opponents (Σ_2P-complete and NP-complete respectively). We also consider a variant of knowledge condition games in which agents do not know which strategies are played, and prove that under this assumption, the presence of opponents does not affect the complexity. The decision problem without opponents is still NP-complete, but requires a different proof.

Key words: complexity, epistemic logic, game theory, imperfect information, knowledge, protocol, strategy

1. Introduction

Strategic interaction, be it in cooperation or in coordination, has been subject of study in economics and game theory (von Neumann and Morgenstern, 1944), the social sciences (Schelling, 1960) and more recently multi-agent systems (van der Hoek and Wooldridge, 2003) and logic (Pauly, 2001). In this paper, we focus on the interplay between *strategic* and *informational* aspects of interaction. Game theorists have for a long time acknowledged the intricate interplay between these two aspects. On the one hand, there is a long list of examples of epistemic conditions that guarantee specific solutions in games, such as Aumann's celebrated account of how common knowledge of rationality justifies the algorithm of backward induction

*Sieuwert van Otterloo thanks the Institute for Logic, Language and Information in Amsterdam for its hospitality during the period that this paper was finalized.

to find Nash equilibria (Aumann, 1995); for a contemporary treatment of several similar issues, see also (de Bruin, 2004). Such examples address general epistemic assumptions about the overall game. On the other hand, there are approaches that attempt to make explicit what the players know in every stage of the game, one of the motivations being that revealing or hiding specific information during an interaction may be of a strategic interest to some of its participants. It is the latter issue that we address in this paper, adding epistemic properties to the state of affair that players or agents want to achieve or, to the contrary, avoid.

Research in multi-agent systems has experienced a flourishing interest in formal approaches to cooperation and interaction, in which languages represent the reasoning of or about agents in coalitions, and models typically represent the effect of agents forming coalitions over time. Since a key feature of agency is *autonomy* cf. (Wooldridge and Rao, 1999), rather than fully determining an agent's behaviour in a deterministic program, the idea is that an agent is only given a pre-defined *protocol*, in which his course of actions may be constrained, but not uniquely prescribed.

The automatic verification of protocols – either real-world protocols such as voting mechanisms, or electronic communication protocols such as electronic auction and network protocols – is an important topic in computer science (Holzmann, 1991). It is becoming ever clearer that the required properties of these protocols often involve the presence or absence of *knowledge*: the information possessed (or not possessed) by the agents that enact the protocol (Fagin et al., 1995). At the same time, it is realised that in many protocols – such as electronic auctions – it is necessary to take into account the *strategic* behaviour of participants. They are not bound to behave as the designer of the protocol desires, but will attempt to obtain the best result possible for themselves. The combination of these two ingredients – knowledge and strategic behaviour – makes formal verification a difficult problem. In this paper, we define a new class of games, which are intended to be an abstract formal model of such protocols. We refer to these games as *knowledge condition games*. In a knowledge condition game, two coalitions of agents enact a protocol. One coalition strives to reach a certain knowledge situation, and the other coalition tries to prevent the first coalition from reaching its goal. In other words, one coalition "wins" if it is able to force a certain condition to hold in the world, where this condition relates to the knowledge (and absence of knowledge) of the agents in the game. Formally, we specify the goal situation (i.e., the epistemic condition that the agents strive to achieve) using epistemic logic, and protocols are modeled as extensive game forms with imperfect information.

One can make many different choices when modeling the knowledge of agents in a protocol. One issue is whether the strategies that agents use are publicly known, or whether they are assumed to be private. We formulate two variants of the knowledge condition games and their associated decision problem. In the first variant all agents know which strategy each agent uses, while in the second variant no agent knows which strategies are being used.

For both variants one can define a decision problem. In these problems one has to decide whether the first coalition has a winning strategy for the knowledge condition game. We then determine the computational complexity of this decision problem for both variants.

The structures over which knowledge condition games are played in this paper are 'game trees', or, more formally, extensive game forms with imperfect information (Osborne and Rubinstein, 1994). These structures specify which agent can act at a point in the protocol, and which actions this agent can choose from. A game tree does not specify any preferences or winning conditions. Instead of defining these winning conditions in a conventional way (by saying how good each single outcome is for each agent), we calculate what each agent knows by the end of the protocols. This depends on the strategies that agents have chosen. We can then specify a certain knowledge property (i.e. 'A knows ϕ and B does not know ψ'), and define a group of agents that wants to make this property true, while another group wants the property not to hold. The first group wins if the knowledge property holds in all reachable outcomes, otherwise the second group of agents wins.

There is an increasing body of work on the *logical* properties of games, and in particular on strategic and epistemic properties (see Section 5). However, in this paper we choose to focus not on the logical properties of knowledge condition games, but on the *computational complexity* of determining who wins a knowledge condition game under various assumptions. There are several reasons for this choice of emphasis.

– First, it is important to know for applications such as automated verification whether these problems are tractable, and if not, what special cases might be tractable.
– Second, the complexity results give us an insight into how these games are different from other games or approaches, and what makes these problems difficult.

The structure of this paper is as follows. In Section 2, we present the definitions required for the remainder of the paper: the section starts off with epistemic logic, then covers interpreted game forms, and ends with strategic games and knowledge condition games. Section 3 provides several extended examples of knowledge condition games. The first example shows how knowledge properties are important in a voting protocol; the second example involves a more playful quiz problem. It shows how signaling can enter into reasoning about knowledge. Section 4 presents four results relating to the complexity of knowledge condition games. We prove the complexity of deciding a knowledge condition game in which strategies are known, first for the restricted case of no opponents, then in general; we then do the same for knowledge condition games in which strategies are unknown. Section 5 discusses some related work, and Section 6 presents some conclusions.

Table I. Summary of key notational conventions.

Notation	Meaning
F	an interpreted game form
G	a strategic game (for instance a knowledge condition game)
Σ	the set of all agents
A, B, C, N, X, Y	individual agents. A, B, C, N are used as actual names of agents in the examples, like 1, 2, 3 are actual numbers. X, Y denote agents in the definitions and proofs
Γ, Ξ	sets of agents
ϕ, ψ, ξ	formulas
$\mathcal{M}$	an epistemic logic model
w	a state in a model
a	a single action
h	a sequence of actions
H	set of sequences of actions
P	a set of atomic propositions
π	an interpretation function
$\mathfrak{U}$	a utility function
u	an update function
m	a model extraction function

2. Preliminary Definitions

In this section we define how one can create a *knowledge condition game G* from an *interpreted game form F.* We begin by defining *epistemic logic, interpreted game forms, strategies* and *updates*, which are all needed in order to define knowledge condition games.

First, note that it seems impossible to find a notation that is consistent with both logical and game theoretical conventions. Therefore the choice of symbols in this paper is arguably rather arbitrary; but it is at least consistent throughout the paper. A summary of our key notational conventions is given in Table I.

2.1. Epistemic Logic

In order to express statements about knowledge we use the language of epistemic propositional logic, along with its $S5_n$ semantics (Fagin et al., 1995; Meyer and van der Hoek, 1995). This language is called $\mathcal{L}$ in this paper. The language $\mathcal{L}$ is parameterised by a finite set Σ of agents and a finite set P of atomic propositions, and where we need to make this clear, we identify the particular language parameterised by Σ and P by $\mathcal{L}(\Sigma, P)$. However, where no confusion is possible we suppress reference to these, simply writing $\mathcal{L}$.

DEFINITION 1. Let Σ be a finite set of agent symbols (with typical element X), and P a finite set of propositional atoms (with typical element p). The language $\mathcal{L}(\Sigma, P)$ (with typical element ϕ is defined through the following BNF grammar:

$$\phi ::= p \mid \bot \mid \phi \to \phi \mid K_X\phi | C\phi$$

An example formula in this language is $\phi_0 = K_B p$. This formula expresses that B knows that p holds. It is useful to define a few more operators in terms of the existing ones. Negation is defined by $\neg\phi = \phi \to \bot$. Disjunction is defined by $\phi \vee \psi = \neg\phi \to \psi$. Conjunction is defined by $\phi \wedge \psi = \neg(\neg\phi \vee \neg\psi)$. Exclusive or is defined by $\phi \nabla \psi = (\phi \to \neg\psi) \wedge (\neg\psi \to \phi)$. Epistemic possibility ('X considers it possible that...') is defined by $M_X\phi = \neg K_X \neg\phi$. The operator $C\phi$ expresses that ϕ is commonly known by all agents.

This language is interpreted over *Kripke models*, the standard semantic structures for modal epistemic logics (Meyer and van der Hoek, 1995).

DEFINITION 2. A (multi-agent) Kripke model $\mathcal{M}$ is a tuple

$$\mathcal{M} = (\Sigma, W, \sim, P, \pi),$$

where:

- Σ is a set of agents;
- W is a set of states or worlds;
- $\sim$ is a collection of equivalence relations $\sim_X \subseteq W \times W$ between states, one for each agent $X \in \Sigma$;
- P is a set of atomic propositions; and
- $\pi : W \to 2^P$ is an interpretation function.

As usual, the equivalence relations capture each agent's knowledge/ignorance about the state of the game: $w \sim_X w'$ means that the states w and w' "look the same" according to agent X. Thus $\sim_X$ relates states that X cannot tell apart. The function π returns for all states w a set $\pi(w) \subseteq P$ with the atomic propositions that are true in w.

An example Kripke model is the model $\mathcal{M}_0$, which has two states w_1 and w_2 and two agents A and B. Agent A can distinguish these states, but agent B cannot: $w_1 \sim_B w_2$. In this model only one atomic proposition occurs: p. This proposition only holds in state w_1.

Epistemic formulas $\phi \in \mathcal{L}$ can be interpreted over Kripke models. Let $\mathcal{M} = (\Sigma, W, \sim, P, \pi)$ be a Kripke model and $w \in W$. We define a relation $\models$ between models with states $\mathcal{M}, w$ and formulas ϕ and write $\mathcal{M}, w \models \phi$ if this relation holds. In this case we say that ϕ is true in w. If it is not the case that $\mathcal{M}, w \models \phi$, then we write $\mathcal{M}, w \not\models \phi$.

DEFINITION 3. Let $\mathcal{M} = (\Sigma, W, \sim, P, \pi)$ be a Kripke model, $w \in W$, $p \in P$, $X \in \Sigma$ and $\phi \in \mathcal{L}(\Sigma, P)$.

$$\begin{array}{lll}
\mathcal{M}, w \models p & \text{iff} & p \in \pi(w) \\
\mathcal{M}, w \models \bot & & \text{never} \\
\mathcal{M}, w \models \phi \to \psi & \text{iff} & \mathcal{M}, w \models \phi \text{ implies } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models K_X\phi & \text{iff} & \forall v \in W : w \sim_X v \text{ implies } \mathcal{M}, v \models \phi \\
\mathcal{M}, w \models C\phi & \text{iff} & \forall v \in W : w \sim^* v \text{ implies } \mathcal{M}, v \models \phi
\end{array}$$

The relation $\sim^*$ is the reflexive, transitive closure of the union $\bigcup_X(\sim_X)$.
The notation $\mathcal{M} \models \phi$ is used to indicate that a formula holds in all states of the given model.

$$(\Sigma, W, \sim, P, \pi) \models \phi \Leftrightarrow \forall w \in W : (\Sigma, W, \sim, P, \pi), w \models \phi$$

We can use this definition to show that in the example model $\mathcal{M}_0$, it is the case that $\mathcal{M}_0, w_1 \models K_A p$ and $\mathcal{M}_0, w_1 \models \neg K_B p$ and $\mathcal{M}_0, w_2 \models K_B(p \vee \neg p)$.

2.2. Interpreted Game Forms

An *interpreted game form* characterises what might be called the "action structure" of a game: the actions that can be performed in any given state of play, and the possible outcomes of these actions. A game form can be described in different but equivalent ways, for instance as a set of sequences of actions, or as a tree. We follow Osborne and Rubinstein (1994) and use the idea of a set H of sequences h. Each sequence h is one possible sequence of actions by agents that is allowed by the rules of the game. The whole set H of them fully describes what can be done in the game.

DEFINITION 4. A set of finite sequences H is *prefix-closed* iff 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. We let $Z(H)$ denote the set of all sequences that cannot be extended. These are called *terminal* histories, and correspond to *outcomes* of the game. The set $A(H, h)$ consists of all actions that can be played in h. Note that 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. Thus one can consider the 'game tree' that is implied by a prefix-closed set.

In game theory, a game tree does not contain any information about the propositions that have been made true at different points of the game. For this reason, we introduce *interpreted* game forms. An interpreted game form is basically a game tree to which an interpretation function for atomic propositions has been added.

DEFINITION 5. An *interpreted game form F* is a tuple

$$F = (\Sigma, H, \text{turn}, \sim, P, \pi)\ ,$$

where:

– Σ is a finite set of agents;
– H is a non-empty, prefix-closed set of finite sequences;
– turn is a function turn: $H \setminus Z(H) \to \Sigma$;
– for each $X \in \Sigma$ the relation $\sim_X \subseteq H \times H$ is an equivalence relation between sequences;
– P is a finite set of atomic propositions; and
– $\pi : Z(H) \to 2^P$ returns the true atomic propositions of any terminal history.

where these components must satisfy the following condition:

$$\text{if turn}(h) = X \text{ and } h' \sim_X h \text{ then also turn}(h') = X \text{ and } A(H, h) = A(H, h').$$

(This definition is adapted from Osborne and Rubinstein (1994, p. 200)). We have extended the information sets such that agents also have information when they are not in charge, which is a common extension for logical purposes (van Benthem, 2001; Bonanno, 2004).

Atomic propositions can be used to refer to certain terminal histories, for instance to histories where an agent achieves a certain goal. The idea of annotating end states or terminal histories with logical propositions has been used before by Harrenstein et al. (2003) and the authors of the present article (van Otterloo et al., 2004).

An example interpreted game form F_0 is depicted in Figure 1. In this example, agent A can make a choice from two alternatives (numbered 1 and 2), one of which satisfies p. After this choice, A can distinguish these situations, but B cannot.

For every interpreted game form F we can calculate a Kripke model $\mathcal{M} = m(F)$ representing the knowledge in the end states of F. We do this by taking all the terminal histories of F as the set of states of $\mathcal{M}$. The states of the model $\mathcal{M}$ are all outcomes of the interpreted game form F, and two outcomes are related in $\mathcal{M}$ iff they are related in F.

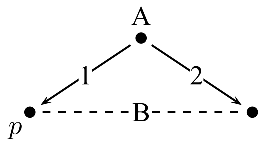

Figure 1. Interpreted game form F_0.

DEFINITION 6. Let $F = (\Sigma, H, \text{turn}, \sim, P, \pi)$ be an interpreted game form. The end situation model m is defined as $m(F) = (\Sigma, Z(H), \sim', P, \pi)$ where for each agent X, $\sim'_X$ is the restriction of $\sim_X$ to $Z(H) \times Z(H)$.

If we apply the function m to the example interpreted game form F_0, we get the example Kripke model $\mathcal{M}_0 = m(F_0)$. The transformation m is used to express when an interpreted game form F makes a formula ϕ true. The function m only uses the epistemic relation between end states. The relations between nonterminal sequences pose constraints on what strategies are allowed in Definition 7.

Note that if game form F is an imperfect recall game form, then the agents can have less information in the end situation than they had halfway in the game. We do not see this as a problem. One plausible situation in which this can happen is the case where the agents are computer programs. Such agents often have very limited memory. (Indeed, while Von Neumann and Morgenstern do consider imperfect recall games, such as two-team bridge (von Neumann and Morgenstern, 1953, p. 53), Selten once claimed that imperfect recall game forms "can be rejected as misspecified models of interpersonal conflict situations" (Selten, 1975).)

2.3. STRATEGIES

Strategies are an important part of every game. Informally a strategy σ_Γ is a function that tells all agents in coalition Γ what to do next in the histories they control. We use *non-deterministic* strategies for our agents. This means that a strategy does not return a unique option that the agent should take, but it returns a set of options, with the intention that the agent should randomly select an element of this set. Our strategies are thus akin to the randomized or 'mixed' strategies, or more correctly the *behavioural strategies*, of game theory (Osborne and Rubinstein, 1994, p. 212), although in this paper, we do not consider the common approach of introducing probability distributions over choices.

DEFINITION 7. Let $F = (\Sigma, H, \text{turn}, \sim, P, \pi)$ be an interpreted game form. A *strategy* σ_Γ is a function that for any node $h \in H \setminus Z(H)$ with $\text{turn}(h) \in \Gamma$ returns a non-empty set $\sigma_\Gamma(h) \subseteq A(H, h)$. A strategy must satisfy the constraint that $h \sim_{\text{turn}(h)} h'$ implies $\sigma_\Gamma(h) = \sigma_\Gamma(h')$.

The second part of the definition states that a strategy should not prescribe different options for histories that an agent cannot distinguish. An agent would not have the knowledge to adhere to a strategy that does not satisfy this condition. Strategies that satisfy the last constraint are sometimes called *uniform*.

For the example interpreted game form F_0 there are three different strategies for agent A. The strategy can either tell the agent to take the first option, or it can prescribe the second option, or the strategy can express that the agent should randomly choose between both options. Formally these possibilities are defined by respectively $\sigma^1_{\{A\}}(\epsilon) = \{1\}$, $\sigma^2_{\{A\}}(\epsilon) = \{2\}$ and $\sigma^3_{\{A\}}(\epsilon) = \{1, 2\}$.

For any strategy σ_Γ for an interpreted game form F we can consider a restricted interpreted game form F' in which the agents $X \in \Gamma$ only choose options that are part of the strategy. The agents $Y \notin \Gamma$ can still do whatever they could do F. Such a restricted interpreted game form models the situation in which coalition Γ is committed to the given strategy. The restricted model F' is computed by an *update function* $F' = u(F, \sigma_\Gamma)$.

DEFINITION 8. Let $F = (\Sigma, H, \text{turn}, \sim, P, \pi)$ be an interpreted game form. The update function u is defined by

$$u(F, \sigma_\Gamma) = (\Sigma, H', \text{turn}', \sim', P, \pi')\ ,$$

where:

- H' is the smallest subset of H such that $\epsilon \in H'$ and for each $h \in H'$ and $a \in A(H, h)$: if $\text{turn}(h) \notin \Gamma$ or $a \in \sigma_\Gamma(h)$ then $ha \in H'$;
- $\sim'$ is such that for all X: $\sim'_X = \sim_X \cap (H' \times H')$; and
- turn' and π' are the same as turn and π, but with their domain restricted to H'.

An update of the example F_0 with strategy $\sigma^3_{\{A\}}$, does not change anything: $u(F_0, \sigma^3_{\{A\}}) = F_0$. An update with $\sigma^1_{\{A\}}$ returns a model F_1 with only two histories: ϵ and 1. This means that the Kripke model of F_1 only has one state, in which p holds: $m(u(F_0, \sigma^1_{\{A\}})), 1 \models K_B p$.

2.4. Strategic Games

A distinction is often made in game theory between *extensive games* and *strategic games*. In an extensive game, agents take turns in selecting actions. In a strategic game the individual actions are not modeled: each player can select a strategy before the game starts, and somehow these strategies of all players together determine an outcome. Knowledge condition games are based on extensive game forms but are defined as strategic games. The next definition of a strategic game is therefore needed.

DEFINITION 9. A *strategic game* G is a tuple:

$$G = (\Sigma, \{S\}_\Sigma, \mathfrak{U}),$$

where:

- Σ is a finite set of agents;
- for each $X \in \Sigma$, S_X is a set of strategies for agent X; and
- $\mathfrak{U} : S^\Sigma \to \mathbb{R}^\Sigma$ a function that, for each choice of strategies, returns a *payoff vector.*

We are only interested in two-player, constant-sum, zero-one games, and in these games only two payoff vectors are possible: $(1, 0)$ which is best for the first player, and $(0, 1)$ which is best for the second player. In these games one can say that an agent can win if it has a strategy that guarantees that the agent gets utility 1. If the first player can win we write $w(G) = 1$.

DEFINITION 10. Let $G = (\{A, B\}, \{S_A, S_B\}, \mathfrak{U})$ be a two player constantsum zero-one game. The winner function w is defined by

$$w(G) = 1 \Leftrightarrow \exists\sigma_A \in S_A \forall\sigma_B \in S_B : \mathfrak{U}(\sigma_A, \sigma_B) = (1, 0)$$

2.5. Knowledge Condition Games

A *knowledge condition game* is two-player (or more accurately: two-team), constant-sum, zero-one strategic game. It is two-player in the sense that we are interested in two coalitions, i.e., sets of agents Γ and Ξ, playing against each other. These sets must be disjoint, but not every agent has to be in one of those sets. If an agent $X \in \Sigma$ is not in $\Gamma \cup \Xi$ then this agent is said to be *neutral*. The agents in Γ are called *proponents*, and the agents in Ξ *opponents*. To define a knowledge condition game, we must give an interpreted game form F and an epistemic logic formula ϕ: the proponents try to make this formula true on F, and the opponents try to make it false on F. Formally:

DEFINITION 11. Let $F = (\Sigma, H, \text{turn}, \sim, P, \pi)$ be an interpreted game form, $\Gamma, \Xi \subseteq \Sigma$ disjoint sets of agents and $\phi \in \mathcal{L}$ a knowledge formula. Define $\text{kcg}(F, \Gamma, \Xi, \phi) = (\{\Gamma, \Xi\}, \{S_\Gamma, S_\Xi\}, \mathfrak{U})$ where S_Γ, S_Ξ contain all strategies of Γ, Ξ in F respectively, and

$$\mathfrak{U}(\sigma_\Gamma, \sigma_\Xi) = \begin{cases} (1, 0) & \text{iff} \quad \forall w \in W : (\Sigma, W, \sim, P, \pi'), w \models \phi \\ (0, 1) & \text{otherwise} \end{cases}$$

where $(\Sigma, W, \sim, P, \pi') = m(u(u(F, \sigma_\Gamma), \sigma_\Xi))$.

In this definition, the proponent coalition Γ has a tougher task than the opponents Ξ, because Γ has to guarantee that ϕ holds in all cases. This has been done with applications in security in mind. In security settings, it is also necessary to secure a system against all possible scenarios and attacks. It therefore makes sense to give Γ the task of making sure that ϕ occurs.

Let F_0 be the example interpreted game form and take $\phi_0 = K_B p$. For the game $G_0 = \text{kcg}(F_0, \{A\}, \emptyset, \phi_0)$ we can compute a payoff matrix. As calculated before, $\{A\}$ has three strategies. The empty coalition only has the unique empty function $f_\emptyset$ as a strategy.

	$\sigma^1_{\{A\}}$	$\sigma^2_{\{A\}}$	$\sigma^3_{\{A\}}$
$f_\emptyset$	$(1, 0)$	$(0, 1)$	$(0, 1)$

We see that for this game, $\{A\}$ has a winning strategy (namely $\sigma^1_{\{A\}}$). Therefore $w(\text{kcg}(F_0, \{A\}, \emptyset, \phi)) = 1$. In the above definition, we use the updated model $m(u(u(F, \sigma_\Gamma), \sigma_\Xi))$ as a model for what all agents know. We have thus implicitly assumed that it is common knowledge to all agents which strategies are used by Γ and Ξ. This is a reasonable assumption if one considers strategies as well known conventions. Also in other circumstances, for instance if the game is played by computer programs that are open for inspection, this is a reasonable assumption. In some circumstances, however, one might not want to make this assumption. Therefore we present below a variant kcg′ of knowledge condition games in which the knowledge formula ϕ is evaluated in the original model $m(F)$. The strategies are used to determine the reachable states w. The proponents win if in all these states w, it is the case that $m(F), w \models \phi$.

DEFINITION 12. Let $F = (\Sigma, H, \text{turn}, \sim, P, \pi)$ be an interpreted extensive game form, Γ, $\Xi \subseteq \Sigma$ disjoint sets of agents and $\phi \in \mathcal{L}$ a knowledge formula. Define $\text{kcg}'(F, \Gamma, \Xi, \phi) = (\{\Gamma, \Xi\}, \{S_\Gamma, S_\Xi\}, \mathfrak{U}')$ where S_Γ, S_Ξ contain all strategies of Γ, Ξ in F respectively, and

$$\mathfrak{U}'(\sigma_\Gamma, \sigma_\Xi) = \begin{cases} (1, 0) & \text{iff} \quad \forall w \in W : m(F), w \models \phi \\ (0, 1) & \text{otherwise} \end{cases}$$

where W is defined by $(\Sigma, W, \sim, P, \pi) = m(u(u(F, \sigma_\Gamma), \sigma_\Xi))$.

The difference between kcg and kcg′ lies in their respective utility function. The function $\mathfrak{U}$ evaluates the formula ϕ in the model $m(u(u(F, \sigma_\Gamma), \sigma_\Xi))$, in all states. The function $\mathfrak{U}'$ evaluates the formula ϕ in the model $m(F)$, thus in the model before the update. This difference reflects the idea that in kcg, strategies are commonly known, whereas in kcg′ they are not known. The function $\mathfrak{U}'$ only evaluates the formula ϕ in states w that occur in the model $m(u(u(F, \sigma_\Gamma), \sigma_\Xi))$. The idea here is that the truth of ϕ only matters in states that are actually reached, and which states are reachable depends on the strategies chosen.

The two types of knowledge condition games that we have defined are two extreme viewpoints. Under the first definition strategies are completely known to everybody. In the definition of kcg′ strategies are completely unknown. One can define other, 'intermediate', variants, where agents can distinguish some but not all strategies, or for instance only know their own strategy. The hardness results for the extreme cases presented later in this paper also hold for some of the intermediate variants. So, one can argue that the extreme cases already cover many interesting situations. Detailed studies of the intermediate situations are therefore left for future work.

2.6. Alternating-Time Temporal Logics

In this subsection we discuss the temporal, epistemic, strategic logic ATEL, since much of the work here is motivated by it. The logic ATEL is an extension of ATL, a well known logic for reasoning about time and strategies.

The combination of knowledge, time and strategies makes ATEL a very rich but also a complicated logic. Our work on knowledge condition games is an attempt to avoid complications by leaving out temporal reasoning.

Alternating-time Temporal Logic (ATL) is a multi-agent extension of the branching time logic CTL (Alur et al., 2002). The language of ATL contains temporal operators similar to CTL, but instead of the path-quantifiers A and E that appear in CTL, strategy operators $\langle\langle\Gamma\rangle\rangle$ are used, where Γ can be any set of agents from a given set Σ. In ATL, a temporal operator is always preceded by a cooperation modality. The formula $\langle\langle\Gamma\rangle\rangle \bigcirc \phi$ for instance, expresses that the coalition Γ can make their choices in such a way that, no matter what the agents in $\Sigma\backslash\Gamma$ do, in the next state ϕ will hold. The CTL-formula $\mathsf{E} \bigcirc\phi$ mirrors ATL's $\langle\langle\Sigma\rangle\rangle \bigcirc\phi$ (the grand coalition Σ can choose actions such that in the next state, ϕ), and the CTL formula $\mathsf{A} \bigcirc\phi$ is the same as $\langle\langle\emptyset\rangle\rangle \bigcirc \phi$: 'no matter what the agents in $\Sigma\backslash\emptyset$ do, ϕ will hold in the next state'. Of course, in between these extremes, ATL can express many more coalitional abilities. ATEL, (where the E stands for epistemic), also contains knowledge operators.

DEFINITION 13. Let Σ be a finite set of agents, and P a finite set of atomic propositions. The logic ATEL contains formulas ϕ generated by the following rule. In this rule, p is a typical element of P, $X \in \Sigma$, $\Gamma \subseteq \sum$ and ψ is a path-formula.

$$\phi ::= p \mid \phi \rightarrow \phi \mid \perp \mid \langle\langle\Gamma\rangle\rangle\psi \mid K_X\phi$$
$$\psi ::= \Box\phi \mid \phi\mathcal{U}\phi \mid \bigcirc \phi$$

This logic is interpreted over alternating epistemic transition systems. These are defined as tuples $(P, \Sigma, Q, \sim, \pi, \delta)$. As usual, P is a set of atomic propositions and

Σ a set of agents. The set Q is a set of states the system can be in, and $\pi : Q \to 2^P$ adds propositions to these states. For any agent X the relation $\sim_X \subseteq Q \times Q$ is an equivalence relation, and $\delta : Q \times \Sigma \to 2^{2^Q}$ assigns to each agent in each state a set of sets of states. Each agent can choose one set of states, and the next state of the system will be from that set.

An example would be a system where $Q = \{0, 1, 2, 3, 4\}$. Suppose that $\delta(0, X) = \{\{1, 2\}, \{3, 4\}\}$ and $\delta(0, Y) = \{\{1, 3\}, \{2, 4\}\}$. Agent X can now choose $\{1, 2\}$ and Y can choose $\{2, 4\}$. They make these choices simultaneously. The next state of the system will be 2, because that is the only common state in their chosen sets. It is necessary to put some constraints on δ so that a next state can always be chosen.

The interpretation of this logic uses the notion of strategy to interpret the coalition operator $\langle\langle\Gamma\rangle\rangle$. A strategy for Γ is any function that makes a choice $\sigma_\Gamma(X, q) \in \delta(q, X)$ for any agent $X \in \Gamma$ in any state $q \in Q$. Based on a strategy σ_Γ, one can define the set of possible walks $\mathcal{W}(\sigma_\Gamma)$ through Q so that all choices for agents $X \in \Gamma$ are made as recommended by the strategy. This set of walks is used in the following interpretation of ATEL.

$$
\begin{array}{lll}
\mathcal{M}, q \models \bot & & \text{never} \\
\mathcal{M}, q \models p & \text{iff} & p \in \pi(v) \text{ where } p \in P \\
\mathcal{M}, q \models \phi \to \psi & \text{iff} & \mathcal{M}, q \models \phi \text{ implies } \mathcal{M}, q \models \psi \\
\mathcal{M}, q \models K_X\phi & \text{iff} & \forall (q, q') \in \sim_X : \mathcal{M}, q' \models \phi \\
\mathcal{M}, q \models \langle\langle\Gamma\rangle\rangle\phi & \text{iff} & \exists \sigma_\Gamma : \forall w = v \ldots \in \mathcal{W}(\sigma_\Gamma) : \mathcal{M}, w \models \phi \\
 & & \\
\mathcal{M}, w \models \bigcirc\phi & \text{iff} & \mathcal{M}, w(n+1) \models \phi \\
\mathcal{M}, w \models \Box\phi & \text{iff} & \forall n > 0 : \mathcal{M}, w(n) \models \phi \\
\mathcal{M}, w \models \phi\mathcal{U}\psi & \text{iff} & \exists m > 0 : \mathcal{M}, w(m) \models \psi \text{ and} \\
 & & \forall m > k > 0 : \mathcal{M}, w(k) \models \phi
\end{array}
$$

A main advantage of ATEL over kcg is that ATEL extends temporal logic, and can thus be used to express different kinds of goals such as eventually achieving something, or avoiding some state forever. When not putting any further constraints on how knowledge and choosing a strategy interfere, the logic has a low model checking complexity (van der Hoek and Wooldridge, 2002). However, this does not remain true if one requires strategies to be *uniform* (see the paragraph following Definition 7). If one demands uniform strategies, model checking becomes NP-complete, even without using the knowledge operator (Schobbens, 2004). Another point of discussion for this logic is the fact that the existence of a strategy, used in the interpretation of $\langle\langle\Gamma\rangle\rangle\phi$, is a very weak condition. One can come up with situations were $\langle\langle X\rangle\rangle\phi$ holds but one would not expect X to achieve ϕ (Jamroga and van der Hoek, 2003; Jonker, 2003; van Otterloo and Jonker, 2004). Thus, it seems that the interpretation of this logic merits further study, and indeed ATEL currently receives a lot of research attention (Agotness, 2004; Roberts et al., 2005).

The framework of knowledge condition games is arguably a less versatile verification framework than ATEL, because kcg does not allow complicated temporal reasoning. Only the special case of knowledge at the outcome stage of the protocol is studied. Knowledge condition games also do not allow for concurrent moves. This has the advantage that knowledge condition games are easier to understand, and that the complications that arise in the interpretation of ATEL do not arise in the context of knowledge condition games. An interesting difference between ATEL and kcg is that in kcg nondeterministic (and hence arguably "richer") strategies are permitted, whereas ATEL assumes deterministic strategies.

3. Examples

In this section, we illustrate the value of knowledge condition games, by presenting several examples of how they can be used to model scenarios of interest.

3.1. Anonymous Voting

A voting protocol can be used when a group of agents has to make a joint decision on a certain issue. A common protocol is *majority voting*: each agent can vote for an option and the option that gets the most votes is the outcome of the protocol. In the example interpreted game form F_V, three agents A, B and C use majority voting to decide whether plan $\mathcal{P}$ should be accepted or not. Each agent has a choice from two actions: support the plan (s), or reject it (r). They vote in alphabetical order, so first A chooses from action s or r, then B (without knowing A's choice) chooses either s or r and finally C does the same, unaware of what A, B did. This protocol thus has eight terminal histories. The proposition p indicates whether $\mathcal{P}$ is accepted and p holds if at least two agents choose s. Furthermore a holds if A chooses s, b if B chooses s and the same for C with c. The interpretation function is thus $\pi(sss) = \{a, b, c, p\}, \pi(ssr) = \{a, b, p\} \ldots \pi(rrr) = \emptyset$. We assume that $s \not\sim X\ s'$ if s and s' differ in the evaluation of the outcome p, or if the vote of X differs in s from that in s'.

The following game results hold.

$$\omega(\text{kcg}(F_V, \{A, B\}, \{C\}, p)) = 1$$
$$w(\text{kcg}(F_V, \{A, B\}, \{C\}, K_B c \vee K_B \neg c)) = 1$$
$$w(\text{kcg}(F_V, \{B\}, \{C\}, K_B c \vee K_B \neg c)) = 0$$

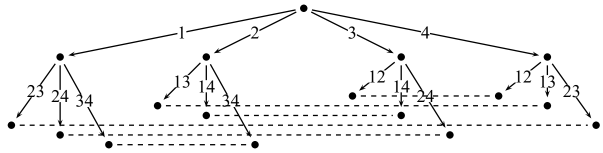

Figure 2. The fifty fifty problem F_Q.

A and B together can ensure that p is true, by voting s and s. What they can also do is vote differently, so that a and $\neg b$ result. In this case the outcome will solely depend on C's choice. They thus learn what C voted. Agent B cannot learn what C did on its own.

One example, described by Schneier (1996, p. 133), is a voting protocol where B would have the option of copying A's (encrypted) vote. In that case one might get

$$w(\text{kcg}(F'_V, \{B\}, \{A, C\}, K_B a \vee K_B \neg a)) = 1$$

This is an unwanted property and thus a 'bug' in the protocol. It is necessary to reason about knowledge to express this bug, so a standard game-theoretic analysis might not have revealed this shortcoming.

3.2. The Fifty-Fifty Problem

Consider the following scenario:

> In a TV quiz show the quiz master asks a candidate the following question: Which day of the week comes directly after Tuesday? Is it (a) Monday, (b) Wednesday, (c) Friday or (d) Saturday. The candidate replies: 'I am not sure. Can I do fifty fifty?'. The quiz master has to remove two options that are not the answer, so he says: 'The answer is not Monday and neither Friday'. Does the candidate know the answer?

(This situation frequently occurs on television in many countries in the 'Millionaire show'.) One can also consider this situation to be a metaphor for a multi-agent information exchange situation. Let us model this in an interpreted game form F_Q involving an agent N (nature) that determines what the right answer is, a quiz master Q that eliminates two answers, and a candidate C. This interpreted game form is depicted in Figure 3. First nature selects one of the answers to be the right answer: it can choose from the actions 1, 2, 3 and 4. The quiz master, who knows the right answer, can then select an action ij that indicates that the two options i and j are eliminated; i, j must be different from the right answer. The terminal histories are thus all histories (k, ij). For such histories, $(k, ij) \sim_C (k', i'j')$ if the same options are eliminated: $ij = i'j'$. The set of atomic propositions is $P = \{a_i \mid 1 \leq i \leq$

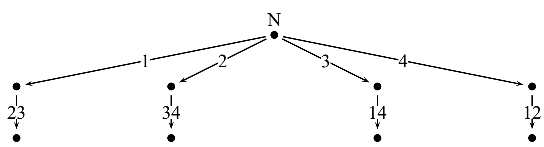

Figure 3. The updated interpreted game form $u(F_Q, \sigma_{\{Q\}})$.

$4\} \cup \{e_i \mid 1 \leq i \leq 4\}$, and each terminal history is interpreted in the following way: $\pi(k, ij)) = \{a_k, e_i, e_j\}$. The question is whether the candidate knows the answer at the end of the protocol. This is expressed by $\psi = K_C a_1 \vee K_C a_2 \vee K_C a_3 \vee K_C a_4$. The following table lists several properties of this situation.

Nature may favour the candidate:	$w(\text{kcg}(F_Q, \{N\}, \emptyset, \psi)) = 1$
Nature may not favour the candidate:	$w(\text{kcg}(F_Q, \{N\}, \emptyset, \neg\psi)) = 1$
The quiz master can help the candidate:	$w(\text{kcg}(F_Q, \{Q\}, \emptyset, \psi)) = 1$

We thus see that whether the candidate knows the answer depends on nature and on the quiz master Q. If nature uses a deterministic strategy, in which for instance a_1 always holds, then the candidate knows that this is the right answer. However, if Nature uses the non-deterministic strategy in which each answer could be the right answer, the candidate will not know the answer.

The situation becomes more interesting if the quiz master gets involved. In this game the quiz master has the ability to signal the right answer to the candidate. Consider, for example, strategy $\sigma_{\{Q\}}$, defined as follows.

$$\sigma_{\{Q\}}(1) = \{23\}$$
$$\sigma_{\{Q\}}(2) = \{34\}$$
$$\sigma_{\{Q\}}(3) = \{14\}$$
$$\sigma_{\{Q\}}(4) = \{12\}$$

This strategy tells the candidate exactly what the right answer is: The answer directly before the two eliminated options (assuming 4 comes before 1). The updated model $u(F_Q, \sigma_{\{Q\}})$ is given in Figure 3. This strategy acts as a code between the candidate and the quiz master. It is the strategy that proves that $w(\text{kcg}(F_Q, \{Q\}, \emptyset, q)) = 1$. (A practical conclusion one can draw is that one should not bet on such a quiz if one does not know what the interests of the quiz master are.)

This example also demonstrates why we prefer to assume that strategies are commonly known. If one would have used the alternative definition kcg$'$, in which agents do not know what strategies are used, then one can obtain the following results.

Nature cannot favour the candidate:	$w(\text{kcg}(F_Q, \{N\}, \emptyset, \psi)) = 0$
The quiz master cannot help the candidate:	$w(\text{kcg}(F_Q, \{Q\}, \emptyset, \psi)) = 0$

These results are counter-intuitive, since signaling in games is a phenomenon that does occur in practice. When proving the security of a protocol, it is a good principle to make the weakest assumptions possible. At first sight, it seems that assuming that strategies are not known is the weakest possible assumption. However in the case of proving ignorance, first sight can be misleading. It is harder to prove that the candidate does not know the answer when he or she knows all strategies that are used, than it is to prove ignorance when he or she does not know the strategies. Therefore the weakest and safest assumption is to assume that he does know the strategies. This shows that it is best to use the definition of kcg rather than the alternative kcg′ for these ignorance proofs. (In fact, this motivates the choice to make kcg the default and call kcg′ the alternative.)

4. Computational Complexity

Looking at computational complexity is interesting for two reasons. First of all it can tell you whether a certain problem is 'tractable', i.e. whether the problem can be reliably solved in practice. Secondly it can tell you more about the problem – for instance whether something is a very general problem (i.e., whether the problem format can be used to formulate questions about many different situations, such as logic), or what features makes a problem difficult. In this section we look at the complexity of the *kcg decision problem*, which is the problem of deciding for a game kcg(F, Γ, Ξ, ϕ) whether the first coalition Γ has a winning strategy. We look at this problem under various assumptions, and report four theorems, as follows:

- The first theorem is concerned with the problem of deciding whether a coalition Γ can win a knowledge condition game with an empty set of opponents. This is called the *no-opponents* knowledge condition game decision problem. It turns out that this problem is already NP-complete, and thus not tractable.
- The second theorem states that the general kcg decision problem is even harder: with opponents the problem is Σ_2P-complete.
- For the other theorems we use the alternative version of knowledge condition games kcg′: In the third theorem we claim that the no-opponents problem is as hard as the general problem. Both problems turn out to be NP-complete, which is the fourth theorem.

Recall that a problem is in the class NP if it can be solved in polynomial time (there is a time bound that is polynomial in the input size) by a nondeterministic Turing machine. In practice this means that we can *check* a solution in reasonable time, but may not be able to *find* a solution in reasonable time (Papadimitriou, 1994; Cormen et al., 1990). Some problems in the class NP are NP-complete: every problem in NP can be expressed as an instance of an NP-complete problem. Thus NP-complete problems are at least as hard as any problem in the class NP. It is widely believed, but has not been proven, that NP-complete problems cannot be solved in polynomial time.

As an aside, we note that we encode interpreted game forms in an explicit way, by listing all histories. In reality protocols are often specified in an implicit way (for instance in some form of source code) and such representations can be exponentially more efficient.

THEOREM 1. *The problem to decide, for a given interpreted game form F, coalition* Γ *and knowledge proprty* ϕ*, whether* $w(\text{kcg}(\text{F}, \Gamma, \emptyset, \phi)) = 1$*, is NP-complete.*

Proof. Assume that F, Γ, ϕ are given. The empty coalition has only one strategy $\sigma_\emptyset$. This strategy is such that $u(F, \sigma_\emptyset) = F$. Therefore

$$w(\text{kcg}(F, \Gamma, \emptyset, \phi)) = 1 \Leftrightarrow \exists \sigma_\Gamma \; m(u(F, \sigma_\Gamma)) \models \phi$$

A nondeterministic polynomial algorithm for this problem exists. Find or guess nondeterministically a strategy σ_Γ. Since a strategy encodes a subset of actions available in F, the size of σ_Γ is smaller than the size of F and thus polynomial in the input size. Now calculate $\mathcal{M} = m(u(F, \sigma_\Gamma))$, and verify for each state w of $\mathcal{M}$ that $\mathcal{M}, w \models \phi$. The number of states in $\mathcal{M}$ is at most the number of terminal histories of F, so $|\mathcal{M}| \leq |F|$. All of this can be done in polynomial time. Therefore, this problem can be solved using a nondeterministic polynomial algorithm and this problem is in NP.

In order to show that the restricted kcg problem of the theorem is as hard as any NP problem, we show that any instance of the 3SAT problem can be transformed into an equivalent restricted kcg instance. Let $\phi^3 = \wedge_i (a_i \vee b_i \vee c_i)$ be a propositional logic formula in conjunctive normal form with three literals per clause. The literal formulas a_i, b_i, c_i must be either atomic propositions or negated atomic propositions. The 3SAT problem is to decide whether a truth-assignment $\mathbb{A}$ for all atomic propositions in ϕ^3 exists such that $\mathbb{A} \models \phi^3$. We can construct an interpreted game form F with a single agent $\Sigma = \{A\}$ and a formula ϕ such that $w(\text{kcg}(F, \{A\}, \emptyset, \phi)) = 1$ if and only if $\exists \mathbb{A} : \mathbb{A} \models \phi^3$.

The model $F = (\{A\}, H, \text{turn}, \sim, P, \pi)$ is constructed in the following way. Let P^3 be the set of atomic propositions occurring in ϕ^3. The new set of atomic propositions P contains two propositions for any old proposition: $P = \{x^+ \mid x \in P^3\} \cup \{x^- \mid x \in P^3\}$. For each new proposition a history is created: $H = \{\epsilon\} \cup \{e_p \mid p \in P\}$. The interpretation function is such that only the corresponding atomic proposition is true: $\pi(e_p) = \{p\}$. Furthermore $\text{turn}(\epsilon) = A$. Agent A cannot distinguish any end state: $e_p \sim A \, e_q$ for all terminal histories e_p and e_q.

The formula $\phi = \phi_1 \wedge \phi_2$ is a conjunction of two parts. The part ϕ_1 expresses that for each original atomic proposition $p \in P^3$, either the positive proposition p^+ is considered possible or the negative p^-, but not both:

$$\phi_1 = \bigwedge_{p \in P^3} (M_A p^+ \vee M_A p^-) \wedge \neg(M_A p^+ \wedge M_A p^-)$$

The idea is that the strategy that A uses is actually an assignment of values to all atomic propositions in P^3. The condition ϕ_1 expresses that such assignment must assign either the truth value true (p^+) or false (p^-) to each proposition p.

The ϕ_2 part encodes the original formula $\phi^3 = \bigwedge_i (a_i \vee b_i \vee c_i)$. In the next definition we use a helper function f defined such that $f(\neg p) = p^-$ and

$f(p) = p^+$. using this function we define B as follows.

$$\phi_2 = \bigwedge_i M_A(f(a_i) \vee f(b_i) \vee f(c_i))$$

It is not hard to see that any strategy $\sigma_{\{A\}}$ such that $m(u(F, \sigma_{\{A\}})) \models \phi_1 \wedge \phi_2$ corresponds to an assignment $\mathbb{A}$ such that $\mathbb{A}(p) = \mathit{true}$ if and only if $p^+ \in \sigma_{\{X\}}(\epsilon)$, and that this assignment satisfies $\mathbb{A} \models \phi^3$. Since the formula and model constructed have a size that is linear with respect to the size of ϕ^3, this is a polynomial reduction. Therefore the restricted kcg problem is NP-hard. Since we have also shown that the problem is in NP, we conclude that the restricted kcg problem is NP-complete. $\square$

As an example, consider the satisfiability of the 3SAT formula $\psi = (p \vee \neg q \vee r) \wedge (\neg q \vee \neg p \vee r)$. This formula contains three propositions, so the corresponding interpreted game form, depicted in Figure 4, contains six terminal histories. The corresponding knowledge formula is ψ_K.

$$\begin{aligned}\psi_K = {} & (M_A p^+ \vee M_A p^-) \wedge \neg(M_A p^+ \wedge M_A p^-) \wedge \\ & (M_A q^+ \vee M_A q^-) \wedge \neg(M_A q^+ \wedge M_A q^-) \wedge \\ & (M_A r^+ \vee M_A r^-) \wedge \neg(M_A r^+ \wedge M_A r^-) \wedge \\ & M_A(p^+ \vee q^- \vee r^+) \wedge M_A(q^- \vee p^- \vee r^+)\end{aligned}$$

A typical NP-complete problem is to determine whether a prepositional logic formula is satisfiable. Suppose ϕ is a formula with atomic propositions $x_1, x_2, \ldots x_n$.

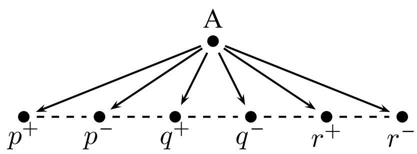

Figure 4. The model of 3SAT formula ψ.

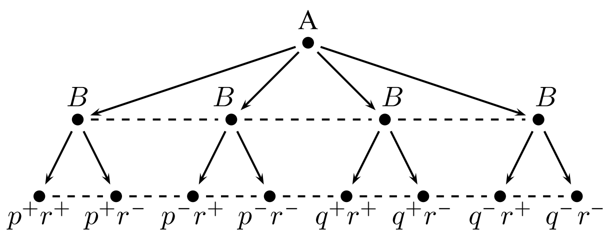

Figure 5. The construction of the Σ_2P proof.

We can thus write $\phi = \phi(\vec{x})$ where the vector $\vec{x}$ consists of all the $\exists x_i$. The satisfaction problem can now be phrased as deciding whether $\exists\vec{x} : \phi(\vec{x})$. In the same way we can formulate more difficult problems, by allowing more quantifiers: $\exists y \forall x : \phi(\vec{x}, \vec{y})$ is the problem where one has to decide whether there is an $\vec{x}$ such that $\phi(\vec{x}, \vec{y})$ is true for all $\vec{y}$. This problem, called SAT_2, is a typical Σ_2P complete problem (Papadimitriou, 1994)(chapter 17). It is widely believed that these problems are strictly more difficult than NP-complete problems.

THEOREM 2. *The problem to decide for given a given interpreted game form F, coalitions Γ and Ξ and property ϕ whether $w(\mathrm{kcg}(F, \Gamma, \Xi, \phi)) = 1$ is Σ_2P-complete.*

Proof. First we have to prove that this problem is indeed in Σ_2P. In order to do this, consider the winning condition of a knowledge condition in more detail.

$$w(\mathrm{kcg}(F, \Gamma, \Xi, \phi)) = 1 \Leftrightarrow \exists\sigma_\Gamma \forall\sigma_\Xi\ m(u(u(F, \sigma_\Gamma), \sigma_\Xi)) \models \phi$$

Suppose that F, Γ, Ξ and ϕ are given. It is possible to encode strategies of Ξ as assignments to a vector of propositional variables $\vec{y}$, and the strategy of Γ as assignments to $\vec{x}$. One can then find a formula $\psi(\vec{x}, \vec{y})$ that is true if $m(u(u(F, \sigma_\Gamma), \sigma_\Xi)) \models \phi$. The size of this formula is polynomial in $|F| + |\phi|$. The kcg decision problem is equivalent to a SAT_2 problem:

$$w(\mathrm{kcg}(F, \Gamma, \Xi, \phi)) = 1 \Leftrightarrow \exists\vec{x}\forall\vec{y} : \psi(\vec{x}, \vec{y})$$

The problem to decide whether $\exists\vec{x}\forall\vec{y} : \psi(\vec{x}, \vec{y})$ is a SAT_2 problem, and is thus in Σ_2P.

The second part of the proof is to show that the kcg decision problem is indeed complete for this class, and this can be done by reducing SAT_2 to a knowledge condition game. The proof is similar to the previous NP-completeness proof, but now involves two agents. Assume that a SAT_2 problem $\exists\vec{y}\forall\vec{x} : \psi(\vec{x}, \vec{y})$ is given. We can assume that ψ is in 3-SAT form: $\psi = \wedge_i(a_i \vee b_i \vee c_i)$. First we define an interpreted game form $F = (\Sigma, H, \mathrm{turn}, \sim, P, \pi)$. Let $\Sigma = \{A, B\}$, and $Z(H) = \{(a, b) | \exists i, j : a = x_i^+ \text{ or } a = x_i^-, b = y_j^+ \text{ or } b = y_j^-\}$. The set H contains all histories of $Z(H)$ and all prefixes of these histories. The function turn is defined such that A moves first, and then B moves: $\mathrm{turn}(\epsilon) = A$ and $\mathrm{turn}((x_i^\pm)) = B$. The relations $\sim_A$ and $\sim_B$ are equal, and defined such that each agent only knows the length of each history: $s \sim_A s' \Leftrightarrow |s| = |s'|$. The set of propositions P of the kcg problem is $\{z^+ \mid z \in (\vec{x} \cup \vec{y})\} \cup \{z^- \mid z \in (\vec{x} \cup \vec{y})\}$. The function π is defined by $\pi(a, b) = \{a, b\}$. This completes the definition of the interpreted game form F. The number of terminal histories of F is $2|\vec{x}| \cdot 2|\vec{y}|$, and thus the size of F is polynomial in the size of the input problem.

We define $\Gamma = \{A\}$ and $\Xi = \{B\}$. Next we define an epistemic logic formula ϕ. such that Γ can win the game $\mathrm{kcg}(F, \Gamma, \Xi, \phi)$ iff $\exists\vec{x}\forall\vec{y} : \psi(\vec{x}, \vec{y})$. Let $\phi =$

$\neg\phi^B \vee (\phi^A \wedge f(\psi(\vec{x}, \vec{y})))$. The part ϕ^B expresses that the strategy of B corresponds to an assignment to $\vec{y}$. The part ϕ^A expresses that the strategy of A corresponds to a strategy for $\vec{x}$. Finally $f(\psi(\vec{x}, \vec{y}))$ is a translation of the input formula $\psi(\vec{x}, \vec{y})$.

$$\phi^B = \bigwedge_j((M_B y_j^+ \vee M_B y_j^-) \wedge \neg(M_B(y_j^+ \wedge y_j^-))$$
$$\phi^A = \bigwedge_i((M_A x_i^+ \vee M_A x_i^-) \wedge \neg(M_A(x_i^+ \wedge x_i^-))$$
$$f(\psi(\vec{x}, \vec{y})) = f(\bigwedge_j(a_i \vee b_i \vee c_i) = \bigwedge_i(f(a_i) \vee f(b_i) \vee f(c_i))$$

The function f is defined such that $f(\neg p) = p^-$ and $f(p) = p^-$. The size of ϕ is linear in the size of ψ. Therefore this is a polynomial reduction. This completes the proof that the knowledge condition game decision problem is Σ_2P-hard. Since it is also in Σ_2P, we conclude that the problem is Σ_2P-complete. □

The construction of a model F is illustrated in Figure 4. This is the model that you would get in the reduction of $\psi(\vec{x}, \vec{y})$ where $\vec{x}$ contains p and q and $\vec{y}$ consists of r. The model is again relatively small: only two actions happen in each play of this interpreted game form. The first one is decided by agent A, the second one by B.

In the two previous proofs, it is essential that the agents are aware of the strategies they choose. Both constructions would not work with the alternative definition kcg′. One can hope that the computational complexity of the kcg′ decision problem would be lower. Indeed one can prove that in this case it does not matter whether there are opponents.

THEOREM 3. *Assume that F, Γ, Ξ and ϕ are given.* $w(\text{kcg}'(F, \Gamma, \Xi, \phi)) = 1$ *if and only if* $w(\text{kcg}'(F, \Gamma, \emptyset, \phi)) = 1$.

Proof. Let $G = \text{kcg}'(F, \Gamma, \Xi, \phi)$ be a kcg′ decision problem. Notice that the goal of coalition Ξ is to choose a strategy σ_Ξ such that $\mathfrak{U}'(\sigma_\Gamma, \sigma_\Xi) = (0, 1)$, where $\mathfrak{U}'$ is the utility function of the game G. Since $\mathfrak{U}'$ is defined using universal quantification over the set of terminal histories of $u(u(G, \sigma_\Gamma), \sigma_\Xi)$ the best thing to do forcoalition Ξ is to make sure that this set is as large as possible. In order to achieve this, σ_Ξ should choose the neutral strategy that allows any action: the strategy σ with $\sigma(h) = A(H, h)$. Since we have assumed that neutral agents can do any action, we might as well assume that the agents $X \in \Xi$ are neutral, and determine the value of the game $w(\text{kcg}'(F, \Gamma, \emptyset, \phi)) = 1$. □

We see thus that the presence of opponents is not relevant, and indeed in ATEL no distinction between opponents and neutral agents is made. The question is now whether solving the kcg′ decision problem is still as hard as the original no-opponents kcg problem. The answer is yes. The no-opponents kcg′ problem is also NP-complete. However the proof is different in an interesting way.

THEOREM 4. *The problem to decide for given interpreted game form F, coalition Γ and knowledge formula ϕ whether* $w(\text{kcg}'(F, \Gamma, \emptyset, \phi)) = 1$ *is NP-complete.*

Proof. We can prove that this problem is in NP by a similar argument as given for Theorem 1. For the hardness result we again show a reduction from 3SAT. Assume that $\phi^3 = \bigwedge_{i=1}^{n}(a_i \vee b_i \vee c_i)$ is a propositional formula in conjunctive normal form with three literals per clause. Let P^3 be the set of atomic propositions occurring in ϕ^3. We define an interpreted game form between two agents: an agent Q that asks questions, and an agent A that answers them. The proponent coalition is $\Gamma = \{A\}$ and Q is assumed to be neutral. Every terminal history is of the form (p, b, i, x), where $p \in P^3, b \in \{0, 1\}, i \in \{1, 2, \ldots, n\}$ and $x \in \{a_i, b_i, c_i\}$. The first action p is chosen by agent Q and must be one atomic proposition of ϕ^3. The agent A must then reply by giving a boolean value b. This indicates what truth value A has in mind for p. Then agent Q chooses one triplet $(a_i \vee b_i \vee c_i)$ that appears in ϕ^3. Agent A then has to choose which of these three parts it thinks should be true: either a_i or b_i or c_i. The trick however is that $\sim_A$ is defined in such a way that for all histories h, h', agent A only knows the length of the histories: $h \sim_A h'$ iff $|h| = |h'|$

A does not know, when making its final decision, which answer it gave on its first turn. The agent thus risks giving inconsistent information. For instance in the history $(p, 1, (\neg p \vee q \vee r), \neg p)$ agent A first says that p is true, and then says that it thinks that $\neg p$ holds. The goal of agent A in the game is to avoid these inconsistent histories. We let $P = \{e\}$ consist of one proposition and define for all $p \in P^3$ the interpretation function such that $\pi((p, 1, i, \neg p)) = \{e\}, \pi((p_j, 0, i, p_j)) = \{e\}$ and $\pi((p, b, i, x)) = \emptyset$ otherwise. One can now consider the knowledge condition game $G = \text{kcg}'(F, \{A\}, \emptyset, \neg e)$. Agent A can win the game G iff there is a satisfying assignment for ϕ^3. □

The proof given above is very similar to a proof given by Schobbens (2004) for the NP-completeness of ATL with imperfect information. This corroborates the claim that this variant of knowledge condition games is closely related to ATL and thus to ATEL. The proof exploits the fact that in games where coalitions do not have perfect recall, it is very difficult for agents and coalitions to coordinate their own actions.

4.1. Tractable Variants

In the previous section we proved that, in general, the kcg decision problem is not tractable. In this section we identify some easier cases.

THEOREM 5. *Let F be an interpreted game form with perfect recall,* Γ *any coalition of agents and* ϕ *an epistemic formula. Deciding whether it is the case that* $w(\text{kcg}'(F, \Gamma, \phi)) = 1$ *can be done in polynomial time.*

Proof. Let $\mathcal{M} = (\Sigma, W, R, P, \pi) = m(F)$ be the end state model of F. We can compute the set $S = \{w \in W \mid \mathcal{M}, w \models \phi\}$ in polynomial time. Define a utility function $\mathfrak{U}$ such that $\mathfrak{U}(w) = 1$ iff $w \in S$ and $\mathfrak{U}(w) = 0$ otherwise. The pair $F, \mathfrak{U}$ is now an extensive game with perfect recall. The optimal solution σ for this game can be computed in polynomial time (Koller and Pfeffer, 1995). If the expected payoff of σ is exactly one, then $w(\text{kcg}'(F, \Gamma, \emptyset, \phi)) = 1$, otherwise $w(\text{kcg}'(F, \Gamma, \emptyset, \phi)) \neq 1$. □

For perfect recall frameworks and the variant kcg′ the decision problem is thus tractable. One might wonder whether the same claim can be made for kcg. The answer is no, because one can modify the NP-completeness proof for kcg in such a way that it uses a perfect recall interpreted game form. The modification is that one has to two agents, A and A', so that A is the agent that chooses a strategy, and A' is the agent that cannot distinguish end states and occurs in the knowledge condition. In general one can always find an perfect recall interpreted game form that is equivalent for the kcg decision problem by choosing a fresh agent for each decision node, and use fresh agents in the knowledge condition.

Instead of asking whether there are interpreted game forms F that make decision problems easy, one can also ask whether there are easy formulas ϕ. The answer to this is question is yes: to see how this works, we first formulate the notion of *positive* and *negative* formulas.

For any $p \in P$, the formula p is both positive and negative. Falsum $\perp$ is also both positive and negative. Iff ϕ is positive and ψ is negative, then $\phi \rightarrow \psi$ is negative. Vice versa, iff ψ is positive and ϕ is negative, then $\phi \rightarrow \psi$ is positive. Iff ϕ is positive then $K_X\phi$ is positive. A formula of the form $K_X\phi$ is not negative.

Positive and negative formulas are both called *monotone* formulas, because one can prove that they preserve truth in the following way. Suppose that $\mathcal{M} = (\Sigma, W, \sim, P, \pi)$ and $\mathcal{M}' = (\Sigma, W', \sim', P, \pi')$ are models such that $W' \subset W$ and $\sim'$ and π' are the restrictions of $\sim$ and π to W'. In this case we say that $\mathcal{M}'$ is a sub-model of $\mathcal{M}$. Suppose ϕ^+ is a positive formula, and ϕ^- is a negative formula. Then the following statements can be proven.

$$\mathcal{M} \models \phi^+ \quad \text{implies} \quad \mathcal{M}' \models \phi^+$$
$$\mathcal{M}' \models \phi^- \quad \text{implies} \quad \mathcal{M} \models \phi^-$$

The proof of these statements is done by induction over the formula structure. The interesting step involves the knowledge operator. Suppose that $\mathcal{M}, w \models K_X\phi^+$. By definition this means that $\forall v \in W$, if $w \sim_X v$ then $\mathcal{M}, v \models \phi^+$. Since W' is a subset of W, this means that $\forall v \in W'$, if $w \sim'_X v$ then $\mathcal{M}', v \models \phi^+$, Using the induction hypothesis we obtain $\forall v \in W'$, if $w \sim'_X v$ then $\mathcal{M}', v \models= \phi^+$, and thus $\mathcal{M}', w \models K_X\phi^+$.

Knowledge condition games with monotone formulas are easier to solve than general knowledge condition games (under standard complexity theoretic assumptions).

THEOREM 6. *The problem to decide for given* F, Γ, Ξ *and a monotone formula* ϕ *whether* $w(\mathrm{kcg}(F, \Gamma, \Xi, \phi)) = 1$ *can be solved in polynomial time.*

Proof. We prove the case where ϕ is a positive formula. The argument for negative formulas is similar. Recall that by definition, $w(\mathrm{kcg}(F, \Gamma, \Xi, \phi)) = 1$ iff $\exists\sigma_\Gamma\forall\sigma_\Xi\forall w \in W$ it is the case that $m(u(u(F, \sigma_\Gamma), \sigma_\Xi)), w \models \phi$ where W is the set of worlds in the model $m(u(u(F, \sigma_\Gamma), \sigma_\Xi))$. The formula ϕ is a positive formula, and thus it follows that $m(u(F, \sigma_\Gamma)), w \models \phi$ implies $m(u(u(F, \sigma_\Gamma), \sigma_\Xi)), w \models \phi$. The best thing for coalition Ξ to do is to use a strategy that does not eliminate any action. They should use a neutral strategy σ^0 such that $u(F, \sigma^0) = F$. This strategy is described by $\sigma^0(h) = A(H, h)$.

For coalition Γ things are exactly opposite. Suppose that σ^1 and σ^2 are strategies so that σ^1 is more specific than σ^2. Formally this means that $\forall h : \sigma^1(h) \subset \sigma^2(h)$. The monotonicity of ϕ implies that $m(u(F, \sigma^2)), w \models \phi$ implies $m(u(F, \sigma^1)), w \models \phi$. Coalition Γ thus does best be choosing the more specific strategy σ^1. For coalition Γ we thus only have to consider the most specific strategies. These most specific strategies are what one can call *pure*, because they select exactly one action at each decision point. A backwards induction argument can be used to show that that there are as many pure strategies for F as there are terminal histories in F. We can try all pure strategies σ^p to see if one satisfies ∀w: $m(u(F, \sigma^p)), w \models \phi$. This gives an algorithm that needs time $\mathcal{O}(|F|^2 \cdot |\phi|)$. The first $|F|$ is caused by the fact that we need to consider all pure strategies. The remaining term $|F| \cdot |\phi|$. is the time needed to determine whether for all w it is the case that $u(\mathcal{M}, \sigma^p), w \models \phi$. The decision problem can thus be done in polynomial time.

For negative formulas the roles of Γ and Ξ are interchanged. For a negative formula ϕ, coalition Γ can use the neutral strategy σ^0. The opponent coalition Ξ should now try all pure strategies σ^p. □

5. Related Work

Our approach to the formalisation of knowledge condition games draws upon a number of disparate areas of work. Our treatment of knowledge is based on epistemic logic (Meyer and van der Hoek, 1995). Epistemic logic originated from philosophy (Hintikka, 1962) but has been successfully used in the domain of computer science to capture the knowledge of agents in interpreted systems (Fagin et al., 1995). It has been combined with temporal logic (Halpern and Vardi, 1989) and coalition logic (van der Hoek and Wooldridge, 2003) in order to capture knowledge development over time, to capture knowledge change after announcements (Baltag et al., 2002) or knowledge evolution in games such as Cluedo (van Ditmarsch, 2000). These frameworks have been applied to a range of different situations, such as the Russian Cards problem (van Ditmarsch, 2003; van Otterloo et al., 2003) and the Dining Cryptographers Problem (van der Meyden and Su, 2004). What we do is in one sense a simplification of these approaches, because using temporal logic one can express more things than just ‘At the end of the protocol. . . ’. In another sense it is an extension, because we explicitly deal with strategies, and take into account that agents may know which strategies are being used. In fact, knowledge condition games have been designed as a response to the problems with ATEL (van der Hoek and Wooldridge, 2003). ATEL is a logic that includes operators for knowledge, time and strategic ability. Knowledge condition games are closely related to model checking problems in ATEL. In particular, the ability of a coalition Γ to win in a knowledge condition game with knowledge condition ϕ and interpreted game form F is more or less captured by the ATEL model checking problem $\mathcal{M}_F \models \langle\langle\Gamma\rangle\rangle \diamond \phi$, where $\mathcal{M}_F$ is the ATEL model corresponding to F. On a technical level, the main difference is perhaps that strategies in ATEL are assumed to be deterministic, while in knowledge condition games, we allow for non-deterministic (and hence arguably "richer") strategies. There are some additional technical differences, for example the way that the underlying transition model is captured is different in the two approaches: in ATEL, a transition function is used, which is rather similar to a labeled transition system, whereas in knowledge condition games we enumerate the set of all histories of the game.

Because of the way it is designed, it is implicitly assumed in ATEL that the epistemic relations between states do not depend on the strategies that are played. In ATEL it is for instance not necessarily the case that agents know their own strategies. In knowledge condition games, we have been able to deal more explicitly with this strategic aspect, and secondly we have explicitly introduced opponents. The long term goal of this research is to get a good understanding of the interaction between knowledge and strategies, and perhaps this will ultimately lead to a refinement of ATEL to incorporate these aspects.

Scenarios like knowledge condition games have appeared many times in the literature. Perhaps the most famous example is the *coordinated attack problem* (Fagin et al., 1995, pp.176–183). In this problem, two army generals desire to reach agreement on when to attack an enemy: neither general wants to attack on their own, but only when they are sure that the other will also attack. They will thus attack only when it is *common knowledge* between them that they will attack. However, in order to reach agreement, they can only communicate through a lossy channel (i.e., where there is a non-zero probability that any given message will be lost). The famous result with respect to this problem says that if the generals will only attack when it is common knowledge that they will attack, then in fact they will *never* attack, because they will never reach common knowledge. This is because, no matter how many rounds of acknowledgments are sent between the two generals, whoever sent the last message can never be certain that it was received. We can think of the coordinated attack problem as a knowledge condition game involving three agents – the two generals forming one coalition on the one hand, and an environment, modelled as an agent who controls the 'lossyness' of the channel, on the other. Then, we can ask whether the two generals can successfully ensure the epistemic condition that it is common knowledge that they will attack. Related work in the game theory literature is the *electronic mail game* (Osborne and Rubinstein, 1994, pp. 81–84).

6. Conclusion

By combining protocols and knowledge conditions into games, one can express properties of multi-agent protocols relating to security and secrecy. In a knowledge condition game one can make fine distinctions between for instance neutral and opponent agents, and one can give examples where this distinction is significant. Therefore these games are a promising direction for future research into the interaction between knowledge and strategies.

The complexity results reported in this paper paint an interesting picture. There seems to be a computational cost for assuming that agents know strategies. The single agent decision problem is already intractable. The presence of opponents makes it even harder to compute whether a coalition can guarantee a property. If we drop this assumption and reformulate the notion of winning a knowledge condition game, then the extra complexity of adding opponents disappears. However the problem without opponents is still NP-complete, and hence intractable, but for different reasons. The complexity proof is no longer based upon formulating a difficult knowledge formula, but on the hardness of coordinating in an interpreted game form without perfect recall.

Future research could focus on comparing decision problems for knowledge condition games to other game-theoretic decision problems, in order to establish what exactly the complexity cost is of considering knowledge goals. It would also be interesting to find out under which assumptions knowledge condition games can be solved in polynomial time. Other directions include looking at knowledge condition games from a logical viewpoint by searching for axioms, and to consider the mechanism design problem to find an interpreted game form with given properties.

References

Agotness, T., 2004, "A note on syntactic characterization of incomplete information in ATEL". in: S. van Otterloo, P. McBurney, W. van der Hoek and M. Wooldridge (eds.): *Proceedings of the First Knowledge and Games Workshop*. University of Liverpool, pp. 34–42.

Alur, R., Henzinger, T. A., and Kupferman, O., 2002, "Alternating-time temporal logic", *Journal of the ACM* **49**(5), 672–713.

Aumann, R., 1995, "Backward induction and common knowledge of rationality", *Games and Economic Behaviour* **8**, 6–19.

Baltag, A., Moss, L., and Solecki, S., 2002, "The logic of public announcements, common knowledge and private suspicions", Originally presented at TARK 98, accepted for publication in Annals of Pure and Applied Logic.

Bonanno, G., 2004, "Memory implies von Neumann-Morgenstern games", *Knowledge Rationality and Action*, to appear.

Cormen, T., Leiserson, C., and Rivest, R., 1990, *Introduction to Algorithms*. The MIT Press: Cambridge, MA.

de Bruin, B.: 2004, "Explaining games- on the logic of game theoretic explanations". Ph.D. Thesis, University of Amsterdam, Amsterdam. ILLC series DS-2004-03.

Fagin, R., Halpern, J., Moses, Y. and Vardi, M.: 1995, *Reasoning About Knowledge*. The MIT Press: Cambridge, MA.

Halpern, J. Y. and Vardi, M. Y., 1989, "The complexity of reasoning about knowledge and time. I. Lower bounds". *Journal of Computer and System Sciences* **38**, 195–237.

Harrenstein, B., van der Hoek, W., Meyer, J.-J. C., and Witteveen, C., 2003, "A modal characterization of Nash Equilibrium", *Fundamenta Informaticae* **57**(2–4), 281–321.

Hintikka, J., 1962, *Knowledge and Belief: An Introduction to the Logic of the Two Notions*. Cornell University Press: Ithaca, NY.

Holzmann, G., 1991, *Design and Validation of Computer Protocols*. Prentice Hall International: Hemel Hempstead, England.

Jamroga, W. and van der Hoek, W., 2003, "Some remarks on alternating-time temporal epistemic logic", submitted.

Jonker, G., 2003, "Feasible strategies in alternating-time temporal epistemic logic", Universiteit Utrecht Master Thesis.

Koller, D. and Pfeffer, A., 1995, "Generating and solving imperfect information games", in: *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI),* Montreal, pp. 1185–1192.

Meyer, J.-J. C. and van der Hoek, W., 1995, *Epistemic Logic for AI and Computer Science,* Cambridge University Press: Cambridge, England.

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

Papadimitriou, C., 1994, *Computational Complexity,* Addison Wesley Longman.

Pauly, M., 2001, "Logic for Social Software". Ph.D. Thesis, University of Amsterdam, ILLC Dissertation Series 2001-10.

Roberts, M., van der Hoek, W., and Wooldridge, M., 2005, "Knowledge and social laws". in *Proceedings of the International Joint Conference on Autonomous Agents and Multi-Agent Systems* (AAMAS), Utrecht.

Schelling, T. C., 1960, *The Strategy of Conflict*, Cambridge, Mass.: Harvard U.P.

Schneier, B., 1996, *Applied Cryptography,* John Wiley & Sons.

Schobbens, P.-Y., 2004, "Alternating-time logic with imperfect recall", in W. van der Hoek, A. Lomuscio, E. de Vink, and M. Wooldrige (eds.): *Electronic Notes in Theoretical Computer Science*, Vol. 85, Elsevier.

Selten, R., 1975, "Reexamination of the perfectness concept for equilibrium points in extensive games", *International Journal of Game Theory* **4**, 25–55.

van Benthem, J., 2001, "Games in dynamic-epistemic logic", *Bulletin of Economic Research* **53**(4), 219–248.

van der Hoek, W. and Wooldridge, M., 2002, "Tractable multiagent planning for epistemic goals", in *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2002)*. Bologna, Italy, pp. 1167–1174.

van der Hoek, W. and Wooldridge, M., 2003, "Cooperation, knowledge, and time: Alternating-time temporal epistemic logic and its applications", *Studia Logica* **75**(4), 125–157.

van der Meyden, R., and Su, K., 2004, "Symbolic model checking the knowledge of the dining cryptographers", under review.

van Ditmarsch, H. P., 2000, "Knowledge games", Ph.D. Thesis, University of Groningen, Groningen.

van Ditmarsch, H. P.: 2003, "The Russian cards problem", *Studia Logica* **75**(4), 31–62.

van Otterloo, S., and Jonker, G., 2004, "On epistemic temporal strategic logic", in *Proceedings of the Second Workshop on Logic and Communication in Multi Agent Systems* (*LCMAS*).

van Otterloo, S., van der Hoek, W., and Wooldridge, M., 2003, "Model checking a knowledge exchange scenario", in *Proceedings of Model Checking and Artificial Intelligence(MoChArt)* Acapulco, pp. 37–44.

van Otterloo, S., van der Hoek, W., and Wooldridge, M., 2004, “Preferences in game logics”, in *AAMAS 2004*, New York.

von Neumann, J. and Morgenstern, O., 1944, *Theory of Games and Economic Behaviour,* Princeton University Press: Princeton, NJ.

von Neumann, J. and Morgenstern, O., 1953, *Theory of Games and Economic Behaviour,* Princeton University Press: Princeton, NJ, 3rd edition.

Wooldridge, M. and Rao, A., (eds.), 1999, *Foundations of Rational Agency,* Kluwer Academic Publishers: Dordrecht, The Netherlands.