Knowledge Condition Games

Sieuwert van Otterloo Wiebe van der Hoek Michael Wooldridge
Department of Computer Science
University of Liverpool

Abstract

Agents often interact strategically to meet conditions involving their own or other agents' knowledge. This interaction can be modeled using a new method of game construction, knowledge condition games, introduced in this paper. Knowledge condition games are games in which a coalition of agents chooses its strategy in an extensive game form such that an epistemic logic condition is fulfilled. In this paper knowledge condition games are defined and applied to two example problems involving voting and a quiz master problem. It is also shown that a restricted class of knowledge condition games is NP-complete.

1. Introduction

Participating in a game can be more important than winning. This is certainly true for game-like situations such as auctions and voting protocols, since agents might enter these situations solely to gain knowledge about for instance the choices other agents make. Similarly, an agent might vote strategically in an election just to find out the preferences of other agents [14]. Other agents might want to use a protocol for the exchange of information, or under the constraint that certain information must remain private [22]. In this paper a new method of game construction, *knowledge condition games*, is presented that can capture these situations.

Extensive game forms with imperfect information [10] are often used as a model for multi agent interaction. Extensive game forms do not specify the preferences of agents, and therefore a game form does not define a game. One can create a game based on a game form by adding preferences for all agents over the different outcomes. Instead of this, we assume that a coalition of agents wants a certain knowledge condition to hold at the end of the protocol. The knowledge condition is expressed in epistemic logic [11]. Epistemic logic originates in philosophy [8] but has been successfully used in the domain of computer systems to capture the knowledge of agents in interpreted systems [6]. It has been combined with temporal logic [17] and coalition logic [19] in order to capture knowledge development over time, to capture knowledge change after announcements [1] or knowledge evolution in games such as Cluedo [21]. These frameworks have been applied to a range of different situations, such as the Russian Cards problem [22, 23] and the Dining Cryptographers Problem [20]. All these approaches mentioned treat the knowledge of agents as arising solely from their observations on the system, assuming that agents have no access to the strategies of other agents. An exception is the approach by Druiven [5] which introduces the term *strategic knowledge* for the knowledge derived from knowing a strategy, and develops a dynamic epistemic logic framework for handling strategic knowledge derived from deterministic strategies.

In knowledge condition games, agents do know the strategies that other agents use. This is a useful feature in for instance security protocols, where the security should not depend on keeping the strategy or algorithm used secret [14] and in protocols that are repeated often, such as daily auctions. In these repeated protocols agents can study the previous behaviour of opponents to form conjectures about their strategies. On the one hand knowledge condition games can be seen as an extension of the 'epistemic logic for multi agent systems' programme, with the goal of modeling adversarial multi agent systems. On the other hand it is a simplification of logics that try to combine features such as knowledge, time and strategies, such as Alternating-time Temporal Epistemic Logic(ATEL) [19]. In ATEL one can combine strategic operators, to say for instance that A knows it is able to prevent B to be able to let C know p. This expressibility comes at a price: ATEL formulas are harder to understand. Knowledge condition games provide a simpler alternative.

The main idea of the paper, expressed in a formal notation, is that an extensive game form interpretation F can be transformed in a knowledge condition game $g(F, \Gamma, \Xi, \phi)$. This is essentially a two player game, where the players are the coalitions of agents Γ and Ξ. Throughout this paper we assume that Γ and Ξ are disjoint. Γ wants ϕ to hold at the end of each run of the protocol, while Ξ wants to prevent

that ϕ holds for at least one run of the protocol. In section 2 this idea is formally defined. Section 3 contains examples of knowledge condition games, while section 4 looks at the computational complexity of knowledge condition games. Section 5 is the conclusion.

2. Definitions

In this section we define how one can create a knowledge condition game G from a game form interpretation F. In order to do this one needs a condition ϕ, a set of agents Γ that want ϕ to hold, and a set of agents Ξ that does not want ϕ to hold. We use the notation $G = g(F, \Gamma, \Xi, \phi)$ for this game. In this section we formally define the notions used and the function g.

We use a notation that is standard for alternating time temporal logic [19] and use Σ for the set of all agents, Γ and Ξ for coalitions of agents, X for a single agent, P for a set of atomic propositions and π for an interpretation function.

In order to express knowledge conditions we use the language of epistemic propositional logic, along with its $S5_n$ semantics [6, 11] and this language is called K in this paper.

Definition 1 *Let P be a finite set of atomic propositions and Σ a finite set of agents. The language K of epistemic logic over P and Σ is the smallest set L such that $P \subset L$ and for any $\phi, \psi \in L$ and $X \in \Sigma$*

$$\begin{aligned} \phi \vee \psi &\in L \\ \neg\phi &\in L \\ K_X\phi &\in L \end{aligned}$$

An example formula in this language is $\phi_0 = K_B p$. This formula expresses that B knows that p holds. Conjunction $\phi \wedge \psi$ is defined as $\neg(\neg\phi \vee \neg\psi)$.

This logic is interpreted over Kripke models [11]. A (multi-agent) Kripke model M is a tuple $M = (\Sigma, S, \sim, P, \pi)$, where Σ is a set of agents, S is a set of states and $\sim$ is a collection of equivalence relations $\sim_X$ between states, one for each agent $X \in \Sigma$. The meaning of the equivalence relations is that it relates states that cannot be distinguished by an agent: $s \sim_X t$ means that the states s and t look the same according to X. P is a set of atomic propositions and π is an interpretation function such that $\pi(s) \subseteq P$ returns the atomic propositions that are true in s.

An example Kripke model is the model M_0, which has two states s_1 and s_2 and two agents A and B. Agent A can distinguish these states, but agent B cannot: $s_1 \sim_B s_2$. In this model only one atomic proposition, proposition p, occurs. This proposition only holds in state s_1.

Epistemic formulas $\phi \in K$ can be interpreted over Kripke models. Let $M = (\Sigma, S, \sim, P, \pi)$ be a Kripke model and $s \in S$. We write $M, s \models \phi$ if ϕ is true in s. This notion is defined in the following way.

$$\begin{array}{lll} M, s \models p & \text{iff} & p \in \pi(s) \\ M, s \models \phi \vee \psi & \text{iff} & \pi, s \models \phi \text{ or } \pi, s \models \psi \\ M, s \models \neg\phi & \text{iff} & \text{not } \pi, s \models \phi \\ M, s \models K_X\phi & \text{iff} & \forall t \in S : s \sim_X t \implies \pi, t \models \phi \end{array}$$

We can use this definition to show the following results for the example model.

$$\begin{aligned} M_0, s_1 &\models K_A p \\ M_0, s_1 &\models \neg K_B p \\ M_0, s_2 &\models K_B(p \vee \neg p) \end{aligned}$$

A game form interpretation is basically a game form to which an interpretation function for atomic propositions has been added. A game form can be described in different but equivalent ways, for instance as a set of runs [12] or as a game tree [10]. We follow Osborne and Rubinstein [12] and use the idea of a set H of runs h. Each run 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. First we define game forms, then we extend this definition with an interpretation function to obtain the definition of a game form interpretation.

Definition 2 *A* game form *is a tuple $(\Sigma, H, Ow, \sim)$, where Σ is a finite set of agents, H is a non-empty set of finite* histories*, such that for any sequence $ha \in H$ also $h \in H$. Let ϵ to denote the empty sequence. The set of all single actions available after h is denoted $A(h) = \{a | ha \in H\}$. A history $h \in H$ is terminal if $A(h) = \emptyset$. The set of all terminal histories of H is denoted $Z(H)$. The function $Ow(h) : H \setminus Z(H) \to \Sigma$ defines which player chooses the next action. Intuitively the agent $Ow(h)$ owns the history h. For each agent $X \in \Sigma$, the relation $\sim_X$ is an equivalence relation between histories, where $h \sim_X j$ expresses the fact that agent X cannot tell the difference between having gone through history h and history j. One condition applies: if $Ow(h) = X$ and $h' \sim_X h$ then also $Ow(h') = X$ and further $A(h) = A(h')$. This condition ensures that an agent knows when it can select an action and that it knows which actions are available.*

This definition is taken from Osborne and Rubinstein [12], definition 200.1, where it is called an extensive game form. 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 [16, 2].

Definition 3 *A* game form interpretation *is a tuple $(\Sigma, H, Ow, \sim, P, \pi)$. The first elements $(\Sigma, H, Ow, \sim)$ form a game form. The set P contains atomic propositions and $\pi : Z(H) \to 2^P$ is a function that assigns to each terminal history the set of atomic propositions that are true in the final state of that history.*

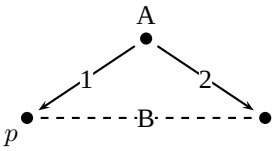

Figure 1. Game form interpretation F_0

The idea is that these propositions can be used to refer to certain histories, for instance to histories where an agent achieves a certain goal. The idea of annotating end states or terminal runs with logical propositions has been used before by Harrenstein e.a. [7] and the authors [24]. Approaches based on temporal logic [17, 18] often annotate all nodes of the model with propositions, so that formulas can be interpreted anywhere in the model.

An example game form interpretation 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. This game form interpretation can be defined as $F_0 = (\{A, B\}, \{\epsilon, 1, 2\}, Ow_0, \sim^0, \{P\}, \pi_0)$, where $Ow_0(\epsilon) = A$, $\pi_0(1) = \{p\}$, $\pi_0(2) = \emptyset$ and $1 \sim^0_B 2$ but not $1 \sim^0_A 2$.

For every game form F we can calculate a Kripke model $M = m(F)$ 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 M. Let $F = (\Sigma, H, Ow, \sim, P, \pi)$. The function m is defined by

$$m(F) = (\Sigma, Z(T), \sim, P, \pi)$$

If we apply the function m to the example game form F_0, we get the example Kripke model $M_0 = m(F_0)$. The transformation m is used to express when a game form F makes a formula ϕ true. We say that a game form F makes a formula ϕ true, written $F \models \phi$, if ϕ holds after every terminal history of F.

$$F \models \phi \Leftrightarrow \forall s \in Z(T)\ m(F), s \models \phi$$

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 nondeterministic 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. A strategy σ_Γ is thus a function such that for any node h with $Ow(h) \in \Gamma$ it returns a non-empty set $\sigma_\Gamma(h) \subseteq A(h)$. An extra condition is that $h \sim_{Ow(h)} h'$ must imply that $\sigma_\Gamma(h) = \sigma_\Gamma(h')$. This condition 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.

A *mixed strategy* is a function that for each node returns a probability function over the available options, with the intention that the agent should randomize over these actions with exactly the probabilities specified for each option. These mixed strategies are often used for the analysis of imperfect information games, because these strategies can provide agents much better expected payoffs then deterministic or pure strategies [12]. A nondeterministic strategy can be seen as an abstraction of a mixed strategy. It indicates which actions should be played with nonzero probability, but ignores the actual probabilities. This abstract view on strategies is sufficient to decide whether agents know something for sure at the end of a game, which is the reason that we use nondeterministic strategies.

For the example game form F_0 there are three different strategies $\sigma_{\{A\}}$ 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 a game form F we can consider a restricted 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 want in F'. Such a restricted 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 $F' = u(F, \sigma_\Gamma)$. Let $F = (\Sigma, H, Ow, \sim, P, \pi)$. We define u by $u(F, \sigma_\Gamma) = (\Sigma, H', Ow', \sim', P, \pi')$. The new set H' is the smallest set such that three conditions are met:

- $\epsilon \in H'$
- for each $h \in H'$ with $Ow(h) \in \Gamma$ we have that $\forall a\ (a \in \sigma_\Gamma(h) \rightarrow ha \in H')$
- for each $h \in H'$ with $Ow(h) \notin \Gamma$ we have that $\forall a\ (ha \in H \rightarrow ha \in H')$

The new equivalence relations $\sim'_X$ are restricted to histories that are part of H': $\sim'_X = \sim_X \cap (H' \times H')$. The functions Ow' and π' are the same as their unprimed counterparts, except that their domain is restricted to histories in 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.

$$m(u(F_0, \sigma^1_{\{A\}})) \models K_B p$$

By adopting the strategy $\sigma^1_{\{A\}}$, the agent A can thus ensure that B knows that p will hold in the outcome of the protocol. Agents are aware of the strategies that other agents use. The assumption implicit in our definition of knowledge is that the game form used is common knowledge, and that every strategy adopted by a coalition of agent is also common knowledge between all agents of the system.

As promised at the start of the section, the function $G = g(F, \Gamma, \Xi, \phi)$ defines a knowledge condition game in which Γ wishes to achieve ϕ, while Ξ hopes to prevent it. The game G is not an extensive game, but a game in normal or strategic form [12]. It is not possible to consider G as an extensive game, because whether the knowledge condition holds is not a local property of each end state, but a property that depends on the strategies as a whole. A strategic game is defined as a tuple $(N, A, \preceq)$ [12]. The knowledge condition game $g(F, \Gamma, \Xi, \phi) = (N, A, \preceq)$ is defined as follows.

- The ordered set of players N is $N = (\Gamma, \Xi)$
- $A = \{A_\Gamma, A_\Xi\}$ contains the actions of each player. In our case A_Γ contains all strategies σ_Γ that can be played in F. Similarly A_Ξ contains all strategies σ_Ξ.
- $\preceq_\Gamma$ and $\preceq_\Xi$ are the preference relations for the agents, over strategy combinations . We define

$$(\sigma_\Gamma, \sigma_\Xi) \preceq_\Gamma (\sigma'_\Gamma, \sigma'_\Xi) \texttt{ iff } v(F, \sigma_\Gamma, \sigma_\Xi) \geq v(F, \sigma'_\Gamma, \sigma'_\Xi)$$

In order to define the function v, assume that at the start of the game, the coalition Γ chooses a strategy σ_Γ and simultaneously Ξ selects a strategy σ_Ξ. An updated game G' is now calculated as $G' = u(u(G, \sigma_\Gamma), \sigma_\Xi)$. The game G is won by Γ if $G' \models \phi$. Otherwise Ξ wins. We write $v(G, \sigma_\Gamma, \sigma_\Xi) = 1$ if Γ wins and $v(G, \sigma_\Gamma, \sigma_\Xi) = 0$ in case Ξ wins.

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

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

We see that for this game, $\{A\}$ has a winning strategy (namely $\sigma^1_{\{A\}}$). The existence of a winning strategy for the first player of a game is denoted by $w(G) = 1$. The formal winning strategy condition for knowledge condition games is

$$w(g(F, \Gamma, \Xi, \phi)) = 1 \Leftrightarrow \exists \sigma_\Gamma \forall \sigma_\Xi \; v(G, \sigma_\Gamma, \sigma_\Xi) = 1$$

Note that for the example game it is the case for any ϕ that $w(g(F_0, \{A\}, \emptyset, \phi)) = 1$ if and only if $w(g(F_0, \{A\}, \{B\}, \phi)) = 1$. The reason is that the second player B only has no choices in this model. In general it does matter whether an agent Y is an opponent ($Y \in \Xi$) or neutral $Y \notin \Gamma \cup \Xi$.

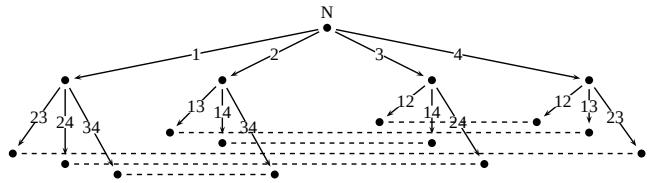

Figure 2. The fifty fifty problem F_Q

3. Examples

3.1. Anonymous Voting

A voting protocol is used when a group of agents has to make a joint decision on a certain issue. A standard 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 game form interpretation F_V, three agents A, B and C use this protocol to decide whether p should be true or not. They vote in alphabetical order, so first A chooses from action p or $\bar{p}$, then B (without knowing A's choice) chooses either p or $\bar{p}$ and finally C does the same, unaware of what A, B did. This protocol thus has eight terminal runs. The proposition p holds if at least two agents choose p. Furthermore a holds if A chooses p, b if B chooses p and the same for C with c. The interpretation function is thus $\pi(ppp) = \{a, b, c, p\}, \pi(pp\bar{p}) = \{a, b, p\} \ldots \pi(\bar{p}\bar{p}\bar{p}) = \emptyset$. 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.

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

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

One example, described by Schneier [14], p. 133, is a voting protocol where B would have the option of copying A's (encrypted) vote. In that case one might get

$$w(g(F_V, \{B\}, \{AC\}, K_B a \vee K_B \neg a)) = 1$$

This is an unwanted property and thus a 'bug' in the protocol. It is easy to overlook such bugs if one does not assume that cooperating agents know each other's strategy, because detection of these bugs may depend on this assumption.

3.2. Fifty fifty Problem

> In a weekly broad casted TV show the quiz master asks a candidate the next question: Which day of the week comes directly after Tuesday? Is that 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 several European countries in the 'Millionaire show'. The stakes in this game start out low, but potentially the stake is a million pounds or euros. Let us model this in a game form interpretation 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 game form interpretation is depicted in figure 2. First nature selects one of the answers to be the right answer. it can chose 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 $P = \{a_i | 1 \leq i \leq 4\} \cup \{e_i | 1 \leq i \leq 4\}$, 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:
$w(g(F_Q, \{N\}, \emptyset, \psi)) = 1$
Nature may not favour the candidate:
$w(g(F_Q, \{N\}, \emptyset, \neg\psi)) = 1$
The quiz master can help the candidate:
$w(g(F_Q, \{Q\}, \emptyset, \psi)) = 1$

We see that the question 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.

It 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. Take for instance this strategy $\sigma_{\{Q\}}$.

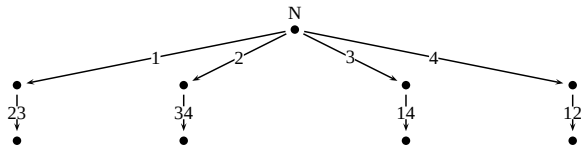

Figure 3. The updated game form interpretation $u(F_Q, \sigma_{\{Q\}})$

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

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 $w(g(F_Q, \{Q\}, \emptyset, q)) = 1$. A practical conclusion one can draw is that one should not bet on this quiz if one does not know what the interests of the quiz master are.

The fifty fifty problem is in fact similar to the Monty Hall dilemma [9]. In that problem there are only three options, and the quiz master (Monty Hall) only removes one option. The role or intentions of the quizmaster, whether he opens doors at random or not, is relevant in both problems. However one needs reasoning about chance in order to solve the Monty Hall dilemma, whereas we could analyse the quiz master problem without probabilities.

4. Complexity

The key decision problem for knowledge condition games is to decide, for a given F, Γ, Ξ and ϕ, whether $w(g(F, \Gamma, \Xi, \phi)) = 1$. Naturally one wants to know how easy or hard it is to decide the problem, since this determines the practical applicability of this new game concept. Problems like these are often called feasible or tractable if they can be solved in polynomial time, and intractable if this cannot be done. In this section we show that even a restricted problem, with $\Xi = \emptyset$, is NP complete. This problem is called the *restricted kcg problem*. It is generally believed that NP-complete problems are not tractable [3, 4].

Theorem 1 *The problem to decide for given F, Γ and ϕ whether $w(g(F, \Gamma, \emptyset, \phi)) = 1$ is NP-complete.*

Proof: To prove this theorem, we must first show that this problem is in NP. Then we show that the NP-complete problem 3SAT can be reduced to this problem.

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

$$w(g(F, \Gamma, \emptyset, \phi)) = 1 \;\Leftrightarrow\; \exists \sigma_\Gamma \; u(F, \sigma_\Gamma) \models \phi$$

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 $M = m(u(F, \sigma_\Gamma))$, and verify for each states s of M that $M, s \models \phi$. The number of states in M is at most the number of end states of F, so $|M| \leq |F|$. 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 restriced 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 restriced kcg instance. Let $\phi^3 = \bigwedge_i (a_i \vee b_i \vee c_i)$ be a propositional logic formula in conjunctive normal form with three literals per clause. The atomic 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 $\mathcal{A}$ for all atomic propositions in ϕ^3 exists such that $\mathcal{A} \models \phi^3$. We can construct a game form interpretation F with a single agent $\Sigma = \{A\}$ and a formula ϕ such that $w(g(F, \{X\}, \emptyset, \phi)) = 1$ if and only if $\exists \mathcal{A} : \; \mathcal{A} \models \phi^3$.

The model $F = (\{A\}, H, Ow, \sim, P, \pi)$ 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: $P = \{x^+ | x \in P^3\} \cup \{x^- | x \in P^3\}$. For each new proposition a history is created: $H = \{\epsilon\} \cup \{e_p | p \in P\}$. The interpretation function is such that only the corresponding atomic proposition is true: $\pi(e_p) = \{p\}$. Furthermore $Ow(n_0) = A$, because there are no other options. Agent A cannot distinguish any end state: $e_p \sim_A e_q$ for all states e_p and e_q.

We use $M_X \phi$ as a shortcut for $\neg K_X \neg \phi$. 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. A merely expresses that such assignment must assign either the truth value true (p^+) or false (p^-) to each proposition p.

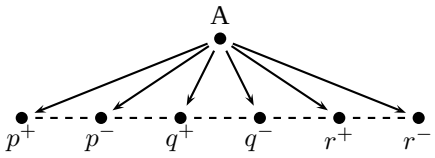

Figure 4. The model of 3SAT formula ψ

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 that takes an atomic proposition: $f(\neg p) = p^-$ 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 $\sigma_{\{X\}}$ such that $m(u(F, \sigma_{\{X\}})) \models \phi_1 \wedge \phi_2$ corresponds to an assignment $\mathcal{A}$ such that $\mathcal{A}(p) = true$ if and only if $p^+ \in \sigma_{\{X\}}(n_0)$, and that this assignment satisfies $\mathcal{A} \models \phi^3$. 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. □

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 game form interpretation, depicted in figure 4 contains six terminal histories. The corresponding knowledge formula is ψ_K.

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

When comparing this result with other complexity results for multi agent logics, one should keep in mind that we have represented the model F explicitly. Other results may use an implicit representation R of a set of histories $H(R)$, such that the size of $H(R)$ is not polynomially bounded in the size of R.

One can make a comparison between ATEL and knowledge condition games. Any protocol can be formulated as a concurrent game structure C suitable for ATEL model checking and as a game form interpretation F. We assume that C contains a atomic proposition `start` that holds in the initial state, and `end` that holds in any end state. Suppose that we want to find out whether coalition Γ can ensure that the epistemic logic formula ϕ holds at the end of the protocol. This can be expressed as the following model checking problem.

$$C \models \Box(\texttt{start} \rightarrow \ll \Gamma \gg \Box(\texttt{end} \rightarrow \phi))$$

One can also formulate this question as a knowledge condition game.

$$w(g(F, \Gamma, \emptyset, \phi)) = 1$$

These two statements have a slightly different interpretation, because in ATEL it is not assumed that agents know which strategies are played. Model checking ATEL under the assumption of uniform strategies is NP-complete [15], but the proof requires a different construction than the proof presented in this paper. The hardness of KCG is caused by the interaction between knowledge and strategies, whereas ATEL model checking is difficult because agents must coordinate.

Since the decidability problem for knowledge condition games is at least NP-hard (the general problem is at least as difficult as the restricted problem), automatic verification is restricted to small problem size, or easy special cases. In the remainder of this section we look at a special case that might help one to solve these problems.

A positive knowledge formulas is a formula in K in which every operator K_X appears within the scope of an even number of negations. Examples of positive formulas are $K_A p$ and $\neg(p \vee \neg K_A K_B \neg p)$. Negative knowledge formulas are the exact opposite: every knowledge operator K_X appears under an odd number of negations. If a knowledge condition game $g(F, \Gamma, \Xi, \phi)$ is played with a positive knowledge formula ϕ, then the best strategy for Ξ is to play any option available. Therefore, Ξ should not commit to anything, and Γ might as well play against the empty coalition. This is expressed in the next theorem.

Theorem 2 *Let F, Γ, Ξ and ϕ be given such that ϕ is a positive knowledge formula. $w(g(F, \Gamma, \Xi, \phi)) = 1$ if and only if $w(g(F, \Gamma, \emptyset, \phi)) = 1$*

Proof: As a first step in the proof, note that for any strategy σ, the Kripke model $m(u(F, \sigma))$ is exactly the same as $m(F)$, except that the set of states might be smaller in $m(u(F, \sigma))$. We say that $m(u(F, \sigma))$ is a submodel of $m(F)$. By induction we can show that if M' is a submodel of M and ϕ a positive knowledge formula, then for any state s existing in M' we have that $M, s \models \phi \Rightarrow M', s \models \phi$. For negative knowledge formulas ϕ it works the other way around: $M, s \models \phi \Leftarrow M', s \models \phi$. For any atomic proposition p this certainly holds and there is even an equality. Assuming that the claims are true for subformulas ψ and ϕ that are either both positive or both negative, one can show that both claims hold for $\phi \vee \psi$. For $M, s \models K_X \phi$ the positive claim is also true, since $\forall t \in S : s \sim t \Rightarrow M, t \models \phi$ implies that $\forall t \in S' : s \sim t \Rightarrow M, t \models \phi$ for the smaller set of states S'. For dealing with negation we first assume inductively that ϕ is a positive knowledge formula, which allows us to show the negative claim for $\neg\phi$: $M, s \models \neg\phi \Leftarrow M', s \models \neg\phi$. Then we assume that ϕ is negative and we can show that $M, s \models \neg\phi \Rightarrow M', s \models \neg\phi$ for the positive knowledge formula $\neg\phi$.

Using this first step, we know that for any strategy σ_Ξ it is the case that $m(F) \models \phi \Rightarrow m(u(F, \sigma_\Xi)) \models \phi$. Using the fact that for the empty strategy $\sigma_\emptyset$ we have that $u(F, \sigma_\emptyset) = F$, we get

$$\exists \sigma_\Gamma \forall \sigma_\Xi u(u(F, \sigma_\Xi), \sigma_\Gamma) \models \phi \Leftrightarrow \exists \sigma_\Gamma u(u(F, \sigma_\emptyset), \sigma_\Gamma) \models \phi$$

which had to be proven. □

A similar theorem regarding positive knowledge formulas has been formulated by Parikh and Ramanujam [13].

Positive knowledge formulas occur in situations where agents want to communicate: they want each other to know certain facts about the outcome. Such situations are often described as cooperative agent systems. Negative or non-positive formulas might occur in situations where agents want to keep each other uninformed. One can think of such examples as adversarial systems. This result thus makes it easier to check knowledge properties of cooperative multi agent systems. The conclusion here is thus similar to the observation made elsewhere that checking knowledge is easier for cooperative than for adversarial systems [23].

5. Conclusion

Knowledge condition games can be used for expressing properties of multi agent protocols that are hard to express in other frameworks. It shows how logic can contribute to game theory, and logic and game theory to the theory of multi agent systems, protocols and verification. One underlying assumption of the kcg framework is that strategies used by agents are known by all agents. This assumption makes knowledge condition games a suitable framework for modeling security protocols or adversarial multi agent systems. The main result we have established is that finding out whether a player has a winning strategy in a knowledge condition game is at least NP-hard. For positive knowledge formulas, which are useful in cooperative multi agent systems, it is in NP. Automatic verification is thus not a very tractable problem, but possible for small systems and simple knowledge properties.

In future work we hope to investigate the relation knowledge condition games and other frameworks. In particular it would be interesting to see whether complexity results for extensive games also apply to knowledge condition games. Furthermore we would like to compare other logical frameworks, such as ATEL and dynamic epistemic logic, to knowledge condition games. In order to do this one needs to make different assumptions regarding strategic knowledge, for instance allowing agents to know something but not everything about strategies. The question here is how general a

framework should be in order to capture realistic verification problems. More case studies based on larger problems would help to answer this question.

At the moment there is no practical implementation that can be used to automatically evaluate knowledge condition games. Since we have shown that this evaluation problem has a high computational complexity, such an implementation should in some way exploit formula structure in order to deal with game forms that are large enough to be interesting. Protocols with large numbers of agents or actions occur often in practical model checking problems, whereas complicated formulas are relatively rare. We speculate that constraint satisfaction techniques might be applicable.

References

[1] A. Baltag, L.S. Moss, and S. Solecki. 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, 2002.

[2] G. Bonanno. Memory implies von neumann-morgenstern games. *Knowledge Rationality and Action*, to appear:1–20, 2004.

[3] S. Cook. The P versus NP problem. *Manuscript prepared for the Clay Mathematics Institute for the Millennium Prize Problems*, 2000.

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

[5] S. Druiven. Knowledge development in games of imperfect information, 2002. University Maastricht Master Thesis.

[6] R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about knowledge*. The MIT Press: Cambridge, MA, 1995.

[7] B. Harrenstein, W. van der Hoek, J.-J. Ch. Meyer, and C. Witteveen. A modal characterization of nash equilibrium. *Fundamenta Informaticae*, 57(2–4):281–321, 2003.

[8] J. Hintikka. *Knowledge and Belief: an introduction to the logic of the two notions*. Cornell University Press: Ithaca, NY, 1962.

[9] B. P. Kooi. *Knowledge, Chance, and Change*. PhD thesis, University of Groningen, Groningen, 2003. ILLC Dissertation Series 2003-01.

[10] H Kuhn. Extensive games and the problem of information. *Contributions to the Theory of Games*, II:193–216, 1953.

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

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

[13] R. Parikh and R. Ramanujam. A knowledge based semantics of messages. *Journal of Logic Language and Information*, 12:453–467, 2003.

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

[15] Pierre-Yves Schobbens. Alternating-time logic with imperfect recall. In Wiebe van der Hoek, Alessio Lomuscio, Erik de Vink, and Mike Wooldrige, editors, *Electronic Notes in Theoretical Computer Science*, volume 85. Elsevier, 2004.

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

[17] W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In D. Bošnački and S. Leue, editors, *Model Checking Software, Proceedings of SPIN 2002 (LNCS Volume 2318)*, pages 95–111. Springer-Verlag: Berlin, Germany, 2002.

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

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

[20] R. van der Meyden and K. Su. Symbolic model checking the knowledge of the dining cryptographers. submitted, 2002.

[21] H. P. van Ditmarsch. *Knowledge Games*. PhD thesis, University of Groningen, Groningen, 2000.

[22] H. P. van Ditmarsch. The russian cards problem. *Studia Logica*, 75(4):31–62, 2003.

[23] S. van Otterloo, W. van der Hoek, and M. Wooldridge. Model checking a knowledge exchange scenario. In *Proceedings of Model Checking and Artificial Intelligence(MoChArt)*, pages 37–44, Acapulco, August 2003.

[24] S. van Otterloo, W. van der Hoek, and M. Wooldridge. Preferences in game logics. In *AAMAS 2004*, New York, July 2004.